КОМПЛЕКС ВЫЧИСЛИТЕЛЬНЫЙ СМ I700

Заводской № 0312 Год выпуска 1989

СМДО СМ I700

Подсистема процессора СМ 2700.2400

Микропрограммные тесты процессора СМ 2700.2400

Часть 2

Текст программы

00076-01 I2 03-2

Книга I0

Утвержден

00076-01 12 03-1-ЛУ

СИСТЕМА МИКРОДИАГНОСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ
ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА СМ 1700

СМДО СМ1700

ПОДСИСТЕМА ПРОЦЕССОРА СМ2700.2400
Микропрограммные тесты процессора СМ 2700.2400

Текст программы
Часть 2

00076-01 12 03-2

Листов 225

1987

Перв. примен.
00076-01

Литера

АННОТАЦИЯ

Настоящий документ содержит тексты программной секции ENKCB, входящей в систему микродиагностического обеспечения вычислительного комплекса СМ 1700 - СМДО СМ1700 и реализующей часть 2 микропрограммных тестов процессора СМ 2700.2400.

Текст программы представлен в загрузочном формате и в форме листинга трансляции, содержащего исходные тексты микропрограмм на языке МИАСС.

Магнитная лента с текстом программы предназначена для создания рабочих кодов микродиагностики на носителе устройства ввода консоли и получения твердой копии документа с исходными текстами, используемого при локализации неисправности, обнаруженной микропрограммными тестами во время их выполнения.

Подробные сведения о структуре микропрограммных тестов содержатся в документе 00076-01 13 01. СИСТЕМА МИКРОДИАГНОСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА СМ 1700 СМДО СМ1700. Описание программы.

Загрузочный модуль на магнитной ленте имеет метку ENKCB.EXE, листинг трансляции - ENKCB.MCR.

; СОДЕРЖАНИЕ

```
;4        .PAGE "ФОРМАТЫ МИКРОИНСТРУКЦИЙ"
;5        .HEXADECIMAL
;6        .LIST
;7        .NOCREF
;8        ;
;9        ;
;10       ;          Следующая таблица изображает 24-битовое микрослово
;11       ;        процессора и соответствует формату слова в управляющей
;12       ;        памяти (WCS). Бит 23 (не показан) представляет собой
;13       ;        бит контроля по паритету (нечет).
;14       ;
;15       ;
;16       ;                  22 21       16 15          9 8 7 6 5 4       0
;17       ;                  +-----+-------+-------+-------+-------+-------+
;18       ;1.BASIC           :1:     DP    :    D ADRS   : B :CC :  SCTL   :
;19       ;  (базовый)       +-+-----------+-------------+---+---+---------+
;20       ;
;21       ;                  22  20 19 17 16            9 8 7 6 5 4       0
;22       ;                  +-----+-------+-------+-------+-------+-------+
;23       ;2.MOVE            :0 1 1: MDP :    XD ADRS    : B :CC :  SCTL   :
;24       ;  (пересылки)     +-----+-----+-------------+---+---+---------+
;25       ;
;26       ;                  22  20 19       14 13 11   9 8 7 6 5 4       0
;27       ;                  +-----+-------+-------+-------+-------+-------+
;28       ;3.EXTENDED        :0 1 0:    XDP    : SI  : A : B :CC :  SCTL   :
;29       ;  (расширенный)   +-----+---------+-----+---+---+---+---------+
;30       ;
;31       ;                  22    19 18 16 15          9 8 7 6 5 4       0
;32       ;                  +-----+-------+-------+-------+-------+-------+
;33       ;4.MEM.REQ         :0 0 1 1: MF1 :    D ADRS   :MF2:DT :  SCTL   :
;34       ;  (запроса        +-------+-----+-------------+---+---+---------+
;35       ;      памяти)
;36       ;                  22    19 18 16 15   12 11  9 8 7 6 5 4       0
;37       ;                  +-----+-------+-------+-------+-------+-------+
;38       ;5.MISC            :0 0 1 0:P:0 0:   M1  : M2  :0:R:0 0:  SCTL   :
;39       ;  (разных         +-------+-+---+-------+-----+-+-+---+---------+
;40       ;      операций)
;41       ;                  22    19 18 17                      4 3       0
;42       ;                  +-----+-------+-------+-------+-------+-------+
;43       ;6.JUMP            :0 0 0 1:OS:    JUMP.ADRS                : JCTL  :
;44       ;  (перехода)      +-------+--+-------------------------------+-------+
;45       ;
;46       ;                  22    19 18 17      12 11  9 8 7 6 5 4 3       0
;47       ;                  +-----+-------+-------+-------+-------+-------+
;48       ;7.DECODE          :0 0 0 0: : DEC.ADRS  :IFUNC: : : : : : JCTL  :
;49       ;  (дешифрации     +-------+++-----------+-----++++++++++-------+
;50       ;    инструкций)          :                    : : : : :
;51       ;                  BACK UP PC          IB REQ---+ : : : +--OPC/SPEC
;52       ;                                                 : : :
;53       ;                            SEL CM HI IR BYTE----+ : +--LOAD RDEST
;54       ;                                                   :
;55       ;                                       LOAD OS--+
;56
```

```
;57      .PAGE "ОПРЕДЕЛЕНИЯ ПОЛЕЙ ОСНОВНОГО МИКРОСЛОВА"
;58      ;
;59      ;    Следующие выражения используются при контроле истинности комбинаций полей
;60      ;
;61      .SET/A.VAL=<.ORI[<.EQLI[<OPC2/>,4]>,<.EQLI[<OPC2/>,5]>]>
;62      .SET/B.VAL=<.ANDI[<.NEQI[<OPC2/>,0]>,<.NEQI[<OPC2/>,1]>,<.NEQI[<OPC2/>,2]>,
;63      <.NEQI[<OPC2/>,3]>]>
;64      .SET/D.VAL=<.ANDI[<.NEQI[<OPC2/>,0]>,<.NEQI[<OPC2/>,1]>,<.NEQI[<OPC2/>,2]>,
;65      <.NEQI[<OPC2/>,4]>,<.NEQI[<OPC2/>,5]>,<.NEQI[<OPC2/>,6]>,
;66      <.NEQI[<OPC2/>,7]>]>
;67      .SET/XD.VAL=<.EQLI[<OPC1/>,<OPC1/MOVE>]>
;68      .SET/CC.VAL=<.ANDI[<.NEQI[<OPC2/>,0]>,<.NEQI[<OPC2/>,2]>,<.NEQI[<OPC2/>,3]>]>
;69      .SET/DT.VAL=<.EQLI[<OPC2/>,<OPC2/MEM.REQ>]>
;70      .SET/SCTL.VAL=<.ANDI[<.NEQI[<OPC2/>,0]>,<.NEQI[<OPC2/>,1]>]>
;71      .SET/JCTL.VAL=<.ORI[<.EQLI[<OPC2/>,0]>,<.EQLI[<OPC2/>,1]>]>
;72      .SET/JUMP.VAL=<.EQLI[<OPC2/>,<OPC2/JUMP>]>
;73      .SET/DECODE.VAL=<.EQLI[<OPC2/>,<OPC2/DECODE>]>
;74      .SET/MISC.VAL=<.EQLI[<OPC2/>,<OPC2/MISC>]>
;75      .SET/DP.VAL=<.EQLI[<OPC/>,<OPC/BASIC>]>
;76      .SET/PAGE.PROB=<.EQLI[<.ANDI[<7FF>,<.>]>,<7FE>]>
;77      .SET/SKIP.LEGAL2=<.ORI[<.EQLI[<SCTL/>,<SCTL/RET.NOERR>]>,<.EQLI[<SCTL/>,
;78      <SCTL/POP.USTACK>]>]>
;79      .SET/SKIP.LEGAL=<.ORI[<.EQLI[<SCTL/>,<SCTL/NO.SKIP>]>,<.EQLI[<SCTL/>,
;80      <SCTL/RETURN>]>,<.EQLI[<SCTL/>,<SCTL/RETURN+1>]>,<SKIP.LEGAL2>]>
;81      .SET/SKIP.PAGE.VAL=<.SELECTI[<.EQLI[<PAGE.PROB>,0]>,<1>,<.EQLI[<PAGE.PROB>,1]>,
;82      <SKIP.LEGAL>]>
;83      .SET/JUMP.CHECK=<.ORI[<.EQLI[<JCTL/>,<JCTL/JSR>]>,
;84      <.EQLI[<JCTL/>,
;85      <JCTL/JSR.VALID>]>]>
;86      .SET/JUMP.PAGE.VAL=<.SELECTI[<.EQLI[<PAGE.PROB>,0]>,<1>,
;87      <.EQLI[<PAGE.PROB>,1]>,<.XORI[1,<JUMP.CHECK>]>]>
;88      ;
;89      ;  Определения кодов операций микроинструкций
;90      ;
;91      OPC/=<22>,.DEFAULT=<OPC/BASIC>
;92              BASIC=1             ; микроинструкция BASIC
;93
;94      OPC1/=<22:20>
;95              EXTENDED=2          ; микроинструкция EXTENDED
;96              MOVE=3              ; микроинструкция MOVE
;97
;98      OPC2/=<22:19>
;99              MEM.REQ=3           ; микроинструкция MEM.REQ
;100             MISC=2              ; микроинструкция MISC
;101             JUMP=1              ; микроинструкция JUMP
;102             DECODE=0            ; микроинструкция DECODE
;103     ;
;104     ;        Определения адресных полей в микроинструкциях:
;105     ;
;106     ;  A.ADRS - обращение к внутренней озу микропроцессора K1804BC1 через порт A
;107     ;  B.ADRS - обращение к внутренней озу микропроцессора K1804BC1 через порт B
;108     ; XB.ADRS
;109     ;
;110     A.ADRS/=<10:9>,.VALIDITY=<A.VAL>
;111             MASK=3              ; маска резервной копии регистров
```

```
;112    B.ADRS/=<8:7>,.VALIDITY=<B.VAL>,.DEFAULT=0
;113            MASK=3          ; маска резервной копии регистров
;114    XB.ADRS/=<8:7>,.VALIDITY=<.EQL[<OPC1/>,<OPC1/MOVE>]>
;115            BPC=0           ; начальное значение счетчика команд (PC)(WR[4])
;116            VA=1            ; адрес обращения к памяти (WR[5])
;117            VAR=1           ; адрес последнего обращения к памяти (WR[5])     6;
;118
;119    ;
;120    ;       Поля управления путями данных
;121    ;
;122    ;  Управляющее поле путей данных микроинструкции BASIC
;123    ;
;124    DP/=<21:16>,.VALIDITY=<DP.VAL>,.DEFAULT=<.SHIFT[.AND[<SDP/Q.CMP.LS>,0FF],-2]>
;125    ;
;126    ;  DP.SMALL применяется в микроинструкциях, к которым можно присоединить XCHG
;127    ;
;128    DP.SMALL/=<20:16>,.VALIDITY=<DP.VAL>
;129    XCHG.BIT/=<21>,.DEFAULT=<0>
;130            YES=1           ; взаимная замена исходного значения рабочего регистра
;131                            ; и ячейки LS
;132            NO=0
;133    ;
;134    ;  Управляющее поле путей данных микроинструкции EXTENDED
;135    ;
;136    XDP/=<19:14>,.VALIDITY=<A.VAL>
;137    ;
;138    ;  Управляющее поле путей данных микроинструкции MOVE
;139    ;
;140    MDP/=<19:17>,.VALIDITY=<XD.VAL>
;141    ;
;142    ;  Поле кодов условий
;143    ;
;144    CC/=<6:5>,.VALIDITY=<CC.VAL>,.DEFAULT=<CC/DT(LONG)&HOLD.ALU.CC>
;145            DT(LONG)&HOLD.ALU.CC=0  ; использование контекста длинных слов
;146                                    ; (32 бита) и сохранение кодов условий АЛУ
;147            DT(LONG)&SET.ALU.CC=1   ; использование контекста длинных слов
;148                                    ; (32 бита) и установка кодов условий АЛУ
;149            DT(SIZE)&SET.ALU.CC=2   ;.использование контекста согласно регистру
;150                                    ; размера и установка кодов условий АЛУ
;151            DT(SIZE)&MAP.ALU.CC.TO.PSL=3 ; аппаратно-зависимая пересылка кодов
;152                                    ; условий АЛУ в коды условий PSL
;153    ;
;154    ;       Поле типа данных для памяти
;155    ;
;156    MDT/=<6:5>,.VALIDITY=<DT.VAL>
;157                    BYTE=0          ; размер = байт
;158                    WORD=1          ; размер = слово
;159                    SIZE=2          ; размер = содержимое регистра размера
;160                    LONG=3          ; размер = длинное слово
;161    ;
;162    ;       Поле входа сдвигов
;163    ;
;164    ;       Это управляющее поле определяет входы сдвигов для следующих операций.
;165    ;       Направление сдвига определяется кодом приемника К1804ВС1
;166    ;
```

```
;167    SI/=<13:11>,.VALIDITY=<A.VAL>
;168                    ROT.WR=0            ; циклический сдвиг рабочего регистра (WR)
;169                    ROT.WR.Q=1          ; циклический сдвиг WR и Q
;170                    ASH.WR=2            ; арифметический сдвиг WR
;171                    ASH.WR.Q=3          ; арифметический сдвиг WR и Q
;172                    MUL.SHF=4           ; операция сдвига для умножения с учетом знака
;173                    UMUL.SHF=5          ; операция сдвига для беззнакового умножения
;174                    SHF.WR.Q=6          ; логический сдвиг WR и Q
;175    ;
;176    ;       Поле управления пропуском микроинструкций
;177    ;
;178    ;       Это поле действительно для всех микроинструкций,за исключением
;179    ;               микроинструкции JUMP
;180    ;               микроинструкции DECODE
;181    ;
;182    SCTL/=<4:0>,.VALIDITY=<.AND[<SCTL.VAL>,<SKIP.PAGE.VAL>]>,.DEFAULT=<SCTL/NO.SKIP>
;183                    N.CLR=0             ; бит N АЛУ равен нулю
;184                    NEQ=1               ; бит Z АЛУ равен нулю
;185                    BIT.SET=1           ; бит Z АЛУ равен нулю
;186                    V.CLR=2             ; бит V АЛУ равен нулю
;187                    C.SET=3             ; бит C АЛУ равен единице
;188                    GEQU=3              ; бит C АЛУ равен единице
;189                    NOT.PSL.C=4         ; бит C PSL равен нулю
;190                    BR.FALSE=5          ; ветвление в команде условного перехода ложно
;191                    R.DST=6             ; признак регистрового режима адресации
;192                    NO.INTERRUPT=7      ; ожидающих прерываний нет
;193                    N.SET=8             ; бит N АЛУ равен единице
;194                    EQL=9               ; бит Z АЛУ равен единице
;195                    BITS.CLR=9          ; бит Z АЛУ равен единице
;196                    V.SET=0A            ; бит V АЛУ равен единице
;197                    C.CLR=0B            ; бит C АЛУ равен нулю
;198                    LSSU=0B             ; бит C АЛУ равен нулю
;199                    STATE.ZERO.SET=0C;  бит 0 регистра STATE истинный
;200                    STATE.ONE.SET=0D ;  бит 1 регистра STATE истинный
;201                    STATE.ZERO.CLR=0E;  бит 0 регистра STATE ложный
;202                    STATE.ONE.CLR=0F ;  бит 1 регистра STATE ложный
;203                    RET.NOERR=10        ; возврат+1,если отсутствует ошибка системы памяти
;204                    JMP.Z.CLR=11        ; переход (по стеку), если бит Z АЛУ равен 0
;205                    POP.USTACK=12       ; сбрасывание верхней записи стека
;206                    JMP.C.SET=13        ; переход (по стеку), если бит C АЛУ равен единице
;207                    RETURN=14           ; возврат к (по микростеку)
;208                    NO.SKIP=15          ; выполнение следующей микроинструкции
;209                    RETURN+1=16         ; возврат (по микростеку)+1
;210                    LOOP.IF.NO.I.AND SYNC=17 ; переход (по стеку), если отсутствуют
;211                                        ; прерывания и требование синхронизации ускорителя
;212                    CONSOLE.ATTN=18     ; "внимание" (ATTN) консоли
;213                    CONSOLE.ACK=19      ; "подтверждение" (ACK) консоли
;214                    NO.FAST.INTERRUPT=1A;ожидающих быстрых прерываний нет
;215                    BACKUP.MASK.VALID=1B;маска для резервной копии регистров истинная
;216                    LSS=1C              ; меньше,чем (с учетом знака)
;217                    MEM.REF.OK=1D       ; ошибки системы памяти нет
;218                    SKIP=1E             ; выполнение микронструкции по адресу плюс один
;219                    SYNC=1F             ; синхронизация ускорителя
;220    ;
;221    ;       Поле управления микроинструкцией JUMP
```

```
;222     ;
;223     ;       Это поле истинно для следующих инструкций:
;224     ;               микроинструкция JUMP
;225     ;               микроинструкция DECODE
;226     ;
;227     JCTL/=<3:0>,.VALIDITY=<.AND[<JCTL.VAL>,<JUMP.PAGE.VAL>]>,.DEFAULT=<JCTL/JUMP.VALID>
;228                     N.CLR=0             ; бит N АЛУ равен нулю
;229                     NEQ=1               ; бит Z АЛУ равен нулю
;230                     BIT.SET=1           ; бит Z АЛУ равен нулю
;231                     V.CLR=2             ; бит V АЛУ равен нулю
;232                     C.SET=3             ; бит C АЛУ равен единице
;233                     GEQU=3              ; бит C АЛУ равен единице
;234                     JUMP=4              ; безусловный переход
;235                     BR.FALSE=5          ; ветвление в команде условного перехода ложно
;236                     R.DST=6             ; признак регистрового режима
;237                     NO.INTERRUPT=7      ; ожидающих прерываний нет
;238                     N.SET=8             ; бит N АЛУ равен единице
;239                     EQL=9               ; бит Z АЛУ равен единице
;240                     BITS.CLR=9          ; бит Z АЛУ равен единице
;241                     V.SET=0A            ; бит V АЛУ равен единице
;242                     C.CLR=0B            ; бит C АЛУ равен нулю
;243                     LSSU=0B             ; бит C АЛУ равен нулю
;244                     JSR=0C              ; безусловный переход и занесение в микростек
;245                     JSR.VALID=0D        ; переход и занесение в микростек,если
;246                                         ; IB VALID=1
;247                     NO.JUMP.TST=0E      ; нет перехода и пропуск,если IB VALID=0
;248                     JUMP.VALID=0F       ; переход,если IB VALID=1
;249     ;
;250     ; Поле адреса перехода
;251     ;
;252     JUMP.ADRS/=<17:4>,.ADDRESS,.VALIDITY=<JUMP.VAL>
;253     ;
;254     ; Разрешение для регистра OS
;255     ;
;256     OS/=<18>,.VALIDITY=<JUMP.VAL>
;257                     NOP=0
;258                     WITH.OS=1           ; присоединение регистра OS к адресу перехода
;259     ;
;260     ; Резервная копия счетчика команд (PC)
;261     ;
;262     BPC/=<18>,.VALIDITY=<DECODE.VAL>,.DEFAULT=<.CASE[<.EQL[<OPC2/>,<OPC2/DECODE>]>]OF[0,1]>
;263                     NOP=1
;264                     SAVE.PC=0           ; запись данных АЛУ в рабочий регистр по адресу
;265                                         ; порта B (PC+1)
;266     ;
;267     ; Адресное поле микроинструкции DECODE
;268     ;
;269     DEC.ADRS/=<17:12>,.VALIDITY=<DECODE.VAL>
;270     ;
;271     ; Функция регистра предвыборки команд (PFR)
;272     ;
;273     IFUNC/=<11:9>,.VALIDITY=<DECODE.VAL>
;274                     CM.EXEC=0           ; начальное ветвление на выполнение в режиме совместимости,
;275                                         ; если старший байт=нулю
;276                     SPEC=0              ; начальное ветвление по спецификатору
```

```
;277                    CM.IRD=1          ; начальное ветвление по дешифрации команды в режиме
;278                                      ; совместимости
;279                    FLOAT=1           ; начальное ветвление по спецификатору плавающего типа
;280                    VAX.EXEC=2        ; начальное ветвление на выполнение в собственном режиме
;281                    ASRC=2            ; начальное ветвление по спецификатору адреса
;282                    VAX.IRD=3         ; начальное ветвление по коду операции в собственном режиме
;283                    VSRC=4            ; начальное ветвление по спецификатору адреса для команд
;284                                      ; битовых полей
;285                    ESRC=5            ; начальное ветвление по спецификатору в индексном режиме
;286                    CM.DST=6          ; начальное ветвление по приемнику в режиме совместимости
;287                    CM.SINGLE=7       ; начальное ветвление в режиме совместимости при одном операнде
;288    ;
;289    ; Запрос заполнения регистра предвыборки команд
;290    ;
;291    IB.REQ/=<B>,.VALIDITY=<DECODE.VAL>,.DEFAULT=0
;292                    NOP=0
;293                    IB.REQ=1          ; разрешение аппаратно-зависимого запроса памяти
;294    ;
;295    ; Выбор кода операции режима совместимости
;296    ;
;297    CM.HI/=<7>..VALIDITY=<DECODE.VAL>,.DEFAULT=0
;298                    LOW.BYTE=0        ; дешифрация битов <7:0> регистра предвыборки инструкций (PFR)
;299                    HI.BYTE=1         ; дешифрация битов <15:6> регистра предвыборки инструкций (PFR)
;300    ;
;301    ; Загрузка регистра OS
;302    ;
;303    LD.OS/=<6>,.VALIDITY=<DECODE.VAL>,.DEFAULT=0
;304                    NOP=0
;305                    LOAD.OS=1         ; загрузка регистра OS из регистра предвыборки инструкций
;306
;307    ; Разрешение установки признака приемника типа регистра
;308    ;
;309    R.DST/=<5>,.VALIDITY=<DECODE.VAL>,.DEFAULT=0
;310                    NOP=0
;311                    ENAB=1
;312
;313    ; Бит кода операции/спецификатора
;314    ;
;315    OPC.SPEC/=<4>,.VALIDITY=<DECODE.VAL>
;316                    CM.EXEC=1         ; выход на выполнение в режиме совместимости
;317                    SPEC=0            ; выход по спецификатору
;318                    CM.IRD=1          ; выход по дешифрации команд в режиме
;319                                      ; совместимости
;320                    FLOAT=0           ; выход по спецификатору плавающего типа
;321                    VAX.EXEC=1        ; выход на выполнение в собственном режиме
;322                    ASRC=0            ; выход по адресу спецификатора
;323                    VAX.IRD=1         ; выход по коду операций в собственном режиме
;324                    VSRC=0            ; выход по спецификатору адреса для команд
;325                                      ; битовых полей
;326                    ESRC=0            ; выход по спецификатору в индексном режиме
;327                    CM.DST=0          ; выход по приемнику в режиме совместимости
;328                    CM.SINGLE=0       ; выход в режиме совместимости при одном операнде
;329    ;
;330    ; Разные функции
;331    ;
```

```
;332    ; Это поле представляет собой признак для возбуждения интерпретации микрослова
;333    ; в качестве инструкции разного назначения или в качестве инструкции порта
;334    ;
;335    MISC.PORT/=<1B>
;336                    MISC=0
;337                    PORT=1
;338    ;
;339    MISC.0/=<17>,.VALIDITY=<MISC.VAL>
;340                    NOP=0               ; операции в АЛУ и с кодами условий отсутствуют
;341    ;
;342    MISC.1/=<15:12>,.VALIDITY=<MISC.VAL>
;343                    NOP=0F              ; отсутствие операции
;344                    SET.CP.ATTN=0E      ; установка для консоли сигнала ATTN (внимание)
;345                                        ; из DAP
;346                    SET.CP.ACK=0D       ; установка для консоли сигнала ACK (подтверждение)
;347                                        ; из DAP
;348                    CLR.CP.ATTN.AND.ACK=0C  ; очистка сигналов DAP ATTN и ACK
;349                    SET.HIGH.PAGE=0B ; установка бита для доступа к старшей половине
;350                                        ; управляющей памяти (WCS)
;351                    CLR.HIGH.PAGE=0A ; очистка бита для доступа к младшей половине
;352                                        ; WCS
;353                    SET.PORT.I.O=9      ; установка режима ввода/вывода порта
;354                    SET.ACC.I.O=8       ; установка режима ввода/вывода ускорителя
;355                    SET.BACKUP.MASK.VALID=7  ; установка признака маски для резервной
;356                                        ; копии регистров
;357                    SET.S1=6            ; возбуждение бита 1 регистра STATE
;358                    SET.S0=5            ; возбуждение бита 0 регистра STATE
;359                    CLR.S1=4            ; очистка бита 1 регистра STATE
;360                    CLR.S0=3            ; очистка бита 0 регистра STATE
;361                    SET.S1&CLR.S0=2     ; возбуждение бита 1 регистра STATE и очистка
;362                                        ; бита 0 регистра STATE
;363                    CLR.S1&SET.S0=1     ; очистка бита 1 регистра STATE и возбуждение
;364                                        ; бита 0 регистра STATE
;365                    CLR=0               ; очистка битов регистра STATE 0 и 1
;366    ;
;367    MISC.2/=<11:9>,.VALIDITY=<MISC.VAL>
;368                    NOP=0               ; отсутствие операции
;369                    MASK.INTERRUPTS=1; маскирование всех прерываний (для дешифрации
;370                                        ; в следующем цикле)
;371                    ASSERT.DA.AND.PASS.WR=2; выдача сигнала наличия данных и пересылка
;372                                        ; WR[0] ускорителю или порту
;373                    TRAP.ACC=3          ; прерывание ускорителя
;374                    READ.ACC.UPC=4      ; чтение счетчика микрокоманд ускорителя
;375                    TRANSFER.GRANT=5 ; опознавание запроса порта
;376                    MASK.HALT.AND.T.BIT=6; маскирование прерываний по останову (HALT;
;377                                        ; и T-биту (для дешифрации двумя циклами позже)
;378                    WRITE.WR.TO.CONSOLE.REGISTER=7 ; пересылка битов <7:0> для
;379                                                   ; микропроцессора консоли
;380    ;
;381    MISC.WR/=<8:7>
;382    MISC.WRL/=<8>
;383    MISC.WRR/=<7>
;384    ;
;385    ;  Выборка устройства порта
;386    ;
```

```
;387    PORT.SELECT/=<17:15>
;388                    IDC=7               ; адресуется контроллер дисков IDC
;389    ;
;390    ;  Функция порта
;391    ;
;392    PORT.FUNC/=<14:13>
;393                    READ=0              ; чтение
;394                    WRITE=1             ; запись
;395                    CONTROL=2           ; управление
;396    ;
;397    ;  Команда для обмена данными между IDC и DAP
;398    ;
;399    R.W.FUNC/=<12:10>
;400                    CSR=0               ; чтение/загрузка управляющего слова IDC
;401                    DAR=1               ; чтение/загрузка адреса данных на диске или
;402                                        ; загрузка команды
;403                    DATA.BYTE=2         ; чтение/загрузка одного байта данных
;404                    DATA.WORD=3         ; чтение/загрузка одного слова данных
;405                    PATTERN=4           ; чтение в DAP информации об ошибке (данные)
;406                    POSITION=5          ; чтение в DAP информации об ошибке (адрес)
;407    ;
;408    ;  Функции управления
;409    ;
;410    CNTL.FUNC/=<12:10>
;411                    CLEAR.FIFO.CNTR=0 ; очистка счетчика управления буферами данных
;412                    RESET.BR=1          ; очистка запроса прерывания BR5
;413                    CLEAR.IDC=3         ; начальная установка IDC
;414                    SET.AUTO=4          ; установка режима автоматической передачи слов
;415                                        ; данных в DAP
;416                    CLEAR.AUTO=5        ; снятие режима автоматической передачи
;417                    SEL.FIFO.A=6        ; выборка буфера A
;418                    SEL.FIFO.B=7        ; выборка буфера B
;419    ;
;420    ;  Поле запроса памяти
;421    ;
;422    ;  Это поле используется для указания нужной операции для управления памяти.
;423    ;  Поле является фиктивным. В действительности оно образовано из двух полей:
;424       M.FUNC1 и M.FUNC2
;425    ;
;426    ;  Особенности некоторых функций памяти.
;427    ;
;428    ;  ROTATE.BYTE.RIGHT - выполняет девятибитовый циклический сдвиг, поэтому ответ
;429    ;                      должен корректироваться действием ROL WRD 3
;430    ;
;431    ;  WORD.SWAP         - выполняет 15-битовый циклический сдвиг, поэтому ответ должен
;432    ;                      корректироваться действием ROL WRD 3
;433    ;
;434       OCTA.WRITE.P      - адрес должен быть выравнен по длинному слову
;435    ;
;436    ;  OCTA.WR.V.WCHK    - адрес должен быть выравнен по длинному слову. Эти данные могут
;437    ;                      пересекать границы страниц
;438    ;
;439    MEM.FUNC/=<7>,.VALIDITY=0
;440            PREFETCH=0                  ; запрос потока команд. вызывает увеличение (+1)
;441                                        ; виртуального адреса перед обращением и заполнение
```

```
;442                                    ; буфера команд
;443             WRITE.TB=1             ; запись в буфер трансляции
;444             TEST.V.RCHK=2          ; проверка защиты по чтению. Возвращает на шину
;445                                    ; МС содержимое буфера трансляции
;446             READ.P=3               ; чтение по физическому адресу
;447             WRITE.P=4              ; запись по физическому адресу
;448             TEST.V.WCHK=5          ; проверка защиты по записи. Возвращает на шину
;449                                    ; МС содержимое буфера трансляции
;450             ROTATE.BYTE.RIGHT=6    ; циклический сдвиг адресных данных на 9 битов
;451                                    ; вправо и возвращение их на шину МС
;452             WORD.SWAP=7            ; циклический сдвиг адресных данных на 15 битов
;453                                    ; влево и возвращение их на шину МС
;454             READ.V.NOCHK=8         ; чтение по виртуальному адресу без проверки
;455                                    ; защиты
;456             READ.V.WCHK=9          ; чтение по виртуальному адресу с проверкой защиты
;457                                    ; по записи
;458             READ.V.RCHK=0A         ; чтение по виртуальному адресу с проверкой защиты
;459                                    ; по чтению
;460             READ.MAINT.VAR.INC=0B  ; проверка инкрементирования виртуального адреса
;461             WRITE.V.NOCHK=0C       ; запись по виртуальному адресу без проверки
;462                                    ; защиты
;463             WRITE.V.WCHK=0D        ; запись по виртуальному адресу с проверкой защиты
;464                                    ; по записи
;465             IB.FILL=0E             ; чтение по виртуальному адресу с проверкой защиты
;466                                    ; по чтению и заполнение регистра предвыборки
;467                                    ; инструкций
;468             READ.V.RCHK.IFILL=0E   ; чтение по виртуальному адресу с проверкой защиты
;469                                    ; по чтению и заполнение регистра предвыборки
;470                                    ; инструкций
;471             WRITE.UBS.MAP=0F       ; запись в буфер трансляции общей шины
;472             OCTA.WRITE.P=10        ; запись восьмикратных слов данных по физическому
;473                                    ; адресу
;474             WRITE.TB.STEP=11         запись в две последовательные ячейки буфера
;475                                    ; трансляции
;476             READ.UBS.MAP=12          чтение из буфера трансляции общей шины
;477             READ.TB=13             ; чтение из буфера трансляции
;478             READ.MAINT.ADRS=14     ; выдача виртуального адреса и чтение его обратно
;479                                      в качестве данных (режим отладки)
;480             MAINT.ECC.DATA=15        проверка всех функций коррекции ошибки (ECC)
;481             READ.CSR=16              чтение регистра управления
;482             READ.CNTRL.REG=16      ; чтение регистра управления
;483             WRITE.CNTRL.REG=17     ; запись в регистр управления
;484             WRITE.CSR=17           ; запись в регистр управления
;485             OCTA.READ.P=18         ; чтение восьмикратных слов данных по физическому
;486                                    ; адресу
;487             READ.V.WCHK.LOCKED=19  ; чтение по виртуальному адресу и проверка защиты
;488                                    ; по записи с блокировкой
;489             OCTA.READ.V.RCHK=1A    ; чтение восьмикратных слов данных по виртуальному
;490                                    ; адресу с проверкой защиты по чтению
;491             READ.MAINT.BR.CHECK=1B; проверка функций ветвления
;492             ISSUE.BG=1C            ; выдача подтверждения для шины
;493             OCTA.WR.V.WCHK=1D      ; запись восьмикратных слов данных по
;494                                    ; виртуальному адресу с проверкой защиты по записи
;495             QUAD.READ.V.RCHK=1E    ; чтение четырехкратных слов данных по
;496                                    ; виртуальному адресу с проверкой защиты по чтению
```

```
;497             READ.MAINT.UBS=1F      ; выдача виртуального адреса на общую шину
;498                                    ; и чтение его в качестве данных (режим отладки)
;499     ;
;500     M.FUNC2/=<18:16>,.VALIDITY=<DT.VAL>
;501     M.FUNC1/=<8:7>,.VALIDITY=<DT.VAL>
;502     ;
;503     PAR/=<23>,.DEFAULT=<.PARITY[<OPC2/>,<OS/>,<JUMP.ADRS/>,<JCTL/>]>
;504     ;
```

```
;505     .PAGE "ОПРЕДЕЛЕНИЯ АДРЕСНЫХ ПОЛЕЙ МЕСТНОЙ ПАМЯТИ"
;506     ;
;507     ;  Это поле задает обращение к младшим 128 ячейкам LS
;508     ;
;509     D.ADRS/=<15:9>,.VALIDITY=<D.VAL>,.DEFAULT=0
;510     ;
;511             SAV.WR0=1                       ; память для WR0
;512             SAV.WR1=2                       ; память для WR1
;513             SAV.WR2=3                       ; память для WR2
;514             SAV.WR3=4                       ; память для WR3
;515             T5.SUB=5                        ; резервировано для общих подпрограмм
;516             T6.SUB=6                        ; резервировано для общих подпрограмм
;517             T7.SUB=7                        ; резервировано для общих подпрограмм
;518             TRANSFER.POINTER=7              ; указатель для программы переноса данных
;519             MEMORY.SIZE=7                   ; запоминание размера памяти в блоках по 256K байт
;520                                             ; после "прощупывания"
;521             FPA.FOUND=7                     ; место запоминания результата проверки
;522                                             ; наличия ускорителя плавающей запятой (FPA)
;523             UBE.FOUND=7                     ; место запоминания результата проверки
;524                                             ; наличия тестера шины (UBE)
;525             T8=8                            ; ячейка для временного хранения 8
;526             T9=9                            ; ячейка для временного хранения 9
;527             T10=0A                          ; ячейка для временного хранения 10
;528             T11=0B                          ; ячейка для временного хранения 11
;529             T12=0C                          ; ячейка для временного хранения 12
;530             T13=0D                          ; ячейка для временного хранения 13
;531             T14=0E                          ; ячейка для временного хранения 14
;532             T15=0F                          ; ячейка для временного хранения 15
;533             PC=10                           ; счетчик инструкций макроуровня
;534             WR.BACKUP=11                    ; резервная копия WR для MOV MEM.DATA TO WR[ ]
;535             MM.LASTADDR+1=12                ; место запоминания последнего адреса основной
;536                                             ; памяти плюс 1 (первый несуществующий адрес памяти)
;537             #FF=13                          ; маска
;538             #FFFF=14                        ; маска
;539             #FF000000=15                    ; маска
;540             #FFFFFF00=16                    ; маска
;541             CSR0.MASK=16                    ; маска
;542             CSR1.MASK=17                    ; маска (01003FFF)
;543             CSR2.MASK=18                    ; маска (7FFE3FFF)
;544             #FE7FFFFF=19                    ; маска
;545             UBS.SET=1A                      ; константа (7FF80000),используемая тестом адреса
;546                                             ; общей шины
;547             UBS.DATA=1B                     ; маска (7FFF8000),используемая в качестве маски
;548                                             ; ошибок для теста данных общей шины
;549             ERR.CON.NA=1E                   ; константа 800500 (16-ричная) с установленными
;550                                             ; битами 23,10,8 для загрузки регистра управления и
;551                                             ; состояния в начальных тестах
;552             ERR.CON=1F                      ; константа 500 (16-ричная). Биты 10 и 8
;553                                             ; установлены для загрузки регистра управления
;554                                             ; и состояния в начальных тестах
;555     ;
;556     ;  Следующие ячейки используются, главным образом, в качестве тестовых наборов
;557     ;  данных (1, перемещаемая в поле нулей)
;558     ;
;559             #1=20                           ; набор 00000001(H)
```

```
;560            #2=21                          ; набор 00000002
;561            #4=22                          ; набор 00000004
;562            #8=23                          ; набор 00000008
;563            #10=24                         ; набор 00000010
;564            #20=25                         ; набор 00000020
;565            #40=26                         ; набор 00000040
;566            #80=27                         ; набор 00000080
;567            #100=28                        ; набор 00000100
;568            #200=29                        ; набор 00000200
;569            #400=2A                        ; набор 00000400
;570            #800=2B                        ; набор 00000800
;571            #1000=2C                       ; набор 00001000
;572            #2000=2D                       ; набор 00002000
;573            #4000=2E                       ; набор 00004000
;574            #8000=2F                       ; набор 00008000
;575            #10000=30                      ; набор 00010000
;576            #20000=31                      ; набор 00020000
;577            #40000=32                      ; набор 00040000
;578            #80000=33                      ; набор 00080000
;579            #100000=34                     ; набор 00100000
;580            #200000=35                     ; набор 00200000
;581            #400000=36                     ; набор 00400000
;582            #800000=37                     ; набор 00800000
;583            #1000000=38                    ; набор 01000000
;584            #2000000=39                    ; набор 02000000
;585            #4000000=3A                    ; набор 04000000
;586            #8000000=3B                    ; набор 08000000
;587            #10000000=3C                   ; набор 10000000
;588            #20000000=3D                   ; набор 20000000
;589            #40000000=3E                   ; набор 40000000
;590            #80000000=3F                   ; набор 80000000
;591            BIT0=20                        ; набор 00000001
;592            BIT1=21                        ; набор 00000002
;593            BIT2=22                        ; набор 00000004
;594            BIT3=23                        ; набор 00000008
;595            BIT4=24                        ; набор 00000010
;596            BIT5=25                        ; набор 00000020
;597            BIT6=26                        ; набор 00000040
;598            BIT7=27                        ; набор 00000080
;599            BIT8=28                        ; набор 00000100
;600            BIT9=29                        ; набор 00000200
;601            BIT10=2A                       ; набор 00000400
;602            BIT11=2B                       ; набор 00000800
;603            BIT12=2C                       ; набор 00001000
;604            BIT13=2D                       ; набор 00002000
;605            BIT14=2E                       ; набор 00004000
;606            BIT15=2F                       ; набор 00008000
;607            BIT16=30                       ; набор 00010000
;608            BIT17=31                       ; набор 00020000
;609            BIT18=32                       ; набор 00040000
;610            BIT19=33                       ; набор 00080000
;611            BIT20=34                       ; набор 00100000
;612            BIT21=35                       ; набор 00200000
;613            BIT22=36                       ; набор 00400000
;614            BIT23=37                       ; набор 00800000
```

```
;615             BIT24=38                        ; набор 01000000
;616             BIT25=39                        ; набор 02000000
;617             BIT26=3A                        ; набор 04000000
;618             BIT27=3B                        ; набор 08000000
;619             BIT28=3C                        ; набор 10000000
;620             BIT29=3D                        ; набор 20000000
;621             BIT30=3E                        ; набор 40000000
;622             BIT31=3F                        ; набор 80000000
;623     ;
;624     ;  Мнемоника адресов CSR контроллера памяти
;625     ;
;626             CSR0=4E                 ; для доступа к CSR0 все биты очищены
;627             CSR1=22                 ; для доступа к CSR1 установлен бит 2
;628             CSR2=23                 ; для доступа к CSR2 установлен бит 3
;629     ;
;630     ;  Биты слова управления и состояния (информация для микромонитора во время
;631     ;  выполнения тестов). Слово управления и состояния доступно микромонитору
;632     ;  по адресу LS 40
;633     ;
;634             PRINT.UBE=26            ; печать,имеется или нет тестер шины UBE (бит 6)
;635                                     ; (индикатор в LS7)
;636             PRINT.R80=27            ; печать,имеется или нет R80 и на каком
;637                                     ; устройстве (бит 7).Индикатор в LS7. Номер
;638                                     ; накопителя в LS6
;639             ERROR=28                ; индикация ошибки теста (бит 8)
;640             SET.PA.ERR=29           ; указывает консольному процессору,что следует
;641                                     ; генерировать ошибку паритета по адресу
;642                                     ; управляющей памяти WCS,указанному в LS7 (бит 9)
;643             CONSOLE.LOE=2A          ; указывает,что консольный процессор выполняет
;644                                     ; зацикливание при ошибке (бит 10) (для начальных
;645                                     ; тестов)
;646             PRINT.MEM.SIZE=2B       ; печать размера памяти в блоках по 256 K
;647                                     ; (бит 11) (размер в LS7)
;648             PRINT.FPA=2C            ; печать, имеется или нет ускоритель плавающей
;649                                     ; запятой FPA (бит 12) (индикатор в LS7)
;650             XFER.DATA=2D            ; указывает перенос данных в местную (LS) или
;651                                     ; основную (MM) память (бит 13)
;652             EOS=2E                  ; конец сегмента (бит 14)
;653             BEGIN.TEST=2F           ; начало теста (бит 15)
;654             INTERUPT.EN=30          ; разрешение прерывания или сигнала, заданного
;655                                     ; битами от 17 до 22 и от 28 до 30 (бит 16)
;656             CONSOLE.HALT=31         ; прерывание по останову от консоли (бит 17)
;657             PWR.FAIL=32             ; прерывание по аварии питания (бит 18)
;658             INTERVAL.TIM=33         ; прерывание от интервального таймера (бит 19)
;659             CONSOLE.ATTN=34         ; прерывание по биту "внимание" (ATTN) консоли
;660                                     ; (бит 20)
;661             CONSOLE.ACK=35          ; прерывание по биту "подтверждение" (ACK) консоли
;662                                     ; (бит 21)
;663             DISABLE.MEM.REF=36      ; запрет обращений к памяти (бит 22)
;664             CINIT=3C                ; сигнал общей шины INIT для контроллера памяти
;665                                     ; (бит 28)
;666             UBS.DCLO=3D             ; индикация общей шины DC LO для контроллера
;667                                     ; памяти (бит 29)
;668             UBS.BBSY=3E             ; индикация общей шины BUS BUSY для контроллера
;669                                     ; памяти (бит 30)
```

```
;670             NA.EXP.REC=37           ; печать N/A под EXP и REC (бит 23)
;671             OTHER.DATA=38           ; печать значений под OTHER (бит 24)
;672             CONSOLE.LOOP=39         ; консоль выполняет зацикливание в соответствии со
;673                                     ; счетчиком циклов в LS (бит 25)
;674             PRINT.CRD=3A            ; печать числа ошибок в одиночных битах (бит 26)
;675             CLR.PA.ERR=3B           ; указывает консоли, что следует очистить ошибку
;676                                       паритета в управляющей памяти WCS по адресу,
;677                                     ; указанному в LS7 (бит 27)
;678             PRINT.IDC=3F            ; печать,имеется или нет встроенный контроллер
;679                                     ; дисков IDC (бит 31)(индикатор в LS7)
;680
;681     ;   Коды модулей
;682
;683             WCS=20                    код модуля для платы управляющей памяти WCS
;684                                       (установлен бит 0)
;685             CPU=21                  ; код модуля для платы путей данных DAP
;686                                     ; (установлен бит 1)
;687             MCT=22                  ; код модуля для платы контроллера памяти MCT
;688                                     ; (установлен бит 2)
;689             FPA=23                  ; код модуля для платы ускорителя плавающей
;690                                     ; запятой FPA (установлен бит 3)
;691             ARRAY=24                ; код модуля для платы ОЗУ (установлен бит 4)
;692             IDC=25                  ; код модуля для платы встроенного контроллера
;693                                     ; дисков IDC (установлен бит 5)
;694
;695     ;   Биты ошибок CSR1 контроллера памяти
;696
;697             VALID.ERR=2E            ; не установлен бит действительности данных (VALID),
;698                                     ; если 1 (бит 14)
;699             TB.PAR.ERR=2F           ; ошибка паритета буфера трансляции (бит 15)
;700             NXM=30                  ; ошибка - несуществующая память (бит 16)
;701             UB.BSY=31               ; общая шина занята (бит 17)
;702             ADP.REG=32              ; выбор регистра адаптера общей шины (бит 18)
;703             WR.ACROSS.PG=33         ; ошибка записи за пределами страницы (бит 19)
;704             OP.ERR=34               ; ошибка операции (бит 20)
;705             TB.MISS=35              ; промах в буфере трансляции (транслированный
;706                                     ; виртуальный адрес в буфере отсутствует)
;707                                     ; (бит 21)
;708             ACCESS.REFUSED=36       ; ошибка защиты при обращении (бит 22)
;709             MODIFY.REFUSED=37       ; ошибка защиты при записи (бит 23)
;710             CRD=3E                  ; исправлена ошибка в одиночном бите (бит 30)
;711             RDS=3F                  ; неисправимая ошибка данных (бит 31)
;712     ;
;713     ;   Управляющие биты CSR1 контроллера памяти
;714     ;
;715             ECC.DIS=39              ; бит запрета коррекции (ECC) в CSR1 (бит 25)
;716             DIAG.CHK=3A             ; бит диагностического контроля CSR1 (бит 26)
;717             MME=3B                  ; бит разрешения диспетчера памяти в CSR1
;718                                     ; (бит 27)
;719             INH.CRD=3C              ; бит запрета сообщения о корректируемых ошибках
;720                                     ; данных (CRD) (бит 28)
;721             TB.PAR.DIAG=3D          ; бит диагностирования паритета буфера трансляции
;722                                     ; (принудительная ошибка паритета TB) (бит 29)
;723     ;
;724     ;   Биты регистров управления тестера шины (UBE)
```

```
;725
;726             Регистр управления 1
;727
 728             GO=20                   ; запуск тестера шины (UBE) (бит 0)
;729             BR4=21                  ; выдача запроса шины на уровне 4 (бит 1)
;730             BR5=22                  ; выдача запроса шины на уровне 5 (бит 2)
;731             BR6=23                  ; выдача запроса шины на уровне 6 (бит 3)
;732             BR7=24                  ; выдача запроса шины на уровне 7 (бит 4)
;733             NPR=25                  ; выдача запроса прямого доступа (NPR) (бит 5)
;734             INT.ODNE=26             ; прерывание по завершению (при переполнении
;735                                     ; счетчика циклов) (бит 6)
;736             RDY=27                  ; бит "готово", устанавливаемый, если есть
;737                                     ; готовность начать функционирование (бит 7)
;738             C00=28                  ; бит C0 операции шины (бит 8)
 739             C01=29                  ; бит C1 операции шины (бит 9)
 740             FUNA=2A                 ; бит функции A тестера шины (бит 10)
;741             FUNB=2B                 ; бит функции B тестера шины (бит 11)
;742             CC+DAT=2C               ; данные из счетчика циклов, если установлен,
 743                                     ; и-из регистра данных, если очищен
;744             INH.DATIP.ROL=2D        ; запрет циклического сдвига при чтении с паузой
;745                                     ; (DATIP) (бит 13)
;746             DATOB.ON.DATIP=2E       ; выполнение записи байта (DATOB) после цикла
;747                                     ; чтения с паузой (DATIP) (бит 14)
;748             ERR=2F                  ; ошибка общей шины или тестера (бит 15)
;749
 750         Регистр управления 2
;751
;752             EXT.MEM0=20             ; бит 0 расширения памяти (бит 0)
;753             EXT.MEM1=21             ; бит 1 расширения памяти (бит 1)
;754             INH.INC.ADDR&CC=22      ; запрет наращивания счетчика циклов и регистра
;755                                       адреса (бит 2)
;756             INH.SACK=23             ; запрет выдачи BUS SACK по BUS GRANT (бит 3)
;757             PWR.DWN.SEQ=24          ; установка ACLO (бит 4)
;758             WNG.CNT.BCK=25          ; не получен BUS GRANT в ответ на запрос
;759                                     ; высшего приоритета (бит 5)
;760             MAX.LATE.ERR=26         ; превышено время задержки для NPR и BR (бит 6)
;761             NO.SACK.ERR=27          ; центральный процессор не зафиксировал таймаута
;762                                     ; при отсутствии SACK (бит 7)
;763             NO.SSYN.ERR=28          ; после выдачи MSYN не получен SSYN (бит 8)
;764             WRG.A.LINES=29          ; ошибка адреса в переданном или принятом
;765                                     ; адресе (бит 9)
;766             NO.GNT+NOT.ONE.GNT=2A   ; отсутствует GRANT или более одного сигнала
;767                                     ; GRANT после запроса (бит 10)
;768             INTR.SSYN.ERR=2B        ; не получен SSYN после прерывания шины (бит 11)
;769             ENB.PB=2C               ; разрешение для бита паритета B (бит 12)
;770             CCOVF=2D                ; переполнение счетчика циклов (бит 13)
;771             TM.DLY=2E               ; запрос для шины каждые 10 мкс (бит 14)
;772     ;
;773     ;   Мнемоника BG6
;774     ;
;775             BG6=23                  ; для адреса BG6 установлен бит 3
;776     ;
;777     ;   Информация об ошибках и управляющая информация для микромонитора
;778     ;
;779             CONTROL.STATUS=40       ; слово управления и состояния
```

```
;780            ERROR.NUMBER=41         ; номер ошибки в текущем тексте
;781            EXPECTED.DATA=42        ; ожидаемый результат текущего теста
;782            RECEIVED.DATA=43        ; действительный результат текущего теста
;783            ADDRESS.DATA=44         ; другие актуальные данные
;784            ERROR.MASK=45           ; биты,подлежащие проверке в результате
;785            MODULE.NUM=46           ; место хранения номера модуля (кодированного)
;786            LOOP.CNT=47             ; место запоминания счета циклов для управления
;787                                    ; зацикливанием из консольного процессора
;788            ERROR.CONTROL=48        ; место запоминания информации управления
;789                                    ; ошибками (зацикливание по ошибке и т.д.)
;790            LOE=20                  ; если установлен,зацикливание на ошибке (бит 0)
;791            NER=21                  ; если установлен,не выдаются сообщения об
;792                                    ; ошибках (бит 1)
;793            BELL=22                 ; если установлен,звуковой сигнал при ошибке
;794                                    ; (бит 2)
;795            HOE=23                  ; если установлен, останов при ошибке (бит 3)
;796            SER=24                  ; если установлен, разрешение сообщений об
;797                                    ; исправимых ошибках данных
;798            APT.PRESENT=25          ; если установлен, имеется APT (бит 5)
;799            LOOP.COMMAND=26         ; если установлен, зацикливание напечатанной
;800                                    ; команды (бит 6)
;801            SPECIAL=27              ; специальный бит для зацикливания в тестах
;802                                    ; "прощупывания" (бит 7)
;803            APT.PARAM=49            ; регистры текущих параметров APT (размер памяти,
;804                                    ; конфигурация аппаратуры,конфигурация програм-
;805                                    ; много обеспечения в байтах 0,1 и 2
;806                                    ; соответственно. Байт 3 для использования в
;807                                    ; будущем)
;808            APT.RESERVED=4A         ; резервировано для будущего использования в APT
;809    ;
;810    ;  Разные константы и ячейки для запоминания
;811    ;
;812            #AAAAAAAA=4C            ; константа
;813            ALTER.ODD=4C            ; константа
;814            #55555555=4D            ; константа
;815            ALTER.EVEN=4D           ; константа
;816            #0=4E                   ; константа
;817            ZERO=4E                 ; константа
;818            #FFFFFFFF=4F            ; константа
;819            ONES=4F                 ; константа
;820            PREVIOUS.ERROR=50       ; номер последней ошибки
;821            DATA.1=51               ; место запоминания данных для ускорителя FPA
;822            DATA.2=52               ; место запоминания данных для ускорителя FPA
;823            DATA.3=53               ; место запоминания данных для ускорителя FPA
;824            FIRST.FLT=54            ; место запоминания данных для ускорителя FPA
;825            SEC.FLT=55              ; место запоминания данных для ускорителя FPA
;826            THIRD.FLT=56            ; место запоминания данных для ускорителя FPA
;827            T57=57                  ; ячейка для временного хранения 57
;828            T58=58                  ; ячейка для временного хранения 58
;829            T59=59                  ; ячейка для временного хранения 59
;830            BEDB=5A                 ; регистр буфера данных тестера шины (UBE)
;831            BECC=5B                 ; регистр счетчика циклов тестера шины (UBE)
;832            BEBA=5C                 ; регистр адреса тестера шины (UBE)
;833            BECR1=5D                ; регистр управления 1 тестера шины (UBE)
;834            CLEAR.ERR.ADDR=5E       ; адрес регистра очистки ошибок тестера шины
```

```
;835                                    ; (UBE)
;836            BECR2=5F                ; регистр управления 2 тестера шины (UBE)
;837            #3(H)=60                ; константа
;838            #12(H)=61               ; константа
;839            #33333333=62            ; константа
;840            #0F0F0F0F=63            ; константа
;841            #00FF00FF=64            ; константа
;842            #162(H)=65              ; константа для указателя переноса данных в LS
;843            BR7.IDENT=66            ; идентификатор BR7 (1010 (двоичн.) в битах 5-2)
;844            BG7=67                  ; адрес BG7 (установлены биты 2 и 3)
;845    ;
;846    ;   Ячейки временного хранения для тестов центрального процессора
;847    ;
;848            L30=30                  ; ячейка временного хранения LS 30 (после
;849                                    ; использования восстанавливается значением
;850                                    ; 10000)
;851            L31=31                  ; ячейка временного хранения LS 31 (после
;852                                    ; использования восстанавливается значением
;853                                    ; 20000)
;854            L53=53                  ; ячейка временного хранения LS 53
;855            L73=73                  ; ячейка временного хранения LS 73
;856    ;
;857    ;   Адреса регистров
;858    ;
;859            CONTROL.OS=78           ; аппаратный индекс для управляющих слов (LS 40-4F)
;860            SHIFT.OS(3-0)=79        ; аппаратный индекс на 16 ячеек для наборов
;861                                    ; тестовых данных со сдвигом (LS 20-2F)
;862            SHIFT.OS(4-0)=7B        ; аппаратный индекс на 32 ячейки для наборов
;863                                    ; тестовых данных со сдвигом (LS 20-3F)
;864            OS=7C                   ; регистр OS
;865            DEC.CON=7D              ; десятичные переносы
;866            SIZE=7E                 ; регистр размера
;867            MDT=7E                  ; размер данных для обращений к памяти
;868            INTERRUPT.VEC=7E        ; аппаратный вектор прерывания
;869    ;
;870    ;   Это слово содержит аппаратные коды условий (CC). Они могут считываться или
;871    ;   записываться сюда. На другие биты PSL не отражается.
;872    ;
;873            PSL.CC=7F                   коды условий PSL
;874    ;
;875    ;   Это поле допускает обращение ко всем 256 ячейкам LS
;876    ;
;877    XD.ADRS/=<16:9>,.VALIDITY=<XD.VAL>
;878    ;
;879            SAV.WR0=1                   ; память для WR0
;880            SAV.WR1=2                   ; память для WR1
;881            SAV.WR2=3                   ; память для WR2
;882            SAV.WR3=4                   ; память для WR3
;883            T5.SUB=5                    ; резервировано для общих подпрограмм
;884            T6.SUB=6                    ; резервировано для общих подпрограмм
;885            T7.SUB=7                    ; резервировано для общих подпрограмм
;886            TRANSFER.POINTER=7          ; указатель для программы переноса данных
;887            MEMORY.SIZE=7               ; запоминание размера памяти в блоках по 256K байт
;888                                        ; после "прощупывания"
;889            FPA.FOUND=7                 ; место запоминания результата проверки
```

```
;890                                    ; наличия ускорителя плавающей запятой (FPA)
;891         UBE.FOUND=7                ; место запоминания результата проверки
;892                                    ; наличия тестера шины (UBE)
;893         T8=8                       ; ячейка для временного хранения 8
;894         T9=9                       ; ячейка для временного хранения 9
;895         T10=0A                     ; ячейка для временного хранения 10
;896         T11=0B                     ; ячейка для временного хранения 11
;897         T12=0C                     ; ячейка для временного хранения 12
;898         T13=0D                     ; ячейка для временного хранения 13
;899         T14=0E                     ; ячейка для временного хранения 14
;900         T15=0F                     ; ячейка для временного хранения 15
;901         PC=10                      ; счетчик инструкций макроуровня
;902         WR.BACKUP=11               ; резервная копия WR для MOV MEM.DATA TO WR[ ]
;903         MM.LASTADDR+1=12           ; место запоминания последнего адреса основной
;904                                    ; памяти плюс 1 (первый несуществующий адрес памяти)
;905         #FF=13                     ; маска
;906         #FFFF=14                   ; маска
;907         #FF000000=15               ; маска
;908         #FFFFFF00=16               ; маска
;909         CSR0.MASK=16               ; маска
;910         CSR1.MASK=17               ; маска (01003FFF)
;911         CSR2.MASK=18               ; маска (7FFE3FFF)
;912         #FE7FFFFF=19               ; маска
;913         UBS.SET=1A                 ; константа (7FF80000),используемая тестом адреса
;914                                    ; общей шины
;915         UBS.DATA=18                ; маска (7FFF8000),используемая в качестве маски
;916                                    ; ошибок для теста данных общей шины
;917         ERR.CON.NA=1E              ; константа 800500 (16-ричная) с установленными
;918                                    ; битами 23,10,8 для загрузки регистра управления и
;919                                    ; состояния в начальных тестах
;920         ERR.CON=1F                 ; константа 500 (16-ричная). биты 10 и 8
;921                                    ; установлены для загрузки регистра управления
;922                                    ; и состояния в начальных тестах
;923     ;
;924     ;  Следующие ячейки используются, главным образом, в качестве тестовых наборов
;925     ;  данных (1, перемещаемая в поле нулей)
;926     ;
;927         #1=20                      ; набор 00000001(H)
;928         #2=21                      ; набор 00000002
;929         #4=22                      ; набор 00000004
;930         #8=23                      ; набор 00000008
;931         #10=24                     ; набор 00000010
;932         #20=25                     ; набор 00000020
;933         #40=26                     ; набор 00000040
;934         #80=27                     ; набор 00000080
;935         #100=28                    ; набор 00000100
;936         #200=29                    ; набор 00000200
;937         #400=2A                    ; набор 00000400
;938         #800=2B                    ; набор 00000800
;939         #1000=2C                   ; набор 00001000
;940         #2000=2D                   ; набор 00002000
;941         #4000=2E                   ; набор 00004000
;942         #8000=2F                   ; набор 00008000
;943         #10000=30                  ; набор 00010000
;944         #20000=31                  ; набор 00020000
```

```
;945            #40000=32                       ; набор 00040000
;946            #80000=33                       ; набор 00080000
;947            #100000=34                      ; набор 00100000
;948            #200000=35                      ; набор 00200000
;949            #400000=36                      ; набор 00400000
;950            #800000=37                      ; набор 00800000
;951            #1000000=38                     ; набор 01000000
;952            #2000000=39                     ; набор 02000000
;953            #4000000=3A                     ; набор 04000000
;954            #8000000=3B                     ; набор 08000000
;955            #10000000=3C                    ; набор 10000000
;956            #20000000=3D                    ; набор 20000000
;957            #40000000=3E                    ; набор 40000000
;958            #80000000=3F                    ; набор 80000000
;959            BIT0=20                         ; набор 00000001
;960            BIT1=21                         ; набор 00000002
;961            BIT2=22                         ; набор 00000004
;962            BIT3=23                         ; набор 00000008
;963            BIT4=24                         ; набор 00000010
;964            BIT5=25                         ; набор 00000020
;965            BIT6=26                         ; набор 00000040
;966            BIT7=27                         ; набор 00000080
;967            BIT8=28                         ; набор 00000100
;968            BIT9=29                         ; набор 00000200
;969            BIT10=2A                        ; набор 00000400
;970            BIT11=2B                        ; набор 00000800
;971            BIT12=2C                        ; набор 00001000
;972            BIT13=2D                        ; набор 00002000
;973            BIT14=2E                        ; набор 00004000
;974            BIT15=2F                        ; набор 00008000
;975            BIT16=30                        ; набор 00010000
;976            BIT17=31                        ; набор 00020000
;977            BIT18=32                        ; набор 00040000
;978            BIT19=33                        ; набор 00080000
;979            BIT20=34                        ; набор 00100000
;980            BIT21=35                        ; набор 00200000
;981            BIT22=36                        ; набор 00400000
;982            BIT23=37                        ; набор 00800000
;983            BIT24=38                        ; набор 01000000
;984            BIT25=39                        ; набор 02000000
;985            BIT26=3A                        ; набор 04000000
;986            BIT27=3B                        ; набор 08000000
;987            BIT28=3C                        ; набор 10000000
;988            BIT29=3D                        ; набор 20000000
;989            BIT30=3E                        ; набор 40000000
;990            BIT31=3F                        ; набор 80000000
;991
;992        Мнемоника адресов CSR контроллера памяти
;993
;994            CSR0=4E                         ; для доступа к CSR0 все биты очищены
;995            CSR1=22                         ; для доступа к CSR1 установлен бит 2
;996            CSR2=23                         ; для доступа к CSR2 установлен бит 3
;997    ;
;998    ;   Биты слова управления и состояния (информация для микромонитора во время
;999    ;   выполнения тестов). Слово управления и состояния доступно микромонитору
```

```
;1000    ;  по адресу LS 40
;1001
;1002            PRINT.UBE=26            ; печать,имеется или нет тестер шины UBE (бит 6)
;1003                                    ; (индикатор в LS7)
;1004            PRINT.RB0=27            ; печать,имеется или нет RB0 и на каком
;1005                                    ; устройстве (бит 7).индикатор в LS7. номер
;1006                                    ; накопителя в LS6
;1007            ERROR=28                ; индикация ошибки теста (бит 8)
;1008            SET.PA.ERR=29           ; указывает консольному процессору,что следует
;1009                                    ; генерировать ошибку паритета по адресу
;1010                                    ; управляющей памяти WCS,указанному в LS7 (бит 9)
;1011            CONSOLE.LOE=2A          ; указывает,что консольный процессор выполняет
;1012                                    ; зацикливание при ошибке (бит 10) (для начальных
;1013                                    ; тестов)
;1014            PRINT.MEM.SIZE=2B       ; печать размера памяти в блоках по 256 К
;1015                                    ; (бит 11) (размер в LS7)
;1016            PRINT.FPA=2C            ; печать, имеется или нет ускоритель плавающей
;1017                                    ; запятой FPA (бит 12) (индикатор в LS7)
 1018            XFER.DATA=2D            ; указывает перенос данных в местную (LS) или
;1019                                    ; основную (мм) память (бит 13)
;1020            EOS=2E                  ; конец сегмента (бит 14)
 1021            BEGIN.TEST=2F           ; начало теста (бит 15)
 1022            INTERUPT.EN=30          ; разрешение прерывания или сигнала, заданного
;1023                                    ; битами от 17 до 22 и от 28 до 30 (бит 16)
 1024            CONSOLE.HALT=31         ; прерывание по останову от консоли (бит 17)
;1025            PWR.FAIL=32             ; прерывание по аварии питания (бит 18)
;1026            INTERVAL.TIM=33         ; прерывание от интервального таймера (бит 19)
;1027            CONSOLE.ATTN=34         ; прерывание по биту "внимание" (ATTN) консоли
;1028                                    ; (бит 20)
;1029            CONSOLE.ACK=35          ; прерывание по биту "подтверждение" (ACK) консоли
;1030                                    ; (бит 21)
;1031            DISABLE.MEM.REF=36      ; запрет обращений к памяти (бит 22)
;1032            CINIT=3C                ; сигнал общей шины INIT для контроллера памяти
;1033                                    ; (бит 28)
;1034            UBS.DCLO=3D             ; индикация общей шины DC LO для контроллера
;1035                                    ; памяти (бит 29)
;1036            UBS.BBSY=3E             ; индикация общей шины BUS BUSY для контроллера
;1037                                    ; памяти (бит 30)
;1038            NA.EXP.REC=37           ; печать N/A под EXP и REC (бит 23)
;1039            OTHER.DATA=38           ; печать значений под OTHER (бит 24)
;1040            CONSOLE.LOOP=39         ; консоль выполняет зацикливание в соответствии со
;1041                                    ; счетчиком циклов в LS (бит 25)
;1042            PRINT.CRD=3A            ; печать числа ошибок в одиночных битах (бит 26)
;1043            CLR.PA.ERR=3B           ; указывает консоли, что следует очистить ошибку
;1044                                    ; паритета в управляющей памяти WCS по адресу,
;1045                                    ; указанному в LS7 (бит 27)
;1046            PRINT.IDC=3F            ; печать,имеется или нет встроенный контроллер
;1047                                    ; дисков IDC (бит 31)(индикатор в LS7)
;1048    ;
;1049    ;  Коды модулей
;1050    ;
;1051            WCS=20                  ; код модуля для платы управляющей памяти WCS
;1052                                    ; (установлен бит 0)
;1053            CPU=21                  ; код модуля для платы путей данных DAP
;1054                                    ; (установлен бит 1)
```

```
;1055            MCT=22                  ; код модуля для платы контроллера памяти MCT
;1056                                    ; (установлен бит 2)
;1057            FPA=23                  ; код модуля для платы ускорителя плавающей
;1058                                    ; запятой FPA (установлен бит 3)
;1059            ARRAY=24                ; код модуля для платы ОЗУ (установлен бит 4)
;1060            IDC=25                  ; код модуля для платы встроенного контроллера
;1061                                    ; дисков IDC (установлен бит 5)
;1062    ;
;1063    ;   Биты ошибок CSR1 контроллера памяти
;1064    ;
;1065            VALID.ERR=2E            ; не установлен бит действительности данных (VALID),
;1066                                    ; если 1 (бит 14)
;1067            TB.PAR.ERR=2F           ; ошибка паритета буфера трансляции (бит 15)
;1068            NXM=30                  ; ошибка - несуществующая память (бит 16)
;1069            UB.BSY=31               ; общая шина занята (бит 17)
;1070            ADP.REG=32              ; выбор регистра адаптера общей шины (бит 18)
;1071            WR.ACROSS.PG=33         ; ошибка записи за пределами страницы (бит 19)
;1072            OP.ERR=34               ; ошибка операции (бит 20)
;1073            TB.MISS=35              ; промах в буфере трансляции (транслированный
;1074                                    ; виртуальный адрес в буфере отсутствует)
;1075                                    ; (бит 21)
;1076            ACCESS.REFUSED=36       ; ошибка защиты при обращении (бит 22)
;1077            MODIFY.REFUSED=37       ; ошибка защиты при записи (бит 23)
;1078            CRD=3E                  ; исправлена ошибка в одиночном бите (бит 30)
;1079            RDS=3F                  ; неисправимая ошибка данных (бит 31)
;1080    ;
;1081    ;   Управляющие биты CSR1 контроллера памяти
;1082    ;
;1083            ECC.DIS=39              ; бит запрета коррекции (ECC) в CSR1 (бит 25)
;1084            DIAG.CHK=3A             ; бит диагностического контроля CSR1 (бит 26)
;1085            MME=3B                  ; бит разрешения диспетчера памяти в CSR1
;1086                                    ; (бит 27)
;1087            INH.CRD=3C              ; бит запрета сообщения о корректируемых ошибках
;1088                                    ; данных (CRD) (бит 28)
;1089            TB.PAR.DIAG=3D          ; бит диагностирования паритета буфера трансляции
;1090                                    ; (принудительная ошибка паритета TB) (бит 29)
;1091    ;
;1092    ;   Биты регистров управления тестера шины (UBE)
;1093    ;
;1094    ;       Регистр управления 1
;1095    ;
;1096            GO=20                   ; запуск тестера шины (UBE) (бит 0)
;1097            BR4=21                  ; выдача запроса шины на уровне 4 (бит 1)
;1098            BR5=22                  ; выдача запроса шины на уровне 5 (бит 2)
;1099            BR6=23                  ; выдача запроса шины на уровне 6 (бит 3)
;1100            BR7=24                  ; выдача запроса шины на уровне 7 (бит 4)
;1101            NPR=25                  ; выдача запроса прямого доступа (NPR) (бит 5)
;1102            INT.ODNE=26             ; прерывание по завершению (при переполнении
;1103                                    ; счетчика циклов) (бит 6)
;1104            RDY=27                  ; бит "готово", устанавливаемый, если есть
;1105                                    ; готовность начать функционирование (бит 7)
;1106            C00=28                  ; бит C0 операции шины (бит 8)
;1107            C01=29                  ; бит C1 операции шины (бит 9)
;1108            FUNA=2A                 ; бит функции A тестера шины (бит 10)
;1109            FUNB=2B                 ; бит функции B тестера шины (бит 11)
```

```
;1110            CC+DAT=2C                ; данные из счетчика циклов, если установлен,
;1111                                     ; и из регистра данных, если очищен
;1112            INH.DATIP.ROL=2D         ; запрет циклического сдвига при чтении с паузой
;1113                                     ; (DATIP) (бит 13)
;1114            DATOB.ON.DATIP=2E        ; выполнение записи байта (DATOB) после цикла
;1115                                     ; чтения с паузой (DATIP) (бит 14)
;1116            ERR=2F                   ; ошибка общей шины или тестера (бит 15)
;1117
;1118          Регистр управления 2
;1119
;1120            EXT.MEM0=20              ; бит 0 расширения памяти (бит 0)
;1121            EXT.MEM1=21              ; бит 1 расширения памяти (бит 1)
;1122            INH.INC.ADDR&CC=22       ; запрет наращивания счетчика циклов и регистра
;1123                                     ; адреса (бит 2)
;1124            INH.SACK=23              ; запрет выдачи BUS SACK по BUS GRANT (бит 3)
;1125            PWR.DWN.SEQ=24           ; установка ACLO (бит 4)
;1126            WNG.GNT.BCK=25           ; не получен BUS GRANT в ответ на запрос
;1127                                     ; высшего приоритета (бит 5)
;1128            MAX.LATE.ERR=26          ; превышено время задержки для NPR и BR (бит 6)
;1129            NO.SACK.ERR=27           ; центральный процессор не зафиксировал таймаута
;1130                                     ; при отсутствии SACK (бит 7)
;1131            NO.SSYN.ERR=28           ; после выдачи MSYN не получен SSYN (бит 8)
;1132            WRG.A.LINES=29           ; ошибка адреса в переданном или принятом
;1133                                     ; адресе (бит 9)
;1134            NO.GNT+NOT.ONE.GNT=2A    ; отсутствует GRANT или более одного сигнала
;1135                                     ; GRANT после запроса (бит 10)
;1136            INTR.SSYN.ERR=2B         ; не получен SSYN после прерывания шины (бит 11)
;1137            ENB.PB=2C                ; разрешение для бита паритета B (бит 12)
;1138            CCOVF=2D                 ; переполнение счетчика циклов (бит 13)
;1139            TM.DLY=2E                ; запрос для шины каждые 10 мкс (бит 14)
;1140    ;
;1141    ;   Мнемоника BG6
;1142    ;
;1143            BG6=23                   ; для адреса BG6 установлен бит 3
;1144    ;
;1145    ;   Информация об ошибках и управляющая информация для микромонитора
;1146    ;
;1147            CONTROL.STATUS=40        ; слово управления и состояния
;1148            ERROR.NUMBER=41          ; номер ошибки в текущем тексте
;1149            EXPECTED.DATA=42         ; ожидаемый результат текущего теста
;1150            RECEIVED.DATA=43         ; действительный результат текущего теста
;1151            ADDRESS.DATA=44          ; другие актуальные данные
;1152            ERROR.MASK=45            ; биты, подлежащие проверке в результате
;1153            MODULE.NUM=46            ; место хранения номера модуля (кодированного)
;1154            LOOP.CNT=47              ; место запоминания счета циклов для управления
;1155                                     ; зацикливанием из консольного процессора
;1156            ERROR.CONTROL=4B         ; место запоминания информации управления
;1157                                     ; ошибками (зацикливание по ошибке и т.д.)
;1158            LOE=20                   ; если установлен, зацикливание на ошибке (бит 0)
;1159            NER=21                   ; если установлен, не выдаются сообщения об
;1160                                     ; ошибках (бит 1)
;1161            BELL=22                  ; если установлен, звуковой сигнал при ошибке
;1162                                     ; (бит 2)
;1163            HOE=23                   ; если установлен, останов при ошибке (бит 3)
;1164            SER=24                   ; если установлен, разрешение сообщений об
```

```
;1165                                   ; исправимых ошибках данных
;1166          APT.PRESENT=25           ; если установлен, имеется APT (бит 5)
;1167          LOOP.COMMAND=26          ; если установлен, зацикливание напечатанной
;1168                                   ; команды (бит 6)
;1169          SPECIAL=27               ; специальный бит для зацикливания в тестах
;1170                                   ; "прощупывания" (бит 7)
;1171          APT.PARAM=49             ; регистры текущих параметров APT (размер памяти,
;1172                                   ; конфигурация аппаратуры,конфигурация програм-
;1173                                   ; много обеспечения в байтах 0,1 и 2
;1174                                   ; соответственно. байт 3 для использования в
;1175                                   ; будущем)
;1176          APT.RESERVED=4A          ; резервировано для будущего использования в APT
;1177
;1178      Разные константы и ячейки для запоминания
;1179
;1180          #AAAAAAAA=4C             ; константа
;1181          ALTER.ODD=4C             ; константа
;1182          #55555555=4D             ; константа
;1183          ALTER.EVEN=4D            ; константа
;1184          #0=4E                    ; константа
;1185          ZERO=4E                  ; константа
;1186          #FFFFFFFF=4F             ; константа
;1187          ONES=4F                  ; константа
;1188          PREVIOUS.ERROR=50        ; номер последней ошибки
;1189          DATA.1=51                ; место запоминания данных для ускорителя FPA
;1190          DATA.2=52                ; место запоминания данных для ускорителя FPA
;1191          DATA.3=53                ; место запоминания данных для ускорителя FPA
;1192          FIRST.FLT=54             ; место запоминания данных для ускорителя FPA
;1193          SEC.FLT=55               ; место запоминания данных для ускорителя FPA
;1194          THIRD.FLT=56             ; место запоминания данных для ускорителя FPA
;1195          T57=57                   ; ячейка для временного хранения 57
;1196          T58=58                   ; ячейка для временного хранения 58
;1197          T59=59                   ; ячейка для временного хранения 59
;1198          BEDB=5A                  ; регистр буфера данных тестера шины (UBE)
;1199          BECC=5B                  ; регистр счетчика циклов тестера шины (UBE)
;1200          BEBA=5C                  ; регистр адреса тестера шины (UBE)
;1201          BECR1=5D                 ; регистр управления 1 тестера шины (UBE)
;1202          CLEAR.ERR.ADDR=5E        ; адрес регистра очистки ошибок тестера шины
;1203                                   ; (UBE)
;1204          BECR2=5F                 ; регистр управления 2 тестера шины (UBE)
;1205          #3(H)=60                 ; константа
;1206          #12(H)=61                ; константа
;1207          #33333333=62             ; константа
;1208          #0F0F0F0F=63             ; константа
;1209          #00FF00FF=64             ; константа
;1210          #162(H)=65               ; константа для указателя переноса данных в LS
;1211          BR7.IDENT=66             ; идентификатор BR7 (1010 (двоичн.) в битах 5-2)
;1212          BG7=67                   ; адрес BG7 (установлены биты 2 и 3)
;1213      ;
;1214      ;   Ячейки временного хранения для тестов центрального процессора
;1215      ;
;1216          L30=30                   ; ячейка временного хранения LS 30 (после
;1217                                   ; использования восстанавливается значением
;1218                                   ; 10000)
;1219          L31=31                   ; ячейка временного хранения LS 31 (после
```

```
;1220                                   ; использования восстанавливается значением
;1221                                   ; 20000)
;1222           L53=53                  ; ячейка временного хранения LS 53
;1223           L73=73                  ; ячейка временного хранения LS 73
;1224    ;
;1225    ;  Адреса регистров
;1226    ;
;1227           CONTROL.OS=78           ; аппаратный индекс для управляющих слов (LS 40-4F)
;1228           SHIFT.OS(3-0)=79        ; аппаратный индекс на 16 ячеек для наборов
;1229                                   ; тестовых данных со сдвигом (LS 20-2F)
;1230           SHIFT.OS(4-0)=78        ; аппаратный индекс на 32 ячейки для наборов
;1231                                   ; тестовых данных со сдвигом (LS 20-3F)
;1232           OS=7C                   ; регистр OS
;1233           DEC.CON=7D              ; десятичные переносы
;1234           SIZE=7E                 ; регистр размера
;1235           MDT=7E                  ; размер данных для обращений к памяти
;1236           INTERRUPT.VEC=7E        ; аппаратный вектор прерывания
;1237    ;
;1238    ;  Это слово содержит аппаратные коды условий (CC). Они могут считываться или
;1239    ;  записываться сюда. На другие биты PSL не отражается.
;1240    ;
;1241           PSL.CC=7F               ; коды условий PSL
;1242    ;
;1243    ;      ОБЩЕЕ ПРИСВОЕНИЕ ЯЧЕЕК LS (регистры)
;1244    ;
;1245    ;  Верхняя половина LS
;1246
;1247           XGPR.OS=0F8             ; аппаратный индекс для регистров общего
;1248                                   ; назначения
;1249           USER.INDEX(3-0)=0F9     ; аппаратный индекс на 16 ячеек для области
;1250                                   ; диагностических данных (ячейки LS от 0A0 до 0AF)
;1251           USER.INDEX(4-0)=0F8     ; аппаратный индекс на 32 ячейки для области
;1252                                   ; диагностических данных (ячейки LS от 0A0 до 0BF)
;1253           CCR=0FC                 ; регистр чтения консоли
;1254           TIMER=0FD               ; значение интервального таймера
;1255           ALU.CC=0FE              ; коды условий АЛУ
;1256
;1257    ;  Эта ячейка представляет собой регистр, предназначенный только для записи.
;1258    ;  Оказывается воздействие на следующие биты PSL:
;1259    ;
;1260    ;      режим совместимости - бит <31>
;1261    ;      текущий режим - биты <25:24>
;1262    ;      уровень приоритета прерываний (IPL) - биты <20:16>
;1263    ;      разрешение прерываний по слежению (T или TP) - бит <4>
;1264    ;      код отрицательного условия - бит <3>
;1265    ;      код условия нуля - бит <2>
;1266    ;      код условия переполнения - бит <1>
;1267    ;      код условия переноса - бит <0>
;1268    ;
;1269           PSL.HW=0FF              ; биты аппаратного PSL
;1270    ;
;1271    ;      СПЕЦИФИЧЕСКОЕ ДЛЯ ПРОГРАММЫ ПРИСВОЕНИЕ ЯЧЕЕК LS (LS 80-F7)
;1272    ;
;1273    ;  Эти адреса доступны только для микроинструкции MOV, начиная с адреса 80. Они не
;1274    ;  используются всеми модулями программного обеспечения, но специально предназна-
```

```
;1275    ;   чены для этого модуля.
;1276    ;
;1277            L90=90                  ; ячейка временного хранения
;1278            ALTER.MIX=91            ; данные: AAAA5555
;1279            HI.IPL=92               ; установленные биты 16-20 (IPL=1F)
;1280            #FFFFFFFC=95            ; часто используемая маска (последний байт)
;1281            #000F=96                ; часто используемые данные
;1282            #00F0=97                ; часто используемые данные
;1283            #0F00=98                ; часто используемые данные
;1284            #F000=99                ; часто используемые данные
;1285            #30000=9A               ; часто используемые данные
;1286            #7FFF8000=9B            ; часто используемые данные
;1287            #540000=9C              ; часто используемые данные
;1288            #F00000=9D              ; часто используемые данные
;1289            #FC0000=9E              ; часто используемые данные
;1290            #3F003FFF=9F            ; часто используемые данные
;1291            #254=0A0                ; указатель для данных обмена с памятью
;1292            LB0=0B0                 ; ячейка временного хранения
;1293            LD0=0D0                 ; ячейка временного хранения
;1294            LF0=0F0                 ; ячейка временного хранения
```

```
;1296    .PAGE  "МАКРООПРЕДЕЛЕНИЯ "
;1297    .UCODE
;1298    .CREF
;1299    ;
;1300    ;  Эта группа имеет двухадресный формат с модификацией приемника.
;1301    ;  Первый операнд комбинируется со вторым операндом и результат записывается по
;1302    ;  второму операнду. Инструкции CMP и BIT содержат только считываемые операнды.
;1303    ;
;1304    ;  Во время всех арифметических и логических инструкций коды условий (CC) АЛУ
;1305    ;  можно загрузить добавлением к микрослову макрооператора
;1306    ;
;1307    ;     DT(SIZE)&SET.ALU.CC
;1308
;1309       Этим указывается, что регистр ALU CC загружается условиями с K1804BC1. Никакие
;1310       внешние манипуляции не происходят. Если операция производится над байтами дан-
;1311       ных, CC загружается в соответствии с битами 7-0. Слова используют биты 15-0, а
;1312    ;  длинные слова используют биты 31-0. Далее следует описание значений CC АЛУ для
;1313    ;  каждой  функции:
;1314
;1315       ADD - двоичное сложение двух операндов
;1316    ;
;1317    ;     N - бит знака
;1318          Z - установлен, если результат равен нулю
;1319    ;     V - арифметическое переполнение
;1320    ;     C - перенос
;1321    ;
;1322    ;  SUB - двоичное сложение первого операнда с дополнением до двух второго операнда
;1323    ;
;1324    ;     N - бит знака
;1325          Z - установлен, если результат равен нулю
;1326          V - арифметическое переполнение
;1327          C - инверсия займа
;1328    ;
;1329    ;  BIS - логическое "ИЛИ" для двух операндов
;1330    ;
;1331          N - бит знака
;1332    ;     Z - установлен, если результат равен нулю
;1333          V - случайное значение (не присвоена никакая функция)
;1334          C - случайное значение (не присвоена никакая функция)
;1335    ;
;1336       BIC - функция "И" для дополнения до единицы (инверсии) первого операнда
;1337    ;        и второго операнда
;1338    ;
;1339          N - бит знака
;1340    ;     Z - установлен, если результат равен нулю
;1341    ;     V - случайное значение (не присвоена никакая функция)
;1342    ;     C - случайное значение (не присвоена никакая функция)
;1343    ;
;1344    ;  AND - логическое "И" для двух операндов
;1345    ;
;1346    ;     N - бит знака
;1347    ;     Z - установлен, если результат равен нулю
;1348    ;     V - случайное значение (не присвоена никакая функция)
;1349    ;     C - случайное значение (не присвоена никакая функция)
;1350    ;
```

```
;1351    ;  XOR - логическое исключающее "ИЛИ" для двух операндов
;1352    ;
;1353    ;     N - бит знака
;1354    ;     Z - установлен, если результат равен нулю
;1355    ;     V - случайное значение (не присвоена никакая функция)
;1356    ;     C - случайное значение (не присвоена никакая функция)
;1357    ;
;1358    ;  CMP - выполняет двоичное сложение первого операнда с дополнением до двух
;1359    ;        второго операнда без запоминания результата
;1360    ;
;1361    ;     N - бит знака
;1362    ;     Z - установлен, если результат равен нулю
;1363    ;     V - арифметическое переполнение
;1364    ;     C - перенос
;1365    ;
;1366    ;  BIT - выполняет логическое умножение (И) для двух операндов без запоминания
;1367    ;        результата
;1368    ;
;1369    ;     N - бит знака
;1370    ;     Z - установлен, если результат равен нулю
;1371    ;     V - случайное значение (не присвоена никакая функция)
;1372    ;     C - случайное значение (не присвоена никакая функция)
;1373    ;
;1374    ;  SWAP - взаимная замена значений двух операндов
;1375    ;
;1376    ;     N - бит знака для значения в рабочем регистре
;1377    ;     Z - установлен, если рабочий регистр в результате имеет значение нуль
;1378    ;     V - случайное значение (не присвоена никакая функция)
;1379    ;     C - случайное значение (не присвоена никакая функция)
;1380    ;
;1381    ;  Макрооператоры формата  WR[B] = WR[B]  операция  LS[D]
;1382    ;
;1383    ADD LS[] TO   WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP.SMALL/<.SHIFT[<.AND[<SDP/LS.PLUS.WR>,07F]>,-2]>"
;1384    SUB LS[] FROM WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP.SMALL/<.SHIFT[<.AND[<SDP/LS.FROM.WR>,07F]>,-2]>"
;1385    BIS LS[] TO   WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.OR.WR>,0FF]>,-2]>"
;1386    BIC LS[] TO   WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.MASK.WR>,0FF]>,-2]>"
;1387    AND LS[] TO   WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP.SMALL/<.SHIFT[<.AND[<SDP/LS.AND.WR>,07F]>,-2]>"
;1388    XOR LS[] TO   WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP.SMALL/<.SHIFT[<.AND[<SDP/LS.XOR.WR>,07F]>,-2]>"
;1389    CMP LS[] WITH WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.CMP.WR>,0FF]>,-2]>"
;1390    BIT LS[] WITH WR[]  "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.BIT.LS>,0FF]>,-2]>"
;1391    SWAP LS[] WITH WR[] "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/SWAP.LS.WR>,0FF]>,-2]>"
;1392    ;
;1393    ;  Следующий макрооператор используется с ADD, SUB, AND, XOR LS[] TO WR[].
;1394    ;  Он берет исходное содержимое WR[] и заносит его в LS[].
;1395    ;
;1396    XCHG                "XCHG.BIT/YES"
;1397    ;
;1398    ;  Макрооператоры формата  LS[D] = LS[D]  операция  Q-регистр
;1399    ;
;1400    ADD  Q TO    LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.PLUS.LS>,0FF]>,-2]>"
;1401    SUB  Q FROM  LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.FROM.LS>,0FF]>,-2]>"
;1402    BIS  Q TO    LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.OR.LS>,0FF]>,-2]>"
;1403    AND  Q TO    LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.AND.LS>,0FF]>,-2]>"
;1404    XOR  Q TO    LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.XOR.LS>,0FF]>,-2]>"
;1405    CMP  Q WITH  LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.CMP.LS>,0FF]>,-2]>"
```

```
;1406   BIT  Q WITH  LS[]   "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.BIT.Q>,0FF]>,-2]>"
;1407   ;
;1408   ;  Макрооператоры формата  Q-регистр = Q-регистр  операция  LS[D]
;1409   ;
;1410   ADD  LS[] TO   Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.PLUS.Q>,0FF]>,-2]>"
;1411   SUB  LS[] FROM Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.FROM.Q>,0FF]>,-2]>"
;1412   BIS  LS[] TO   Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.OR.Q>,0FF]>,-2]>"
;1413   BIC  LS[] TO   Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.MASK.Q>,0FF]>,-2]>"
;1414   AND  LS[] TO   Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.AND.Q>,0FF]>,-2]>"
;1415   XOR  LS[] TO   Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.XOR.Q>,0FF]>,-2]>"
;1416   CMP  LS[] WITH Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.CMP.Q>,0FF]>,-2]>"
;1417   BIT  LS[] WITH Q     "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.BIT.Q>,0FF]>,-2]>"
;1418
;1419      Макрооператоры формата  LS[D] = LS[D]  операция  WR[B]
;1420
;1421   ADD  WR[] TO   LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.PLUS.LS>,0FF]>,-2]>"
;1422   SUB  WR[] FROM LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.FROM.LS>,0FF]>,-2]>"
;1423   BIS  WR[] TO   LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.OR.LS>,0FF]>,-2]>"
;1424   AND  WR[] TO   LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.AND.LS>,0FF]>,-2]>"
;1425   XOR  WR[] TO   LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.XOR.LS>,0FF]>,-2]>"
;1426   CMP  WR[] WITH LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.CMP.LS>,0FF]>,-2]>"
;1427   BIT  WR[] WITH LS[]  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.BIT.LS>,0FF]>,-2]>"
;1428   SWAP WR[] WITH LS[]  "SWAP LS[@2] WITH WR[@1]"
;1429   ;
;1430   ;  Макрооператоры формата  WR[B] = WR[B]  операция  WR[A]
;1431   ;
;1432   ADD  WR[] TO   WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.PLUS.WR>,03F]>"
;1433   SUB  WR[] FROM WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.FROM.WR>,03F]>"
;1434   BIS  WR[] TO   WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.OR.WR>,03F]>"
;1435   BIC  WR[] TO   WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.MASK.WR>,03F]>"
;1436   AND  WR[] TO   WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.AND.WR>,03F]>"
;1437   XOR  WR[] TO   WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.XOR.WR>,03F]>"
;1438   CMP  WR[] WITH WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.CMP.WR>,03F]>"
;1439   BIT  WR[] WITH WR[]  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.BIT.WR>,03F]>"
;1440   ;
;1441   ;  Макрооператоры формата  WR[B] = WR[B]  операция  Q-регистр
;1442   ;
;1443   ADD  Q TO    WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.PLUS.WR>,03F]>"
;1444   SUB  Q FROM  WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.FROM.WR>,03F]>"
;1445   BIS  Q TO    WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.OR.WR>,03F]>"
;1446   AND  Q TO    WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.AND.WR>,03F]>"
;1447   XOR  Q TO    WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.XOR.WR>,03F]>"
;1448   CMP  Q WITH  WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.CMP.WR>,03F]>"
;1449   BIT  Q WITH  WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/Q.BIT.WR>,03F]>"
;1450   ;
;1451   ;  Макрооператоры формата  Q-регистр = Q-регистр  операция  WR[B]
;1452   ;
;1453   ADD  WR[] TO   Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.PLUS.Q>,03F]>"
;1454   SUB  WR[] FROM Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.FROM.Q>,03F]>"
;1455   BIS  WR[] TO   Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.OR.Q>,03F]>"
;1456   BIC  WR[] TO   Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.MASK.Q>,03F]>"
;1457   AND  WR[] TO   Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.AND.Q>,03F]>"
;1458   XOR  WR[] TO   Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.XOR.Q>,03F]>"
;1459   CMP  WR[] WITH Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/WR.CMP.Q>,03F]>"
;1460   BIT  WR[] WITH Q     "OPC1/EXTENDED,A.ADRS/@1,XDP/<.AND[<SDP/Q.BIT.WR>,03F]>"
```

```
;1461   ;
;1462   ;  Форматы этой группы трехадресного типа. Первый и второй операнды
;1463   ;  считываются, а результат записывается по третьему операнду
;1464   ;
;1465   ;  Макрооператоры формата  Q-регистр = WR[B]  операция  WR[A]
;1466   ;
;1467   ADD  WR[] PLUS WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.PLUS.WR.Q>,03F]>"
;1468   SUB  WR[] FROM WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.FROM.WR.Q>,03F]>"
;1469   BIS  WR[] WITH WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.OR.WR.Q>,03F]>"
;1470   BIC  WR[] WITH WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.MASK.WR.Q>,03F]>"
;1471   AND  WR[] WITH WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.AND.WR.Q>,03F]>"
;1472   XOR  WR[] WITH WR[] TO Q  "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.XOR.WR.Q>,03F]>"
;1473   ;
;1474   ;   Макрооператоры формата  Q-регистр = WR[B]  операция  LS[D]
;1475   ;
;1476   ADD  WR[] PLUS LS[] TO Q  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.PLUS.LS.Q>,0FF]>,-2]>"
;1477   SUB  WR[] FROM LS[] TO Q  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.FROM.LS.Q>,0FF]>,-2]>"
;1478   BIS  WR[] WITH LS[] TO Q  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.OR.LS.Q>,0FF]>,-2]>"
;1479   AND  WR[] WITH LS[] TO Q  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.AND.LS.Q>,0FF]>,-2]>"
;1480   XOR  WR[] WITH LS[] TO Q  "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.XOR.LS.Q>,0FF]>,-2]>"
;1481   ;
;1482   ;   Макрооператоры формата  Q-регистр = LS[D]  операция  WR[B]
;1483   ;
;1484   ADD LS[] PLUS WR[] TO Q    "ADD WR[@2] PLUS LS[@1] TO Q"
;1485   SUB LS[] FROM WR[] TO Q    "OPC/BASIC,B.ADRS/@2,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.FROM.WR.Q>,0FF]>,-2]>"
;1486   BIS LS[] WITH WR[] TO Q    "BIS WR[@2] WITH LS[@1] TO Q"
;1487   AND LS[] WITH WR[] TO Q    "AND WR[@2] WITH LS[@1] TO Q"
;1488   XOR LS[] WITH WR[] TO Q    "XOR WR[@2] WITH LS[@1] TO Q"
;1489   ;
;1490   ;   Операции  ADD/SUB  специального назначения
;1491   ;
;1492   ADD (WR[] TO WR[])+1       "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.PLUS.WR+1>,03F]>"
;1493   ADD (LS[] TO WR[])+1       "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.PLUS.WR+1>,0FF]>,-2]>"
;1494   ADD (WR[] TO LS[])+1       "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.PLUS.LS+1>,0FF]>,-2]>"
;1495   ADD OS PLUS WR[] TO LS[]   "OPC1/MOVE,B.ADRS/@1,XD.ADRS/@2,MDP/<.SHIFT[<.AND[<SDP/WR.PLUS.OS>,3F]>,-3]>"
;1496   SUB (WR[] FROM WR[])-1     "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.FROM.WR-1>,03F]>"
;1497   SUB (LS[] FROM WR[])-1     "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.FROM.WR-1>,0FF]>,-2]>"
;1498   SUB (WR[] FROM LS[])-1     "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.FROM.LS-1>,0FF]>,-2]>"
;1499   SUB WRB[] FROM WRA[] TO WRB[]  "OPC1/EXTENDED,B.ADRS/@1,A.ADRS/@2,XDP/<.AND[<SDP/WR.FROM.WR.WR>,03F]>"
;1500   SUB LS[] FROM WR[] TO LS       "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.FROM.WR.LS>,0FF]>,-2]
;1501   SUB WR[] FROM LS[] TO WR       "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WRA.FROM.LS.WRB>,0FF]>,-
;1502   SUB WRA[] FROM Q TO WRB[]      "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.FROM.Q.WR>,03F]>"
;1503   SUB LS[] FROM Q TO LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/LS.FROM.Q.LS>,0FF]>,-2]>"
;1504   SUB Q FROM WR[] TO Q         "OPC1/EXTENDED,B.ADRS/@1,XDP/<.AND[<SDP/Q.FROM.WR.Q>,03F]>"
;1505   SUB Q FROM LS[] TO Q         "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.FROM.LS.Q>,0FF]>,-2]>"
;1506   ADD Q PLUS WR[] TO LS[]      "OPC/BASIC,D.ADRS/@2,B.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.PLUS.WR.TO.LS>,0FF]>,-2]>
;1507   SUB Q FROM WR[] TO LS[]      "OPC/BASIC,D.ADRS/@2,B.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.FROM.WR.TO.LS>,0FF]>,-2]>
;1508   ADD Q PLUS LS[] TO WR[]      "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/Q.PLUS.LS.TO.WR>,0FF]>,-2]>
;1509   ADD Q TO WR[] XCHG TO LS[]   "OPC/BASIC,D.ADRS/@2,B.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/WR.PLUS.Q.XCHG>,0FF]>,-2]>"
;1510   ;
;1511   ;  Макроинструкции пересылки
;1512   ;
;1513   ;  Во время макроинструкций пересылки можно загружать коды условий (CC) АЛУ
;1514   ;  путем добавления к микрослову макрооператора
;1515   ;
```

```
;1516    ;       DT(SIZE)&SET.ALU.CC
;1517    ;
;1518    ;  Далее следует описание значений кодов условий АЛУ для каждой функций:
;1519    ;  ..
;1520    ;  MOV - занесение первого операнда на место второго операнда
;1521    ;
;1522    ;       N - бит знака
;1523    ;       Z - установлен, если результат равен нулю
;1524    ;       V - случайное значение (не присвоена никакая функция)
;1525    ;       C - случайное значение (не присвоена никакая функция)
;1526    ;
;1527    ;  MCOM - занесение дополнения до единицы (инверсии) первого операнда
;1528    ;         на место второго операнда
;1529    ;
;1530    ;       N - бит знака
;1531    ;       Z - установлен, если результат равен нулю
;1532    ;       V - случайное значение (не присвоена никакая функция)
;1533    ;       C - случайное значение (не присвоена никакая функция)
;1534    ;
;1535    ;  MNEG - занесение дополнения до двух первого операнда на место второго
;1536    ;         операнда
;1537    ;
;1538    ;       N - бит знака
;1539    ;       Z - установлен, если результат нулевой
;1540    ;       V - арифметическое переполнение
 1541    ;       C - перенос
;1542    ;
;1543    MOV Q  TO LS[]              "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/MOV.Q.LS>,0FF]>,-2]>"
;1544    MOV Q  TO WR[]              "OPC1/EXTENDED,B.ADRS/@1,XDP/<.AND[<SDP/MOV.Q.WR>,03F]>"
;1545    MOV LS[] TO Q               "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/MOV.LS.Q>,0FF]>,-2]>"
;1546    MOV Q TO WR[] XCHG TO LS[]  "OPC/BASIC,D.ADRS/@2,B.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/MOV.Q.WR&WR.LS>,0FF]>,-2]>"
;1547    MOV IB.DATA TO OS           "OPC2/DECODE,IFUNC/SPEC,R.DST/NOP,IB.REQ/IB.REQ,JCTL/NO.JUMP.TST,LD.OS/LOAD.OS"
;1548    MOV CM.IB.DATA TO OS        "OPC2/DECODE,IFUNC/SPEC,R.DST/NOP,IB.REQ/NOP,JCTL/NO.JUMP.TST,LD.OS/LOAD.OS"
;1549    MOV MEM.DATA TO WR[]        "OPC1/MOVE,B.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.MEM.WR>,3F]>,-3]>,XD.ADRS/WR.BAC
;1550    MOV MEM.DATA TO WR[] XCHG TO LS[]    "OPC1/MOVE,B.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.MEM.WR>,3F]>,-3]>,XD.AD
;1551    MOV MEM.DATA TO LS[]        "OPC1/MOVE,XD.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.MEM.LS>,3F]>,-3]>"
;1552    MOV LS[] TO WR[]    "OPC1/MOVE,XD.ADRS/@1,B.ADRS/@2,MDP/<.SHIFT[<.AND[<SDP/MOV.LS.WR>,3F]>,-3]>"
;1553    MOV WR[] TO LS[]    "OPC1/MOVE,B.ADRS/@1,XD.ADRS/@2,MDP/<.SHIFT[<.AND[<SDP/MOV.WR.LS>,3F]>,-3]>"
;1554    MOV WR[] TO WR[]    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.PLUS.0.TO.WR>,03F]>"
;1555    MOV WR[] TO Q       "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.OR.WR.Q>,03F]>"
;1556    MOV XWR[] TO Q      "OPC1/MOVE,XB.ADRS/@1,XD.ADRS/0,MDP/<.SHIFT[<.AND[<SDP/MOV.XWR.Q>,3F]>,-3]>"
;1557    MOV FPA TO LS[]     "OPC1/MOVE,XD.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.ACC.LS>,3F]>,-3]>"
;1558    MOV PORT TO LS[]    "OPC1/MOVE,XD.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.ACC.LS>,3F]>,-3]>,CC/DT(LONG)&HOLD.ALU.
;1559    MCOM WR[] TO LS[]   "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.COM.LS>,0FF]>,-2]>"
;1560    MCOM LS[] TO WR[]   "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.COM.WR>,0FF]>,-2]>"
;1561    MNEG WR[] TO LS[]   "OPC/BASIC,B.ADRS/@1,D.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/WR.NEG.LS>,0FF]>,-2]>"
;1562    MNEG LS[] TO WR[]   "OPC/BASIC,D.ADRS/@1,B.ADRS/@2,DP/<.SHIFT[<.AND[<SDP/LS.NEG.WR>,0FF]>,-2]>"
;1563    MNEG WR[] TO WR[]   "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.NEG.WR>,03F]>"
;1564    MCOM WR[] TO WR[]   "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.COM.WR>,03F]>"
;1565    MINC WR[] TO WR[]   "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.INC.WR>,03F]>"
;1566    MDEC WR[] TO WR[]   "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,XDP/<.AND[<SDP/WR.DEC.WR>,03F]>"
;1567    MNEG Q TO LS[]      "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/Q.NEG.LS>,0FF]>,-2]>"
;1568    ;
;1569    ;
;1570    ;       Операции с одним операндом
```

```
;1571    ;
;1572    ;  Инструкции этого формата выполняют требуемую операцию над заданным
;1573    ;  операндом:
;1574    ;                очистку
;1575    ;                отрицание
;1576    ;                инкрементирование
;1577    ;                декрементирование
;1578    ;                дополнение
;1579    ;                проверку
;1580    ;
;1581    ;   Во время инструкций с одним операндом коды условий (CC) АЛУ можно
;1582    ;   загрузить с использованием в микрослове макрооператора
;1583    ;
;1584    ;              DT(SIZE)&SET.ALU.CC
;1585    ;
;1586    ;   Далее следует описание кодов условий АЛУ каждой функции:
;1587    ;
;1588    ;  CLR - запоминание нуля в операнде
;1589    ;
;1590    ;      N - бит знака
;1591    ;      Z - установлен, если результат равен нулю
;1592    ;      V - случайное значение (не присвоена никакая функция)
;1593    ;      C - случайное значение (не присвоена никакая функция)
;1594    ;
;1595    ;  NEG - выполнение для операнда дополнения до двух
;1596    ;
;1597    ;      N - бит знака
;1598    ;      Z - установлен, если результат равен нулю
;1599    ;      V - арифметическое переполнение
;1600    ;      C - перенос
;1601    ;
;1602    ;  INC - прибавление единицы к операнду
;1603    ;
;1604    ;      N - бит знака
;1605    ;      Z - установлен, если результат равен нулю
;1606    ;      V - арифметическое переполнение
;1607    ;      C - перенос
;1608
;1609       DEC - вычитание единицы из операнда
;1610    ;
;1611    ;      N - бит знака
;1612    ;      Z - установлен, если результат равен нулю
;1613    ;      V - арифметическое переполнение
;1614    ;      C - перенос
;1615    ;
;1616    ;  COM - выполнение для операнда дополнения до единицы (XNOR с нулем)
;1617    ;
;1618    ;      N - бит знака
;1619    ;      Z - установлен, если результат равен нулю
;1620    ;      V - случайное значение (не присвоена никакая функция)
;1621    ;      C - случайное значение (не присвоена никакая функция)
;1622    ;
;1623    ;  TST - прибавление нуля к операнду с целью получения кодов условий
;1624    ;
;1625    ;      N - бит знака
```

```
;1626    ;       Z - установлен,если результат равен нулю
;1627    ;       V - арифметическое переполнение (в данном случае всегда нуль)
;1628    ;       C - перенос (в данном случае нуль)
;1629    ;
;1630    CLR Q           "OPC1/EXTENDED,A.ADRS/0,B.ADRS/0,XDP/<.AND[<SDP/WR.XOR.WR.Q>,03F]>"
;1631    CLR LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/CLR.LS>,0FF]>,-2]>"
;1632    CLR WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.XOR.WR>,03F]>"
;1633    NEG Q           "OPC1/EXTENDED,XDP/<.AND[<SDP/NEG.Q>,03F]>"
;1634    NEG LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/NEG.LS>,0FF]>,-2]>"
;1635    NEG WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.NEG.WR>,03F]>"
;1636    INC Q           "OPC1/EXTENDED,XDP/<.AND[<SDP/INC.Q>,03F]>"
;1637    INC LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/INC.LS>,0FF]>,-2]>"
;1638    INC WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.INC.WR>,03F]>"
;1639    DEC Q           "OPC1/EXTENDED,XDP/<.AND[<SDP/DEC.Q>,03F]>"
;1640    DEC LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/DEC.LS>,0FF]>,-2]>"
;1641    DEC WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.DEC.WR>,03F]>"
;1642    COM Q           "OPC1/EXTENDED,XDP/<.AND[<SDP/COM.Q>,03F]>"
;1643    COM LS[]        "OPC/BASIC,D.ADRS/@1,DP/<.SHIFT[<.AND[<SDP/COM.LS>,0FF]>,-2]>"
;1644    COM WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.COM.WR>,03F]>"
;1645    TST WR[]        "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,XDP/<.AND[<SDP/WR.PLUS.0.TO.WR>,03F]>"
;1646    TST Q           "OPC1/EXTENDED,XDP/<.AND[<SDP/Q.PLUS.0>,03F]>"
;1647    NOP             "OPC/BASIC,D.ADRS/0,B.ADRS/0,DP/<.SHIFT[<.AND[<SDP/Q.CMP.LS>,0FF]>,-2]>"
;1648    ;
;1649    ;    Макрооператоры для операций сдвига WR[B] и/или Q-регистра
;1650    ;
;1651    ;  Во время операций сдвига можно загрузить коды условий (CC) АЛУ использованием
;1652    ;  в микрослове макрооператора
;1653    ;
;1654    ;       DT(SIZE)&SET.ALU.CC
;1655    ;
;1656    ;  Далее следует описание кодов условий АЛУ для каждой функции:
;1657    ;
;1658    ;  ROR WR[] - циклический сдвиг 32-битового значения на одну позицию вправо
;1659    ;
;1660    ;       N - знак входных данных
;1661    ;       Z - установлен, если входные данные равны нулю
;1662    ;       V - случайное значение (не присвоена никакая функция)
;1663    ;       C - случайное значение (не присвоена никакая функция)
;1664    ;
;1665    ;  ROL WR[] - циклический сдвиг 32-битового значения на одну позицию влево
;1666    ;
;1667    ;       N - бит знака входных данных
;1668    ;       Z - установлен,если входные данные равны нулю
;1669    ;       V - случайное значение (не присвоена никакая функция)
;1670    ;       C - случайное значение (не присвоена никакая функция)
;1671    ;
;1672    ;  ASHR WR[] - сдвиг 32-битового значения на одну позицию вправо с расширением знака
;1673    ;
;1674    ;       N - знак входных данных
;1675    ;       Z - установлен, если входные данные равны нулю
;1676    ;       V - случайное значение (не присвоена никакая функция)
;1677    ;       C - случайное значение (не присвоена никакая функция)
;1678    ;
;1679    ;  ASHL WR[] - сдвиг 32-битового значения на одну позицию влево с установкой нуля в
;1680    ;              младшем бите
```

```
;1681    ;
;1682    ;       N - знак входных данных
;1683    ;       Z - установлен, если входные данные нулевые
;1684    ;       V - случайное значение (не присвоена никакая функция)
;1685    ;       C - случайное значение (не присвоена никакая функция)
;1686    ;
;1687    ;   ASHRC WR[].Q - сдвиг WR и Q, как 64-битовых данных, на одну позицию влево с
;1688    ;                  расширением знака
;1689    ;
;1690    ;       N - знак входных данных в WR[]
;1691    ;       Z - установлен, если входные данные в WR[] равны нулю
;1692    ;       V - случайное значение (не присвоена никакая функция)
;1693    ;       C - случайное значение (не присвоена никакая функция)
;1694    ;
;1695    ;   RORC WR[].Q - циклический сдвиг WR и Q, как 64-битовых данных, на одну
;1696    ;                 позицию вправо
;1697    ;
;1698    ;       N - знак входных данных в WR[]
;1699    ;       Z - установлен, если входные данные в WR[] равны нулю
;1700    ;       V - случайное значение (не присвоена никакая функция)
;1701    ;       C - случайное значение (не присвоена никакая функция)
;1702    ;
;1703    ;   ASHLC WR[].Q - сдвиг WR и Q, как 64-битовых данных, на одну позицию влево
;1704    ;                  с занесением нуля
;1705    ;
;1706    ;       N - знак входных данных в WR[]
;1707    ;       Z - установлен, если входные данные в WR[] равны нулю
;1708    ;       V - случайное значение (не присвоена никакая функция)
;1709    ;       C - случайное значение (не присвоена никакая функция)
;1710    ;
;1711    ;   ROLC WR[].Q - циклический сдвиг WR и Q, как 64-битовых данных, на одну
;1712    ;                 позицию влево
 1713
;1714    ;       N - знак входных данных в WR[]
;1715    ;       Z - установлен, если входные данные в WR[] нулевые
;1716    ;       V - случайное значение (никакая функция не присвоена)
;1717    ;       C - случайное значение (никакая функция не присвоена)
;1718    ;
;1719    ;   SHL2 WR[] - сдвиг 32-битового значения на 2 позиции влево с занесением нулей в
;1720    ;               младшие биты
;1721    ;
;1722    ;       N - знак входных данных WR[]+WR[]
;1723    ;       Z - установлен , если входные данные WR[]+WR[] равны нулю
;1724    ;       V - переполнение по входным данным WR[]+WR[]
;1725    ;       C - перенос из входных данных WR[]+WR[]
;1726    ;
;1727    ROR WR[]        "OPC1/EXTENDED,B.ADRS/@1,SI/ROT.WR,XDP/<.AND[<SDP/ASHR>,03F]>"
;1728    ROL WR[]        "OPC1/EXTENDED,B.ADRS/@1,SI/ROT.WR,XDP/<.AND[<SDP/ASHL>,03F]>"
;1729    ASHR WR[]       "OPC1/EXTENDED,B.ADRS/@1,SI/ASH.WR,XDP/<.AND[<SDP/ASHR>,03F]>"
;1730    ASHL WR[]       "OPC1/EXTENDED,B.ADRS/@1,SI/ASH.WR,XDP/<.AND[<SDP/ASHL>,03F]>"
;1731    ASHRC WR[].Q    "OPC1/EXTENDED,B.ADRS/@1,SI/ASH.WR.Q,XDP/<.AND[<SDP/ASHRC>,03F]>"
;1732    RORC WR[].Q     "OPC1/EXTENDED,B.ADRS/@1,SI/ROT.WR.Q,XDP/<.AND[<SDP/ASHRC>,03F]>"
;1733    ASHLC WR[].Q    "OPC1/EXTENDED,B.ADRS/@1,SI/ASH.WR.Q,XDP/<.AND[<SDP/ASHLC>,03F]>"
;1734    ROLC WR[].Q     "OPC1/EXTENDED,B.ADRS/@1,SI/ROT.WR.Q,XDP/<.AND[<SDP/ASHLC>,03F]>"
;1735    SHL2 WR[]       "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@1,SI/2,XDP/<.AND[<SDP/SHL2>,03F]>"
```

```
;1736   COND.ADD&SHIFT WR[] TO WR[].Q    "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,SI/MUL.SHF,XDP/<.AND[<SDP/COND.ADD&SHIFT>
;1737   COND.SUB&SHIFT WR[] FROM WR[].Q "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,SI/MUL.SHF,
;1738                                    XDP/<.AND[<SDP/COND.SUB&SHIFT>,03F]>"
;1739   UNSIGNED.MUL WR[] TO WR[].Q      "OPC1/EXTENDED,A.ADRS/@1,B.ADRS/@2,SI/UMUL.SHF,
;1740                                    XDP/<.AND[<SDP/COND.ADD&SHIFT>,03F]>"
;1741   SHR WR[]          "OPC1/EXTENDED,B.ADRS/@1,SI/SHF.WR.Q,XDP/<.AND[<SDP/ASHR>,03F]>"
;1742   SHL WR[]          "ASHL WR[@1]"
;1743   SHRC WR[].Q       "OPC1/EXTENDED,B.ADRS/@1,SI/SHF.WR.Q,XDP/<.AND[<SDP/ASHRC>,03F]>"
;1744   SHLC WR[].Q       "ASHLC WR[].Q"
;1745   ;
;1746   ;   Макрооператор запроса памяти
;1747   ;
;1748   ; *  Этот макрооператор не предназначен для общего использования.     *
;1749   ;  *  Он предназначен только для использования в следующем макрооператоре *
;1750   ;
;1751   MEM.FUNC[]                "M.FUNC2/<.SHIFT[<MEM.FUNC/@1>,-2]>,M.FUNC1/<.AND[<MEM.FUNC/@1>,3]>"
;1752   MEM.REQ[] ADRS[] DT[]     "OPC2/MEM.REQ,MEM.FUNC[@1],D.ADRS/@2,MDT/@3"
;1753   ;
;1754   WRITE.MEM LS[]            "OPC1/MOVE,XD.ADRS/@1,MDP/<.SHIFT[<.AND[<SDP/MOV.LS.MEM>,3F]>,-3]>"
;1755   ;
;1756   ;
;1757   ;  Макрооператоры для разных операций
;1758   ;
;1759   ;  В языке микропрограммирования на уровне схем существует необходимость в раз-
;1760   ;  личных уникальных действиях для функционирования. Они задаются инструкциями
;1761   ;  MISC. Большинство функций выполняются указанием запроса внутри макрооперато-
;1762   ;  ров MISC[] и MISC2[].
;1763   ;
;1764   MISC []                   "OPC2/MISC,MISC.1/@1,MISC.2/NOP,MISC.PORT/MISC"
;1765   MISC2 []                  "OPC2/MISC,MISC.1/NOP,MISC.2/@1,MISC.PORT/MISC"
;1766   MISC [] WR[]              "MISC [@1],MISC.WRL/0,MISC.WRR/@2"
;1767   MISC2 [] WR[]             "MISC2 [@1],MISC.WRL/0,MISC.WRR/@2"
;1768   PORT.CONTROL []           "OPC2/MISC,MISC.PORT/PORT,CNTL.FUNC/@1,PORT.SELECT/IDC,PORT.FUNC/CONTROL"
;1769   PORT.WRITE PORT.ADRS[] WITH WR[]   "OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/@1,
;1770                                       PORT.SELECT/IDC,PORT.FUNC/WRITE,MISC.WRL/0,MISC.WRR/@2"
;1771   PORT.READ PORT.ADRS[]     "OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/@1,PORT.SELECT/IDC,PORT.FUNC/READ"
;1772   ;
;1773   ;  В нескольких случаях разные функции вызываются как уникальные функции набора
;1774   ;  языка микропрограммирования. Они перечислены ниже.
;1775   ;
;1776   ;  Следующие макрооператоры пересылают данные во внешние ускорители.
;1777   ;
;1778   ASSERT.DA.AND.PASS.WR0          "MISC2 [ASSERT.DA.AND.PASS.WR] WR[0]"
;1779   ASSERT.DA.AND.PASS.WR1          "MISC2 [ASSERT.DA.AND.PASS.WR] WR[1]"
;1780   ;
;1781   ;  Макрооператоры регистра STATE - возможные изменения битов STATE задаются как
;1782   ;  индивидуальные функции.
;1783   ;
;1784   CLR.STATE.ZERO                   "OPC2/MISC,MISC.1/CLR.S0,MISC.2/NOP,MISC.PORT/MISC"
;1785   CLR.STATE.ONE                    "OPC2/MISC,MISC.1/CLR.S1,MISC.2/NOP,MISC.PORT/MISC"
;1786   SET.STATE.ZERO                   "OPC2/MISC,MISC.1/SET.S0,MISC.2/NOP,MISC.PORT/MISC"
;1787   SET.STATE.ONE                    "OPC2/MISC,MISC.1/SET.S1,MISC.2/NOP,MISC.PORT/MISC"
;1788   SET.STATE.ZERO&CLR.STATE.ONE     "OPC2/MISC,MISC.1/CLR.S1&SET.S0,MISC.2/NOP,MISC.PORT/MISC"
;1789   SET.STATE.ONE&CLR.STATE.ZERO     "OPC2/MISC,MISC.1/SET.S1&CLR.S0,MISC.2/NOP,MISC.PORT/MISC"
;1790   CLR.STATE.ONE&CLR.STATE.ZERO     "OPC2/MISC,MISC.1/CLR,MISC.2/NOP,MISC.PORT/MISC"
```

```
;1791   SET.BACKUP.MASK.VALID          "OPC2/MISC,MISC.1/SET.BACKUP.MASK.VALID,MISC.2/NOP,MISC.PORT/MISC"
;1792   ;
;1793   ;  Макрооператоры размера контекста и кодов условий
;1794   ;
;1795   DT(SIZE)&MAP.ALU.CC.TO.PSL      "CC/DT(SIZE)&MAP.ALU.CC.TO.PSL"
;1796   DT(LONG)&SET.ALU.CC             "CC/DT(LONG)&SET.ALU.CC"
;1797   DT(SIZE)&SET.ALU.CC             "CC/DT(SIZE)&SET.ALU.CC"
;1798   ;
;1799   ;  Макрооператоры управления переходами и микросеквенсером
;1800   ;
;1801   ;  Следующие макрооператоры используются для управления прохождением программ
;1802   ;
;1803   ;  JMP []          - безусловный переход к новому адресу
;1804   ;  JMP.OS []       - безусловный переход к адресу, полученному в результате логического
;1805   ;                    объединения по "ИЛИ" адреса в инструкции и регистра OS
;1806   ;  JMP.IF[] TO []  - переход к новому адресу только если выполнено условие перехода
;1807   ;  JSR []          - безусловный переход к новому адресу и занесение адреса возврата в стек
;1808   ;                    секвенсера
;1809   ;  JSR.OS []       - переход к адресу, полученному в результате логического "ИЛИ" адреса
;1810   ;                    в инструкции и регистра OS. Адрес возврата заносится также в стек
;1811   ;                    секвенсера
;1812   ;  RETURN          - переход к адресу из вершины стека секвенсера и продвижение стека
;1813   ;                    вперед
;1814   ;  RETURN+1        - переход к адресу из вершины стека секвенсера плюс один и продвижение
;1815   ;                    стека вперед
;1816   ;  RETURN+1.IF(MEM.REF.OK) - если в самом последнем запросе памяти не обнаружено
;1817   ;                    ошибок, тогда переход к адресу из вершины стека секвенсера плюс один
;1818   ;                    и продвижение стека вперед. Если была ошибка, тогда пропускается
;1819   ;                    следующая инструкция
;1820   ;  LOOP.IF(NEQ)    - если бит Z АЛУ равен нулю (последнее значение, вычисленное с
;1821                       занесением в ALU.CC, не показывает результата, равного нулю), то
;1822   ;                    переход к адресу из вершины стека секвенсера. Стек не продвигается.
;1823   ;                    Если бит Z АЛУ равен единице, тогда продолжением является следующая
;1824   ;                    инструкция (UPC+1) и стек продвигается вперед
;1825   ;  LOOP.IF(GEQU)   - если бит "C" АЛУ равен единице (последнее значение, вычисленное с
;1826   ;                    занесением в ALU.CC, было больше или равно нулю), то переход к
;1827   ;                    адресу из вершины стека секвенсера, стек не продвигается. Если бит "C"
;1828   ;                    АЛУ равен нулю, то продолжением является следующая инструкция (UPC+1)
;1829   ;                    и стек продвигается вперед
;1830   ;  LOOP.IF(NO INTERRUPT AND NO SYNC) - если условие прерывания не является истинным
;1831   ;                    и условие синхронизации с ускорителем не является истинным, тогда
;1832   ;                    переход к адресу из вершины стека секвенсера, стек не продвигается.
;1833   ;                    Если условие прерывания истинно, то продолжением является следующая
;1834   ;                    инструкция (UPC+1) и стек секвенсера не продвигается. Если является
;1835   ;                    истинным условие синхронизации с ускорителем (SYNC), то продолжением
;1836   ;                    является следующая инструкция и стек секвенсера продвигается вперед
;1837   ;  SKIP.IF[]       - если условие пропуска удовлетворено, то переход к UPC+2, иначе
;1838   ;                    переход к UPC+1
;1839   ;
;1840   JMP []                          "OPC2/JUMP,JUMP.ADRS/@1,JCTL/JUMP"
;1841   JMP.OS []                       "OPC2/JUMP,JUMP.ADRS/@1,OS/WITH.OS,JCTL/JUMP"
;1842   JMP.IF[] TO []                  "OPC2/JUMP,JCTL/@1,JUMP.ADRS/@2"
;1843   JSR []                          "OPC2/JUMP,JCTL/JSR,JUMP.ADRS/@1"
;1844   JSR.OS []                       "OPC2/JUMP,JCTL/JSR,JUMP.ADRS/@1,OS/WITH.OS"
;1845   RETURN                          "SCTL/RETURN"
```

```
;1846   RETURN+1                          "SCTL/RETURN+1"
;1847   RETURN+1.IF(MEM.REF.OK)           "SCTL/RET.NOERR"
;1848   LOOP.IF(NEQ)                      "SCTL/JMP.Z.CLR"
;1849   LOOP.IF(GEQU)                     "SCTL/JMP.C.SET"
;1850   LOOP.IF(NO INTERRUPT AND NO SYNC) ELSE SKIP.IF(SYNC)    "SCTL/LOOP.IF.NO.I.AND.SYNC"
;1851   SKIP.IF[]                         "SCTL/@1"
;1852   SKIP                              "SCTL/SKIP"
;1853   ;
;1854   ;  Макрооператоры потока инструкций
;1855   ;
;1856   ; * Этот макрооператор не предназначен для общего использования.          *
;1857   ; * Он предназначен только для использования в следующих макрооператорах *
;1858   ;
;1859   DEC[ ]                    "OPC2/DECODE,DEC.ADRS/<.SHIFT[<JUMP.ADRS/@1>,-8]>"
;1860   ;
;1861   DECODE.SPEC ADRS[]        "DEC[@1],IFUNC/SPEC,BPC/NOP,CM.HI/LOW.BYTE,IB.REQ/IB.REQ,LD.OS/LOAD.OS,R.DST/ENAB,OPC
;1862   DECODE.ASRC ADRS[]        "DEC[@1],IFUNC/ASRC,BPC/NOP,CM.HI/LOW.BYTE,IB.REQ/IB.REQ,LD.OS/LOAD.OS,RDST/ENAB,OPC.
;1863   DECODE.ESRC ADRS[]        "DEC[@1],IFUNC/ESRC,BPC/NOP,CM.HI/LOW.BYTE,IB.REQ/IB.REQ,LD.OS/LOAD.OS,R DST/ENAB.OPC
;1864   DECODE.VSRC ADRS[]        "DEC[@1],IFUNC/VSRC,BPC/NOP,CM.HI/LOW.BYTE,IB.REQ/IB.REQ,LD.OS/LOAD.OS,R.DST/ENAB,OPC
;1865   DECODE.FLOAT ADRS[]       "DEC[@1],IFUNC/FLOAT,BPC/NOP,CM.HI/LOW.BYTE.IB.REQ/IB.REQ,LD.OS/LOAD.OS.R.DST/ENAB,OP
;1866   DECODE.OPC1 ADRS[]        "DEC[@1],IFUNC/VAX.IRD,R.DST/NOP,CM.HI/LOW.BYTE,IB.REQ/IB.REQ,OPC.SPEC/VAX.IRD"
;1867   EXEC.DISPATCH ADRS[]      "DEC[@1],IFUNC/VAX.EXEC,IB.REQ/NOP,R.DST/NOP,CM.HI/LOW.BYTE,JCTL/JSR,
;1868                              OPC.SPEC/VAX.EXEC"
;1869   DECODE.CM.OPC ADRS[]      "DEC[@1],IFUNC/CM.IRD,CM.HI/HI.BYTE,OPC.SPEC/CM.IRD,LD.OS/LOAD.OS"
;1870   DECODE.CM.DST ADRS[]      "DEC[@1],IFUNC/CM.DST,OPC.SPEC/CM.DST,LD.OS/LOAD.OS,R.DST/ENAB"
;1871   DECODE.CM.SINGLE ADRS[ ]      "DEC[@1],IFUNC/CM.SINGLE,OPC.SPEC/CM.SINGLE"
;1872   CM.EXEC ADRS[]            "DEC[@1],IFUNC/CM.EXEC,JCTL/JUMP,OPC.SPEC/CM.EXEC,LD.OS/LOAD.OS"
;1873   SAVE.PC WR0               "BPC/SAVE.PC"
;1874   SAVE.PC WR4               "BPC/SAVE.PC"
;1875   SAVE.CM.PC WR0            "BPC/SAVE.PC"
;1876   SAVE.CM.PC WR4            "BPC/SAVE.PC"
;1877   ;
;1878   ;  Эти условия пропуска используются микроинструкцией DECODE
;1879   ;
;1880   SKIP.IF(IB VALID)         "JCTL/NO.JUMP.TST"
;1881   JMP.IF(IB VALID)          "JCTL/JUMP.VALID"
;1882   JSR.IF(IB VALID)          "JCTL/JSR.VALID"
;1883   ;
;1884   ;  Эти условия пропуска должны использоваться с макрооператорами режима
;1885   ;  совместимости
;1886   ;
;1887   JMP.TO []                 "JCTL/JUMP,DEC[@1]"
;1888   JSR.TO []                 "JCTL/JSR,DEC[@1]"
;1889
;1890   ;       МАКРООПРЕДЕЛЕНИЯ ДЛЯ ДИАГНОСТИКИ
;1891   ;
;1892   LS.DATA.WORD/=<15:0>
;1893   UPPER.BYTE/=<23:16>
;1894   ;
;1895   WORD[]          "UPPER.BYTE/0,LS.DATA.WORD/@1"
;1896   COUNT.LS[]      "UPPER.BYTE/0,LS.DATA.WORD/@1"
;1897   COUNT.MM[]      "UPPER.BYTE/1,LS.DATA.WORD/@1"
;1898   START.ADR[]     "UPPER.BYTE/0,LS.DATA.WORD/@1"
;1899   ;
;1900   ASCII.LIT/=<15:0>
```

```
;1901            CA=4341
;1902            CB=4342
;1903            CC=4343
;1904            CD=4344
;1905            CE=4345
;1906            CF=4346
;1907            CG=4347
;1908            CH=4348
;1909    ;
;1910    ASCII [] "UPPER.BYTE/0,ASCII.LIT/@1"
;1911
```

```
;1912    .PAGE "ОПРЕДЕЛЕНИЯ ПОЛЕЙ УПРАВЛЯЮЩЕГО МИКРОСЛОВА ПУТЕЙ ДАННЫХ "
;1913    .BIN
;1914    ;
;1915    ;  Этот раздел относится к ПЗУ и ПМЛ управления операцией (функция, источник (R),
;1916    ;  приемник (S), входной перенос) микропроцессоров К1804ВС1 платы DAP. ПЗУ УПР.
;1917    ;  ИСТ.ПРИЕМН.АЛУ и ПМЛ УПР. ФУНКЦИЙ АЛУ рассматриваются как одна управляющая
;1918    ;  память на 512 адресов с шириной слова 10 битов
;1919    ;
;1920    .CCODE
;1921    SDP/=<8:0>,.ADDRESS,.VALIDITY=0
;1922    ;
;1923    ; Это поле соответствует входам управления функцией АЛУ IN5, IN4 и IN3
;1924    ;
;1925    ALU/=<2:0>,.DEFAULT=<ALU/R.OR.S>
;1926    ;
;1927          R.PLUS.S=0                ; сложение R и S
;1928          S.MINUS.R=1               ; вычитание R из S
;1929          R.MINUS.S=2               ; вычитание S из R
;1930          R.OR.S=3                  ; "ИЛИ" для R и S
;1931          R.AND.S=4                 ; "И" для R и S
;1932          NOTR.AND.S=5              ; инвертирование R и затем "И" с S
;1933          R.XOR.S=6                 ; исключающее "ИЛИ" для R и S
;1934          R.XNOR.S=7                ; исключающее "ИЛИ" для R и S, а затем
;1935                                    ; инвертирование результата
;1936    ;
;1937    ;  Это поле соответствует входам управления приемником АЛУ
;1938    ;  IN8, IN7 и IN6
;1939    ;
;1940    DST/=<5:3>,.DEFAULT=<DST/NOP>
;1941    ;
;1942          LOAD.Q=0                  ; загрузка Q-регистра
;1943          NOP=1                     ; сохранение всех регистров
;1944          WRITE.B.A=2               ; запись во внутреннее ЗУ по адресу на B-порте
;1945                                    ; и вывод из внутреннего ЗУ по адресу на A-порте
;1946          WRITE.B.F=3               ; запись во внутреннюю память по B-порту и
;1947                                    ; вывод АЛУ
;1948          RSHF.RAM.Q=4              ; сдвиг вправо внутренней памяти и Q
;1949          RSHF.RAM=5                ; сдвиг вправо внутренней памяти
;1950          LSHF.RAM.Q=6              ; сдвиг влево внутренней памяти и Q
;1951          LSHF.RAM=7                ; сдвиг влево внутренней памяти
;1952
;1953    ;  Это поле соответствует входам управления источником IN2, IN1 и IN0
;1954    ;
;1955    SRC/=<8:6>,.DEFAULT=<SRC/D.0>
;1956    ;
;1957          A.Q=0                     ; R=A    S=Q
;1958          A.B=1                     ; R=A    S=B
;1959          0.Q=2                     ; R=0    S=Q
;1960          0.B=3                     ; R=0    S=B
;1961          D.Q=4                     ; R=D    S=Q
;1962          D.0=5                     ; R=D    S=0
;1963          0.A=6                     ; R=0    S=A
;1964          D.A=7                     ; R=D    S=A
;1965    ;
;1966    ;  Входной перенос
```

```
;1967   ;
;1968   CIN/=<9>,.DEFAULT=0
;1969   ;
;1970           NO.CIN=0                ; входной перенос АЛУ отсутствует
;1971           CIN=1 ,                 ; имеется входной перенос в АЛУ
;1972   ;
;1973   ;
;1974   ;
```

```
                ;1975    .PAGE  "СОДЕРЖИМОЕ УПРАВЛЯЮЩЕЙ ПАМЯТИ ПУТЕЙ ДАННЫХ "
                ;1976    .SEQUENTIAL
                ;1977    ;
                ;1978    ;  Здесь представлено содержимое управляющей памяти путей данных.
                ;1979    ;
                ;1980    ;  Следующие ячейки используются микроинструкцией DECODE.IS.  Эта микроинструкция выпол-
                ;1981    ;  няет операцию PC+1 и резервирует копию счетчика команд PC во внутренней
                ;1982    ;  памяти
                ;1983    ;
                ;1984    000:
                ;1985    DECODE.IS:
C 0000,  3D8    ;1986            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0001,  3D8    ;1987            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0002,  3D8    ;1988            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0003,  3D8    ;1989            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0004,  3D8    ;1990            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0005,  3D8    ;1991            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0006,  3D8    ;1992            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0007,  3D8    ;1993            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0008,  3D8    ;1994            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0009,  3D8    ;1995            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000A,  3D8    ;1996            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000B,  3D8    ;1997            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000C,  3D8    ;1998            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000D,  3D8    ;1999            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000E,  3D8    ;2000            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 000F,  3D8    ;2001            SRC/D.0,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
                ;2002
                ;2003    ;  PC не резервируется
                ;2004    ;
C 0010,  3C8    ;2005            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0011,  3C8    ;2006            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0012,  3C8    ;2007            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0013,  3C8    ;2008            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0014,  3C8    ;2009            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0015,  3C8    ;2010            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0016,  3C8    ;2011            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0017,  3C8    ;2012            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0018,  3C8    ;2013            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 0019,  3C8    ;2014            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 001A,  3C8    ;2015            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 001B,  3C8    ;2016            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 001C,  3C8    ;2017            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 001D,  3C8    ;2018            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 001E,  3C8    ;2019            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
  001F,  3C8    ;2020            SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
                ;2021    ;
                ;2022    ;
                ;2023    ;  Эти адреса используются микроинструкцией JUMP
                ;2024    ;
                ;2025    ;  Следующие управляющие коды гарантируют, что во время операций JUMP или JSR
                ;2026    ;  не будут разрушены никакие данные внутри K1804BC1
                ;2027    ;
                ;2028    020:
                ;2029    JUMP:
```

```
C 0020, 3CB      ;2030          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0021, 3CB      ;2031          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0022, 3CB      ;2032          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0023, 3CB      ;2033          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0024, 3CB      ;2034          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0025, 3CB      ;2035          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0026, 3CB      ;2036          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0027, 3CB      ;2037          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0028, 3CB      ;2038          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0029, 3CB      ;2039          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002A, 3CB      ;2040          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002B, 3CB      ;2041          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002C, 3CB      ;2042          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002D, 3CB      ;2043          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002E, 3CB      ;2044          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 002F, 3CB      ;2045          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0030, 3CB      ;2046          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0031, 3CB      ;2047          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0032, 3CB      ;2048          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0033, 3CB      ;2049          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0034, 3CB      ;2050          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0035, 3CB      ;2051          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0036, 3CB      ;2052          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0037, 3CB      ;2053          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0038, 3CB      ;2054          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 0039, 3CB      ;2055          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003A, 3CB      ;2056          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003B, 3CB      ;2057          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003C, 3CB      ;2058          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003D, 3CB      ;2059          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003E, 3CB      ;2060          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
C 003F, 3CB      ;2061          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/CIN
                 ;2062  ;
                 ;2063  ;
                 ;2064  ;  Эти ячейки используются микроинструкцией MISC
                 ;2065  ;
                 ;2066  ;  Следующие управляющие коды гарантируют, что во время выполнения инструкций
                 ;2067  ;  MISC не будут разрушены никакие данные внутри K1804BC1
                 ;2068  ;
                 ;2069  040:
                 ;2070  MISC:
C 0040, 313      ;2071          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0041, 313      ;2072          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0042, 313      ;2073          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0043, 313      ;2074          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0044, 313      ;2075          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0045, 313      ;2076          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0046, 313      ;2077          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0047, 313      ;2078          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0048, 313      ;2079          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0049, 313      ;2080          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 004A, 313      ;2081          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 004B, 313      ;2082          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 004C, 313      ;2083          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 004D, 313      ;2084          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
```

```
004E, 313      ;2085          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
004F, 313      ;2086          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0050, 313      ;2087          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0051, 313      ;2088          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0052, 313      ;2089          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0053, 313      ;2090          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0054, 313      ;2091          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0055, 313      ;2092          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0056, 313      ;2093          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0057, 313      ;2094          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0058, 313      ;2095          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
0059, 313      ;2096          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005A, 313      ;2097          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005B, 313      ;2098          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005C, 313      ;2099          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005D, 313      ;2100          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005E, 313      ;2101          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
005F, 313      ;2102          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
               ;2103
               ;2104
               ;2105      Эти адреса используются микроинструкцией MEM.REQ
               ;2106
               ;2107  ;   Этой инструкцией вызывается загрузка WR[5] виртуальным адресом обращения к
               ;2108  ;   памяти.
               ;2109  ;
               ;2110  060:
               ;2111  MEM.REQ
0060, 3DB      ;2112          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0061, 3DB      ;2113          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0062, 3DB      ;2114          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0063, 3DB      ;2115          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0064, 3DB      ;2116          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0065, 3DB      ;2117          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0066, 3DB      ;2118          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0067, 3DB      ;2119          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0068, 3DB      ;2120          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0069, 3DB      ;2121          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006A, 3DB      ;2122          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006B, 3DB      ;2123          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006C, 3DB      ;2124          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006D, 3DB      ;2125          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006E, 3DB      ;2126          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
006F, 3DB      ;2127          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0070, 3DB      ;2128          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0071, 3DB      ;2129          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0072, 3DB      ;2130          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0073, 3DB      ;2131          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0074, 3DB      ;2132          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0075, 3DB      ;2133          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0076, 3DB      ;2134          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0077, 3DB      ;2135          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0078, 3DB      ;2136          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
0079, 3DB      ;2137          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
007A, 3DB      ;2138          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
007B, 3DB      ;2139          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
```

```
C 007C, 3DB      ;2140          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 007D, 3DB      ;2141          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 007E, 3DB      ;2142          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 007F, 3DB      ;2143          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                 ;2144  ...;
                 ;2145  ;  Эти адреса, начиная от 80 до BF, являются адресами для микроинструкций EXTENDED
                 ;2146  ;
                 ;2147  080:
                 ;2148  WR.PLUS.0.TO.WR:
C 0080, 118      ;2149          SRC/0.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
                 ;2150  WR.INC.WR:
C 0081, 318      ;2151          SRC/0.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
                 ;2152  WR.NEG.WR:
C 0082, 31A      ;2153          SRC/0.A,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
                 ;2154  WR.COM.WR:
C 0083, 31F      ;2155          SRC/0.A,ALU/R.XNOR.S,DST/WRITE.B.F,CIN/CIN
                 ;2156  WR.DEC.WR:
C 0084, 119      ;2157          SRC/0.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
                 ;2158  FILL85:
C 0085, 108      ;2159          SRC/0.A
                 ;2160  MOV.Q.WR:
C 0086, 298      ;2161          SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                 ;2162  FILL87:
C 0087, 088      ;2163          SRC/0.Q
                 ;2164  COND.ADD&SHIFT:
C 0088, 060      ;2165          SRC/A.B,ALU/R.PLUS.S,DST/RSHF.RAM.Q,CIN/NO.CIN
                 ;2166  COND.SUB&SHIFT:
C 0089, 261      ;2167          SRC/A.B,ALU/S.MINUS.R,DST/RSHF.RAM.Q,CIN/CIN
                 ;2168  FILL8A:
C 008A, 048      ;2169          SRC/A.B
                 ;2170  SHL2:
C 008B, 078      ;2171          SRC/A.B,ALU/R.PLUS.S,DST/LSHF.RAM,CIN/NO.CIN
                 ;2172  ASHRC:
C 008C, 0E3      ;2173          SRC/0.B,ALU/R.OR.S,DST/RSHF.RAM.Q,CIN/NO.CIN
                 ;2174  ASHR:
C 008D, 2EB      ;2175          SRC/0.B,ALU/R.OR.S,DST/RSHF.RAM,CIN/CIN
                 ;2176  ASHLC:
C 008E, 2F3      ;2177          SRC/0.B,ALU/R.OR.S,DST/LSHF.RAM.Q,CIN/CIN
                 ;2178  ASHL:
C 008F, 2FB      ;2179          SRC/0.B,ALU/R.OR.S,DST/LSHF.RAM,CIN/CIN
                 ;2180  Q.PLUS.0:
C 0090, 088      ;2181          SRC/0.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
                 ;2182  FILL91:
C 0091, 088      ;2183          SRC/0.Q
                 ;2184  WR.CMP.Q:
C 0092, 20A      ;2185          SRC/A.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                 ;2186  Q.CMP.WR:
C 0093, 209      ;2187          SRC/A.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                 ;2188  DEC.Q:
C 0094, 081      ;2189          SRC/0.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/NO.CIN
                 ;2190  INC.Q:
C 0095, 280      ;2191          SRC/0.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/CIN
                 ;2192  NEG.Q:
C 0096, 282      ;2193          SRC/0.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
                 ;2194  COM.Q:
```

```
C 0097, 287      ;2195          SRC/0.Q,ALU/R.XNOR.S,DST/LOAD.Q,CIN/CIN
                 ;2196   WR.BIT.WR:
C 0098, 04C      ;2197          SRC/A.B,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
                 ;2198   WR.CMP.WR:
C 0099, 24A      ;2199          SRC/A.B,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                 ;2200   FILL9A:
C 009A, 04B      ;2201          SRC/A.B
                 ;2202   WR.PLUS.WR+1:
C 009B, 258      ;2203          SRC/A.B,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
                 ;2204   FILL9C:
C 009C, 04B      ;2205          SRC/A.B
                 ;2206   FILL9D:
C 009D, 04B      ;2207          SRC/A.B
                 ;2208   FILL9E:
C 009E, 0CB      ;2209          SRC/0.B
                 ;2210   FILL9F:
C 009F, 0CB      ;2211          SRC/0.B
                 ;2212   WR.PLUS.Q:
C 00A0, 000      ;2213          SRC/A.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
                 ;2214   WR.FROM.Q:
C 00A1, 201      ;2215          SRC/A.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
                 ;2216   Q.FROM.WR.Q:
C 00A2, 202      ;2217          SRC/A.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
                 ;2218   WR.OR.Q:
C 00A3, 203      ;2219          SRC/A.Q,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
                 ;2220   WR.AND.Q:
C 00A4, 004      ;2221          SRC/A.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
                 ;2222   WR.MASK.Q:
C 00A5, 205      ;2223          SRC/A.Q,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/CIN
                 ;2224   WR.XOR.Q:
C 00A6, 206      ;2225          SRC/A.Q,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
                 ;2226   FILLA7:
C 00A7, 00B      ;2227          SRC/A.Q
                 ;2228   WR.PLUS.WR.Q:
C 00A8, 040      ;2229          SRC/A.B,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
                 ;2230   WR.FROM.WR.Q:
C 00A9, 241      ;2231          SRC/A.B,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
                 ;2232   FILLAA:
C 00AA, 04B      ;2233          SRC/A.B
                 ;2234   WR.OR.WR.Q:
C 00AB, 243      ;2235          SRC/A.B,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
                 ;2236   WR.AND.WR.Q:
C 00AC, 044      ;2237          SRC/A.B,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
                 ;2238   WR.MASK.WR.Q:
C 00AD, 245      ;2239          SRC/A.B,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/CIN
                 ;2240   WR.XOR.WR.Q:
C 00AE, 246      ;2241          SRC/A.B,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
                 ;2242   FILLAF:
C 00AF, 04B      ;2243          SRC/A.B
                 ;2244   Q.PLUS.WR:
C 00B0, 018      ;2245          SRC/A.Q,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
                 ;2246   WR.FROM.Q.WR:
C 00B1, 219      ;2247          SRC/A.Q,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
                 ;2248   Q.FROM.WR:
C 00B2, 21A      ;2249          SRC/A.Q,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
```

```
                    ;2250   Q.OR.WR:
C 00B3, 21B         ;2251           SRC/A.Q,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                    ;2252   Q.AND.WR:
C 00B4, 01C         ;2253           SRC/A.Q,ALU/R.AND.S,DST/WRITE.B.F,CIN/NO.CIN
                    ;2254   FILLB5:
C 00B5, 00B         ;2255           SRC/A.Q
                    ;2256   Q.XOR.WR:
C 00B6, 21E         ;2257           SRC/A.Q,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
                    ;2258   Q.BIT.WR:
C 00B7, 20C         ;2259           SRC/A.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
                    ;2260   WR.PLUS.WR:
C 00B8, 058         ;2261           SRC/A.B,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
                    ;2262   WR.FROM.WR:
C 00B9, 259         ;2263           SRC/A.B,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
                    ;2264   WR.FROM.WR.WR:
C 00BA, 25A         ;2265           SRC/A.B,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
                    ;2266   WR.OR.WR:
C 00BB, 258         ;2267           SRC/A.B,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                    ;2268   WR.FROM.WR-1:
C 00BC, 059         ;2269           SRC/A.B,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
                    ;2270   WR.MASK.WR:
C 00BD, 25D         ;2271           SRC/A.B,ALU/NOTR.AND.S,DST/WRITE.B.F,CIN/CIN
                    ;2272   WR.XOR.WR:
C 00BE, 25E         ;2273           SRC/A.B,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
                    ;2274   WR.AND.WR:
C 00BF, 25C         ;2275           SRC/A.B,ALU/R.AND.S,DST/WRITE.B.F,CIN/CIN
                    ;2276   ;
                    ;2277   ;
                    ;2278   ;  Здесь представлены управляющие коды путей данных для микроинструкций MOVE
                    ;2279   ;
                    ;2280   0C0:
                    ;2281   MOV.MEM.WR:
C 00C0, 1D3         ;2282           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C1, 1D3         ;2283           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C2, 1D3         ;2284           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C3, 1D3         ;2285           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C4, 1D3         ;2286           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C5, 1D3         ;2287           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C6, 1D3         ;2288           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 00C7, 1D3         ;2289           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
                    ;2290   MOV.LS.MEM:
C 00C8, 1CB         ;2291           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00C9, 1CB         ;2292           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CA, 1CB         ;2293           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CB, 1CB         ;2294           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CC, 1CB         ;2295           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CD, 1CB         ;2296           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CE, 1CB         ;2297           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00CF, 1CB         ;2298           SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
                    ;2299   MOV.XWR.Q:
C 00D0, 0C3         ;2300           SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D1, 0C3         ;2301           SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D2, 0C3         ;2302           SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D3, 0C3         ;2303           SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D4, 0C3         ;2304           SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
```

```
C 00D5,  0C3      ;2305          SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D6,  0C3      ;2306          SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 00D7,  0C3      ;2307          SRC/0.B,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
                  ;2308    MOV.LS.WR:
C 00D8,  1DB      ;2309          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00D9,  1DB      ;2310          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DA,  1DB      ;2311          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DB,  1DB      ;2312          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DC,  1DB      ;2313          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DD,  1DB      ;2314          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DE,  1DB      ;2315          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
C 00DF,  1DB      ;2316          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/NO.CIN
                  ;2317    MOV.MEM.LS:
C 00E0,  1CB      ;2318          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E1,  1CB      ;2319          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E2,  1CB      ;2320          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E3,  1CB      ;2321          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E4,  1CB      ;2322          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E5,  1CB      ;2323          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E6,  1CB      ;2324          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E7,  1CB      ;2325          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
                  ;2326    MOV.ACC.LS:
C 00E8,  1CB      ;2327          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00E9,  1CB      ;2328          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00EA,  1CB      ;2329          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00EB,  1CB      ;2330          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00EC,  1CB      ;2331          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00ED,  1CB      ;2332          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00EE,  1CB      ;2333          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00EF,  1CB      ;2334          SRC/D.0,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
                  ;2335    WR.PLUS.OS:
C 00F0,  148      ;2336          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F1,  148      ;2337          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F2,  148      ;2338          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F3,  148      ;2339          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F4,  148      ;2340          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F5,  148      ;2341          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F6,  148      ;2342          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 00F7,  148      ;2343          SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
                  ;2344    MOV.WR.LS:
C 00F8,  10B      ;2345          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00F9,  10B      ;2346          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FA,  10B      ;2347          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FB,  10B      ;2348          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FC,  10B      ;2349          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FD,  10B      ;2350          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FE,  10B      ;2351          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
C 00FF,  10B      ;2352          SRC/0.A,ALU/R.OR.S,DST/NOP,CIN/NO.CIN
                  ;2353    ;
                  ;2354    ;
                  ;2355    ;  Эта группа ячеек предназначена для микроинструкций BASIC.
                  ;2356    ;  Она использует адреса от 100 до 1FF
                  ;2357    ;
                  ;2358    100:
                  ;2359    LE PLUS.WR:
```

```
C 0100, 158     ;2360          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0101, 158     ;2361          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0102, 158     ;2362          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0103, 158     ;2363          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
                ;2364    LS.FROM.WR:
C 0104, 359     ;2365          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 0105, 359     ;2366          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 0106, 359     ;2367          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 0107, 359     ;2368          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
                ;2369    LS.AND.WR:
C 0108, 35C     ;2370          SRC/D.A,ALU/R.AND.S,DST/WRITE.B.F,CIN/CIN
C 0109, 35C     ;2371          SRC/D.A,ALU/R.AND.S,DST/WRITE.B.F,CIN/CIN
C 010A, 35C     ;2372          SRC/D.A,ALU/R.AND.S,DST/WRITE.B.F,CIN/CIN
C 010B, 35C     ;2373          SRC/D.A,ALU/R.AND.S,DST/WRITE.B.F,CIN/CIN
                ;2374    LS.XOR.WR:
C 010C, 35E     ;2375          SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
C 010D, 35E     ;2376          SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
C 010E, 35E     ;2377          SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
C 010F, 35E     ;2378          SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.F,CIN/CIN
                ;2379    LS.FROM.WR-1:
C 0110, 159     ;2380          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
C 0111, 159     ;2381          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
C 0112, 159     ;2382          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
C 0113, 159     ;2383          SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/NO.CIN
                ;2384    LS.MASK.WR:
C 0114, 35D     ;2385          SRC/D.A,ALU/NOTR.AND.S,DST/WRITE.B.F,CIN/CIN
C 0115, 35D     ;2386          SRC/D.A,ALU/NOTR.AND.S,DST/WRITE.B.F,CIN/CIN
C 0116, 35D     ;2387          SRC/D.A,ALU/NOTR.AND.S,DST/WRITE.B.F,CIN/CIN
C 0117, 35D     ;2388          SRC/D.A,ALU/NOTR.AND.S,DST/WRITE.B.F,CIN/CIN
                ;2389    LS.PLUS.WR+1:
C 0118, 358     ;2390          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 0119, 358     ;2391          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 011A, 358     ;2392          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
C 011B, 358     ;2393          SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/CIN
                ;2394    LS.OR.WR:
C 011C, 35B     ;2395          SRC/D.A,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 011D, 35B     ;2396          SRC/D.A,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 011E, 35B     ;2397          SRC/D.A,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 011F, 35B     ;2398          SRC/D.A,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                ;2399    WR.PLUS.LS.Q:
C 0120, 140     ;2400          SRC/D.A,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0121, 140     ;2401          SRC/D.A,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0122, 140     ;2402          SRC/D.A,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0123, 140     ;2403          SRC/D.A,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
                ;2404    LS.FROM.WR.Q:
C 0124, 341     ;2405          SRC/D.A,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0125, 341     ;2406          SRC/D.A,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0126, 341     ;2407          SRC/D.A,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0127, 341     ;2408          SRC/D.A,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
                ;2409    WR.FROM.LS.Q:
C 0128, 342     ;2410          SRC/D.A,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 0129, 342     ;2411          SRC/D.A,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 012A, 342     ;2412          SRC/D.A,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 012B, 342     ;2413          SRC/D.A,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
                ;2414    WR.OR.LS.Q:
```

```
C 012C, 343     ;2415          SRC/D.A,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 012D, 343     ;2416          SRC/D.A,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 012E, 343     ;2417          SRC/D.A,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 012F, 343     ;2418          SRC/D.A,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
                ;2419   WR.AND.LS.Q:
C 0130, 144     ;2420          SRC/D.A,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0131, 144     ;2421          SRC/D.A,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0132, 144     ;2422          SRC/D.A,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0133, 144     ;2423          SRC/D.A,ALU/R.AND.S,DST/LOAD.Q,CIN/NO.CIN
                ;2424   WR.XOR.LS.Q:
C 0134, 346     ;2425          SRC/D.A,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0135, 346     ;2426          SRC/D.A,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0136, 346     ;2427          SRC/D.A,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0137, 346     ;2428          SRC/D.A,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
                ;2429   LS.CMP.WR:
C 0138, 34A     ;2430          SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 0139, 34A     ;2431          SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 013A, 34A     ;2432          SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 013B, 34A     ;2433          SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                ;2434   WR.CMP.LS:
C 013C, 349     ;2435          SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 013D, 349     ;2436          SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 013E, 349     ;2437          SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 013F, 349     ;2438          SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                ;2439   LS.PLUS.Q:
C 0140, 180     ;2440          SRC/D.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0141, 180     ;2441          SRC/D.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0142, 180     ;2442          SRC/D.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
C 0143, 180     ;2443          SRC/D.Q,ALU/R.PLUS.S,DST/LOAD.Q,CIN/NO.CIN
                ;2444   LS.FROM.Q:
C 0144, 381     ;2445          SRC/D.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0145, 381     ;2446          SRC/D.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0146, 381     ;2447          SRC/D.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
C 0147, 381     ;2448          SRC/D.Q,ALU/S.MINUS.R,DST/LOAD.Q,CIN/CIN
                ;2449   Q.FROM.LS.Q:
C 0148, 382     ;2450          SRC/D.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 0149, 382     ;2451          SRC/D.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 014A, 382     ;2452          SRC/D.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
C 014B, 382     ;2453          SRC/D.Q,ALU/R.MINUS.S,DST/LOAD.Q,CIN/CIN
                ;2454   LS.OR.Q:
C 014C, 383     ;2455          SRC/D.Q,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 014D, 383     ;2456          SRC/D.Q,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 014E, 383     ;2457          SRC/D.Q,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
C 014F, 383     ;2458          SRC/D.Q,ALU/R.OR.S,DST/LOAD.Q,CIN/CIN
                ;2459   LS.MASK.Q:
C 0150, 185     ;2460          SRC/D.Q,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0151, 185     ;2461          SRC/D.Q,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0152, 185     ;2462          SRC/D.Q,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/NO.CIN
C 0153, 185     ;2463          SRC/D.Q,ALU/NOTR.AND.S,DST/LOAD.Q,CIN/NO.CIN
                ;2464   LS.XOR.Q:
C 0154, 386     ;2465          SRC/D.Q,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0155, 386     ;2466          SRC/D.Q,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0156, 386     ;2467          SRC/D.Q,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
C 0157, 386     ;2468          SRC/D.Q,ALU/R.XOR.S,DST/LOAD.Q,CIN/CIN
                ;2469   LS.AND.Q:
```

```
C 0158,  384      ;2470           SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 0159,  384      ;2471           SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 015A,  384      ;2472           SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 015B,  384      ;2473           SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
                  ;2474   LS.BIT.Q:
C 015C,  38C      ;2475           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015D,  38C      ;2476           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015E,  38C      ;2477           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015F,  38C      ;2478           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
                  ;2479   MOV.LS.Q:
C 0160,  1C3      ;2480           SRC/D.0,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 0161,  1C3      ;2481           SRC/D.0,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 0162,  1C3      ;2482           SRC/D.0,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
C 0163,  1C3      ;2483           SRC/D.0,ALU/R.OR.S,DST/LOAD.Q,CIN/NO.CIN
                  ;2484   WR.BIT.LS:
C 0164,  14C      ;2485           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 0165,  14C      ;2486           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 0166,  14C      ;2487           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 0167,  14C      ;2488           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
                  ;2489   LS.CMP.Q:
C 0168,  38A      ;2490           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 0169,  38A      ;2491           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 016A,  38A      ;2492           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 016B,  38A      ;2493           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                  ;2494   Q.CMP.LS:
C 016C,  389      ;2495           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 016D,  389      ;2496           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 016E,  389      ;2497           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 016F,  389      ;2498           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                  ;2499   Q.PLUS.LS.TO.WR:
C 0170,  198      ;2500           SRC/D.Q,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0171,  198      ;2501           SRC/D.Q,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0172,  198      ;2502           SRC/D.Q,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
C 0173,  198      ;2503           SRC/D.Q,ALU/R.PLUS.S,DST/WRITE.B.F,CIN/NO.CIN
                  ;2504   WRA.FROM.LS.WRB:
C 0174,  35A      ;2505           SRC/D.A,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
C 0175,  35A      ;2506           SRC/D.A,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
C 0176,  35A      ;2507           SRC/D.A,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
C 0177,  35A      ;2508           SRC/D.A,ALU/R.MINUS.S,DST/WRITE.B.F,CIN/CIN
                  ;2309   LS.NEG.WR:
C 0178,  3D9      ;2510           SRC/D.0,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 0179,  3D9      ;2511           SRC/D.0,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 017A,  3D9      ;2512           SRC/D.0,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
C 017B,  3D9      ;2513           SRC/D.0,ALU/S.MINUS.R,DST/WRITE.B.F,CIN/CIN
                  ;2514   LS.COM.WR:
C 017C,  3DF      ;2515           SRC/D.0,ALU/R.XNOR.S,DST/WRITE.B.F,CIN/CIN
C 017D,  3DF      ;2516           SRC/D.0,ALU/R.XNOR.S,DST/WRITE.B.F,CIN/CIN
C 017E,  3DF      ;2517           SRC/D.0,ALU/R.XNOR.S,DST/WRITE.B.F,CIN/CIN
C 017F,  3DF      ;2518           SRC/D.0,ALU/R.XNOR.S,DST/WRITE.B.F,CIN/CIN
                  ;2519   ;
                  ;2520   ; Следующие ячейки задают в качестве приемника местную память.
                  ;2521   ; Этот факт используется аппаратурой для генерации импульсов записи
                  ;2522   ; для местной памяти
                  ;2523   ;
                  ;2524   180:
```

```
                        ;2525   LS.PLUS.WR.XCHG:
C 0180,  150            ;2526           SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 0181,  150            ;2527           SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 0182,  150            ;2528           SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 0183,  150            ;2529           SRC/D.A,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
                        ;2530   LS.FROM.WR.XCHG:
C 0184,  351            ;2531           SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.A,CIN/CIN
C 0185,  351            ;2532           SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.A,CIN/CIN
C 0186,  351            ;2533           SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.A,CIN/CIN
C 0187,  351            ;2534           SRC/D.A,ALU/S.MINUS.R,DST/WRITE.B.A,CIN/CIN
                        ;2535   LS.AND.WR.XCHG:
C 0188,  354            ;2536           SRC/D.A,ALU/R.AND.S,DST/WRITE.B.A,CIN/CIN
C 0189,  354            ;2537           SRC/D.A,ALU/R.AND.S,DST/WRITE.B.A,CIN/CIN
C 018A,  354            ;2538           SRC/D.A,ALU/R.AND.S,DST/WRITE.B.A,CIN/CIN
C 018B,  354            ;2539           SRC/D.A,ALU/R.AND.S,DST/WRITE.B.A,CIN/CIN
                        ;2540   LS.XOR.WR.XCHG:
C 018C,  356            ;2541           SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.A,CIN/CIN
C 018D,  356            ;2542           SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.A,CIN/CIN
C 018E,  356            ;2543           SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.A,CIN/CIN
C 018F,  356            ;2544           SRC/D.A,ALU/R.XOR.S,DST/WRITE.B.A,CIN/CIN
                        ;2545   SWAP.LS.WR:
C 0190,  1D3            ;2546           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 0191,  1D3            ;2547           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 0192,  1D3            ;2548           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
C 0193,  1D3            ;2549           SRC/D.0,ALU/R.OR.S,DST/WRITE.B.A,CIN/NO.CIN
                        ;2550   CLR.LS:
C 0194,  3CC            ;2551           SRC/D.0,ALU/R.AND.S,DST/NOP,CIN/CIN
C 0195,  3CC            ;2552           SRC/D.0,ALU/R.AND.S,DST/NOP,CIN/CIN
C 0196,  3CC            ;2553           SRC/D.0,ALU/R.AND.S,DST/NOP,CIN/CIN
C 0197,  3CC            ;2554           SRC/D.0,ALU/R.AND.S,DST/NOP,CIN/CIN
                        ;2555   MOV.Q.WR&WR.LS:
C 0198,  293            ;2556           SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0199,  293            ;2557           SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 019A,  293            ;2558           SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 019B,  293            ;2559           SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
                        ;2560   WR.PLUS.Q.XCHG:
C 019C,  010            ;2561           SRC/A.Q,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 019D,  010            ;2562           SRC/A.Q,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 019E,  010            ;2563           SRC/A.Q,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
C 019F,  010            ;2564           SRC/A.Q,ALU/R.PLUS.S,DST/WRITE.B.A,CIN/NO.CIN
                        ;2565   WR.FROM.LS-1:
C 01A0,  14A            ;2566           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01A1,  14A            ;2567           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01A2,  14A            ;2568           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01A3,  14A            ;2569           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
                        ;2570   LS.FROM.WR.LS:
C 01A4,  349            ;2571           SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01A5,  349            ;2572           SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01A6,  349            ;2573           SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01A7,  349            ;2574           SRC/D.A,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                        ;2575   WR.PLUS.LS+1:
C 01A8,  348            ;2576           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01A9,  348            ;2577           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01AA,  348            ;2578           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01AB,  348            ;2579           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/CIN
```

```
                        ;2580   WR.FROM.LS:
C 01AC,  34A            ;2581           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01AD,  34A            ;2582           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01AE,  34A            ;2583           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01AF,  34A            ;2584           SRC/D.A,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                        ;2585   WR.PLUS.LS:
C 01B0,  148            ;2586           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01B1,  148            ;2587           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01B2,  148            ;2588           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01B3,  148            ;2589           SRC/D.A,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
                        ;2590   WR.OR.LS:
C 01B4,  348            ;2591           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B5,  348            ;2592           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B6,  348            ;2593           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B7,  348            ;2594           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
                        ;2595   WR.AND.LS:
C 01B8,  340            ;2596           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01B9,  340            ;2597           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01BA,  340            ;2598           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01BB,  340            ;2599           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
                        ;2600   WR.XOR.LS:
C 01BC,  34E            ;2601           SRC/D.A,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01BD,  34E            ;2602           SRC/D.A,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01BE,  34E            ;2603           SRC/D.A,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01BF,  34E            ;2604           SRC/D.A,ALU/R.XOR.S,DST/NOP,CIN/CIN
                        ;2605   Q.PLUS.LS:
C 01C0,  188            ;2606           SRC/D.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01C1,  188            ;2607           SRC/D.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01C2,  188            ;2608           SRC/D.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01C3,  188            ;2609           SRC/D.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
                        ;2610   LS.FROM.Q.LS:
C 01C4,  389            ;2611           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01C5,  389            ;2612           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01C6,  389            ;2613           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01C7,  389            ;2614           SRC/D.Q,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                        ;2615   Q.FROM.LS:
C 01C8,  38A            ;2616           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01C9,  38A            ;2617           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01CA,  38A            ;2618           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01CB,  38A            ;2619           SRC/D.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                        ;2620   Q.OR.LS:
C 01CC,  38B            ;2621           SRC/D.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01CD,  38B            ;2622           SRC/D.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01CE,  38B            ;2623           SRC/D.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01CF,  38B            ;2624           SRC/D.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
                        ;2625   Q.AND.LS:
C 01D0,  18C            ;2626           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 01D1,  18C            ;2627           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 01D2,  18C            ;2628           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
C 01D3,  18C            ;2629           SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/NO.CIN
                        ;2630   Q.XOR.LS:
C 01D4,  38E            ;2631           SRC/D.Q,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01D5,  38E            ;2632           SRC/D.Q,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01D6,  38E            ;2633           SRC/D.Q,ALU/R.XOR.S,DST/NOP,CIN/CIN
C 01D7,  38E            ;2634           SRC/D.Q,ALU/R.XOR.S,DST/NOP,CIN/CIN
```

```
                    ;2635   MOV.Q.LS:
C 01D8,  28B        ;2636           SRC/0.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01D9,  28B        ;2637           SRC/0.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01DA,  28B        ;2638           SRC/0.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01DB,  28B        ;2639           SRC/0.Q,ALU/R.OR.S,DST/NOP,CIN/CIN
                    ;2640   Q.NEG.LS:
C 01DC,  28A        ;2641           SRC/0.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01DD,  28A        ;2642           SRC/0.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01DE,  28A        ;2643           SRC/0.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01DF,  28A        ;2644           SRC/0.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                    ;2645   WR.COM.LS:
C 01E0,  0CF        ;2646           SRC/0.B,ALU/R.XNOR.S,DST/NOP,CIN/NO.CIN
C 01E1,  0CF        ;2647           SRC/0.B,ALU/R.XNOR.S,DST/NOP,CIN/NO.CIN
C 01E2,  0CF        ;2648           SRC/0.B,ALU/R.XNOR.S,DST/NOP,CIN/NO.CIN
C 01E3,  0CF        ;2649           SRC/0.B,ALU/R.XNOR.S,DST/NOP,CIN/NO.CIN
                    ;2650   WR.NEG.LS:
C 01E4,  2CA        ;2651           SRC/0.B,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01E5,  2CA        ;2652           SRC/0.B,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01E6,  2CA        ;2653           SRC/0.B,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01E7,  2CA        ;2654           SRC/0.B,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                    ;2655   Q.PLUS.WR.TO.LS:
C 01E8,  008        ;2656           SRC/A.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01E9,  008        ;2657           SRC/A.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01EA,  008        ;2658           SRC/A.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
C 01EB,  008        ;2659           SRC/A.Q,ALU/R.PLUS.S,DST/NOP,CIN/NO.CIN
                    ;2660   Q.FROM.WR.TO.LS:
C 01EC,  20A        ;2661           SRC/A.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01ED,  20A        ;2662           SRC/A.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01EE,  20A        ;2663           SRC/A.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
C 01EF,  20A        ;2664           SRC/A.Q,ALU/R.MINUS.S,DST/NOP,CIN/CIN
                    ;2665   DEC.LS:
C 01F0,  1CA        ;2666           SRC/D.0,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01F1,  1CA        ;2667           SRC/D.0,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01F2,  1CA        ;2668           SRC/D.0,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
C 01F3,  1CA        ;2669           SRC/D.0,ALU/R.MINUS.S,DST/NOP,CIN/NO.CIN
                    ;2670   COM.LS:
C 01F4,  3CF        ;2671           SRC/D.0,ALU/R.XNOR.S,DST/NOP,CIN/CIN
C 01F5,  3CF        ;2672           SRC/D.0,ALU/R.XNOR.S,DST/NOP,CIN/CIN
C 01F6,  3CF        ;2673           SRC/D.0,ALU/R.XNOR.S,DST/NOP,CIN/CIN
C 01F7,  3CF        ;2674           SRC/D.0,ALU/R.XNOR.S,DST/NOP,CIN/CIN
                    ;2675   NEG.LS:
C 01F8,  3C9        ;2676         - SRC/D.0,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01F9,  3C9        ;2677           SRC/D.0,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01FA,  3C9        ;2678           SRC/D.0,ALU/S.MINUS.R,DST/NOP,CIN/CIN
C 01FB,  3C9        ;2679           SRC/D.0,ALU/S.MINUS.R,DST/NOP,CIN/CIN
                    ;2680   INC.LS:
C 01FC,  3C8        ;2681           SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01FD,  3C8        ;2682           SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01FE,  3C8        ;2683           SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
C 01FF,  3C8        ;2684           SRC/D.0,ALU/R.PLUS.S,DST/NOP,CIN/CIN
                    ;2685   ;
                    ;2686      .UCODE
                    ;2687      .SEQUENTIAL
```

```
                    ;2688      .PAGE "УКАЗАТЕЛИ ТЕСТОВ"
                    ;2689      ;
                    ;2690      ;   Эта часть содержит указатели для всех тестов данного сегмента. Адрес WCS для
                    ;2691      ; инструкции JUMP соответствует номеру теста, т.е., WCS 5 содержит JUMP к тесту 5.
                    ;2692      ; Все неиспользованные номера тестов содержат переход к программе обработки оши-
                    ;2693      ; бок. Эта часть имеет длину 80 ячеек, что позволяет иметь до 80 тестов на сег-
                    ;2694      ; мент, начиная от адреса 1 WCS до адреса 4F.
                    ;2695      ;
                    ;2696     1:                                                  ; указатели тестов начинаются адресом WCS 1
U 0001, 8871,94     ;2697              JMP [T.1]                                  ; переход к тесту 1
U 0002, 8952,04     ;2698              JMP [T.2]                                  ; переход к тесту 2
U 0003, 0956,F4     ;2699              JMP [T.3]                                  ; переход к тесту 3
U 0004, 895A,24     ;2700              JMP [T.4]                                  ; переход к тесту 4
U 0005, 8960,14     ;2701              JMP [T.5]                                  ; переход к тесту 5
U 0006, 8965,24     ;2702              JMP [T.6]                                  ; переход к тесту 6
U 0007, 8972,14     ;2703              JMP [T.7]                                  ; переход к тесту 7
U 0008, 9978,74     ;2704              JMP [T.8]                                  ; переход к тесту 8
U 0009, 8983,54     ;2705              JMP [T.9]                                  ; переход к тесту 9
U 000A, 0985,24     ;2706              JMP [T.A]                                  ; переход к тесту A
U 0008, 898A,64     ;2707              JMP [T.B]                                  ; переход к тесту B
U 000C, 098C,B4     ;2708              JMP [T.C]                                  ; переход к тесту C
U 000D, 8993,B4     ;2709              JMP [T.D]                                  ; переход к тесту D
U 000E, 8996,74     ;2710              JMP [T.E]                                  ; переход к тесту E
U 000F, 0997,E4     ;2711              JMP [T.F]                                  ; переход к тесту F
U 0010, 886B,E4     ;2712              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2713                                                         ; в данном сегменте
U 0011, 886B,E4     ;2714              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2715                                                         ; в данном сегменте
U 0012, 886B,E4     ;2716              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2717                                                         ; в данном сегменте
U 0013, 886B,E4     ;2718              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2719                                                         ; в данном сегменте
U 0014, 886B,E4     ;2720              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2721                                                         ; в данном сегменте
U 0015, 886B,E4     ;2722              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2723                                                         ; в данном сегменте
U 0016, 886B,E4     ;2724              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2725                                                         ; в данном сегменте
U 0017, 886B,E4     ;2726              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2727                                                         ; в данном сегменте
U 0018, 886B,E4     ;2728              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2729                                                         ; в данном сегменте
U 0019, 886B,E4     ;2730              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2731                                                         ; в данном сегменте
U 001A, 886B,E4     ;2732              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2733                                                         ; в данном сегменте
U 001B, 886B,E4     ;2734              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2735                                                         ; в данном сегменте
U 001C, 886B,E4     ;2736              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2737                                                         ; в данном сегменте
U 001D, 886B,E4     ;2738              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2739                                                         ; в данном сегменте
U 001E, 886B,E4     ;2740              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
                    ;2741                                                         ; в данном сегменте
U 001F, 886B,E4     ;2742              JMP [ERR.NO.TEST]                          ; переход к программе обработки ошибок.Тест отсутствует
```

```
                    ;2743                                   ; в данном сегменте
U 0020, 886B,E4     ;2744       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2745                                   ; в данном сегменте
U 0021, 886B,E4     ;2746       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2747                                   ; в данном сегменте
U 0022, 886B,E4     ;2748       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2749                                   ; в данном сегменте
U 0023, 886B,E4     ;2750       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2751                                   ; в данном сегменте
U 0024, 886B,E4     ;2752       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2753                                   ; в данном сегменте
U 0025, 886B,E4     ;2754       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2755                                   ; в данном сегменте
U 0026, 886B,E4     ;2756       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2757                                   ; в данном сегменте
U 0027, 886B,E4     ;2758       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2759                                   ; в данном сегменте
U 0028, 886B,E4     ;2760       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2761                                   ; в данном сегменте
U 0029, 886B,E4     ;2762       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2763                                   ; в данном сегменте
U 002A, 886B,E4     ;2764       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2765                                   ; в данном сегменте
U 002B, 886B,E4     ;2766       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2767                                   ; в данном сегменте
U 002C, 886B,E4     ;2768       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2769                                   ; в данном сегменте
U 002D, 886B,E4     ;2770       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2771                                   ; в данном сегменте
U 002E, 886B,E4     ;2772       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2773                                   ; в данном сегменте
U 002F, 886B,E4     ;2774       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2775                                   ; в данном сегменте
U 0030, 886B,E4     ;2776       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2777                                   ; в данном сегменте
U 0031, 886B,E4     ;2778       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2779                                   ; в данном сегменте
U 0032, 886B,E4     ;2780       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2781                                   ; в данном сегменте
U 0033, 886B,E4     ;2782       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2783                                   ; в данном сегменте
U 0034, 886B,E4     ;2784       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2785                                   ; в данном сегменте
U 0035, 886B,E4     ;2786       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2787                                   ; в данном сегменте
U 0036, 886B,E4     ;2788       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2789                                   ; в данном сегменте
U 0037, 886B,E4     ;2790       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2791                                   ; в данном сегменте
U 0038, 886B,E4     ;2792       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2793                                   ; в данном сегменте
U 0039, 886B,E4     ;2794       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2795                                   ; в данном сегменте
U 003A, 886B,E4     ;2796       JMP [ERR.NO.TEST]           ; переход к программе обработки ошибок.Тест отсутствует
                    ;2797                                   ; в данном сегменте
```

```
U 003B, 886B,E4 ;2798          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2799                                     ; в данном сегменте
U 003C, 886B,E4 ;2800          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2801                                     ; в данном сегменте
U 003D, 886B,E4 ;2802          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2803                                     ; в данном сегменте
U 003E, 886B,E4 ;2804          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2805                                     ; в данном сегменте
U 003F, 886B,E4 ;2806          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2807                                     ; в данном сегменте
U 0040, 886B,E4 ;2808          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2809                                     ; в данном сегменте
U 0041, 886B,E4 ;2810          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2811                                     ; в данном сегменте
U 0042, 886B,E4 ;2812          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2813                                     ; в данном сегменте
U 0043, 886B,E4 ;2814          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2815                                     ; в данном сегменте
U 0044, 886B,E4 ;2816          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2817                                     ; в данном сегменте
U 0045, 886B,E4 ;2818          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2819                                     ; в данном сегменте
  0046, 886B,E4 ;2820          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2821                                     ; в данном сегменте
  0047, 886B,E4 ;2822          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2823                                     ; в данном сегменте
U 0048, 886B,E4 ;2824          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2825                                     ; в данном сегменте
U 0049, 886B,E4 ;2826          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2827                                     ; в данном сегменте
U 004A, 886B,E4 ;2828          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2829                                     ; в данном сегменте
U 004B, 886B,E4 ;2830          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2831                                     ; в данном сегменте
U 004C, 886B,E4 ;2832          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2833                                     ; в данном сегменте
U 004D, 886B,E4 ;2834          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2835                                     ; в данном сегменте
U 004E, 886B,E4 ;2836          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2837                                     ; в данном сегменте
U 004F, 886B,E4 ;2838          JMP [ERR.NO.TEST]          ; переход к программе обработки ошибок.Тест отсутствует
                ;2839                                     ; в данном сегменте
                ;2840
                ;2841
```

```
                ;2842     .PAGE  "ДИАГНОСТИЧЕСКИЕ УКАЗАТЕЛИ"
                ;2843     ;
                ;2844     ;   Этот раздел содержит все указатели, необходимые для данного диагностического
                ;2845     ; сегмента. Первый указатель (по адресу WCS 0) указывает на программу переноса
                ;2846     ; данных. Это первая ячейка, работающая после новой загрузки WCS. Инструкция, раз-
                ;2847     ; мещенная здесь, является переходом (JUMP) к TRANSFER.POINT, где могут содержать-
                ;2848     ; ся инструкции для переноса данных. Если отсутствуют данные, подлежащие переносу,
                ;2849     ; или по завершению переноса данных, центральный процессор выставляет CPU ATTN,
                ;2850     ; причем все биты слова управления и состояния очищены.
                ;2851     ;   Следующие 80 ячеек (от ячейки WCS 1 до ячейки WCS 4F) содержат указатели для
                ;2852     ; каждого теста в этом сегменте. Указателями являются инструкции JUMP к номеру
                ;2853     ; теста, соответствующему номеру ячейки, т.е, ячейка 5 будет содержать переход к
                ;2854     ; тесту 5. Все неиспользованные номера тестов содержат переход к программе обрабо-
                ;2855     ; тки ошибок.
                ;2856     ;   Наконец, следующие 32 ячейки (от ячейки WCS 50 до 6F) содержат указатели
                ;2857     ; (инструкции JUMP) к подпрограммам WCS, которые подлежат использованию диагности-
                ;2858     ; ческим монитором консольного процессора.
                ;2859     ;
                ;2860    0:
U 0000, 086C,24 ;2861             JMP [TRANSFER.POINT]                  ; переход к подпрограмме, которая загружает LS 7
                ;2862                                                   ; указателем секции данных и выдает CPU ATTN с
                ;2863                                                   ; установленным битом переноса данных
                ;2864    50:                                            ; начало указателей
                ;2865                                                   ; подпрограмм в ячейке WCS 50
                ;2866     ;
                ;2867     ;         Указатели подпрограмм
                ;2868     ;
U 0050, 0860,84 ;2869             JMP [DEPOSIT.CSR]                     ; переход к подпрограмме WCS, которая выполняет запись в
                ;2870                                                   ; регистры управления и состояния MCT по команде DEPOSIT
                ;2871                                                   ; CSR
U 0051, 0860,D4 ;2872             JMP [EXAMINE.CSR]                     ; переход к подпрограмме WCS, выполняющей чтение
                ;2873                                                   ; регистров управления и состояния MCT по команде
                ;2874                                                   ; EXAMINE CSR
U 0052, 0861,F4 ;2875             JMP [DEPOSIT.TB]                      ; переход к подпрограмме WCS, выполняющей запись в буфер
                ;2876                                                   ; трансляции MCT по команде DEPOSIT для буфера
                ;2877                                                   ; трансляции
U 0053, 8863,A4 ;2878             JMP [EXAMINE.TB]                      ; переход к подпрограмме WCS, выполняющей чтение буфера
                ;2879                                                   ; трансляции MCT по команде EXAMINE для буфера
                ;2880                                                   ; трансляции
U 0054, 8865,C4 ;2881             JMP [DEPOSIT.MM]                      ; переход к подпрограмме WCS, которая записывает в
                ;2882                                                   ; основную память по команде DEPOSIT для основной
                ;2883                                                   ; памяти. Используется также для пересылки данных из
                ;2884                                                   ; WCS в основную память
U 0055, 0866,24 ;2885             JMP [EXAMINE.MM]                      ; переход к подпрограмме WCS, которая читает из основной
                ;2886                                                   ; памяти по команде EXAMINE для основной памяти
U 0056, 0868,94 ;2887             JMP [SAVE.WR]                         ; переход к подпрограмме WCS, которая запоминает
                ;2888                                                   ; содержимое рабочих регистров WR0-WR3 в местной памяти
                ;2889                                                   ; LS 0-3
U 0057, 8868,E4 ;2890             JMP [RESTORE.WR]                      ; переход к подпрограмме WCS, которая восстанавливает
                ;2891                                                   ; содержимое WR0-WR3 из LS 0-3
U 0058, 8864,24 ;2892             JMP [DEPOSIT.UBS]                     ; переход к подпрограмме WCS, которая записывает в буфер
                ;2893                                                   ; трансляции общей шины контроллера памяти
U 0059, 0865,44 ;2894             JMP [EXAMINE.UBS]                     ; переход к подпрограмме WCS, которая читает из буфера
                ;2895                                                   ; трансляции общей шины контроллера памяти
U 005A, 0866,84 ;2896             JMP [EXAMINE.ICSR]                    ; переход к подпрограмме WCS, которая читает CSR IDC
```

```
U 005B, 0866,B4 ;2897           JMP [EXAMINE.IDAR]              ; переход к подпрограмме WCS, которая читает DAR IDC
U 005C, 0866,E4 ;2898           JMP [EXAMINE.DBUF]              ; переход к подпрограмме WCS, которая читает DBUF IDC
U 005D, 8867,14 ;2899           JMP [EXAMINE.PATT]              ; переход к подпрограмме WCS, которая читает код
                ;2900                                           ; коррекции ECC из IDC
U 005E, 8867,44 ;2901           JMP [EXAMINE.POSIT]             ; переход к подпрограмме WCS, которая читает адрес
                ;2902                                           ; ошибки ECC из IDC
U 005F, 0867,94 ;2903           JMP [DEPOSIT.ICSR]              ; переход к подпрограмме WCS, которая записывает CSR
                ;2904                                           ; IDC
U 0060, 0867,C4 ;2905           JMP [DEPOSIT.IDAR]              ; переход к подпрограмме WCS, которая записывает DAR
                ;2906                                           ; IDC
U 0061, 0867,F4 ;2907           JMP [DEPOSIT.DBUF]              ; переход к подпрограмме WCS, которая записывает DBUF
                ;2908                                           ; IDC
U 0062, 8868,24 ;2909           JMP [CLEAR.FIFO.ADDR]           ; переход к подпрограмме WCS, которая очищает адрес FIFO
                ;2910                                           ; IDC
U 0063, 8868,44 ;2911           JMP [SELECT.FIFO.A]             ; переход к подпрограмме WCS, которая выбирает FIFO A
U 0064, 0868,64 ;2912           JMP [SELECT.FIFO.B]             ; переход к подпрограмме WCS, которая выбирает FIFO B
U 0065, DB00,15 ;2913           NOP                             ; не используется
U 0066, DB00,15 ;2914           NOP                             ; не используется
U 0067, DB00,15 ;2915           NOP                             ; не используется
U 0068, DB00,15 ;2916           NOP                             ; не используется
U 0069, DB00,15 ;2917           NOP                             ; не используется
U 006A, DB00,15 ;2918           NOP                             ; не используется
U 006B, DB00,15 ;2919           NOP                             ; не используется
U 006C, DB00,15 ;2920           NOP                             ; не используется
U 006D, DB00,15 ;2921           NOP                             ; не используется
U 006E, DB00,15 ;2922           NOP                             ; не используется
U 006F, DB00,15 ;2923           NOP                             ; не используется
                ;2924
```

```
                        .PAGE  "СЕКЦИЯ ДАННЫХ МЕСТНОЙ ПАМЯТИ"
                        .REGION/70,5FF
                        ;
                        ;
                        ;       Этот раздел перечисляет данные, которые будут переданы в LS консольным
                        ;   процессором в качестве части процедуры инициации, выполняемой тестом 0.
                        ;   Программа переноса данных ссылается на эту область данных. LS содержит
                        ;   константы, слова управления и состояния и ячейки временного хранения, ис-
                        ;   пользуемые и подпрограммами микродиагностики.
                        ;       LS имеет ширину 32 бита. Каждое из слов здесь имеет ширину 16 битов,
                        ;   так, что два последовательных слова, перечисленные здесь, будут загружать-
                        ;   ся в каждую последовательную ячейку LS. Первое перечисленное слово будет
                        ;   занимать биты 0-15 ячейки LS. Второе перечисленное слово будет составлять
                        ;   биты 16-31 той же ячейки LS.
                        ;       Ячейки 78-7F первой половины LS и F8-FF второй половины LS не являются
                        ;   в действительности ячейками LS. Они задают доступ к одному из нескольких
                        ;   регистров центрального процессора или реализуют средство индексации других
                        ;   ячеек LS. По этой причине они не записываются посредством программы пере-
                        ;   носа данных.
                        ;       ПРИМЕЧАНИЕ: эта таблица задана в шестнадцатеричном формате, т.е. каждый
                        ;   разряд представляется четырьмя битами, так, что четыре разряда представ-
                        ;   ляют полное 16-битовое слово. Микроассемблер требует, чтобы шестнадцате-
                        ;   ричному значению, которое начинается буквой (например, FFFF), предшество-
                        ;   вал 0 (0FFFF). Таким образом, некоторые значения в таблице содержат 5
                        ;   шестнадцатеричных разрядов. Пятый разряд в действительности не использу-
                        ;   ются.
                        ;
                        ;---
                        ;
                        TRANSFER.DATA.LS1:
                        ;
0070. 0000.78           ;       COUNT.LS[0078]                  ; счетчик длинных слов (число длинных слов, подлежащих
                        ;                                       ; пересылке в LS, равно 120, десятичн.) приемником
                        ;                                       ; является LS
0071. 0000.00           ;       START.ADR[0000]                 ; начальный адрес приемника (начинается при LS=0)
                        ;
0072. 0000.00           ;       WORD[0000]                      ; ячейка 0
0073. 0000.00           ;       WORD[0000]                      ;
0074  0000,00           ;       WORD[0000]                      ; ячейка 1
0075  0000,00           ;       WORD[0000]                      ;
0076. 0000.00           ;       WORD[0000]                      ; ячейка 2
0077  0000,00           ;       WORD[0000]                      ;
0078. 0000,00           ;       WORD[0000]                      ; ячейка 3
0079. 0000.00           ;       WORD[0000]                      ;
007A. 0000,00           ;       WORD[0000]                      ; ячейка 4
007B, 0000,00           ;       WORD[0000]                      ;
007C, 0000,00           ;       WORD[0000]                      ; ячейка 5
007D. 0000,00           ;       WORD[0000]                      ;
007E. 0000,00           ;       WORD[0000]                      ; ячейка 6
007F. 0000,00           ;       WORD[0000]                      ;
0080, 0000,00           ;       WORD[0000]                      ; ячейка 7
0081, 0000,00           ;       WORD[0000]                      ;
0082, 0000,00           ;       WORD[0000]                      ; ячейка 8
0083, 0000,00           ;       WORD[0000]                      ;
0084, 0000,00           ;       WORD[0000]                      ; ячейка 9
```

```
U 0085, 0000,00 ;2980          WORD[0000]                    ;
U 0086, 0000,00 ;2981          WORD[0000]                    ; ячейка 0A
U 0087, 0000,00 ;2982          WORD[0000]                    ;
U 0088, 0000,00 ;2983          WORD[0000]                    ; ячейка 0B
U 0089, 0000,00 ;2984          WORD[0000]                    ;
U 008A, 0000,00 ;2985          WORD[0000]                    ; ячейка 0C
U 008B, 0000,00 ;2986          WORD[0000]                    ;
U 008C, 0000,00 ;2987          WORD[0000]                    ; ячейка 0D
U 008D, 0000,00 ;2988          WORD[0000]                    ;
U 008E, 0000,00 ;2989          WORD[0000]                    ; ячейка 0E
U 008F, 0000,00 ;2990          WORD[0000]                    ;
U 0090, 0000,00 ;2991          WORD[0000]                    ; ячейка 0F
U 0091, 0000,00 ;2992          WORD[0000]                    ;
U 0092, 0000,00 ;2993          WORD[0000]                    ; ячейка 10
U 0093, 0000,00 ;2994          WORD[0000]                    ;
U 0094, 0000,00 ;2995          WORD[0000]                    ; ячейка 11
U 0095, 0000,00 ;2996          WORD[0000]
U 0096, 0000,00 ;2997          WORD[0000]                    ; ячейка 12
U 0097, 0000,00 ;2998          WORD[0000]                    ;
U 0098, 0000,FF ;2999          WORD[00FF]                    ; ячейка 13
U 0099, 0000,00 ;3000          WORD[0000]                    ;
U 009A, 00FF,FF ;3001          WORD[0FFFF]                   ; ячейка 14
U 009B, 0000,00 ;3002          WORD[0000]                    ;
U 009C, 0000,00 ;3003          WORD[0000]                    ; ячейка 15
U 009D, 00FF,00 ;3004          WORD[0FF00]                   ;
U 009E, 00FF,00 ;3005          WORD[0FF00]                   ; ячейка 16
U 009F, 00FF,FF ;3006          WORD[0FFFF]                   ;
U 00A0, 003F,FF ;3007          WORD[3FFF]                    ; ячейка 17
U 00A1, 0001,00 ;3008          WORD[0100]                    ;
U 00A2, 003F,FF ;3009          WORD[3FFF]                    ; ячейка 18
U 00A3, 00FF,FE ;3010          WORD[0FFFE]                   ;
U 00A4, 00FF,FF ;3011          WORD[0FFFF]                   ; ячейка 19
U 00A5, 00FE,7F ;3012          WORD[0FE7F]                   ;
U 00A6, 0000,00 ;3013          WORD[0000]                    ; ячейка 1A
U 00A7, 007F,F8 ;3014          WORD[7FF8]                    ;
U 00A8, 0080,00 ;3015          WORD[8000]                    ; ячейка 1B
U 00A9, 007F,FF ;3016          WORD[7FFF]                    ;
U 00AA, 0000,00 ;3017          WORD[0000]                    ; ячейка 1C
U 00AB, 0000,00 ;3018          WORD[0000]                    ;
U 00AC, 0000,00 ;3019          WORD[0000]                    ; ячейка 1D
U 00AD, 0000,00 ;3020          WORD[0000]                    ;
U 00AE, 0005,00 ;3021          WORD[0500]                    ; ячейка 1E
U 00AF, 0000,80 ;3022          WORD[0080]                    ;
U 00B0, 0005,00 ;3023          WORD[0500]                    ; ячейка 1F
U 00B1, 0000,00 ;3024          WORD[0000]                    ;
U 00B2, 0000,01 ;3025          WORD[0001]                    ; ячейка 20
U 00B3, 0000,00 ;3026          WORD[0000]                    ;
U 00B4, 0000,02 ;3027          WORD[0002]                    ; ячейка 21
U 00B5, 0000,00 ;3028          WORD[0000]                    ;
U 00B6, 0000,04 ;3029          WORD[0004]                    ; ячейка 22
U 00B7, 0000,00 ;3030          WORD[0000]                    ;
U 00B8, 0000,08 ;3031          WORD[0008]                    ; ячейка 23
U 00B9, 0000,00 ;3032          WORD[0000]                    ;
U 00BA, 0000,10 ;3033          WORD[0010]                    ; ячейка 24
U 00BB, 0000,00 ;3034          WORD[0000]                    ;
```

```
U 00BC, 0000,20 ;3035          WORDI[0020]                    ; ячейка 25
U 00BD, 0000,00 ;3036          WORDI[0000]                    ;
U 00BE, 0000,40 ;3037          WORDI[0040]                    ; ячейка 26
U 00BF, 0000,00 ;3038          WORDI[0000]                    ;
U 00C0, 0000,B0 ;3039          WORDI[00B0]                    ; ячейка 27
U 00C1, 0000,00 ;3040          WORDI[0000]                    ,
U 00C2, 0001,00 ;3041          WORDI[0100]                    ; ячейка 28
U 00C3, 0000,00 ;3042          WORDI[0000]                    ,
U 00C4, 0002,00 ;3043          WORDI[0200]                    ; ячейка 29
U 00C5, 0000,00 ;3044          WORDI[0000]                    ;
U 00C6, 0004,00 ;3045          WORDI[0400]                    ; ячейка 2A
U 00C7, 0000,00 ;3046          WORDI[0000]                    ;
U 00C8, 0008,00 ;3047          WORDI[0800]                    ; ячейка 2B
U 00C9, 0000,00 ;3048          WORDI[0000]                    ,
U 00CA, 0010,00 ;3049          WORDI[1000]                    ; ячейка 2C
U 00CB, 0000,00 ;3050          WORDI[0000]                    ;
U 00CC, 0020,00 ;3051          WORDI[2000]                    ; ячейка 2D
U 00CD, 0000,00 ;3052          WORDI[0000]                    ;
U 00CE, 0040,00 ;3053          WORDI[4000]                    ; ячейка 2E
U 00CF, 0000,00 ;3054          WORDI[0000]                    ;
U 00D0, 0080,00 ;3055          WORDI[8000]                    ; ячейка 2F
U 00D1, 0000,00 ;3056          WORDI[0000]                    ,
U 00D2, 0000,00 ;3057          WORDI[0000]                    ; ячейка 30
U 00D3, 0000,01 ;3058          WORDI[0001]                    ;
U 00D4, 0000,00 ;3059          WORDI[0000]                    ; ячейка 31
U 00D5, 0000,02 ;3060          WORDI[0002]                    ;
U 00D6, 0000,00 ;3061          WORDI[0000]                    ; ячейка 32
U 00D7, 0000,04 ;3062          WORDI[0004]                    ;
U 00D8, 0000,00 ;3063          WORDI[0000]                    ; ячейка 33
U 00D9, 0000,08 ;3064          WORDI[0008]                    ,
U 00DA, 0000,00 ;3065          WORDI[0000]                    ; ячейка 34
U 00DB, 0000,10 ;3066          WORDI[0010]                    ;
U 00DC, 0000,00 ;3067          WORDI[0000]                    ; ячейка 35
U 00DD, 0000,20 ;3068          WORDI[0020]                    ;
U 00DE, 0000,00 ;3069          WORDI[0000]                    ; ячейка 36
U 00DF, 0000,40 ;3070          WORDI[0040]                    ;
U 00E0, 0000,00 ;3071          WORDI[0000]                    ; ячейка 37
U 00E1, 0000,80 ;3072          WORDI[0080]                    ,
U 00E2, 0000,00 ;3073          WORDI[0000]                    ; ячейка 38
U 00E3, 0001,00 ;3074          WORDI[0100]                    ,
U 00E4, 0000,00 ;3075          WORDI[0000]                    ; ячейка 39
U 00E5, 0002,00 ;3076          WORDI[0200]                    ;
U 00E6, 0000,00 ;3077          WORDI[0000]                      ячейка 3A
U 00E7, 0004,00 ;3078          WORDI[0400]                    ;
U 00E8, 0000,00 ;3079          WORDI[0000]                    ; ячейка 3B
U 00E9, 0008,00 ;3080          WORDI[0800]                    ;
U 00EA, 0000,00 ;3081          WORDI[0000]                    ; ячейка 3C
U 00EB, 0010,00 ;3082          WORDI[1000]                    ;
U 00EC, 0000,00 ;3083          WORDI[0000]                    ; ячейка 3D
U 00ED, 0020,00 ;3084          WORDI[2000]                    ;
U 00EE, 0000,00 ;3085          WORDI[0000]                    ; ячейка 3E
U 00EF, 0040,00 ;3086          WORDI[4000]
U 00F0, 0000,00 ;3087          WORDI[0000]                    ; ячейка 3F
U 00F1, 0080,00 ;3088          WORDI[8000]                    ;
U 00F2, 0000,00 ;3089          WORDI[0000]                    ; ячейка 40
```

```
U 00F3, 0000,00 ;3090           WORD[0000]                              ;
U 00F4, 0000,00 ;3091           WORD[0000]                              ; ячейка 41
U 00F5, 0000,00 ;3092           WORD[0000]                              ;
U 00F6, 0000,00 ;3093           WORD[0000]                              ; ячейка 42
U 00F7, 0000,00 ;3094           WORD[0000]                              ;
U 00F8, 0000,00 ;3095           WORD[0000]                              ; ячейка 43
U 00F9, 0000,00 ;3096           WORD[0000]                              ;
U 00FA, 0000,00 ;3097           WORD[0000]                              ; ячейка 44
U 00FB, 0000,00 ;3098           WORD[0000]                              ;
U 00FC, 0000,00 ;3099           WORD[0000]                              ; ячейка 45
U 00FD, 0000,00 ;3100           WORD[0000]                              ;
U 00FE, 0000,00 ;3101           WORD[0000]                              ; ячейка 46
U 00FF, 0000,00 ;3102           WORD[0000]                              ;
U 0100, 0000,00 ;3103           WORD[0000]                              ; ячейка 47
U 0101, 0000,00 ;3104           WORD[0000]                              ;
U 0102, 0000,00 ;3105           WORD[0000]                              ; ячейка 48
U 0103, 0000,00 ;3106           WORD[0000]                              ;
U 0104, 0000,00 ;3107           WORD[0000]                              ; ячейка 49
U 0105, 0000,00 ;3108           WORD[0000]                              ;
U 0106, 0000,00 ;3109           WORD[0000]                              ; ячейка 4A
U 0107, 0000,00 ;3110           WORD[0000]                              ;
U 0108, 0055,55 ;3111           WORD[5555]                              ; ячейка 4B
U 0109, 00AA,AA ;3112           WORD[0AAAA]                             ;
U 010A, 00AA,AA ;3113           WORD[0AAAA]                             ; ячейка 4C
U 010B, 00AA,AA ;3114           WORD[0AAAA]                             ;
U 010C, 0055,55 ;3115           WORD[5555]                              ; ячейка 4D
U 010D, 0055,55 ;3116           WORD[5555]                              ;
U 010E, 0000,00 ;3117           WORD[0000]                              ; ячейка 4E
U 010F, 0000,00 ;3118           WORD[0000]                              ;
U 0110, 00FF,FF ;3119           WORD[0FFFF]                             ; ячейка 4F
U 0111, 00FF,FF ;3120           WORD[0FFFF]                             ;
U 0112, 0000,00 ;3121           WORD[0000]                              ; ячейка 50
U 0113, 0000,00 ;3122           WORD[0000]                              ;
U 0114, 0000,00 ;3123           WORD[0000]                              ; ячейка 51
U 0115, 0000,00 ;3124           WORD[0000]                              ;
U 0116, 0000,00 ;3125           WORD[0000]                              ; ячейка 52
U 0117, 0000,00 ;3126           WORD[0000]                              ;
U 0118, 0000,00 ;3127           WORD[0000]                              ; ячейка 53
U 0119, 0000,00 ;3128           WORD[0000]                              ;
U 011A, 0000,00 ;3129           WORD[0000]                              ; ячейка 54
U 011B, 0000,00 ;3130           WORD[0000]                              ;
U 011C, 0000,00 ;3131           WORD[0000]                              ; ячейка 55
U 011D, 0000,00 ;3132           WORD[0000]                              ;
U 011E, 0000,00 ;3133           WORD[0000]                              ; ячейка 56
U 011F, 0000,00 ;3134           WORD[0000]                              ;
U 0120, 0000,00 ;3135           WORD[0000]                              ; ячейка 57
U 0121, 0000,00 ;3136           WORD[0000]                              ;
U 0122, 0000,00 ;3137           WORD[0000]                              ; ячейка 58
U 0123, 0000,00 ;3138           WORD[0000]                              ;
U 0124, 0000,00 ;3139           WORD[0000]                              ; ячейка 59
U 0125  0000,00 ;3140           WORD[0000]                              ;
U 0126, 0000,00 ;3141           WORD[0000]                              ; ячейка 5A
U 0127, 0000,00 ;3142           WORD[0000]                              ;
U 0128, 0000,00 ;3143           WORD[0000]                              ; ячейка 5B
U 0129, 0000,00 ;3144           WORD[0000]                              ;
```

```
U 012A, 0000,00 ;3145          WORD[0000]                    ; ячейка 5C
U 012B, 0000,00 ;3146          WORD[0000]                    ;
U 012C, 0000,00 ;3147          WORD[0000]                    ; ячейка 5D
U 012D, 0000,00 ;3148          WORD[0000]                    ;
U 012E, 0000,00 ;3149          WORD[0000]                    ; ячейка 5E
U 012F, 0000,00 ;3150          WORD[0000]                    ;
U 0130, 0000,00 ;3151          WORD[0000]                    ; ячейка 5F
U 0131, 0000,00 ;3152          WORD[0000]                    ;
U 0132, 0000,03 ;3153          WORD[0003]                    ; ячейка 60
U 0133, 0000,00 ;3154          WORD[0000]                    ;
U 0134, 0000,12 ;3155          WORD[0012]                    ; ячейка 61
U 0135, 0000,00 ;3156          WORD[0000]                    ;
U 0136, 0033,33 ;3157          WORD[3333]                    ; ячейка 62
U 0137, 0033,33 ;3158          WORD[3333]                    ;
U 0138, 000F,0F ;3159          WORD[0F0F]                    ; ячейка 63
U 0139, 000F,0F ;3160          WORD[0F0F]                    ;
U 013A, 0000,FF ;3161          WORD[00FF]                    ; ячейка 64
U 013B, 0000,FF ;3162          WORD[00FF]                    ;
U 013C, 0001,62 ;3163          WORD[0162]                    ; ячейка 65
U 013D, 0000,00 ;3164          WORD[0000]                    ;
U 013E, 0000,00 ;3165          WORD[0000]                    ; ячейка 66
U 013F, 0000,00 ;3166          WORD[0000]                    ;
U 0140, 0000,00 ;3167          WORD[0000]                    ; ячейка 67
U 0141, 0000,00 ;3168          WORD[0000]                    ;
U 0142, 0000,00 ;3169          WORD[0000]                    ; ячейка 68
U 0143, 0000,00 ;3170          WORD[0000]                    ;
U 0144, 0000,00 ;3171          WORD[0000]                    ; ячейка 69
U 0145, 0000,00 ;3172          WORD[0000]                    ;
U 0146, 0000,00 ;3173          WORD[0000]                    ; ячейка 6A
U 0147, 0000,00 ;3174          WORD[0000]                    ;
U 0148, 0000,00 ;3175          WORD[0000]                    ; ячейка 6B
U 0149, 0000,00 ;3176          WORD[0000]                    ;
U 014A, 0000,00 ;3177          WORD[0000]                    ; ячейка 6C
U 014B, 0000,00 ;3178          WORD[0000]                    ;
U 014C, 0000,00 ;3179          WORD[0000]                    ; ячейка 6D
U 014D, 0000,00 ;3180          WORD[0000]                    ;
U 014E, 0000,00 ;3181          WORD[0000]                    ; ячейка 6E
U 014F, 0000,00 ;3182          WORD[0000]                    ;
U 0150, 0000,00 ;3183          WORD[0000]                    ; ячейка 6F
U 0151, 0000,00 ;3184          WORD[0000]                    ;
U 0152, 0000,00 ;3185          WORD[0000]                    ; ячейка 70
U 0153, 0000,00 ;3186          WORD[0000]                    ;
U 0154, 0000,00 ;3187          WORD[0000]                    ; ячейка 71
U 0155, 0000,00 ;3188          WORD[0000]                    ;
U 0156, 0000,00 ;3189          WORD[0000]                    ; ячейка 72
U 0157, 0000,00 ;3190          WORD[0000]                    ;
U 0158, 0000,00 ;3191          WORD[0000]                    ; ячейка 73
U 0159, 0000,00 ;3192          WORD[0000]                    ;
U 015A, 0000,00 ;3193          WORD[0000]                    ; ячейка 74
U 015B, 0000,00 ;3194          WORD[0000]                    ;
U 015C, 0000,00 ;3195          WORD[0000]                    ; ячейка 75
U 015D, 0000,00 ;3196          WORD[0000]                    ;
U 015E, 0000,00 ;3197          WORD[0000]                    ; ячейка 76
U 015F, 0000,00 ;3198          WORD[0000]                    ;
U 0160, 0000,00 ;3199          WORD[0000]                    ; ячейка 77
```

```
U 0161, 0000,00 ;3200            WORD[0000]                                ;
                ;3201
                ;3202               СЕКЦИЯ ДАННЫХ, СПЕЦИФИЧЕСКИХ ДЛЯ ПРОГРАММЫ (LS 80-F7)
                ;3203
                ;3204       ;       Этот раздел задает вторую половину LS. Она доступна только для инструк-
                ;3205       ;       ции MOV или такой, которая использует формат микроинструкции MOVE.
                ;3206       ;       Эта секция загружается отдельной пересылкой.
                ;3207       ;
                ;3208    TRANSFER.DATA.LS2:
  0162, 0000,78 ;3209            COUNT.LS[0078]                            ; счетчик длинных слов (число длинных слов,подлежащи
                ;3210                                                      ; пересылке в LS, равно 120 десятичн.) приемником
                ;3211                                                      ; является LS
U 0163, 0000,80 ;3212            START.ADR[0080]                           ; начальный адрес приемника (начинается с LS 80
                ;3213                                                      ; (шестнадцатеричн.))
U 0164, 0000,3C ;3214            WORD[003C]                                ; ячейка 80
U 0165, 0000,00 ;3215            WORD[0000]                                ;
U 0166, 0000,43 ;3216            WORD[0043]                                ; ячейка 81
U 0167, 0000,00 ;3217            WORD[0000]                                ;
U 0168, 0000,64 ;3218            WORD[0064]                                ; ячейка 82
U 0169, 0000,00 ;3219            WORD[0000]                                ;
U 016A, 0000,25 ;3220            WORD[0025]                                ; ячейка 83
U 016B, 0000,00 ;3221            WORD[0000]                                ;
U 016C, 0000,26 ;3222            WORD[0026]                                ; ячейка 84
U 016D, 0000,00 ;3223            WORD[0000]                                ;
U 016E, 0000,20 ;3224            WORD[0020]                                ; ячейка 85
U 016F, 0000,00 ;3225            WORD[0000]                                ;
U 0170, 0000,54 ;3226            WORD[0054]                                ; ячейка 86
U 0171, 0000,00 ;3227            WORD[0000]                                ;
U 0172, 0000,7D ;3228            WORD[007D]                                ; ячейка 87
U 0173, 0000,00 ;3229            WORD[0000]                                ;
U 0174, 0000,00 ;3230            WORD[0000]                                ; ячейка 88
U 0175, 0000,00 ;3231            WORD[0000]                                ;
U 0176, 0000,00 ;3232            WORD[0000]                                ; ячейка 89
U 0177, 0000,00 ;3233            WORD[0000]                                ;
U 0178, 0000,00 ;3234            WORD[0000]                                ; ячейка 8A
U 0179, 0000,00 ;3235            WORD[0000]                                ;
U 017A, 0000,00 ;3236            WORD[0000]                                ; ячейка 8B
U 017B, 0000,00 ;3237            WORD[0000]                                ;
U 017C, 0000,00 ;3238            WORD[0000]                                ; ячейка 8C
U 017D, 0000,00 ;3239            WORD[0000]                                ;
U 017E, 0000,00 ;3240            WORD[0000]                                ; ячейка 8D
U 017F, 0000,00 ;3241            WORD[0000]                                ;
U 0180, 0000,00 ;3242            WORD[0000]                                ; ячейка 8E
U 0181, 0000,00 ;3243            WORD[0000]                                ;
U 0182, 0000,00 ;3244            WORD[0000]                                ; ячейка 8F
U 0183, 0000,00 ;3245            WORD[0000]                                ;
U 0184, 0000,00 ;3246            WORD[0000]                                ; ячейка 90
U 0185, 0000,00 ;3247            WORD[0000]                                ;
U 0186, 0055,55 ;3248            WORD[5555]                                ; ячейка 91
U 0187, 00AA,AA ;3249            WORD[0AAAA]                               ;
U 0188, 0000,00 ;3250            WORD[0000]                                ; ячейка 92
U 0189, 0000,1F ;3251            WORD[001F]                                ;
U 018A, 0000,00 ;3252            WORD[0000]                                ; ячейка 93
U 018B, 0000,00 ;3253            WORD[0000]                                ;
U 018C, 0000,00 ;3254            WORD[0000]                                ; ячейкА 94
```

```
U 018D, 0000,00 ;3255          WORD[0000]                    ;
U 018E, 00FF,FC ;3256          WORD[0FFFC]                   ; ячейка 95
U 018F, 00FF,FF ;3257          WORD[0FFFF]                   ;
U 0190, 0000,0F ;3258          WORD[000F]                    ; ячейка 96
U 0191, 0000,00 ;3259          WORD[0000]                    ;
U 0192, 0000,F0 ;3260          WORD[00F0]                    ; ячейка 97
U 0193, 0000,00 ;3261          WORD[0000]                    ;
U 0194, 000F,00 ;3262          WORD[0F00]                    ; ячейка 98
U 0195, 0000,00 ;3263          WORD[0000]                    ;
U 0196, 00F0,00 ;3264          WORD[0F000]                   ; ячейка 99
U 0197, 0000,00 ;3265          WORD[0000]                    ;
U 0198, 0000,00 ;3266          WORD[0000]                    ; ячейка 9A
U 0199, 0000,03 ;3267          WORD[0003]                    ;
U 019A, 0080,00 ;3268          WORD[8000]                    ; ячейка 9B
U 019B, 007F,FF ;3269          WORD[7FFF]                    ;
U 019C, 0000,00 ;3270          WORD[0000]                    ; ячейка 9C
U 019D, 0000,54 ;3271          WORD[0054]                    ;
U 019E, 0000,00 ;3272          WORD[0000]                    ; ячейка 9D
U 019F, 0000,F0 ;3273          WORD[00F0]                    ;
U 01A0, 0000,00 ;3274          WORD[0000]                    ; ячейка 9E
U 01A1, 0000,FC ;3275          WORD[00FC]                    ;
U 01A2, 003F,FF ;3276          WORD[3FFF]                    ; ячейка 9F
U 01A3, 003F,00 ;3277          WORD[3F00]                    ;
U 01A4, 0002,54 ;3278          WORD[0254]                    ; ячейка 0A0
U 01A5, 0000,00 ;3279          WORD[0000]                    ;
U 01A6, 0000,00 ;3280          WORD[0000]                    ; ячейка 0A1
U 01A7, 0000,00 ;3281          WORD[0000]                    ;
U 01A8, 0000,00 ;3282          WORD[0000]                    ; ячейка 0A2
U 01A9, 0000,00 ;3283          WORD[0000]                    ;
U 01AA, 0000,00 ;3284          WORD[0000]                    ; ячейка 0A3
U 01AB, 0000,00 ;3285          WORD[0000]                    ;
U 01AC, 0000,00 ;3286          WORD[0000]                    ; ячейка 0A4
U 01AD, 0000,00 ;3287          WORD[0000]                    ;
U 01AE, 0000,00 ;3288          WORD[0000]                    ; ячейка 0A5
U 01AF, 0000,00 ;3289          WORD[0000]                    ;
U 01B0, 0000,00 ;3290          WORD[0000]                    ; ячейка 0A6
U 01B1, 0000,00 ;3291          WORD[0000]                    ;
U 01B2, 0000,00 ;3292          WORD[0000]                    ; ячейка 0A7
U 01B3, 0000,00 ;3293          WORD[0000]                    ;
U 01B4, 0000,00 ;3294          WORD[0000]                    ; ячейка 0A8
U 01B5, 0000,00 ;3295          WORD[0000]                    ;
U 01B6, 0000,00 ;3296          WORD[0000]                    ; ячейка 0A9
U 01B7, 0000,00 ;3297          WORD[0000]                    ;
U 01B8, 0000,00 ;3298          WORD[0000]                    ; ячейка 0AA
  01B9, 0000,00 ;3299          WORD[0000]                    ;
U 01BA, 0000,00 ;3300          WORD[0000]                    ; ячейка 0AB
U 01BB, 0000,00 ;3301          WORD[0000]                    ;
U 01BC, 0000,00 ;3302          WORD[0000]                    ; ячейка 0AC
U 01BD, 0000,00 ;3303          WORD[0000]                    ;
U 01BE, 0000,00 ;3304          WORD[0000]                    ; ячейка 0AD
U 01BF, 0000,00 ;3305          WORD[0000]                    ;
U 01C0, 0000,00 ;3306          WORD[0000]                    ; ячейка 0AE
U 01C1, 0000,00 ;3307          WORD[0000]                    ;
U 01C2, 0000,00 ;3308          WORD[0000]                    ; ячейка 0AF
U 01C3, 0000,00 ;3309          WORD[0000]                    ;
```

```
U 01C4, 0000,00 ;3310          WORD[0000]                          ; ячейка 0B0
U 01C5, 0000,00 ;3311          WORD[0000]                          ;
U 01C6, 0000,00 ;3312          WORD[0000]                          ; ячейка 0B1
U 01C7, 0000,00 ;3313          WORD[0000]                          ;
U 01C8, 0000,00 ;3314          WORD[0000]                          ; ячейка 0B2
U 01C9, 0000,00 ;3315          WORD[0000]                          ;
U 01CA, 0000,00 ;3316          WORD[0000]                          ; ячейка 0B3
U 01CB, 0000,00 ;3317          WORD[0000]                          ;
U 01CC, 0000,00 ;3318          WORD[0000]                          ; ячейка 0B4
U 01CD, 0000,00 ;3319          WORD[0000]                          ;
U 01CE, 0000,00 ;3320          WORD[0000]                          ; ячейка 0B5
U 01CF, 0000,00 ;3321          WORD[0000]                          ;
U 01D0, 0000,00 ;3322          WORD[0000]                          ; ячейка 0B6
U 01D1, 0000,00 ;3323          WORD[0000]                          ;
U 01D2, 0000,00 ;3324          WORD[0000]                          ; ячейка 0B7
U 01D3, 0000,00 ;3325          WORD[0000]                          ;
U 01D4, 0000,00 ;3326          WORD[0000]                          ; ячейка 0B8
U 01D5, 0000,00 ;3327          WORD[0000]                          ;
U 01D6, 0000,00 ;3328          WORD[0000]                          ; ячейка 0B9
U 01D7, 0000,00 ;3329          WORD[0000]                          ;
U 01D8, 0000,00 ;3330          WORD[0000]                          ; ячейка 0BA
U 01D9, 0000,00 ;3331          WORD[0000]                          ;
U 01DA, 0000,00 ;3332          WORD[0000]                          ; ячейка 0BB
U 01DB, 0000,00 ;3333          WORD[0000]                          ;
U 01DC, 0000,00 ;3334          WORD[0000]                          ; ячейка 0BC
U 01DD, 0000,00 ;3335          WORD[0000]                          ;
U 01DE, 0000,00 ;3336          WORD[0000]                          ; ячейка 0BD
U 01DF, 0000,00 ;3337          WORD[0000]                          ;
U 01E0, 0000,00 ;3338          WORD[0000]                          ; ячейка 0BE
U 01E1, 0000,00 ;3339          WORD[0000]                          ;
U 01E2, 0000,00 ;3340          WORD[0000]                          ; ячейка 0BF
U 01E3, 0000,00 ;3341          WORD[0000]                          ;
U 01E4, 0000,00 ;3342          WORD[0000]                          ; ячейка 0C0
U 01E5, 0000,00 ;3343          WORD[0000]                          ;
U 01E6, 0000,00 ;3344          WORD[0000]                          ; ячейка 0C1
U 01E7, 0000,00 ;3345          WORD[0000]                          ;
U 01E8, 0000,00 ;3346          WORD[0000]                          ; ячейка 0C2
U 01E9, 0000,00 ;3347          WORD[0000]                          ;
U 01EA, 0000,00 ;3348          WORD[0000]                          ; ячейка 0C3
U 01EB, 0000,00 ;3349          WORD[0000]                          ;
U 01EC, 0000,00 ;3350          WORD[0000]                          ; ячейка 0C4
U 01ED, 0000,00 ;3351          WORD[0000]                          ;
U 01EE, 0000,00 ;3352          WORD[0000]                          ; ячейка 0C5
U 01EF, 0000,00 ;3353          WORD[0000]                          ;
U 01F0, 0000,00 ;3354          WORD[0000]                          ; ячейка 0C6
U 01F1, 0000,00 ;3355          WORD[0000]                          ;
U 01F2, 0000,00 ;3356          WORD[0000]                          ; ячейка 0C7
U 01F3, 0000,00 ;3357          WORD[0000]                          ;
U 01F4, 0000,00 ;3358          WORD[0000]                          ; ячейка 0C8
U 01F5, 0000,00 ;3359          WORD[0000]                          ;
U 01F6, 0000,00 ;3360          WORD[0000]                          ; ячейка 0C9
U 01F7, 0000,00 ;3361          WORD[0000]                          ;
U 01F8, 0000,00 ;3362          WORD[0000]                          ; ячейка 0CA
U 01F9, 0000,00 ;3363          WORD[0000]                          ;
U 01FA, 0000,00 ;3364          WORD[0000]                          ; ячейка 0CB
```

```
U 01F8, 0000,00 ;3365           WORD[0000]                      ;
U 01FC, 0000,00 ;3366           WORD[0000]                      ; ячейка 0CC
U 01FD, 0000,00 ;3367           WORD[0000]                      ;
U 01FE, 0000,00 ;3368           WORD[0000]                      ; ячейка 0CD
U 01FF, 0000,00 ;3369           WORD[0000]                      ;
U 0200, 0000,00 ;3370           WORD[0000]                      ; ячейка 0CE
U 0201, 0000,00 ;3371           WORD[0000]                      ;
U 0202, 0000,00 ;3372           WORD[0000]                      ; ячейка 0CF
U 0203, 0000,00 ;3373           WORD[0000]                      ;
U 0204, 0000,00 ;3374           WORD[0000]                      ; ячейка 0D0
U 0205, 0000,00 ;3375           WORD[0000]                      ;
U 0206, 0000,00 ;3376           WORD[0000]                      ; ячейка 0D1
U 0207, 0000,00 ;3377           WORD[0000]                      ;
U 0208, 0000,00 ;3378           WORD[0000]                      ; ячейка 0D2
U 0209, 0000,00 ;3379           WORD[0000]                      ;
U 020A, 0000,00 ;3380           WORD[0000]                      ; ячейка 0D3
U 020B, 0000,00 ;3381           WORD[0000]                      ;
U 020C, 0000,00 ;3382           WORD[0000]                      ; ячейка 0D4
U 020D, 0000,00 ;3383           WORD[0000]                      ;
U 020E, 0000,00 ;3384           WORD[0000]                      ; ячейка 0D5
U 020F, 0000,00 ;3385           WORD[0000]                      ;
U 0210, 0000,00 ;3386           WORD[0000]                      ; ячейка 0D6
U 0211, 0000,00 ;3387           WORD[0000]                      ;
U 0212, 0000,00 ;3388           WORD[0000]                      ; ячейка 0D7
U 0213, 0000,00 ;3389           WORD[0000]                      ;
U 0214, 0000,00 ;3390           WORD[0000]                      ; ячейка 0D8
U 0215, 0000,00 ;3391           WORD[0000]                      ;
U 0216, 0000,00 ;3392           WORD[0000]                      ; ячейка 0D9
U 0217, 0000,00 ;3393           WORD[0000]                      ;
U 0218, 0000,00 ;3394           WORD[0000]                      ; ячейка 0DA
U 0219, 0000,00 ;3395           WORD[0000]                      ;
U 021A, 0000,00 ;3396           WORD[0000]                      ; ячейка 0DB
U 021B, 0000,00 ;3397           WORD[0000]                      ;
U 021C, 0000,00 ;3398           WORD[0000]                      ; ячейка 0DC
U 021D, 0000,00 ;3399           WORD[0000]                      ;
U 021E, 0000,00 ;3400           WORD[0000]                      ; ячейка 0DD
U 021F, 0000,00 ;3401           WORD[0000]                      ;
U 0220, 0000,00 ;3402           WORD[0000]                      ; ячейка 0DE
U 0221, 0000,00 ;3403           WORD[0000]                      ;
U 0222, 0000,00 ;3404           WORD[0000]                      ; ячейка 0DF
U 0223, 0000,00 ;3405           WORD[0000]                      ;
U 0224, 0000,00 ;3406           WORD[0000]                      ; ячейка 0E0
U 0225, 0000,00 ;3407           WORD[0000]                      ;
U 0226, 0000,00 ;3408           WORD[0000]                      ; ячейка 0E1
U 0227, 0000,00 ;3409           WORD[0000]                      ;
U 0228, 0000,00 ;3410           WORD[0000]                      ; ячейка 0E2
U 0229, 0000,00 ;3411           WORD[0000]                      ;
U 022A, 0000,00 ;3412           WORD[0000]                      ; ячейка 0E3
U 022B, 0000,00 ;3413           WORD[0000]                      ;
U 022C, 0000,00 ;3414           WORD[0000]                      ; ячейка 0E4
U 022D, 0000,00 ;3415           WORD[0000]                      ;
U 022E, 0000,00 ;3416           WORD[0000]                      ; ячейка 0E5
U 022F, 0000,00 ;3417           WORD[0000]                      ;
U 0230, 0000,00 ;3418           WORD[0000]                      ; ячейка 0E6
U 0231, 0000,00 ;3419           WORD[0000]                      ;
```

```
U 0232,  0000,00 ;3420            WORD[0000]                          ; ячейка 0E7
U 0233,  0000,00 ;3421            WORD[0000]                          ;
U 0234,  0000,00 ;3422            WORD[0000]                          ; ячейка 0E8
U 0235,  0000,00 ;3423            WORD[0000]                          ;
U 0236,  0000,00 ;3424            WORD[0000]                          ; ячейка 0E9
U 0237,  0000,00 ;3425            WORD[0000]                          ;
U 0238,  0000,00 ;3426            WORD[0000]                          ; ячейка 0EA
U 0239,  0000,00 ;3427            WORD[0000]                          ;
U 023A,  0000,00 ;3428            WORD[0000]                          ; ячейка 0EB
U 023B,  0000,00 ;3429            WORD[0000]                          ;
U 023C,  0000,00 ;3430            WORD[0000]                          ; ячейка 0EC
U 023D,  0000,00 ;3431            WORD[0000]                          ;
U 023E,  0000,00 ;3432            WORD[0000]                          ; ячейка 0ED
U 023F,  0000,00 ;3433            WORD[0000]                          ;
U 0240,  0000,00 ;3434            WORD[0000]                          ; ячейка 0EE
U 0241,  0000,00 ;3435            WORD[0000]                          ;
U 0242,  0000,00 ;3436            WORD[0000]                          ; ячейка 0EF
U 0243,  0000,00 ;3437            WORD[0000]                          ;
U 0244,  0000,00 ;3438            WORD[0000]                          ; ячейка 0F0
U 0245,  0000,00 ;3439            WORD[0000]                          ;
U 0246,  0000,00 ;3440            WORD[0000]                          ; ячейка 0F1
U 0247,  0000,00 ;3441            WORD[0000]                          ;
U 0248,  0000,00 ;3442            WORD[0000]                          ; ячейка 0F2
U 0249,  0000,00 ;3443            WORD[0000]                          ;
U 024A,  0000,00 ;3444            WORD[0000]                          ; ячейка 0F3
U 024B,  0000,00 ;3445            WORD[0000]                          ;
U 024C,  0000,00 ;3446            WORD[0000]                          ; ячейка 0F4
U 024D,  0000,00 ;3447            WORD[0000]                          ;
U 024E,  0000,00 ;3448            WORD[0000]                          ; ячейка 0F5
U 024F,  0000,00 ;3449            WORD[0000]                          ;
U 0250,  0000,00 ;3450            WORD[0000]                          ; ячейка 0F6
U 0251,  0000,00 ;3451            WORD[0000]                          ;
U 0252,  0000,00 ;3452            WORD[0000]                          ; ячейка 0F7
U 0253,  0000,00 ;3453            WORD[0000]                          ;
                 ;3454
                 ;3455    3FF:
                 ;3456    RETURN.3FF:
U 03FF,  5B00,14 ;3457            RETURN                              ; используется тестом F
                 ;3458
                 ;3459    .REGION/600,6FF
```

```
                    ;3460     .PAGE "ПОДПРОГРАММЫ В УПРАВЛЯЮЩЕЙ ПАМЯТИ (WCS), ИСПОЛЬЗУЕМЫЕ МИКРОМОНИТОРОМ"
                    ;3461     ;
                    ;3462     ;              Программа генерации данных
                    ;3463     ;
                    ;3464     ;          Эта программа используется диагностическим монитором для загрузки рабо-
                    ;3465     ;        чего регистра WR0 кодом данных из консольного процессора. Монитор устанав-
                    ;3466     ;        ливает CONSOLE ATTN, если должна генерироваться единица, или очищает
                    ;3467     ;        CONSOLE ATTN, если должен генерироваться 0. Затем он проходит эту программу
                    ;3468     ;        один раз с целью вдвигания бита данных в WR0. Эта программа загружает 32-
                    ;3469     ;        битовые значения намного быстрее, чем было бы при вдвигании каждой инструк-
                    ;3470     ;        ции в CSR из монитора, и используется, главным образом, для формирования
                    ;3471     ;        данных, подлежащих загрузке в LS в качестве констант.
                    ;3472     ;
                    ;3473     ;                          *** ПРЕДУПРЕЖДЕНИЕ ***
                    ;3474     ;
                    ;3475     ;        Эта программа должна начинаться адресом 600 и заканчиваться 605.
                    ;3476     ;
                    ;3477     ;
                    ;3478     GEN.XFER.DATA:
U 0600, 2F80,15     ;3479             CLR WR[0]                          ; очистка рабочего регистра
U 0601, A0C0,15     ;3480             COM WR[0]                          ; рабочий регистр WR0 содержит все единицы
                    ;3481     LOOP.XFER:
U 0602, 5B00,18     ;3482             SKIP.IF[CONSOLE.ATTN]              ; пропуск следующей инструкции, если установлен CONSOLE
                    ;3483                                                ; ATTN (генерация единицы)
                    ;3484               ASHL WR[0],                      ; вдвигание 0 в бит 0 WR0
U 0603, A3D0,1E     ;3485             SKIP                               ; пропуск инструкции ROL
U 0604, A3C0,15     ;3486             ROL WR[0]                          ; вдвигание единицы в бит 0 WR0
U 0605, 0B60,24     ;3487             JMP [LOOP.XFER]                    ; повторение
                    ;3488     ;
                    ;3489     ;        Здесь хранится номер версии диагностики
                    ;3490     ;
                    ;3491     ;                          *** ПРЕДУПРЕЖДЕНИЕ ***
                    ;3492     ;
                    ;3493     ;        Эти слова должны быть в ячейках 606 и 607.
                    ;3494     ;
                    ;3495
U 0606, 0003,00     ;3496             WORD[0300]                         ; версия 03.00
U 0607, 0043,42     ;3497             ASCII [CB]                         ; последние две буквы имени файла [ENKCB]
                    ;3498
                    ;3499
                    ;3500     ;              Программа записи в регистр управления и состояния MCT
                    ;3501     ;***
                    ;3502     ;
                    ;3503     ; Описание функционирования:
                    ;3504     ;
                    ;3505     ;          Программа записи CSR используется для записи в регистры управления и
                    ;3506     ;        состояния контроллера памяти. Контроллер памяти имеет три регистра уп-
                    ;3507     ;        равления и состояния, идентифицируемые как CSR0, CSR1 и CSR2.
                    ;3508     ;          CSR0 содержит контрольные биты или биты синдрома, полученные из
                    ;3509     ;        логических схем корректирующего кода (ECC) после определенных программ
                    ;3510     ;        диагностирования. Его нельзя непосредственно записывать при помощи этой
                    ;3511     ;        программы.
                    ;3512     ;          CSR1 содержит биты ошибок, установленные, если произошла ошибка во
                    ;3513     ;        время обращения к памяти. Он содержит 5 битов управления (29-25),
                    ;3514     ;        допускающих запись. имеется также 7 проверочных битов (биты 6-0),
```

```
                    ;3515      ;       которые допускают запись, но не считываются. Эти биты используются для
                    ;3516      ;       записи контрольных битов в микросхемы кодов коррекции (ECC).
                    ;3517      ;         CSR2 содержит биты ошибок, установленные, если произошла ошибка во
                    ;3518      ;       время обращения к общей шине. Эти биты только считываются и не могут
                    ;3519      ;       записываться данной программой.
                    ;3520      ;         Поэтому CSR1 является единственным регистром управления и состояния,
                    ;3521      ;       допускающим запись, и только биты 25-29 могут как записываться, так и
                    ;3522      ;       считываться. Таким образом, эта программа при записи CSR вынуждает
                    ;3523      ;       запись в CSR1.
                    ;3524      ;         Эта программа заносит в LS5 данные для обращения к CSR1, затем запи-
                    ;3525      ;       сывает в CSR данные, хранящиеся в LS6. Затем она выдает CPU ATTN с целью
                    ;3526      ;       информирования монитора, что она завершила свою функцию.
                    ;3527      ;         ПРИМЕЧАНИЕ: биты ошибок в CSR1 очищаются при любой записи в CSR1. Эти-
                    ;3528      ;       ми битами являются 31,30, и 23-14. Биты 13-0 и бит 24 не используются.
                    ;3529      ;
                    ;3530      ;  Процедура вызова:
                    ;3531      ;       Консольный процессор загружает адрес указателя этой программы (инструк-
                    ;3532      ;       ции JMP в ячейке WCS 50) в счетчик микроинструкций (UPC) и запускает
                    ;3533      ;       центральный процессор.
                    ;3534      ;
                    ;3535      ;  Входные параметры:
                    ;3536      ;       В ячейку LS6 до выполнения этой программы должны быть записаны из кон-
                    ;3537      ;       соли требуемые данные, подлежащие занесению в CSR1.
                    ;3538      ;
                    ;3539      ;  Неявные входные данные:
                    ;3540      ;       Отсутствуют.
                    ;3541      ;
                    ;3542      ;  Параметры выхода:
                    ;3543      ;       В биты CSR 29-25 и 6-0 записаны данные из LS6.
                    ;3544      ;
                    ;3545      ;  Неявные выходы:
                    ;3546      ;       отсутствуют.
                    ;3547      ;
                    ;3548      ;  Побочный эффект:
                    ;3549      ;       Этой программой модифицируется WR0.
                    ;3550      ;       Этой программой модифицируется LS5.
                    ;3551      ;       Биты CSR1 31,30 и 23-14 очищаются.
                    ;3552      ;
                    ;3553      ;---
                    ;3554      DEPOSIT.CSR:
U 0608, 3644,15     ;3555              MOV LS[CSR1] TO WR[0]                    ; установка бита 2 в WR0 в качестве номера CSR,в который
                    ;3556                                                       ; будет производиться запись (биты 2 и 3 являются
                    ;3557                                                       ; номерами CSR)
U 0609, BE0A,15     ;3558              MOV WR[0] TO LS[T5.SUB]                   ; занесение  номера CSR в ячейку LS5 (CSR1)
U 060A, 1D0B,F5     ;3559              MEM.REQ[WRITE.CSR] ADRS[T5.SUB] DT[LONG]   ; выдача запроса памяти для доступа к CSR1
U 060B, B20C,15     ;3560              WRITE.MEM LS[T6.SUB]                     ; пересылка данных из ячейки 6 в CSR1
U 060C, 8868,74     ;3561              JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                    ;3562      ;
                    ;3563      ;         Программа индикации регистра управления и состояния МСТ
                    ;3564      ;***
                    ;3565      ;
                    ;3566      ; Описание функционирования:
                    ;3567      ;
                    ;3568      ;           Программа индикации CSR читает регистр управления и состояния контрол-
                    ;3569      ;       лера памяти. Существуют 3 регистра управления и состояния, идентифицируе-
```

```
                ;3570      ;       мые как CSR0, CSR1 и CSR2.
                ;3571      ;         CSR0 содержит контрольные биты или биты синдрома из логических схем
                ;3572      ;       корректирующего кода (ECC). Они содержатся в CSR в битах 6-0. Другие биты
                ;3573      ;       непредсказуемые. Биты 6-0 допускают только запись специальными диагнос-
                ;3574      ;       тическими программами (они не могут записываться командой DEPOSIT CSR).
                ;3575      ;         CSR1 содержит биты ошибок и биты управления. Битами ошибок являются
                ;3576      ;       31,30 и 23-14. Битами управления являются биты 29-25. другие биты не-
                ;3577      ;       предсказуемые. Биты ошибок записываются во время выполнения функций па-
                ;3578      ;       мяти и не могут записываться посредством команды DEPOSIT CSR. Биты управ-
                ;3579      ;       ления допускают запись. Заметим, что команда DEPOSIT CSR записывает биты
                ;3580      ;       управления и очищает все биты ошибок.
                ;3581      ;         CSR2 содержит 3 бита (биты 16-14), которые могут считываться. Они уста-
                ;3582      ;       навливаются, если произошла ошибка общей шины при обращении к общей шине.
                ;3583      ;       Они не могут записываться непосредственно командой DEPOSIT CSR.
                ;3584      ;         Консольный процессор загружает LS5 номером CSR (0,1 или 2), который
                ;3585      ;       подлежит считывать. Для правильного его расположения эта программа сдви-
                ;3586      ;       гает LS5 на два места влево. Затем она выполняет чтение CSR и помещает
                ;3587      ;       данные в LS6 для считывания с консоли.
                ;3588      ;
                ;3589      ; Процедура вызова:
                ;3590      ;
                ;3591      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                ;3592      ;       указателя этой программы (инструкции JMP в ячейке WCS 51) и запускает
                ;3593      ;       центральный процессор.
                ;3594      ;
                ;3595      ; Входные параметры:
                ;3596      ;
                ;3597      ;       LS5 содержит номер CSR, подлежащего чтению.
                ;3598      ;
                ;3599      ; Неявные входные данные:
                ;3600      ;
                ;3601      ;       Отсутствуют.
                ;3602      ;
                ;3603      ; Выходные параметры:
                ;3604      ;
                ;3605      ;       Данные, считанные с выбранного CSR, в LS6.
                ;3606      ;
                ;3607      ; Неявные выходные данные:
                ;3608      ;
                ;3609      ;       Отсутствуют.
                ;3610      ;
                ;3611      ; Побочный эффект:
                ;3612      ;
                ;3613      ;        LS5 смещена на 2 места влево.
                ;3614      ;
                ;3615      ;---
                ;3616      EXAMINE.CSR:
U 060D, 360A,15 ;3617              MOV LS[T5.SUB] TO WR[0]                 ; выборка из LS5 номера CSR
U 060E, A2D0,15 ;3618              SHL2 WR[0]                              ; сдвиг числа на 2 места влево для получения номера CSR
                ;3619                                                      ; в битах 2 и 3
U 060F, BE0A,15 ;3620              MOV WR[0] TO LS[T5.SUB]                 ; занесение сдвинутого числа в LS5
U 0610, 9D0B,75 ;3621              MEM.REQ[READ.CSR] ADRS[T5.SUB] DT[LONG]  ; выдача запроса памяти для доступа к заданному CSR
U 0611, 3022,15 ;3622              MOV MEM.DATA TO WR[0]                   ; чтение выбранного CSR
                ;3623      CHECK.CSR2:
U 0612, 860A,95 ;3624              MOV LS[T5.SUB] TO WR[1]                 ; выборка сдвинутого номера CSR
```

```
                    ;3625              CMP LS[#8] WITH WR[1],                  ; проверка, был ли выбран CSR2
U 0613,  CE46,B5    ;3626             DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 0614,  0861,61    ;3627             JMP.IF[NEQ] TO [CHECK.CSR1]              ; переход, если выбран не CSR2
U 0615,  C530,15    ;3628             BIC LS[CSR2.MASK] TO WR[0]               ; очистка неиспользуемых (непредсказуемых) битов в CSR2
                    ;3629          CHECK.CSR1:
                    ;3630              CMP LS[#4] WITH WR[1],                  ; проверка был ли выбран CSR1
U 0616,  4E44,B5    ;3631             DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 0617,  0861,91    ;3632             JMP.IF[NEQ] TO [CHECK.CSR0]              ; переход, если выбран не CSR1
U 0618,  C52E,15    ;3633             BIC LS[CSR1.MASK] TO WR[0]               ; очистка неиспользуемых (непредсказуемых) битов в CSR1
                    ;3634          CHECK.CSR0:
                    ;3635              CMP LS[#0] WITH WR[1],                  ; проверка был ли выбран CSR0
U 0619,  4E9C,B5    ;3636             DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 061A,  8861,D1    ;3637             JMP.IF[NEQ] TO [LOAD.LS6]                ; переход, если выбран не CSR0
U 061B,  452C,15    ;3638             BIC LS[CSR0.MASK] TO WR[0]               ; очистка неиспользуемых (непредсказуемых) битов в CSR0
U 061C,  C54E,15    ;3639             BIC LS[BIT7] TO WR[0]                    ; бит 7 также не используется
                    ;3640          LOAD.LS6:
U 061D,  BE0C,15    ;3641             MOV WR[0] TO LS[T6.SUB]                  ; загрузка данных CSR в LS6 для печати
U 061E,  8868,74    ;3642             JMP [WAIT.SUB]                          ; выдача CPU ATTN и ожидание ответа
                    ;3643          ;
                    ;3644          ;        Программа записи в буфер трансляции MCT
                    ;3645          ;***
                    ;3646          ;
                    ;3647          ; Описание функционирования:
                    ;3648          ;
                    ;3649          ;          Эта программа дает возможность записывать в ту часть контроллера памя-
                    ;3650          ;        ти, которая составляет буфер трансляции. Область буфера содержит 128 дес.
                    ;3651          ;        ячеек, адресуемых от 0 до 7F (шестнадцатер.). Остаток памяти буфера транс-
                    ;3652          ;        ляции (TB) либо не используется, либо содержит данные отображения адресов
                    ;3653          ;        общей шины. Для записи в ту часть ОЗУ TB, которая содержит отображение
                    ;3654          ;        общей шины, используется команда DEPOSIT UB. Адрес, заданный командой
                    ;3655          ;        DEPOSIT, является действительным адресом обращения к ОЗУ буфера. Эта прог-
                    ;3656          ;        рамма выполняет все необходимые сдвиги заданного адреса для правильного
                    ;3657          ;        его расположения. Заданный адрес загружен в ячейку LS5 до выполнения
                    ;3658          ;        этой программы. Адрес должен быть в шестнадцатеричном формате.
                    ;3659          ;          Требуемые данные должны размещаться консольным процессором в LS6 до вы-
                    ;3660          ;        полнения этой программы. Биты 07-01 являются битами VALID (бит действи-
                    ;3661          ;        тельности), PROT (биты защиты), MODIFY (бит модификации) и BYTE OFFSET
                    ;3662          ;        (бит смещения байта)(битов PROT имеется четыре). Биты 22-08 содержат би-
                    ;3663          ;        ты физического адреса (PA) от PA 23 до PA 09 соответственно. Биты от 31
                    ;3664          ;        до 23 и бит 0 не используются. Эта программа делает все сдвиги, необходи-
                    ;3665          ;        мые, чтобы записать биты в буфер трансляции так, как описано. Нужные
                    ;3666          ;        данные должны представляться в шестнадцатеричном формате.
                    ;3667          ;
                    ;3668          ; Процедура вызова:
                    ;3669          ;
                    ;3670          ;          Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                    ;3671          ;        указателя для данной программы (инструкции JMP в ячейке WCS 52) и запус-
                    ;3672          ;        кает центральный процессор.
                    ;3673          ;
                    ;3674          ; Входные параметры:
                    ;3675          ;
                    ;3676          ;        LS5 - адрес буфера трансляци, по которому предстоит записывать (диапазон
                    ;3677          ;              от 0 до 7F).
                    ;3678          ;        LS6 - данные, подлежащие записи (22 бита, от 1 до 22). Биты 0 и 23-31
                    ;3679          ;              игнорируются.
```

```
                ;3680     ;
                ;3681     ; Неявные входные данные:
                ;3682     ;
                ;3683     ;       Отсутствуют.
                ;3684     ;
                ;3685     ; Выходные параметры:
                ;3686     ;
                ;3687     ;       В ТВ по заданному адресу будет записана заданная информация.
                ;3688     ;
                ;3689     ; Неявные выходные данные:
                ;3690     ;
                ;3691     ;       Отсутствуют.
                ;3692     ;
                ;3693     ; Побочный эффект:
                ;3694     ;
                ;3695     ;       WR0 - будет модифицирован
                ;3696     ;       WR1 - будет модифицирован
                ;3697     ;
                ;3698     ;--
                ;3699     DEPOSIT.TB:
U 061F, 8862,FC ;3700             JSR [SHIFT.TB.ADR]                      ; переход к подпрограмме правильного расположения адреса
                ;3701                                                     ; ТВ
U 0620, 360C,15 ;3702             MOV LS[T6.SUB] TO WR[0]                 ; выборка данных из LS6
U 0621, 4526,15 ;3703             BIC LS[#FF] TO WR[0]                    ; очистка младшего байта заданной информации. Младший
                ;3704                                                     ; байт будет позднее размещен в битах 24-31
U 0622, 8862,AC ;3705             JSR [SHIFT.LOOP]                        ; переход к подпрограмме для сдвига WR0 на 8 позиций
                ;3706                                                     ; вправо
U 0623, E40C,15 ;3707             SWAP LS[T6.SUB] WITH WR[0]              ; замена исходных данных модифицированными
                ;3708                                                     ; данными. Теперь LS6 содержит биты 8-22 в битах 0-14,
                ;3709                                                     ; как и требовалось. WR0 содержит заданные исходные
                ;3710                                                     ; данные
U 0624, 452C,15 ;3711             BIC LS[#FFFFFF00] TO WR[0]              ; очистка всех заданных исходных данных, кроме младшего
                ;3712                                                     ; байта
U 0625, 8862,AC ;3713             JSR [SHIFT.LOOP]                        ; переход к подпрограмме сдвига WR0 на 8 позиций
                ;3714                                                     ; вправо. После возврата WR0 будет содержать биты 0-7 на
                ;3715                                                     ; месте битов 24-31. Другие биты очищены
U 0626, ED0C,15 ;3716             BIS WR[0],TO LS[T6.SUB]                 ; теперь LS6 содержит данные, подлежащие записи, в
                ;3717                                                     ; правильном формате, т.е., биты PA в битах 0-14, BYTE
                ;3718                                                     ; OFFSET, MODIFY, биты PROT от A до D и VALID в битах
                ;3719                                                     ; 25-31
U 0627, 980A,F5 ;3720             MEM.REQ[WRITE.TB] ADRS[T5.SUB] DTI[LONG]  ; выдача запроса памяти для записи в ТВ
U 0628, 820C,15 ;3721             WRITE.MEM LS[T6.SUB]                    ; запись данных из LS6 в ТВ
U 0629, 8868,74 ;3722             JMP [WAIT.SUB]                          ; переход к выдаче CPU ATTN и ожиданию
                ;3723     ;
                ;3724     ;       Подпрограмма сдвига ТВ на 8 мест вправо
                ;3725     ;
                ;3726     SHIFT.LOOP:
U 062A, 3646,95 ;3727             MOV LS[#8] TO WR[1]                     ; в WR1 находится счетчик сдвига на 8 позиций
                ;3728     SHIFT.LOOP.1:
U 062B, 2340,15 ;3729             ROR WR[0]                               ; сдвиг WR0 на одну позицию вправо
                ;3730               DEC WR[1],                            ; уменьшение счетчика и установка
U 062C, A102,B5 ;3731             DT(LONG)&SET.ALU.CC                     ; кодов условий
U 062D, 8862,B1 ;3732             JMP.IF[NEQ] TO [SHIFT.LOOP.1]           ; повторение до тех пор, пока будет выполнено 8
                ;3733                                                     ; сдвигов, затем
U 062E, 5B00,14 ;3734             RETURN                                  ; возврат
```

```
                    ;3735     ;
                    ;3736     ;       Подпрограмма размещения адреса буфера TB в битах 9-14 и бите 31
                    ;3737     ;       (бит 0 адреса TB должен находиться на месте бита 31)
                    ;3738     ;
                    ;3739     SHIFT.TB.ADR:
U 062F,  360A,15    ;3740             MOV LS[T5.SUB] TO WR[0]                 ; выборка адреса TB из LS5
U 0630,  452C,15    ;3741             BIC LS[#FFFFFF00] TO WR[0]              ; очистка всех битов, кроме младшего байта (8-битовый
                    ;3742                                                     ; адрес)
U 0631,  A2D0,15    ;3743             SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0632,  A2D0,15    ;3744             SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0633,  A2D0,15    ;3745             SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0634,  A2D0,15    ;3746             SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0635,  23D0,15    ;3747             ASHL WR[0]                              ; сдвиг еще на одну позицию. Теперь биты 0-6 находятся в
                    ;3748                                                     ; битах 9-15
                    ;3749               BIT LS[BIT15] WITH WR[0],             ; проверка, установлен или очищен бит 15 (старший бит)
U 0636,  595E,35    ;3750             D?(LONG)&SET.ALU.CC                     ; для правильной адресации TB старший бит должен перейти
                    ;3751                                                     ; в 31
U 0637,  5B00,09    ;3752             SKIP.IF[EQL]                            ; пропуск следующей инструкции, если старший бит равен
                    ;3753                                                     ; нулю
U 0638,  477E,15    ;3754             BIS LS[BIT31] TO WR[0]                  ; в противном случае поместить бит 31 в старший бит.
                    ;3755               MOV WR[0] TO LS[T5.SUB].              ; теперь адрес TB в правильной форме находится в ячейке
                    ;3756                                                     ; LS5
U 0639,  3E0A,14    ;3757             RETURN                                  ; возврат к основному коду
                    ;3758     ;
                    ;3759     ;       Программа индикации  буфера трансляции MOT
                    ;3760     ;***
                    ;3761     ;
                    ;3762     ; Описание функционирования:
                    ;3763     ;
                    ;3764     ;          Эта программа выполняет чтение из буфера трансляции (TB) контроллера
                    ;3765     ;       памяти. Содержимое TB будет получено с битами BYTE OFFSET, MODIFY, PROT
                    ;3766     ;       от A до D и VALID в битах 01-07 соответственно. Биты PA 09-23 поступают
                    ;3767     ;       в битах 08-22 соответственно. Биты 23-31 и бит 0 не используются, поэтому
                    ;3768     ;       будут очищены до печати результата. Результат для печати на консоли будет
                    ;3769     ;       помещен в LS6.
                    ;3770     ;          Программа выдает запрос к памяти с функцией памяти READ.TB и типом дан-
                    ;3771     ;       ных LONGWORD (длинное слово). Консольный процессор загружает LS5 заданным
                    ;3772     ;       адресом прежде, чем выполняется эта программа. Эта программа сдвигает ад-
                    ;3773     ;       рес так, как требуется для адресации заданной ячейки TB. Заданный адрес
                    ;3774     ;       должен быть в 16-ричном формате.
                    ;3775     ;          TB состоит из 128 дес.ячеек (адреса от 0 до 7F). Оставшиеся ячейки ОЗУ
                    ;3776     ;       буфера трансляции либо не используются, либо используются для отображения
                    ;3777     ;       адресов общей шины. Адреса из области отображения общей шины считываются
                    ;3778     ;       посредством программы EXAMINE UB.
                    ;3779     ;
                    ;3780     ; Процедура вызова:
                    ;3781     ;
                    ;3782     ;          Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                    ;3783     ;       указателя этой программы (инструкции JMP в ячейке WCS 53).
                    ;3784     ;
                    ;3785     ; Входные параметры:
                    ;3786     ;
                    ;3787     ;       LS5 - адрес буфера трансляции (0-7F) в шестнадцатеричном формате.
                    ;3788     ;
                    ;3789     ; Неявные входные данные:
```

```
                ;3790     ;
                ;3791     ;       Отсутствуют.
                ;3792     ;
                ;3793     ; Выходные параметры:
                ;3794     ;
                ;3795     ;       LS6 - данные из TB по заданному адресу
                ;3796     ;
                ;3797     ; Неявные выходные данные:
                ;3798     ;
                ;3799     ;       Отсутствуют.
                ;3800     ;
                ;3801     ; Побочный эффект:
                ;3802     ;
                ;3803     ;       LS5 - модифицируется
                ;3804     ;       WR0 - модифицируется
                ;3805     ;
                ;3806     ;---
                ;3807     EXAMINE.TB:
U 063A, 8862,FC ;3808             JSR [SHIFT.TB.ADR]                      ; заданный в LS5 адрес сдвигается в правильное положение
                ;3809                                                     ; и замещает исходный адрес в LS5
U 063B, 9C0B,F5 ;3810             MEM.REQ[READ.TB] ADRS[T5.SUB] DT[LONG]  ; выдача запроса к памяти для чтения TB по адресу,
                ;3811                                                     ; содержащемуся в LS5
U 063C, 3022,15 ;3812             MOV MEM.DATA TO WR[0]                   ; занесение результата в WR0
U 063D, 452A,15 ;3813             BIC LS[#FF000000] TO WR[0]              ; очистка старшего байта результата (старший байт не
                ;3814                                                     ; является частью TB и является непредсказуемым)
U 063E, 456E,15 ;3815             BIC LS[BIT23] TO WR[0]                  ; то же действие для бита 23
U 063F, 4540,15 ;3816             BIC LS[BIT0] TO WR[0]                   ; то же действие для бита 0. WR0 теперь содержит
                ;3817                                                     ; результат, готовый к печати
U 0640, BE0C,15 ;3818             MOV WR[0] TO LS[T6.SUB]                 ; результат занесен в LS6
U 0641, 8868,74 ;3819             JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;3820     ;
                ;3821     ;       Программа записи в буфер трансляции адресов общей шины
                ;3822     ;***
                ;3823     ;
                ;3824     ; Описание функционирования.
                ;3825     ;
                ;3826     ;          Эта программа позволяет записывать в ту часть буфера трансляции (TB)
                ;3827     ;       контроллера памяти, которая содержит данные отображения адресов общей ши-
                ;3828     ;       ны. Область отображения содержит 512 дес. ячеек, адресуемых от 200 до 3FF.
                ;3829     ;       Оставшаяся часть ОЗУ TB либо не используется, либо содержит информацию бу-
                ;3830     ;       фера трансляции. Для записи в ОЗУ буфера трансляции используется команда
                ;3831     ;       DEPOSIT TB. Адрес, заданный командой DEPOSIT, является действительным ад-
                ;3832     ;       ресом обращения к ОЗУ. Эта программа выполняет все сдвиги заданного адреса,
                ;3833     ;       необходимые для правильного его расположения. Заданный адрес загружается
                ;3834     ;       консольным процессором в ячейку LS5 прежде, чем выполняется эта программа.
                ;3835     ;       Адрес должен иметь 16-ричный формат.
                ;3836     ;          Требуемые данные размещаются консольным процессором в LS6 до выполне-
                ;3837     ;       ния этой программы. Биты 07-01 являются битами VALID, PROT, MODIFY и BYTE
                ;3838     ;       OFFSET (имеются четыре бита PROT). Биты 22-08 содержат биты PA от PA23 до
                ;3839     ;       PA9 соответственно. Биты от 31 до 23 и бит 0 не используются. Эта програм-
                ;3840     ;       ма выполняет все сдвиги, необходимые для записи этих битов в TB, как опи-
                ;3841     ;       сано. Заданная информация должна быть в 16-ричном формате.
                ;3842     ;
                ;3843     ; Процедура вызова:
                ;3844
```

```
                    ;3845      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                    ;3846      ;       зателя этой программы (инструкции JMP в ячейке 58) и запускает централь-
                    ;3847      ;       ный процессор.
                    ;3848      ;
                    ;3849      ; Входные параметры:
                    ;3850      ;
                    ;3851      ;       LS5 - адрес области отображения общей шины в ТВ (должен находиться в диа-
                    ;3852      ;             пазоне 200-3FF шестнадцатер.).
                    ;3853      ;       LS6 - в битах 1-22 содержит данные, подлежащие записи в область отобра-
                    ;3854      ;             жения общей шины.
                    ;3855      ;
                    ;3856      ; Неявные входные данные:
                    ;3857      ;
                    ;3858      ;       Отсутствуют.
                    ;3859      ;
                    ;3860      ; Выходные параметры:
                    ;3861      ;
                    ;3862      ;       Часть ОЗУ ТВ, составляющая область отображения общей шины, записывается
                    ;3863      ;       данными из LS6 по адресу, заданному в LS5.
                    ;3864      ;
                    ;3865      ; Неявные выходные данные:
                    ;3866      ;
                    ;3867      ;       Отсутствуют.
                    ;3868      ;
                    ;3869      ; Побочный эффект:
                    ;3870      ;
                    ;3871      ;       Будет модифицирован WR0.
                    ;3872      ;       Будет модифицирован WR1.
                    ;3873      ;
                    ;3874      ;---
                    ;3875      DEPOSIT.UBS:
U 0642, 0864,DC     ;3876              JSR [SHIFT.UB.ADR]                      ; переход к подпрограмме для получения правильного
                    ;3877                                                      ; адреса области отображения общей шины
U 0643, 360C,15     ;3878              MOV LS[T6.SUB] TO WR[0]                 ; выборка данных из LS6
U 0644, 4526,15     ;3879              BIC LS[#FF] TO WR[0]                    ; очистка младшего байта заданной информации. Младший
                    ;3880                                                      ; байт будет позднее размещен в битах 24-31
U 0645, 8862,AC     ;3881              JSR [SHIFT.LOOP]                        ; переход к подпрограмме сдвига WR0 на 8 позиций вправо
U 0646, E40C,15     ;3882              SWAP LS[T6.SUB] WITH WR[0]              ; взаимная замена заданных исходных данных с
                    ;3883                                                      ; модифицированными данными. LS6 теперь содержит биты
                    ;3884                                                      ; 8-22 в битах 0-14, как и требовалось. WR0 содержит
                    ;3885                                                      ; заданные исходные данные
U 0647, 452C,15     ;3886              BIC LS[#FFFFFF00] TO WR[0]               ; очистка всех битов за исключением младшего байта
                    ;3887                                                      ; заданных исходных данных
U 0648, 8862,AC     ;3888              JSR [SHIFT.LOOP]                        ; переход к подпрограмме сдвига WR0 на 8 позиций
                    ;3889                                                      ; вправо. После возврата WR0 будет содержать биты 0-7 на
                    ;3890                                                      ; месте битов 24-31. Другие биты очищены
U 0649, ED0C,15     ;3891              BIS WR[0] TO LS[T6.SUB]                  ; теперь LS6 содержит данные, подлежащие записи в ТВ, в
                    ;3892                                                      ; правильном формате, т.е. биты PA в битах 0-14 и BYTE
                    ;3893                                                      ; OFFSET, MODIFY, PROT от A до D и VALID в битах 25-31
U 064A, 1B0B,F5     ;3894              MEM.REQ[WRITE.UBS.MAP] ADRS[T5.SUB] DT[LONG]    ; выдача запроса к памяти для записи в область
                    ;3895                                                      ; отображения общей шины
U 064B, B20C,15     ;3896              WRITE.MEM LS[T6.SUB]                    ; запись данных из LS6 в область отображения общей шины
U 064C, 8868,74     ;3897              JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                    ;3898      ;
                    ;3899      ;       Подпрограмма правильного позиционирования заданного адреса для
```

```
                ;3900     ;       адресации области отображения адресов общей шины.
                ;3901     ;
                ;3902     SHIFT.UB.ADR:
64D, 360A.15    ;3903             MOV LS[T5.SUB] TO WR[0]                 ; выборка заданного адреса в WR0
64E, A2D0,15    ;3904             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
64F, A2D0,15    ;3905             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
650, A2D0,15    ;3906             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
651, A2D0,15    ;3907             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
652, 23D0,15    ;3908             ASHL WR[0]                              ; сдвиг еще на 1 позицию. Теперь адрес находится в битах
                ;3909                                                     ; 9-17 WR0
                ;3910               MOV WR[0] TO LS[T5.SUB],              ; теперь LS5 содержит правильный адрес
653, 3E0A.14    ;3911             RETURN                                  ; возврат к основной программе
                ;3912     ;
                ;3913     ;       Программа индикации буфера трансляции адресов общей шины
                ;3914     ;***
                ;3915     ;          Эта программа читает из той части контроллера памяти, которая составляет
                ;3916     ;       область отображения общей шины. Содержимое области отображения общей шины
                ;3917     ;       будет получено с битами BYTE OFFSET, MODIFY, PROT от A до D и VALID в битах
                ;3918     ;       01-07 соответственно. Биты PA 09-23 поступают в битах 08-22 соответственно.
                ;3919     ;       Биты 23-31 и бит 0 не используются, поэтому будут очищены до печати резуль-
                ;3920     ;       тата. результат для печати на консоли будет помещен в LS6.
                ;3921     ;          Эта программа выдает запрос к памяти с функцией памяти READ.UBS.MAP и
                ;3922     ;       типом данных=LONGWORD (длинное слово). Консольный процессор загружает LS5
                ;3923     ;       заданным адресом прежде, чем выполняется данная программа. Эта программа
                ;3924     ;       сдвигает адрес так, как требуется для адресации заданной ячейки области
                ;3925     ;       отображения. Заданный адрес должен быть в 16-ричном формате.
                ;3926     ;          Область отображения ОШ состоит из 512 дес. ячеек (адреса от 200 до 3FF).
                ;3927     ;       Оставшиеся ячейки в ОЗУ ТВ либо не используются, либо используются буфером
                ;3928     ;       трансляции. Содержимое буфера трансляции считывается посредством программы
                ;3929     ;       EXAMINE TB.
                ;3930
                ;3931     ; Процедура вызова:
                ;3932     ;
                ;3933     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;3934     ;       зателя этой программы (инструкции JUMP в ячейке WCS 59) и запускает цент-
                ;3935     ;       ральный процессор.
                ;3936     ;
                ;3937     ; Входные параметры:
                ;3938     ;
                ;3939     ;       LS5 - адрес ТВ в шестнадцатеричном формате.
                ;3940     ;
                ;3941     ; Неявные входные данные:
                ;3942     ;
                ;3943     ;       Отсутствуют.
                ;3944     ;
                ;3945     ; Выходные параметры:
                ;3946     ;
                ;3947     ;       LS6 - данные из области отображения общей шины по заданному адресу.
                ;3948     ;
                ;3949     ; Неявные выходные данные:
                ;3950     ;
                ;3951     ;       Отсутствуют.
                ;3952     ;
                ;3953     ; Побочный эффект:
                ;3954     ;
```

```
                    ;3955       ;       LS5 - будет модифицироваться.
                    ;3956       ;       WR0 - будет модифицироваться.
                    ;3957       ;
                    ;3958       ;---
                    ;3959       EXAMINE.UBS:
U 0654,  0864,DC    ;3960               JSR [SHIFT.UB.ADR]                      ; сдвиг адреса, заданного в LS5, в правильное положение и
                    ;3961                                                       ; замена LS5
U 0655,  1C0B,75    ;3962               MEM.REQ[READ.UBS.MAP] ADRS[T5.SUB] DT[LONG]   ; выдача запроса к памяти для чтения области
                    ;3963                                                       ; отображения UB по адресу, содержащемуся в LS5
U 0656,  3022,15    ;3964               MOV MEM.DATA TO WR[0]                   ; занесение результата в WR0
U 0657,  452A,15    ;3965               BIC LS[#FF000000] TO WR[0]              ; очистка старшего байта результата (старший байт не
                    ;3966                                                       ; является частью TB и является непредсказуемым)
U 0658,  456E,15    ;3967               BIC LS[BIT23] TO WR[0]                  ; то же для бита 23
U 0659,  4540,15    ;3968               BIC LS[BIT0] TO WR[0]                   ; то же для бита 0. WR0 теперь содержит результат, готовый
                    ;3969                                                       ; для печати
U 065A,  BE0C,15    ;3970               MOV WR[0] TO LS[T6.SUB]                 ; результат находится в LS6
  065B,  8B6B,74    ;3971               JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                    ;3972       ;
                    ;3973       ;       Программа записи в основную память
                    ;3974       ;***
                    ;3975       ;
                    ;3976       ; Описание функционирования:
                    ;3977       ;
                    ;3978       ;           Эта программа записывает в основную память по заданному адресу длинное
                    ;3979       ;       слово данных. Адрес не обязательно должен быть на границе длинного слова.
                    ;3980       ;       Консольный процессор использует эту программу для пересылки данных в ос-
                    ;3981       ;       новную память, а также и для команды DEPOSIT.MM. Консольный процессор дол-
                    ;3982       ;       жен загружать адрес в LS5 и данные в LS6 до выполнения этой программы.
                    ;3983       ;           Эта программа выдает запрос для памяти с функцией памяти WRITE.P и ти-
                    ;3984       ;       пом данных=длинное слово. Затем она выдает инструкцию WRITE.MEM LS[6] для
                    ;3985               записи данных в память. Для того, чтобы убедиться, что запись произошла
                    ;3986               без ошибок, контролируется ERR.SUM. Если установлен сигнал ERR.SUM, то
                    ;3987               центральный процессор, прежде, чем выставить CPU ATTN, устанавливает CPU
                    ;3988               ACK для информирования консольного процессора, что произошла ошибка.
                    ;3989       ;
                    ;3990       ; Процедура вызова:
                    ;3991       ;
                    ;3992       ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                    ;3993       ;       зателя этой программы (инструкции JMP в ячейке WCS 54) и запускает цент-
                    ;3994       ;       ральный процессор.
                    ;3995       ;
                    ;3996       ; Входные параметры:
                    ;3997       ;
                    ;3998       ;       LS5 - физический адрес основной памяти
                    ;3999       ;       LS6 - длинное слово данных для основной памяти
                    ;4000       ;
                    ;4001       ; Неявные входные данные:
                    ;4002       ;
                    ;4003       ;       Отсутствуют
                    ;4004       ;
                    ;4005       ; Выходные параметры:
                    ;4006       ;
                    ;4007       ;       В основной памяти по заданному адресу записаны данные из LS6
                    ;4008       ;
                    ;4009       ; Неявные выходные данные:
```

```
                ;4010     ;
                ;4011     ;       Ошибка памяти вызывает установку CPU ACK для информирования консольного
                ;4012     ;       процессора об ошибке записи
                ;4013     ;
                ;4014     ; Побочный эффект:
                ;4015     ;
                ;4016     ;       Отсутствует
                ;4017     ;
                ;4018     ;---
                ;4019     DEPOSIT.MM:
U 065C, 990A,75 ;4020             MEM.REQ[WRITE.P] ADRS[T5.SUB] DTILONG]   ; инициирование записи в основную память с
                ;4021                                                     ; использованием физического адреса, хранящегося в LS5
                ;4022               WRITE.MEM LS[T6.SUB],                 ; запись данных из LS6 в основную память и
U 065D, 320C,1D ;4023             SKIP.IF[MEM.REF.OK]                     ; пропуск следующей инструкции, если ошибки памяти
                ;4024                                                     ; отсутствуют (нет ERR SUM)
U 065E, 10D0,15 ;4025             MISC [SET.CP.ACK]                       ; установка CPU ACK, если выставлен сигнал ERR.SUM
U 065F, 3644,15 ;4026             MOV LS[#4] TO WR[0]                     ; занесение числа 4 в WR0
  0660, 6C0A,15 ;4027             ADD WR[0] TO LS[T5.SUB]                 ; увеличение LS5 для записи следующего длинного слова
  0661, 8868,74 ;4028             JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4029     ;
                ;4030     ;       Программа индикации основной памяти
                ;4031     ;***
                ;4032     ;
                ;4033     ; Описание функционирования:
                ;4034     ;
                ;4035     ;          Эта программа используется для чтения из основной памяти длинного сло-
                ;4036     ;       ва по заданному физическому адресу. Адрес не обязательно должен быть на
                ;4037     ;       границе длинного слова. Эта программа используется консольным процессором
                ;4038     ;       при команде EXAMINE.MM. Прежде, чем выполняется эта программа, консольный
                ;4039     ;       процессор должен загрузить в LS5 заданный физический адрес.
                ;4040     ;          Эта программа производит вначале запись в CSR1 для установки бита INH
                ;4041     ;       REP CRD (запрет сообщения об исправимых ошибках чтения), чтобы препятст-
                ;4042     ;       вовать появлению ошибки системы памяти (ERR SUM) при одиночной ошибке, ко-
                ;4043     ;       торая была исправлена. Затем программа выдает запрос к памяти с функцией
                ;4044     ;       READ.P и типом данных=длинное слово. Затем она читает ячейку памяти и по-
                ;4045     ;       мещает результат в LS6. Она также проверяет ERR SUM и, если произошла
                ;4046     ;       ошибка, выдает CPU ACK (исправленные ошибки в одиночных битах не вызывают
                ;4047     ;       ERR SUM). Если произошла ошибка, то бит CPU ACK будет установлен до выдачи
                ;4048     ;       CPU ATTN для информирования консольного процессора, что имеется ошибка.
                ;4049     ;
                ;4050     ; Процедура вызова:
                ;4051     ;
                ;4052     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;4053     ;       зателя этой программы (инструкция JMP в ячейке WCS 55) и запускает цент-
                ;4054     ;       ральный процессор.
                ;4055     ;
                ;4056     ; Входные параметры:
                ;4057     ;
                ;4058     ;       LS5 - физический адрес основной памяти, по которому будет происходить
                ;4059     ;             чтение
                ;4060     ;
                ;4061     ; Неявные входные данные:
                ;4062     ;
                ;4063     ;       LS[INH.CRD] - данные для записи в CSR1 с целью запрета ERR SUM для
                ;4064     ;                     исправимых ошибок чтения (CRD)
```

```
                ;4065     ;       LS[CSR1]    - адрес для записи CSR1
                ;4066     ;
                ;4067     ; Выходные параметры:
                ;4068     ;
                ;4069     ;       LS6 - длинное слово данных, считанное по заданному физическому адресу
                ;4070     ;             памяти
                ;4071     ;
                ;4072     ; Неявные выходные данные:
                ;4073     ;
                ;4074     ;       CPU ACK, если зафиксирована ошибка
                ;4075     ;
                ;4076     ; Побочный эффект:
                ;4077     ;
                ;4078     ;       Отсутствует
                ;4079     ;
                ;4080     ;---
                ;4081     EXAMINE.MM:
U 0662, 1D45,F5 ;4082           MEM.REQ[WRITE.CSR] ADRS[CSR1] DTILONG]   ; выдача запроса к памяти для записи CSR1
U 0663, B278,15 ;4083           WRITE.MEM LS[INH.CRD]                    ; установка в CSR1 бита INH REP CRD и очистка запрета
                ;4084                                                    ; коррекции ошибок (ECC)
U 0664, 180B,F5 ;4085           MEM.REQ[READ.PJ ADRSITS.SUB] DTILONG]    ; выдача запроса к памяти для чтения длинного слова
                ;4086                                                    ; данных из основной памяти. Физический адрес находится
                ;4087                                                    ; в LS5
                ;4088             MOV MEM.DATA TO LSIT6.SUB],            ; чтение данных и занесение в LS6
U 0665, 380C,1D ;4089           SKIP.IF[MEM.REF.OK]                      ; пропуск следующей инструкции, если ошибки памяти
                ;4090                                                    ; отсутствуют (нет ERR SUM)
U 0666, 10D0,15 ;4091           MISC [SET.CP.ACK]                        ; установка CPU ACK, если произошла ошибка
U 0667, 8868,74 ;4092           JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;4093     ;
                ;4094     ;       Программа индикации регистров IDC
                ;4095     ;***
                ;4096     ;
                ;4097     ; Описание функционирования:
                ;4098     ;
                ;4099     ;       Следующие программы используются для чтения данных из регистров необяза-
                ;4100     ;       тельного модуля IDC. Результат для печати на консоли заносится в LS6.
                ;4101     ;
                ;4102     ; Процедура вызова:
                ;4103     ;
                ;4104     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;4105     ;       зателя этой программы (инструкции JMP в ячейках WCS от 5A до 5E) и запус-
                ;4106     ;       кает центральный процессор
                ;4107     ;
                ;4108     ; Входные параметры:
                ;4109     ;
                ;4110     ;       Отсутствуют
                ;4111     ;
                ;4112     ; Неявные входные данные:
                ;4113     ;
                ;4114     ;       Отсутствуют
                ;4115     ;
                ;4116     ; Выходные параметры:
                ;4117     ;
                ;4118     ;       LS6.- данные из заданного регистра IDC
                ;4119     ;
```

```
                    ;4120     ; Неявные выходные данные:
                    ;4121     ;
                    ;4122     ;       Отсутствуют
                    ;4123     ;
                    ;4124     ; Побочный эффект:
                    ;4125     ;
                    ;4126     ;       Отсутствует
                    ;4127     ;
                    ;4128     ;---
                    ;4129     EXAMINE.ICSR:
U 0668, 9090,15     ;4130             MISC [SET.PORT.I.0]                   ; выборка порта на шину
U 0669, 9780,15     ;4131             PORT.READ PORT.ADRS[CSR]              ; пересылка команды чтения
U 066A, 0867,64     ;4132             JMP [READ.IDC]                        ; переход к чтению данных
                    ;4133     EXAMINE.IDAR:
U 066B, 9090,15     ;4134             MISC [SET.PORT.I.0]                   ; выборка порта на шину
U 066C, 1784,15     ;4135             PORT.READ PORT.ADRS[DAR]              ; пересылка команды чтения
U 066D, 0867,64     ;4136             JMP [READ.IDC]                        ; переход к чтению данных
                    ;4137     EXAMINE.DBUF:
U 066E, 9090,15     ;4138             MISC [SET.PORT.I.0]                   ; выборка порта на шину
U 066F, 978C,15     ;4139             PORT.READ PORT.ADRS[DATA.WORD]        ; пересылка команды чтения
U 0670, 0867,64     ;4140             JMP [READ.IDC]                        ; переход к чтению данных
                    ;4141     EXAMINE.PATT:
U 0671, 9090,15     ;4142             MISC [SET.PORT.I.0]                   ; выборка порта на шину
U 0672, 1790,15     ;4143             PORT.READ PORT.ADRS[PATTERN]          ; пересылка команды чтения
U 0673, 0867,64     ;4144             JMP [READ.IDC]                        ; переход к чтению данных
                    ;4145     EXAMINE.POSIT:
U 0674, 9090,15     ;4146             MISC [SET.PORT.I.0]                   ; выборка порта на шину
U 0675, 9794,15     ;4147             PORT.READ PORT.ADRS[POSITION]         ; пересылка команды чтения
                    ;4148     READ.IDC:
U 0676, 3A0C,15     ;4149             MOV PORT TO LS[T6.SUB]                ; чтение данных и занесение в LS6
U 0677, 1080,15     ;4150             MISC [SET.ACC.I.0]                    ; возврат к выборке ускорителя
U 0678, 8868,74     ;4151             JMP [WAIT.SUB]                        ; переход к установке ATTN и ожиданию
                    ;4152     ;
                    ;4153     ;       Программы записи в регистры IDC
                    ;4154     ;***
                    ;4155     ;
                    ;4156     ; Описание функционирования:
                    ;4157     ;
                    ;4158     ;       Следующие программы используются для записи данных в регистры необяза-
                    ;4159     ;       тельного модуля IDC. Данные берутся из LS6, которая предварительно заг-
                    ;4160     ;       ружается монитором, если выполняется команда DEPOSIT.
                    ;4161     ;
                    ;4162     ; Процедура вызова:
                    ;4163     ;
                    ;4164     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) указатель
                    ;4165     ;       этой программы (инструкции JMP в ячейках WCS от 5F до 61) и запускает
                    ;4166     ;       центральный процессор.
                    ;4167     ;
                    ;4168     ; Входные параметры:
                    ;4169     ;
                    ;4170     ;       LS6 - код данных, подлежащих записи
                    ;4171     ;
                    ;4172     ; Неявные входные данные:
                    ;4173     ;
                    ;4174     ;       Отсутствуют
```

```
                ;4175     ;
                ;4176     ; Выходные параметры:
                ;4177     ;
                ;4178     ;       Отсутствуют
                ;4179     ;
                ;4180     ; Неявные выходные параметры:
                ;4181     ;
                ;4182     ;       Отсутствуют
                ;4183     ;
                ;4184     ; Побочный эффект:
                ;4185     ;
                ;4186     ;       Отсутствует
                ;4187     ;
                ;4188     ;---
                ;4189    DEPOSIT.ICSR:
U 0679, 360C,15 ;4190            MOV LS[T6.SUB] TO WR[0]                 ; выборка данных, подлежащих записи
U 067A, 17A0,15 ;4191            PORT.WRITE PORT.ADRS[CSR] WITH WR[0]    ; запись в IDC
U 067B, 8868,74 ;4192            JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4193    DEPOSIT.IDAR:
U 067C, 360C,15 ;4194            MOV LS[T6.SUB] TO WR[0]                 ; выборка данных, подлежащих записи
U 067D, 97A4,15 ;4195            PORT.WRITE PORT.ADRS[DAR] WITH WR[0]    ; запись в IDC
U 067E, 8868,74 ;4196            JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4197    DEPOSIT.DBUF:
U 067F, 360C,15 ;4198            MOV LS[T6.SUB] TO WR[0]                 ; выборка данных, подлежащих записи
U 0680, 17AC,15 ;4199            PORT.WRITE PORT.ADRS[DATA.WORD] WITH WR[0]  ; запись в IDC
U 0681, 8868,74 ;4200            JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4201    CLEAR.FIFO.ADDR:
U 0682, 17C0,15 ;4202            PORT.CONTROL [CLEAR.FIFO.CNTR]          ; очистка адреса FIFO A или FIFO B
U 0683, 8868,74 ;4203            JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4204    SELECT.FIFO.A:
U 0684, 17D8,15 ;4205            PORT.CONTROL [SEL.FIFO.A]               ; выборка FIFO A
U 0685, 8868,74 ;4206            JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4207    SELECT.FIFO.B:
U 0686, 97DC,15 ;4208            PORT.CONTROL [SEL.FIFO.B]               ; выборка FIFO B
                ;4209    WAIT.SUB:
U 0687, 10E0,15 ;4210            MISC [SET.CP.ATTN]                      ; выдача консольному процессору сигнала CPU ATTN
                ;4211    WAIT.DE.EX:
U 0688, 8868,84 ;4212            JMP [WAIT.DE.EX]                        ; зацикливание на себя до тех пор, пока консольный
                ;4213                                                    ; процессор возьмет управление
                ;4214     ;
                ;4215     ;       Программа запоминания рабочих регистров
                ;4216     ;***
                ;4217     ;
                ;4218     ; Описание функционирования:
                ;4219     ;
                ;4220     ;          Эта программа запоминает рабочие регистры 0-3 в ячейках местной памяти
                ;4221     ;       LS 1-4. Если консольный процессор забирает управление от центрального про-
                ;4222     ;       цессора, он вызывает эту программу для того, чтобы рабочие регистры могли
                ;4223     ;       использоваться консольным процессором. Рабочие регистры восстанавливаются
                ;4224     ;       другой программой, вызываемой консольным процессором прежде, чем централь-
                ;4225     ;       ный процессор возобновляет выполнение (например, при команде CONTINUE).
                ;4226     ;          Эта программа просто пересылает данные из WR 0-3 в ячейки LS 1-4 и за-
                ;4227     ;       тем выдает для консольного процессора CPU ATTN. Затем она зацикливается
                ;4228     ;       инструкцией JMP на себя до тех пор, пока консольный процессор возьмет
                ;4229     ;       управление.
```

```
                ;4230     ;
                ;4231     ; Процедура вызова:
                ;4232     ;
                ;4233     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) указатель
                ;4234     ;       этой программы (инструкцию JMP в ячейке WCS 46) и запускает центральный
                ;4235     ;       процессор.
                ;4236     ;
                ;4237     ; Входные параметры:
                ;4238     ;
                ;4239     ;       Отсутствуют
                ;4240     ;
                ;4241     ; Неявные входные данные:
                ;4242     ;
                ;4243     ;       Отсутствуют
                ;4244     ;
                ;4245     ; Выходные параметры:
                ;4246     ;
                ;4247     ;       LS1 - содержимое WR0
                ;4248     ;       LS2 - содержимое WR1
                ;4249     ;       LS3 - содержимое WR2
                ;4250     ;       LS4 - содержимое WR3
                ;4251     ;
                ;4252     ; Неявные выходные данные:
                ;4253
                ;4254     ;       Отсутствуют
                ;4255     ;
                ;4256     ; Побочный эффект:
                ;4257
                ;4258     ;       Отсутствует
                ;4259
                ;4260     ;--
                ;4261     SAVE.WR:
U 0689, 3E02,15 ;4262             MOV WR[0] TO LS[SAV.WR0]          ; запоминание WR0 в ячейке LS1
U 068A, BE04,95 ;4263             MOV WR[1] TO LS[SAV.WR1]          ; запоминание WR1 в ячейке LS2
U 068B, 3E07,15 ;4264             MOV WR[2] TO LS[SAV.WR2]          ; запоминание WR2 в ячейке LS3
U 068C, 3E09,95 ;4265             MOV WR[3] TO LS[SAV.WR3]          ; запоминание WR3 в ячейке LS4
U 068D, 8868,74 ;4266             JMP [WAIT.SUB]                   ; переход к установке ATTN и ожиданию
                ;4267
                ;4268     ;       Программа восстановления рабочих регистров
                ;4269     ;***
                ;4270
                ;4271     ; Описание функционирования:
                ;4272     ;
                ;4273     ;          Эта программа восстанавливает рабочие регистры К1804ВС1, хранящиеся в
                ;4274     ;       LS 1-4. Консольный процессор вызывает эту программу прежде, чем запускает-
                ;4275     ;       ся центральный процессор после останова по инициативе консольного процес-
                ;4276     ;       сора.
                ;4277     ;          Эта программа просто пересылает данные из ячеек LS 1-4 в рабочие регис-
                ;4278     ;       тры (WR) 0-3 и затем выдает для консольного процессора CPU ATTN. После
                ;4279     ;       этого выполняется инструкция JMP на себя, пока консольный процессор не
                ;4280     ;       возобновит управление.
                ;4281     ;
                ;4282     ; Процедура вызова:
                ;4283     ;
                ;4284     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
```

```
                ;4285      ;       зателя этой программы (инструкций JMP в ячейке WCS 47) и запускает цент-
                ;4286      ;       ральный процессор.
                ;4287      ;
                ;4288      ; Входные параметры:
                ;4289      ;
                ;4290      ;       LS1 содержит данные для восстановления WR0
                ;4291      ;       LS2 содержит данные для восстановления WR1
                ;4292      ;       LS3 содержит данные для восстановления WR2
                ;4293      ;       LS4 содержит данные для восстановления WR3
                ;4294      ;
                ;4295      ; Неявные входные данные:
                ;4296      ;
                ;4297      ;       Отсутствуют
                ;4298      ;
                ;4299      ; Выходны   араметры:
                ;4300      ;
                ;4301      ;       WR0 получает данные из LS1
                ;4302      ;       WR1 получает данные из LS2
                ;4303      ;       WR2 получает данные из LS3
                ;4304      ;       WR3 получает данные из LS4
                ;4305      ;
                ;4306      ; Неявные выходные данные:
                ;4307      ;
                ;4308      ;       Отсутствуют
                ;4309      ;
                ;4310      ; Побочный эффект:
                ;4311      ;
                ;4312      ;       Отсутствует
                ;4313      ;
                ;4314      ;---
                ;4315      RESTORE.WR:
U 068E,  B602,15 ;4316             MOV LS[SAV.WR0] TO WR[0]                ; восстановление WR0
U 068F,  3604,95 ;4317             MOV LS[SAV.WR1] TO WR[1]                ; восстановление WR1
U 0690,  B607,15 ;4318             MOV LS[SAV.WR2] TO WR[2]                ; восстановление WR2
U 0691,  B609,95 ;4319             MOV LS[SAV.WR3] TO WR[3]                ; восстановление WR3
U 0692,  8868,74 ;4320             JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4321
```

```
;4322      .PAGE "ПОДПРОГРАММЫ WCS ДЛЯ НУЖД ЦЕНТРАЛЬНОГО ПРОЦЕССОРА"
;4323      ;
;4324      ;            Программа обработки ошибок
;4325      ;
;4326      ; Описание функционирования:
;4327      ;
;4328      ;     Эта программа используется для обнаружения ошибок и выдачи диагностических
;4329      ;   сообщений об ошибках, появляющихся во время выполнения микродиагностики, ба-
;4330      ;   зируемой на WCS. Микрокод WCS устанавливает параметры, перечисленные ниже, и
;4331      ;   вызывает эту программу. Эта программа сравнивает WR1 (ожидаемые данные) и
;4332      ;   WR0 (полученные данные) в соответствии с маской ошибок. Установленные биты
;4333      ;   маски указывают те биты, которые не проверяются на наличие ошибки.
;4334      ;     Если WR0 и WR1 равны (ошибок нет), программа выполняет возврат+1 к точке
;4335      ;   вызова. Единственным исключением является условие, когда установлено зацикли-
;4336      ;   вание на ошибке. В этом случае проверяется номер ошибки, чтобы убедиться,
;4337      ;   происходила ли эта ошибка ранее. Если да, то вынуждается цикл по ошибке, пу-
;4338      ;   тем выполнения возврата. Таким образом фиксируется цикл для неустойчивых
;4339      ;   ошибок.
;4340      ;     Если WR0 и WR1 не равны (ошибка), тогда в слове управления и состояния уста-
;4341      ;   навливается бит 8, ожидаемые и полученные данные запоминаются в LS для печа-
;4342      ;   ти и проверяются признаки в слове управления ошибками. Оно управляет запре-
;4343      ;   том печати ошибок, разрешением зацикливания на ошибке и т.д. После печати
;4344      ;   ошибки выполняется возврат+1 к точке вызова. Зацикливание на ошибке вместо
;4345      ;   возврата+1 вызывает возврат, чтобы вынудить зацикливание по ошибке. Также, в
;4346      ;   конце программы переключается CPU ACK для образования точки синхронизации
;4347      ;   осциллографа.
;4348      ;
;4349      ; Процедура вызова:
;4350      ;
;4351      ;   JSR [CHECK.RESULT]
;4352      ;
;4353      ; Входные параметры:
;4354      ;
;4355      ;   WR[0] = полученные данные, т.е. действительный результат теста
;4356      ;   WR[1] = ожидаемые данные, т.е. такой результат, каким он должен быть
;4357      ;
;4358      ; Неявные входные данные:
;4359      ;
;4360      ;   LS[ERROR.MASK]      = используется для маскирования битов, которые не подле-
;4361      ;                         жат проверке
;4362      ;   LS[ERROR.NUMBER]    = номер текущей ошибки
;4363      ;   LS[PREVIOUS.ERROR]  = номер ошибки предыдущих данных
;4364      ;   LS[CONTROL.STATUS]  = слово управления и состояния. Содержит информацию для
;4365      ;                         монитора о необходимых действиях, записанную централь-
;4366      ;                         ным процессором
;4367      ;   LS[ERROR.CONTROL]   = информация такого типа, как зацикливание по ошибке,
;4368      ;                         звуковой сигнал при ошибке, запрет сообщений об ошиб-
;4369      ;                         ках и останов при ошибке.
;4370      ;
;4371      ; Выходные параметры:
;4372      ;
;4373      ;   Отсутствуют
;4374      ;
;4375      ; Неявные выходные данные:
;4376      ;
```

```
                    ;4377     ;    LS[CONTROL.STATUS] = слово управления и состояния с новой информацией
                    ;4378     ;    LS[PREVIOUS.ERROR] = номер последней ошибки (модифицируется только при ошибке,
                    ;4379     ;                         если разрешено зацикливание при ошибке)
                    ;4380     ;    LS[EXPECTED.DATA]  = ожидаемый результат, если была обнаружена ошибка
                    ;4381     ;    LS[RECEIVED.DATA]  = действительный результат, если была обнаружена ошибка
                    ;4382     ;
                    ;4383     ; Побочный эффект:
                    ;4384     ;
                    ;4385     ;    WR0, WR1 и регистр Q модифицированы и не восстановлены перед возвратом.
                    ;4386     ;    Если разрешено зацикливание при ошибке и должен выполняться цикл по ошибке,
                    ;4387     ;    будет переключен CPU ACK для синхронизации осциллографа.
                    ;4388     ;
                    ;4389     CHECK.RESULT:
U 0693, 458A,15     ;4390              BIC LS[ERROR.MASK] TO WR[0]                ; маскирование непроверяемых битов (очистка) для
                    ;4391                                                         ; полученных данных
U 0694, C58A.95     ;4392              BIC LS[ERROR.MASK] TO WR[1]                ; то же для ожидаемых данных
                    ;4393                XOR WR[0] WITH WR[1] TO Q,               ; сравнение полученных данных в WR[0] с ожидаемыми
                    ;4394                                                         ; данными в WR[1] для обнаружения ошибки
U 0695, AB80,B5     ;4395              DT(LONG)&SET.ALU.CC                        ;
U 0696, 8869,F1     ;4396              JMP.IF[NEQ] TO [ERROR]                     ; переход, если ошибка
U 0697, 3690,15     ;4397              MOV LS[ERROR.CONTROL] TO WR[0]             ; выборка из LS слова управления ошибками
                    ;4398                BIT WR[0] WITH LS[LOE],                  ; проверка, разрешено ли зацикливание по ошибке и
U 0698, 5940,35     ;4399              DT(LONG)&SET.ALU.CC                        ; установка кодов условий
U 0699, DB00,01     ;4400              SKIP.IF[BIT.SET]                           ; пропуск следующей инструкции, если да
U 069A, DB00,16     ;4401              RETURN+1                                   ; иначе возврат+1 (отсутствует ошибка и не разрешено
                    ;4402                                                         ; зацикливание)
U 069B, 36A0,15     ;4403              MOV LS[PREVIOUS.ERROR] TO WR[0]            ; если отсутствует ошибка, но зацикливание по ошибке
                    ;4404                                                         ; разрешено, проверка, была ли предыдущая ошибка
                    ;4405                XOR WR[0] WITH LS[ERROR.NUMBER] TO Q,;
U 069C, CD82,35     ;4406              DT(LONG)&SET.ALU.CC                        ; является ли это место тем же, в котором раньше была
                    ;4407                                                         ; ошибка (неустойчивая ошибка)?
U 069D, 086B,91     ;4408              JMP.IF[NEQ] TO [RET+1]                     ; если нет, переход к возврату без зацикливания по
                    ;4409                                                         ; ошибке (возврат+1)
U 069E, 086B,C4     ;4410              JMP [RET]                                  ; иначе цикл при ошибке (зацикливание будет происходить
                    ;4411                                                         ; даже если ошибка изчезла)
                    ;4412     ERROR:
U 069F, 3EB6,15     ;4413              MOV WR[0] TO LS[RECEIVED.DATA]              ; занесение результата теста в LS для печати
U 06A0, 3690,15     ;4414              MOV LS[ERROR.CONTROL] TO WR[0]              ; выборка битов управления ошибками для проверки
                    ;4415                                                         ; разрешения зацикливания
                    ;4416                BIT WR[0] WITH LS[LOOP.COMMAND],         ; проверка, установлен ли бит 6
U 06A1, 594C,35     ;4417              DT(LONG)&SET.ALU.CC                        ; и установка кодов условий
U 06A2, 086B,C1     ;4418              JMP.IF[BIT.SET] TO [RET]                   ; переход на зацикливание, если установлен (быстрый
                    ;4419                                                         ; цикл)
U 06A3, 3EB4,95     ;4420              MOV WR[1] TO LS[EXPECTED.DATA]              ; занесение в LS ожидаемых данных для печати
U 06A4, 3680,95     ;4421              MOV LS[CONTROL.STATUS] TO WR[1]             ; выборка из LS слова управления и состояния
U 06A5, C532,95     ;4422              BIC LS[#FE7FFFFF] TO WR[1]                 ; очистка всех битов слова управления и состояния, кроме
                    ;4423                                                         ; 23 и 24
U 06A6, C750,95     ;4424              BIS LS[ERROR] TO WR[1]                      ; установка в слове управления и состояния бита 8
                    ;4425                                                         ; (указывает, что обнаружена ошибка)
U 06A7, BE80,95     ;4426              MOV WR[1] TO LS[CONTROL.STATUS]             ; запись откорректированного слова управления и
                    ;4427                                                         ; состояния обратно в LS 40
                    ;4428                BIT WR[0] WITH LS[BELL],                 ; проверка, установлен ли бит звукового сигнала при
                    ;4429                                                         ; ошибке (WR0 все еще содержит слово управления
                    ;4430                                                         ; ошибками)
U 06A8, D944,35     ;4431              DT(LONG)&SET.ALU.CC                        ; установка кодов условий
```

```
U 06A9, 886A,D9 ;4432            JMP.IF[BITS.CLR] TO [CHECK.NER]          ; если бит "звуковой сигнал при ошибке" не установлен,
                ;4433                                                     ; переход для проверки, запрещаются или нет сообщения об
                ;4434                                                     ; ошибках
U 06AA, 10E0,15 ;4435            MISC [SET.CP.ATTN]                       ; иначе установка CPU ATTN (для звукового сигнала)
                ;4436    WAIT.1:
U 06AB, 086A,B4 ;4437            JMP [WAIT.1]                             ; ожидание ответа консольного процессора
U 06AC, 086B,64 ;4438            JMP [LOOP]                               ; переход к проверке цикла при ошибке после того, как
                ;4439                                                     ; консольный процессор закончил
                ;4440    CHECK.NER:
                ;4441            BIT WR[0] WITH LS[NER],                  ; если не звуковой сигнал, проверка, запрещены ли
                ;4442                                                     ; сообщения об ошибках
U 06AD, D942,35 ;4443            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06AE, 886B,21 ;4444            JMP.IF[BIT.SET] TO [CHECK.HALT]          ; если сообщения об ошибках запрещены, переход к
                ;4445                                                     ; проверке останова по ошибке
U 06AF, 10E0,15 ;4446            MISC [SET.CP.ATTN]                       ; установка CPU ATTN (сообщение об ошибке)
                ;4447    WAIT.2:
U 06B0, 086B,04 ;4448            JMP [WAIT.2]                             ; ожидание ответа консольного процессора
U 06B1, 086B,64 ;4449            JMP [LOOP]                               ; переход к проверке зацикливания по ошибке после того,
                ;4450                                                     ; как консольный процессор закончил
                ;4451    CHECK.HALT:
                ;4452                BIT WR[0] WITH LS[HOE],              ; проверка, разрешен ли останов при ошибке и
U 06B2, 5946,35 ;4453            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06B3, 886B,69 ;4454            JMP.IF[BITS.CLR] TO [LOOP]               ; если нет, переход к проверке зацикливания при ошибке
U 06B4, 10E0,15 ;4455            MISC [SET.CP.ATTN]                       ; иначе установка CPU ATTN (останов при ошибке)
                ;4456    WAIT.3:
U 06B5, 086B,54 ;4457            JMP [WAIT.3]                             ; ожидание ответа консольного процессора
                ;4458    LOOP:
U 06B6, 3690,15 ;4459            MOV LS[ERROR.CONTROL] TO WR[0]           ; выборка битов управления ошибками (NER, HOE и т.д.)
                ;4460              BIT WR[0] WITH LS[LOE],                ; проверка, разрешено ли зацикливание при ошибке и
U 06B7, 5940,35 ;4461            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06B8, DB00,01 ;4462            SKIP.IF[BIT.SET]                         ; пропуск инструкции, если разрешено зацикливание при
                ;4463                                                     ; ошибке
                ;4464    RET+1:
U 06B9, DB00,16 ;4465            RETURN+1                                 ; иначе возврат+1 в цикл по отсутствию ошибок
U 06BA, 36B2,15 ;4466            MOV LS[ERROR.NUMBER] TO WR[0]            ; если зацикливание, выборка номера ошибки
U 06BB, BEA0,15 ;4467            MOV WR[0] TO LS[PREVIOUS.ERROR]          ; запоминание его по месту хранения предыдущей ошибки
                ;4468    RET:
U 06BC, 10D0,15 ;4469            MISC [SET.CP.ACK]                        ; установка CPU ACK для синхронизации осциллографа
                ;4470              MISC [CLR.CP.ATTN.AND.ACK],            ; и его очистка
U 06BD, 10C0,14 ;4471            RETURN                                   ; возврат к тесту и цикл при ошибке
                ;4472     ;
                ;4473     ;
                ;4474    ERR.NO.TEST:
U 06BE, DB00,15 ;4475            NOP                                      ; этот NOP вызывает ошибку в следовании тестов
U 06BF, B65E,15 ;4476            MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;4477                                                     ; состояния
U 06C0, 3EB0,15 ;4478            MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
                ;4479                                                     ; 15 указывает конс.процессору начало теста
U 06C1, 8868,74 ;4480            JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;4481     ;
                ;4482     ;
                ;4483     ;                Программа инициации переноса данных
                ;4484     ;
                ;4485     ;    Эта программа загружает LS данными, необходимыми для выполнения тестов. За-
                ;4486     ;    тем она выдает CPU ATTN при очищенном слове управления и состояния, что ука-
```

```
                ;4487     ;    зывает, что все пересылки завершены.
                ;4488     ;
                ;4489     ;    РЕЗУЛЬТАТ: LS загружена константами. Монитор информирован, что перенос дан-
                ;4490     ;               ных завершен
                ;4491     .LIST
                ;4492     TRANSFER.POINT:
U 06C2, 2F80,15 ;4493             CLR WR[0]                          ; начало. Очистка WR0
U 06C3, BFFE,15 ;4494             MOV WR[0] TO LS[PSL.HW]             ; обеспечение условия, что бит "РЕЖИМ
                ;4495                                                ; СОВМЕСТИМОСТИ". очищен
U 06C4, 2040,15 ;4496             INC WR[0]                          ; WR0 теперь содержит 1
U 06C5, 23D0,15 ;4497             ASHL WR[0]                         ; 10(B)
U 06C6, 2040,15 ;4498             INC WR[0]                          ; 11(B)
U 06C7, 23D0,15 ;4499             ASHL WR[0]                         ; 110(B)
U 06C8, 2040,15 ;4500             INC WR[0]                          ; 111(B)
U 06C9, 23D0,15 ;4501             ASHL WR[0]                         ; 1110(B)
U 06CA, 23D0,15 ;4502             ASHL WR[0]                         ; 1 1100(B)
U 06CB, 23D0,15 ;4503             ASHL WR[0]                         ; 11 1000(B)
U 06CC, 23D0,15 ;4504             ASHL WR[0]                         ; 111 0000(B). WR0 теперь содержит 70(H). Это правильный
                ;4505                                                ; начальный адрес секции данных (включая счетчик длинных
                ;4506                                                ; слов, приемник и адрес приемника)
U 06CD, 3E0E,15 ;4507             MOV WR[0] TO LS[TRANSFER.POINTER]   ; загрузка LS 7 указателем
                ;4508                                                ; для пересылки секции данных
U 06CE, 2F80,15 ;4509             CLR WR[0]                          ; 0 в WR0
U 06CF, 2040,15 ;4510             INC WR[0]                          ; 1 в WR0
U 06D0, 23D0,15 ;4511             ASHL WR[0]                         ; 10(B)
U 06D1, 23D0,15 ;4512             ASHL WR[0]                         ; 100(B)
U 06D2, 23D0,15 ;4513             ASHL WR[0]                         ; 1000(B)
U 06D3, 23D0,15 ;4514             ASHL WR[0]                         ; 1 0000(B)
U 06D4, 23D0,15 ;4515             ASHL WR[0]                         ; 10 0000(B)
U 06D5, 23D0,15 ;4516             ASHL WR[0]                         ; 100 0000(B)
U 06D6, 23D0,15 ;4517             ASHL WR[0]                         ; 1000 0000(B)
U 06D7, 23D0,15 ;4518             ASHL WR[0]                         ; 1 0000 0000(B)
U 06D8, 23D0,15 ;4519             ASHL WR[0]                         ; 10 0000 0000(B)
U 06D9, 23D0,15 ;4520             ASHL WR[0]                         ; 100 0000 0000(B)
U 06DA, 23D0,15 ;4521             ASHL WR[0]                         ; 1000 0000 0000(B)
U 06DB, 23D0,15 ;4522             ASHL WR[0]                         ; 1 0000 0000 0000(B)
U 06DC, 23D0,15 ;4523             ASHL WR[0]                         ; 10 0000 0000 0000(B)
U 06DD, 3E80,15 ;4524             MOV WR[0] TO LS[CONTROL.STATUS]     ; установка бита 13 в слове управления
                ;4525                                                ; и состояния в LS (бит 13 указывает, что консольный
                ;4526                                                ; процессор должен выполнять перенос данных)
U 06DE, 10E0,15 ;4527             MISC [SET.CP.ATTN]                 ; выдача CPU ATTN для консольного процессора
                ;4528                                                ; (выполнение первой пересылки в LS в ячейки от 0 до 77(H))
                ;4529     WAIT.XFER1:
U 06DF, 086D,F4 ;4530             JMP [WAIT.XFER1]                   ; зацикливание здесь в ожидании ответа консольного
                ;4531                                                ; процессора
U 06E0, 36CA,15 ;4532             MOV LS[#162(H)] TO WR[0]           ; выборка начального адреса второй половины
                ;4533                                                ; данных в LS
U 06E1, 3E0E,15 ;4534             MOV WR[0] TO LS[TRANSFER.POINTER]   ; загрузка в LS начального
                ;4535                                                ; адреса для консольного процессора
U 06E2, 365A,15 ;4536             MOV LS[XFER.DATA] TO WR[0]          ; BIT 13 указывает на необходимость
                ;4537                                                ; переноса данных
U 06E3, 3E80,15 ;4538             MOV WR[0] TO LS[CONTROL.STATUS]     ; установка слова управления и
                ;4539                                                ; состояния
U 06E4, 10E0,15 ;4540             MISC [SET.CP.ATTN]                 ; выдача CPU ATTN для консольного процессора
                ;4541                                                ; (выполнить вторую пересылку в LS в ячейки от 80 до F7(H))
```

```
                    ;4542    WAIT.XFER2:
U 06E5,  0B6E,54    ;4543            JMP [WAIT.XFER2]                        ; зацикливание здесь в ожидании ответа консольного
                    ;4544                                                    ; процессора
U 06E6,  B724,15    ;4545            MOV LS[HI.IPL] TO WR[0]
U 06E7,  BFFE,15    ;4546            MOV WR[0] TO LS[PSL.HW]                  ; установка IPL=1F
                    ;4547     ;
                    ;4548     ;
U 06EB,  17CC,15    ;4549            PORT.CONTROL [CLEAR.IDC]                ; установка исходного состояния IDC
U 06E9,  17CC,15    ;4550            PORT.CONTROL [CLEAR.IDC]                ;
U 06EA,  17CC,15    ;4551            PORT.CONTROL [CLEAR.IDC]                ;
U 06EB,  17CC,15    ;4552            PORT.CONTROL [CLEAR.IDC]                ; IDC остановлен
                    ;4553     ;
U 06EC,  B64E,95    ;4554            MOV LS[#80] TO WR[1]                    ; установка бита 7 в WR1
U 06ED,  97A0,95    ;4555            PORT.WRITE PORT.ADRS[CSR] WITH WR[1]    ; сброс CRDY в IDC
U 06EE,  17D4,15    ;4556            PORT.CONTROL [CLEAR.AUTO]               ; сброс автономного режима в IDC
U 06EF,  97C4,15    ;4557            PORT.CONTROL [RESET.BR]                 ; сброс BR 7 в IDC
                    ;4558     ;
U 06F0,  E580,15    ;4559            CLR LS[CONTROL.STATUS]                  ; очистка слова управления и состояния
U 06F1,  10E0,15    ;4560            MISC [SET.CP.ATTN]                      ; вызов консольного процессора
                    ;4561    WAIT.XFER:
U 06F2,  0B6F,24    ;4562            JMP [WAIT.XFER]                         ; ожидание ответа конс. процессора
U 06F3,  5B00,14    ;4563            RETURN                                  ; используется только когда пересылка вызывается
                    ;4564                                                    ; из микродиагностики
                    ;4565     .LIST
                    ;4566     ;
```

```
                ;4567       .PAGE
                ;4568       .REGION/700,3FFF
```

```
                ;4569     .PAGE "ОБЩИЕ ПОДПРОГРАММЫ ТЕСТОВ"
                ;4570     ;
                ;4571     ;    Эта подпрограмма читает номер ошибки и увеличивает на единицу
                ;4572     ;
                ;4573     INC.EN:
U 0700,  3682,15 ;4574            MOV LS[ERROR.NUMBER] TO WR[0]          ; чтение номера ошибки
U 0701,  2040,15 ;4575            INC WR[0]                              ; и увеличение
U 0702,  BE82,15 ;4576            MOV WR[0] TO LS[ERROR.NUMBER]          ; его
U 0703,  5B00,14 ;4577            RETURN                                 ;
                ;4578     ;
                ;4579     ;    Эта подпрограмма читает номер ошибки и уменьшает на единицу
                ;4580     ;
                ;4581     DEC.EN:
U 0704,  3682,15 ;4582            MOV LS[ERROR.NUMBER] TO WR[0]          ; чтение номера ошибки
U 0705,  2100,15 ;4583            DEC WR[0]                              ; и уменьшение
U 0706,  BE82,15 ;4584            MOV WR[0] TO LS[ERROR.NUMBER]          ; его
U 0707,  5B00,14 ;4585            RETURN                                 ;
                ;4586     ;
                ;4587     ;    Эта подпрограмма устанавливает признак N/A (не применимо) для полей
                ;4588     ;    ожидаемых и полученных данных
                ;4589     ;
                ;4590     ASSERT.NA:
U 0708,  B66E,15 ;4591            MOV LS[BIT23] TO WR[0]                 ;
U 0709,  6D80,15 ;4592            BIS WR[0] TO LS[CONTROL.STATUS]        ; слово управления и состояния = бит 23
U 070A,  5B00,14 ;4593            RETURN                                 ;
                ;4594     ;
                ;4595     ;    Эта подпрограмма сбрасывает признак N/A (не применимо) для полей
                ;4596     ;    ожидаемых и полученных данных
                ;4597     ;
                ;4598     DEASSERT.NA:
U 070B,  B680,15 ;4599            MOV LS[CONTROL.STATUS] TO WR[0]        ;
U 070C,  456E,15 ;4600            BIC LS[BIT23] TO WR[0]                 ;
U 070D,  3E80,15 ;4601            MOV WR[0] TO LS[CONTROL.STATUS]        ;
U 070E,  5B00,14 ;4602            RETURN                                 ;
                ;4603     ;
                ;4604     ;    Эта подпрограмма устанавливает признак поля "ДРУГИЕ ДАННЫЕ"
                ;4605     ;
                ;4606     ASSERT.OTHER:
U 070F,  B670,15 ;4607            MOV LS[BIT24] TO WR[0]                 ;
U 0710,  6D80,15 ;4608            BIS WR[0] TO LS[CONTROL.STATUS]        ;
U 0711,  5B00,14 ;4609            RETURN                                 ;
                ;4610     ;
                ;4611     ;    Эта подпрограмма сбрасывает признак поля "ДРУГИЕ ДАННЫЕ"
                ;4612     ;
                ;4613     DEASSERT.OTHER:
U 0712,  B680,15 ;4614            MOV LS[CONTROL.STATUS] TO WR[0]        ;
U 0713,  4570,15 ;4615            BIC LS[BIT24] TO WR[0]                 ;
U 0714,  3E80,15 ;4616            MOV WR[0] TO LS[CONTROL.STATUS]        ;
U 0715,  5B00,14 ;4617            RETURN                                 ;
                ;4618     ;
                ;4619     718:                                           ; эта ячейка резервирована для использования в тесте F
                ;4620     RETURN.718:
U 0718,  5B00,14 ;4621            RETURN
                ;4622
                ;4623
```

```
;4624      .PAGE "ТЕСТ 1 – тест данных памяти стека (модуль DAP)"
;4625      ;
;4626      ; ОПИСАНИЕ ТЕСТА
;4627      ;
;4628      ;   Этот тест проверяет 16 ячеек стека памяти произвольной выборки на целост-
;4629      ;   ность данных. Одна ячейка стека проверена в тестах JSR и RETURN и использова-
;4630      ;   лась для вызова всех подпрограмм в предыдущих тестах. Другие ячейки памяти
;4631      ;   стека проверяются выполнением выгрузки верхней записи до выполнения JSR.
;4632      ;   Каждый раз, когда стек выгружается, используемый адрес стека для UPC возвра-
;4633      ;   та является другим. Когда тест будет выполнен 16 раз, все ячейки стека будут
;4634      ;   проверены на целостность данных. Функция POP в начале теста не проверяется,
;4635      ;   но если она не выполняется, это значит что та же самая ячейка будет исполь-
;4636      ;   зоваться 16 раз. Инструкция POP проверяется в конце этого теста. Этот тест
;4637      ;   зацикливается 16 раз, выполняя одну функцию POP перед каждым циклом для из-
;4638      ;   менения адреса стека памяти произвольной выборки. Пути данных работают сле-
;4639      ;   дующим образом: после начальной установки выполняется POP; микроинструкции
;4640      ;   JSR меняют адреса таким образом, что инструкцией JSR "0" сдвигается через
;4641      ;   поле единиц в памяти стека.
;4642      ;
;4643      ;   инструкция возврата "RETURN" будет выполняться по адресу приемника для
;4644      ;   выгрузки стека по последнему адресу и он используется как PC возврата.
;4645      ;   Такая процедура повторяется, пока "0" не будет сдвинут через разряды 0-13
;4646      ;   UPC. После этого тест повторяется с новым адресом стека памяти. Последней
;4647      ;   проверяется инструкция POP, чтобы убедиться, что по крайней мере две
;4648      ;   ячейки памяти стека доступны, и что сама инструкция POP выполняется.
;4649      ;   Инструкция POP проверяется выполнением двух JSR, инструкции POP и выпол-
;4650      ;   нением RETURN. Тогда выполняется проверка, чтобы убедиться, что возврат
;4651      ;   выполняется правильно к ячейке+1 первой инструкции JSR.
;4652      ;
;4653      ; ПРЕДПОЛОЖЕНИЯ:
;4654      ;
;4655      ;   Предполагается, что предыдущий тест JSR и RETURN выполнен успешно.
;4656      ;   Так как тест проверяет возврат к специфическому адресу, а не данные,
;4657      ;   ожидаемые и полученные данные в распечатке ошибки должны игнорироваться.
;4658      ;
;4659      ;   ПРИМЕЧАНИЕ: Неисправная работа стека в начальной фазе каждого цикла может
;4660      ;   отправить микропрограмму в неиспользуемую зону. Отыскание неисправности
;4661      ;   такого типа на этом этапе возможно только логическим методом.
;4662      ;
;4663      ; ШАГИ ТЕСТА:
;4664      ;
;4665      ;   1. Установка маски, номера и номера модуля в местной памяти
;4666      ;      (для распечатки ошибок) и очистка предыдущего номера в местной
;4667      ;      памяти.
;4668      ;   2. Выполнение инструкции POP для уменьшения адреса указателя стека.
;4669      ;   3. Выполнение JSR при микроадресе, который загружает в стек единицы, за
;4670      ;      исключением одного разряда.
;4671      ;   4. Выполнение RETURN по адресу приемника. Неисправность на этом этапе
;4672      ;      безусловно вызовет неправильное продолжение микропрограммы.
;4673      ;   5. В адресе возврата переход на ячейку, которая сдвигает нуль через выходы
;4674      ;      стека и выполняет JSR.
;4675      ;   6. Выполняется RETURN по адресу приемника. Сообщение об ошибке, если
;4676      ;      выполнен возврат на старый адрес.
;4677      ;   7. Повторение шагов 5 и 6, пока нуль не будет сдвинут через разряды адреса
;4678      ;      0 – 13.
```

```
;4679      ;    8. Повторение шагов от 2 до 7, пока не будут проверены все 16 ячеек памяти
;4680      ;       стека.
;4681      ;    9. Выполнение двух JSR и затем выполнение инструкции POP.
;4682      ;   10. Выполнение RETURN и проверка, что выполнен возврат к ячейке+1 первой JSR.
;4683      ;
;4684      ; ОШИБКИ:
;4685      ;
;4686      ;    ошибка 1 - неисправность памяти стека при возврате:
;4687      ;                  разряд 13.
;4688      ;    ошибка 2 - неисправность памяти стека при возврате:
;4689      ;                  разряд 12.
;4690      ;    ошибка 3 - неисправность памяти стека при возврате:
;4691      ;                  разряд 11.
;4692      ;    ошибка 4 - неисправность памяти стека при возврате:
;4693      ;                  разряд 10.
;4694      ;    ошибка 5 - неисправность памяти стека при возврате:
;4695      ;                  разряд 9.
;4696      ;    ошибка 6 - неисправность памяти стека при возврате:
;4697      ;                  разряд 8.
;4698      ;    ошибка 7 - неисправность памяти стека при возврате:
;4699      ;                  разряд 7.
;4700      ;    ошибка 8 - неисправность памяти стека при возврате:
;4701      ;                  разряд 6.
;4702      ;    ошибка 9 - неисправность памяти стека при возврате:
;4703      ;                  разряд 5.
;4704      ;    ошибка A - неисправность памяти стека при возврате:
;4705      ;                  разряд 4.
;4706      ;    ошибка B - неисправность памяти стека при возврате:
;4707      ;                  разряд 3.
;4708      ;    ошибка C - неисправность памяти стека при возврате:
;4709      ;                  разряд 2.
;4710      ;    ошибка D - неисправность памяти стека при возврате:
;4711      ;                  разряд 1.
;4712      ;    ошибка E - неисправность памяти стека при возврате:
;4713      ;                  разряд 0.
;4714      ;    ошибка F - инструкция POP неправильно уменьшила адрес указателя стека.
;4715      ;
;4716      ;    НАЛАДКА:
;4717      ;
;4718      ;         В случае неисправности конкретный адрес стека, на котором появи-
;4719      ;    лась неисправность, микропрограммой не может быть определен. Останов по
;4720      ;    ошибке и просмотр текущего адреса является единственным способом для его
;4721      ;    определения. Однако, фактический адрес не имеет значения, так как прове-
;4722      ;    ряются только разряды данных на выходе памяти стека.
;4723      ;
;4724      ;    ОШИБКИ С 1 ПО E - Скорее всего, подозреваемой является сама память стека,
;4725      ;    хотя функции JSR и RETURN уже были проверены. Неисправный разряд в памяти
;4726      ;    стека может быть определен по номеру ошибки.
;4727      ;
;4728      ;    ОШИБКА F - Эта ошибка связана с ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ или ПМЛ УКАЗАТЕЛЬ
;4729      ;    МИАСС ТЕКА. Выход ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ ENABLE SP L должен быть низким
;4730      ;    во время микроинструкции POP. Если этот выход некорректный, необходимо
;4731      ;    проверить наличие 12(H) на входах ПМЛ CSR 4-0. Если эти входы корректные,
;4732      ;    ПМЛ является дефектной. Если ENABLE SP L корректный, надо проверить этот
;4733      ;    сигнал на входе ПМЛ УКАЗАТЕЛЬ МИАСС ТЕКА. Также надо проверить CSR 03 H.
```

```
                    ;4734     ;   Если эти входы корректные, может быть дефектной ПМЛ УКАЗАТЕЛЬ МИАСС ТЕКА.
                    ;4735     ;   Также необходимо проверить адресные входы памяти стека, чтобы убедиться в
                    ;4736     ;   отсутствии обрыва (неодходимо проверить контакты с 3 по 6 каждой микросхемы
                    ;4737     ;   памяти стека).
                    ;4738     ;
                    ;4739     T.1:
U 0719, B65E,15     ;4740             MOV LS[BEGIN.TEST] TO WR[0]                ; установка разряда 15 в WR0 для слова управления и
                    ;4741                                                        ; состояния
U 071A, 3E80,15     ;4742             MOV WR[0] TO LS[CONTROL.STATUS]            ; установка разряда 15 в слове управления и
                    ;4743                                                        ; состояния. Разряд 15 указывает для консольного
                    ;4744                                                        ; процессора начало теста
U 071B, 10E0,15     ;4745             MISC [SET.CP.ATTN]                         ; выдача сигнала CPU ATTN для консольного процессора
                    ;4746     WAIT.T1.0:
U 071C, 8871,C4     ;4747             JMP [WAIT.T1.0]                            ; зацикливание ожидания ответа консольного процессора
U 071D, E58A,15     ;4748             CLR LS[ERROR.MASK]                         ; очистка маски ошибок.
U 071E, 65A0,15     ;4749             CLR LS[PREVIOUS.ERROR]                     ; очистка номера предыдущей ошибки
U 071F, 2F80,15     ;4750             CLR WR[0]                                  ;
U 0720, 2040,15     ;4751             INC WR[0]                                  ; установка 1 в WR0
U 0721, BE82,15     ;4752             MOV WR[0] TO LS[ERROR.NUMBER]              ; установка номера для первой ошибки
U 0722, A3C0,15     ;4753             ROL WR[0]                                  ; установка 2 в WR0
U 0723, 3E8C,15     ;4754             MOV WR[0] TO LS[MODULE.NUM]                ; установка кода модуля для модуля DAP
U 0724, B66E,15     ;4755             MOV LS[BIT23] TO WR[0]                     ;
U 0725, 3E80,15     ;4756             MOV WR[0] TO LS[CONTROL.STATUS]            ; бит 23 = ожидаемые и полученные данные не применяются
U 0726, 2F85,15     ;4757             CLR WR[2]                                  ; установка счетчика большого цикла на 0
                    ;4758     BIGLOOP.T1:
U 0727, 2045,15     ;4759             INC WR[2]                                  ; увеличение счетчика большого цикла
U 0728, 5B00,12     ;4760             SKIP.IF[POP.USTACK]                        ; ПРИМЕЧАНИЕ: Это не микроинструкция SKIP. Это
                    ;4761                                                        ; микроинструкция POP
                    ;4762     LOOP.T1.1:
U 0729, B640,15     ;4763             MOV LS[#1] TO WR[0]                        ;
U 072A, BE82,15     ;4764             MOV WR[0] TO LS[ERROR.NUMBER]              ; номер ошибки = 1
U 072B, B9FF,E4     ;4765             JMP [L.1FFE]                               ;
                    ;4766     1FFE:
                    ;4767     L.1FFE:
                    ;4768      ; следующий переход вызовет ошибку истинности:
;????VALIDITY FAILURE
U 1FFE, 8950,AC     ;4769             JSR [SUB.T1]                               ; загрузка адреса возврата 1FFF в стек
U 1FFF, 8870,0C     ;4770             JSR [INC.EN]                               ; увеличение номера ошибки до 2
U 2000, 8AFF,E4     ;4771             JMP [T1.2]                                 ;
                    ;4772     2FFE:
                    ;4773     T1.2:
                    ;4774      ; следующий переход вызовет ошибку истинности:
;????VALIDITY FAILURE
U 2FFE, 8950,AC     ;4775             JSR [SUB.T1]                               ; загрузка адреса возврата 2FFF в стек
U 2FFF, 8870,0C     ;4776             JSR [INC.EN]                               ; увеличение номера ошибки до 3
U 3000, 8B7F,E4     ;4777             JMP [T1.3]                                 ;
                    ;4778     37FE:
                    ;4779     T1.3:
                    ;4780      ; следующий переход вызовет ошибку истинности:
;????VALIDITY FAILURE
U 37FE, 8950,AC     ;4781             JSR [SUB.T1]                               ; загрузка адреса возврата 37FF в стек
U 37FF, 8870,0C     ;4782             JSR [INC.EN]                               ; увеличение номера ошибки до 4
U 3800, 88BF,E4     ;4783             JMP [T1.4]                                 ;
                    ;4784     38FE:
                    ;4785     T1.4:
```

```
U 3BFE, 8950,AC ;4786          JSR [SUB.T1]                    ; загрузка адреса возврата 3BFF в стек
U 3BFF, 8870,0C ;4787          JSR [INC.EN]                    ; увеличение номера ошибки до 5
U 3C00, 8BDF,E4 ;4788          JMP [T1.5]                      ;
                ;4789    3DFE:
                ;4790    T1.5:
U 3DFE, 8950,AC ;4791          JSR [SUB.T1]                    ; загрузка адреса возврата 3DFF в стек
U 3DFF, 8870,0C ;4792          JSR [INC.EN]                    ; увеличение номера ошибки до 6
U 3E00, 8BEF,E4 ;4793          JMP [T1.6]                      ;
                ;4794    3EFE:
                ;4795    T1.6:
U 3EFE, 8950,AC ;4796          JSR [SUB.T1]                    ; загрузка адреса возврата 3EFF в стек
U 3EFF, 8870,0C ;4797          JSR [INC.EN]                    ; увеличение номера ошибки до 7
U 3F00, 8BF7,E4 ;4798          JMP [T1.7]                      ;
                ;4799    3F7E:
                ;4800    T1.7:
U 3F7E, 8950,AC ;4801          JSR [SUB.T1]                    ; загрузка адреса возврата 3F7F в стек
U 3F7F, 8870,0C ;4802          JSR [INC.EN]                    ; увеличение номера ошибки до 8
U 3F80, 8BFB,E4 ;4803          JMP [T1.8]                      ;
                ;4804    3FBE:
                ;4805    T1.8:
U 3FBE, 8950,AC ;4806          JSR [SUB.T1]                    ; загрузка адреса возврата 3FBF в стек
U 3FBF, 8870,0C ;4807          JSR [INC.EN]                    ; увеличение номера ошибки до 9
U 3FC0, 8BFD,E4 ;4808          JMP [T1.9]                      ;
                ;4809    3FDE:
                ;4810    T1.9:
U 3FDE, 8950,AC ;4811          JSR [SUB.T1]                    ; загрузка адреса возврата 3FDF в стек
U 3FDF, 8870,0C ;4812          JSR [INC.EN]                    ; увеличение номера ошибки до A
U 3FE0, 8BFE,E4 ;4813          JMP [T1.A]                      ;
                ;4814    3FEE:
                ;4815    T1.A:
U 3FEE, 8950,AC ;4816          JSR [SUB.T1]                    ; загрузка адреса возврата 3FEF в стек
U 3FEF, 8870,0C ;4817          JSR [INC.EN]                    ; увеличение номера ошибки до B
U 3FF0, 8BFF,64 ;4818          JMP [T1.B]                      ;
                ;4819    3FF6:
                ;4820    T1.B:
U 3FF6, 8950,AC ;4821          JSR [SUB.T1]                    ; загрузка адреса возврата 3FF7 в стек
U 3FF7, 8870,0C ;4822          JSR [INC.EN]                    ; увеличение номера ошибки до C
U 3FF8, 8BFF,A4 ;4823          JMP [T1.C]                      ;
                ;4824    3FFA:
                ;4825    T1.C:
U 3FFA, 8950,AC ;4826          JSR [SUB.T1]                    ; загрузка адреса возврата 3FFB в стек
U 3FFB, 8870,0C ;4827          JSR [INC.EN]                    ; увеличение номера ошибки до D
U 3FFC, 8B5F,C4 ;4828          JMP [T1.D]                      ;
                ;4829    35FC:
                ;4830    T1.D:
U 35FC, 8950,AC ;4831          JSR [SUB.T1]                    ; загрузка адреса возврата 3FFD в стек
U 35FD, 8870,0C ;4832          JSR [INC.EN]                    ; увеличение номера ошибки до E
U 35FE, 0B6F,D4 ;4833          JMP [T1.E]                      ;
                ;4834    36FD:
                ;4835    T1.E:
U 36FD, 8950,AC ;4836          JSR [SUB.T1]                    ; загрузка адреса возврата 36FE в стек
U 36FE, 0950,54 ;4837          JMP [T1.EA]                     ;
                ;4838    3FFF:
U 3FFF, 0950,04 ;4839          JMP [T1.ERR.1]
                ;4840    1500:
```

```
                ;4841    T1.ERR.1:
U 1500, B69E,15 ;4842            MOV LS[ONES] TO WR[0]                  ; ошибка, если переход сюда, поэтому умышленно
                ;4843                                                   ; устанавливаются ошибочные данные для индикации ошибки
U 1501, 2F82,95 ;4844            CLR WR[1]                              ;
U 1502, 0869,3C ;4845            JSR [CHECK.RESULT]                     ;
U 1503, 8872,94 ;4846            JMP [LOOP.T1.1]                        ; зацикливание при ошибке, если разрешено
U 1504, 0950,54 ;4847            JMP [T1.EA]                            ;
                ;4848    T1.EA:
                ;4849            XOR LS[#10] WITH WR[2] TO Q,           ;
U 1505, 4D49,35 ;4850            DT(LONG)&SET.ALU.CC                    ; счетчик большого цикла = 16?
U 1506, 0872,71 ;4851            JMP.IF[NEQ] TO [BIGLOOP.T1]            ; если нет - возврат к циклу
U 1507, 8870,0C ;4852            JSR [INC.EN]                           ; увеличение номера ошибки до F
                ;4853    LOOP.T1.F:
U 1508, 0950,BC ;4854            JSR [SUB.T1.A]                         ; переход к подпрограмме A
U 1509, 8952,04 ;4855            JMP [END.T1]                           ; переход к концу теста
                ;4856    SUB.T1:
U 150A, 5B00,14 ;4857            RETURN                                 ; фиктивная подпрограмма
                ;4858    SUB.T1.A:
U 150B, 8951,1C ;4859            JSR [SUB.T1.B]                         ; вызов подпрограммы B
U 150C, 369E,95 ;4860            MOV LS[ONES] TO WR[1]                  ; ошибка, если переход сюда, поэтому
U 150D, 2F80,15 ;4861            CLR WR[0]                              ; умышленно устанавливаются ошибочные данные для
                ;4862                                                   ; индикации ошибки
U 150E, 0869,3C ;4863            JSR [CHECK.RESULT]                     ;
U 150F, 8950,B4 ;4864            JMP [LOOP.T1.F]                        ; зацикливание при ошибке, если разрешено
U 1510, 8952,04 ;4865            JMP [END.T1]                           ;
                ;4866    SUB.T1.B:
U 1511, 5B00,12 ;4867            SKIP.IF[POP.USTACK]                    ; ПРИМЕЧАНИЕ: Это не микроинструкция SKIP. Это
                ;4868                                                   ; микроинструкция POP
U 1512, 5B00,14 ;4869            RETURN                                 ;
                ;4870    818:                                           ; эта ячейка резервирована для теста F
                ;4871    RETURN.818:
U 0818, 5B00,14 ;4872            RETURN                                 ;
                ;4873    C18:                                           ; эта ячейка резервирована для теста F
                ;4874    RETURN.C18:
U 0819, 5B00,14 ;4875            RETURN
                ;4876    1018:                                          ; эта ячейка резервирована для теста F
                ;4877    RETURN.1018:
U 101B, 5B00,14 ;4878            RETURN                                 ;
                ;4879    1418:                                          ; эта ячейка резервирована для теста F
                ;4880    RETURN.1418:
U 1418, 5B00,14 ;4881            RETURN                                 ;
                ;4882    1A18:                                          ; эта ячейка резервирована для теста F
                ;4883    RETURN.1A18:
U 1A18, 5B00,14 ;4884            RETURN                                 ;
                ;4885    1C18:                                          ; эта ячейка резервирована для теста F
                ;4886    RETURN.1C18:
U 1C18, 5B00,14 ;4887            RETURN                                 ;
                ;4888    2018:                                          ; эта ячейка резервирована для теста F
                ;4889    RETURN.2018:
U 2018, 5B00,14 ;4890            RETURN                                 ;
                ;4891    2418:                                          ; эта ячейка резервирована для теста F
                ;4892    RETURN.2418:
U 2418, 5B00,14 ;4893            RETURN                                 ;
                ;4894    2818:                                          ; эта ячейка резервирована для теста F
                ;4895    RETURN.2818:
```

```
U 2818,  5B00,14 ;4896            RETURN                                ;
                 ;4897    2C18:                                        ; эта ячейка резервирована для теста F
                 ;4898    RETURN.2C18:
U 2C18,  5B00,14 ;4899            RETURN                                ;
                 ;4900    3018:                                        ; эта ячейка резервирована для теста F
                 ;4901    RETURN.3018:
U' 3018, 5B00,14 ;4902            RETURN                                ;
                 ;4903    3418:                                        ; эта ячейка резервирована для теста F
                 ;4904    RETURN.3418:
U 3418,  5B00,14 ;4905            RETURN                                ;
                 ;4906    3818:                                        ; эта ячейка резервирована для теста F
                 ;4907    RETURN.3818:
U 3818,  5B00,14 ;4908            RETURN                                ;
                 ;4909    3C18:                                        ; эта ячейка резервирована для теста F
                 ;4910    RETURN.3C18:
U 3C18,  5B00,14 ;4911            RETURN
                 ;4912    1520:
                 ;4913    END.Ti:
```

```
;4914      .PAGE  "ТЕСТ 2 - тест гнездования подпрограмм (модуль DAP)"
;4915      ;
;4916      ; ОПИСАНИЕ ТЕСТА:
;4917      ;
;4918      ;   Этот тест проверяет ПМЛ УКАЗАТЕЛЬ МИАСС ТЕКА и память стека, чтобы убедиться,
;4919      ;   что 16 равноценных ячеек могут быть доступны для гнездования подпрограмм. За
;4920      ;   серией из 16 выполненных JSR следует 16 инструкций RETURN. Каждый раз при
;4921      ;   выполнении RETURN увеличивается WR0 и в конце проверяется корректность значе-
;4922      ;   ния. Это позволяет избежать бесконечного цикла, если адреса памяти наклады-
;4923      ;   ваются. После проверки адресации стека тест проверяет, что инструкция
;4924      ;   RETURN+1 также работает.
;4925      ;
;4926      ; ПРЕДПОЛОЖЕНИЯ:
;4927      ;
;4928      ;   Предполагается, что предыдущий тест памяти стека выполнен успешно.
;4929      ;
;4930      ; ШАГИ ТЕСТА:
;4931      ;
;4932      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;4933      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной па-
;4934      ;      мяти.
;4935      ;   2. Сброс WR0 и затем выполнение 16 инструкций JSR.
;4936      ;   3. Увеличечние WR0 последней инструкции JSR и выполнение RETURN.
;4937      ;   4. Снова увеличение WR0 и выполнение RETURN.
;4938      ;   5. Повторение шага 4 до тех пор, пока WR0=16 (дес.) и проверка правильности
;4939      ;      WR0.
;4940      ;   6. Выполнение двух инструкций JSR и двух инструкций RETURN+1 для проверки,
;4941      ;      что RETURN+1 также уменьшает адрес стека.
;4942      ;
;4943      ; ОШИБКИ:
;4944      ;
;4945      ;   ошибка 1 - отказ памяти стека или ПМЛ УКАЗАТЕЛЬ МИАСС ТЕКА.
;4946      ;   ошибка 2 - инструкция RETURN+1 неправильно уменьшила указатель стека.
;4947      ;
;4948      ; НАЛАДКА:
;4949      ;
;4950      ;   ОШИБКА 1 - Ожидаемые данные в распечатке ошибки указывают, в которую ячейку
;4951      ;   должен быть выполнен возврат. Вероятнее всего, ошибка появляется из-за того,
;4952      ;   что ячейки памяти стека были повторно записаны во время 16 инструкций JSR.
;4953      ;   Если сигнал на линии адреса совсем не меняется, тогда будет дважды записано
;4954      ;   в одну и ту же ячейку. Одной из возможностей является то, что ПМЛ УПР.МИКРО-
;4955      ;   СЕКВЕНСЕРОМ не вырабатывает ENABLE SP L в инструкции возврата. Необходимо
;4956      ;   проверить сигнал ENABLE SP L на ПМЛ УКАЗАТЕЛЬ МИАСС ТЕКА. Он должен быть
;4957      ;   низким во время инструкции RETURN. Если нет, ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ не-
;4958      ;   исправна. Другой возможностью может быть то, что неисправна ПМЛ УКАЗАТЕЛЬ
;4959      ;   МИАСС ТЕКА. Самый простой способ проверки этой ПМЛ заключается в выполнении
;4960      ;   инструкции JSR шагами с проверкой адресных линий. Такая же процедура может
;4961      ;   быть выполнена во время инструкций RETURN, если JSR работает правильно.
;4962      ;   Третьей возможностью может быть то, что адресные линии не доведены до памяти
;4963      ;   стека или сама память стека неисправна внутри.
;4964      ;
;4965      ;   ОШИБКА 2 - Если это первая ошибка, наиболее подозрительной является ПМЛ УПР.
;4966      ;   МИАСС ЕКВЕНСЕРОМ. Эта ошибка показывает, что инструкция RETURN правильно
;4967      ;   выполняет увеличение указателя после выгрузки, а RETURN+1 нет.
;4968      T.2:
```

```
U 1520, B65E,15 ;4969            MOV LSI[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                ;4970                                                  ; состояния
U 1521, 3EB0,15 ;4971            MOV WR[0] TO LSI[CONTROL.STATUS]      ; установка бита 15 в слове управления и состояния. Бит
                ;4972                                                  ; 15 указывает начало теста для консольного процессора
U 1522, 10E0,15 ;4973            MISC [SET.CP.ATTN]                    ; выдача сигнала CPU ATTN для консольного процессора
                ;4974    WAIT.T2.0:
U 1523, 8952,34 ;4975            JMP [WAIT.T2.0]                       ; зацикливание ожидания ответа консольного процессора
U 1524, E5BA,15 ;4976            CLR LSI[ERROR.MASK]                   ; очистка маски ошибки
U 1525, 65A0,15 ;4977            CLR LSI[PREVIOUS.ERROR]
U 1526, 2FB0,15 ;4978            CLR WR[0]
U 1527, 2040,15 ;4979            INC WR[0]                             ; установка 1 в WR0
U 1528, BEB2,15 ;4980            MOV WR[0] TO LSI[ERROR.NUMBER]        ; установка номера ошибки для первой ошибки
U 1529, A3C0,15 ;4981            ROL WR[0]                             ; установка 2 в WR0
U 152A, 3E8C,15 ;4982            MOV WR[0] TO LSI[MODULE.NUM]          ; установка кода модуля для модуля DAP
U 152B, B66E,15 ;4983            MOV LSI[BIT23] TO WR[0]
U 152C, 3EB0,15 ;4984            MOV WR[0] TO LSI[CONTROL.STATUS]       бит 23=ожидаемые и получаемые данные не применяются
                ;4985    LOOP.T2.1:
U 152D, 2FB0,15 ;4986            CLR WR[0]                              установка 0 в WR0
U 152E, B954,1C ;4987            JSR [T2.SUB.1]                         выдача первого из 16 гнездованных вызовов
U 152F, B648,95 ;4988            MOV LSI[#10] TO WR[1]                  ожидаемые данные = 17
U 1530, 2042,95 ;4989            INC WR[1]
U 1531, 0869,3C ;4990            JSR [CHECK.RESULT]                     WR0 увеличен 16 раз?
U 1532, 0952,D4 ;4991            JMP [LOOP.T2.1]                       ; зацикливание при ошибке, если разрешено
U 1533, 8870,0C ;4992            JSR [INC.EN]                          ; увеличение номера ошибки до 2
                ;4993    LOOP.T2.2:
U 1534, 2FB0,15 ;4994            CLR WR[0]
U 1535, 0953,BC ;4995            JSR [T2.SUB.17]                        выдача 1-го из двух гнездованных вызовов
U 1536, 2FB0,15 ;4996            CLR WR[0]                              ошибка, если переход сюда
U 1537, B642,95 ;4997            MOV LSI[#2] TO WR[1]                   ожидаемые данные=2
U 1538, 0869,3C ;4998            JSR [CHECK.RESULT]                     проверка, что обе инструкции RETURN+1 работают
U 1539, 8953,44 ;4999            JMP [LOOP.T2.2]                        зацикливание при ошибке, если разрешено
U 153A, 095B,F4 ;5000            JMP [END.T2]                           все выполнено - пропуск всех подпрограмм
                ;5001    T2.SUB.17:
U 153B, 8953,FC ;5002            JSR [T2.SUB.18]                       ; выдача второго из двух гнездованных вызовов
U 153C, 2FB0,15 ;5003            CLR WR[0]                             ; ошибка, если переход сюда
U 153D, 2040,15 ;5004            INC WR[0]                             ; увеличение счетчика
U 153E, DB00,16 ;5005            RETURN+1
                ;5006    T2.SUB.18:
U 153F, 2040,15 ;5007            INC WR[0]                              увеличение счетчика
U 1540, DB00,16 ;5008            RETURN+1
                ;5009    T2.SUB.1:
U 1541, 8954,4C ;5010            JSR [T2.SUB.2]                        ; вызов следующей подпрограммы в гнезде
U 1542, 2040,15 ;5011            INC WR[0]                             ; увеличение счетчика для подтверждения, что был переход
                ;5012                                                  ; сюда
U 1543, 5B00,14 ;5013            RETURN
                ;5014    T2.SUB.2:
U 1544, 8954,7C ;5015            JSR [T2.SUB.3]                        ; вызов следующей подпрограммы в гнезде
U 1545, 2040,15 ;5016            INC WR[0]                             ; увеличение счетчика для подтверждения, что был переход
                ;5017                                                  ; сюда
U 1546, 5B00,14 ;5018            RETURN
                ;5019    T2.SUB.3:
U 1547, 0954,AC ;5020            JSR [T2.SUB.4]                        ; вызов следующей подпрограммы в гнезде
U 1548, 2040,15 ;5021            INC WR[0]                             ; увеличение счетчика для подтверждения, что был переход
                ;5022                                                  ; сюда
U 1549, 5B00,14 ;5023            RETURN
```

```
                ;5024    T2.SUB.4:
U 154A, 8954,DC ;5025            JSR [T2.SUB.5]                 ; вызов следующей подпрограммы в гнезде
U 154B, 2040,15 ;5026            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5027                                           ; сюда
U 154C, 5B00,14 ;5028            RETURN                         ;
                ;5029    T2.SUB.5:
U 154D, 8955,0C ;5030            JSR [T2.SUB.6]                 ; вызов следующей подпрограммы в гнезде
U 154E, 2040,15 ;5031            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5032                                           ; сюда
U 154F, 5B00,14 ;5033            RETURN                         ;
                ;5034    T2.SUB.6:
U 1550, 8955,3C ;5035            JSR [T2.SUB.7]                 ; вызов следующей подпрограммы в гнезде
U 1551, 2040,15 ;5036            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5037                                           ; сюда
U 1552, 5B00,14 ;5038            RETURN                         ;
                ;5039    T2.SUB.7:
U 1553, 8955,6C ;5040            JSR [T2.SUB.8]                 ; вызов следующей подпрограммы в гнезде
U 1554, 2040,15 ;5041            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5042                                           ; сюда
U 1555, 5B00,14 ;5043            RETURN                         ;
                ;5044    T2.SUB.8:
U 1556, 8955,9C ;5045            JSR [T2.SUB.9]                 ; вызов следующей подпрограммы в гнезде
U 1557, 2040,15 ;5046            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5047                                           ; сюда
U 1558, 5B00,14 ;5048            RETURN                         ;
                ;5049    T2.SUB.9:
U 1559, 8955,CC ;5050            JSR [T2.SUB.10]                ; вызов следующей подпрограммы в гнезде
U 155A, 2040,15 ;5051            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5052                                           ; сюда
U 155B, 5B00,14 ;5053            RETURN                         ;
                ;5054    T2.SUB.10:
U 155C, 8955,FC ;5055            JSR [T2.SUB.11]                ; вызов следующей подпрограммы в гнезде
U 155D, 2040,15 ;5056            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5057                                           ; сюда
U 155E, 5B00,14 ;5058            RETURN                         ;
                ;5059    T2.SUB.11:
U 155F, 0956,2C ;5060            JSR [T2.SUB.12]                ; вызов следующей подпрограммы в гнезде
U 1560, 2040,15 ;5061            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5062                                           ; сюда
U 1561, 5B00,14 ;5063            RETURN                         ;
                ;5064    T2.SUB.12:
U 1562, 8956,5C ;5065            JSR [T2.SUB.13]                ; вызов следующей подпрограммы в гнезде
U 1563, 2040,15 ;5066            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5067                                           ; сюда
U 1564, 5B00,14 ;5068            RETURN                         ;
                ;5069    T2.SUB.13:
U 1565, 0956,8C ;5070            JSR [T2.SUB.14]                ; вызов следующей подпрограммы в гнезде
U 1566, 2040,15 ;5071            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5072                                           ; сюда
U 1567, 5B00,14 ;5073            RETURN                         ;
                ;5074    T2.SUB.14:
U 1568, 0AA8,0C ;5075            JSR [T2.SUB.15]                ; вызов следующей подпрограммы в гнезде
U 1569, 2040,15 ;5076            INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5077                                           ; сюда
U 156A, 5B00,14 ;5078            RETURN                         ;
```

```
                ;5079    2A80:
                ;5080    T2.SUB.15:
U 2A80,  8AA8,4C ;5081           JSR [T2.SUB.16]                ; вызов следующей подпрограммы в гнезде
U 2A81,  2040,15 ;5082           INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5083                                           ; сюда
U 2A82,  2040,15 ;5084           INC WR[0]                      ;
U 2A83,  5B00,14 ;5085           RETURN                         ;
                ;5086    T2.SUB.16:
U 2A84,  2040,15 ;5087           INC WR[0]                      ; увеличение счетчика для подтверждения, что был переход
                ;5088                                           ; сюда
U 2A85,  5B00,14 ;5089           RETURN                         ;
                ;5090    156F:
                ;5091    END.T2;
```

```
;5092      .PAGE "ТЕСТ 3 - тесты управления циклами (модуль DAP)"
;5093      ;
;5094      ;   ОПИСАНИЕ ТЕСТА
;5095      ;
;5096      ;   Этот тест проверяет функции управления пропуском JMP.Z.CLR и JMP.C.SET.
;5097      ;   Эти инструкции выполняют переход по верхней записи стека, если условие
;5098      ;   подтверждается, и не выгружают стека. Таким образом, цикл может выполняться,
;5099      ;   пока условие не пропадет. Когда это случается, стек выгружается, изменяя
;5100      ;   точку входа, и микросеквенсер переходит на следующий по порядку адрес. Тест
;5101      ;   загружает коды условий АЛУ нулями, выполняет 3 инструкции JSR для получения
;5102      ;   3 адресов в стеке и выполняет JMP.Z.CLR. Это вызовет возврат микрокода назад
;5103      ;   на ячейку+1 последней инструкции JSR, где выполняется проверка. Бит Z АЛУ
;5104      ;   устанавливается через шину Y и снова выполняется JMP.Z.CLR. Сейчас бит Z АЛУ
;5105      ;   установлен, JMP.Z.CLR выгружает стек и выполняется следующая инструкция. Тест
;5106      ;   обнаруживает, если не происходит правильный цикл, или цикл выгружает стек, или
;5107      ;   стек не выгружается после пропадания условия. Инструкция JMP.C.SET проверяет-
;5108      ;   ся таким же образом.
;5109      ;
;5110      ;   ПРЕДПОЛОЖЕНИЯ:
;5111      ;
;5112      ;   Предполагается, что все предыдущие тесты JSR, RETURN и памяти стека выполнены
;5113      ;   успешно.
;5114      ;
;5115      ;   ШАГИ ТЕСТА:
;5116      ;
;5117      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;5118      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;5119      ;      памяти.
;5120      ;   2. Сброс всех кодов условий АЛУ и выполнение 2 инструкций JSR.
;5121      ;   3. На месте назначения второй инструкции JSR выполняется JSR К инструкции
;5122      ;      JMP.Z.CLR.
;5123      ;   4. JMP.Z.CLR выполняет возврат к ячейке+1 последней инструкции JSR, где
;5124      ;      проверяется Z. Если бит Z сброшен, микрокод устанавливает Z и опять вы-
;5125      ;      полняет JMP.Z.CLR. В противном случае сообщается об ошибке (инструкция
;5126      ;      JMP.Z.CLR выполнила зацикливание при установленном Z).
;5127      ;   5. Сейчас, когда Z установлен, JMP.Z.CLR выгружает стек и происходит переход
;5128      ;      к следующей инструкции. Если JMP.Z.CLR опять вызывает зацикливание, ошибка
;5129      ;      обнаруживается в шаге 4.
;5130      ;   6. Если JMP.Z.CLR происходит, проверяется бит Z. Если Z не установлен, сооб-
;5131      ;      щается об ошибке (инструкция JMP.Z.CLR не выполнила зацикливания).
;5132      ;   7. Меняется номер ошибки и выполняется RETURN. Если выгрузка выполняется,
;5133      ;      RETURN выполняет переход на ячейку+1 второй инструкции JSR и тест JMP.Z.CLR
;5134      ;      завершается.
;5135      ;   8. Устанавливаются все коды условий АЛУ и выполняются инструкции JSR.
;5136      ;   9. На месте назначения второй инструкции JSR выполняется JSR к инструкции
;5137      ;      JMP.C.SET.
;5138      ;   10. JMP.C.SET выполняет возврат к ячейке+1 последней инструкции JSR, где про-
;5139      ;       веряется бит C. Если "C" установлен, микрокод сбрасывает "C" и опять вы-
;5140      ;       полняет JMP.C.SET. В противном случае сообщается об ошибке (инструкция
;5141      ;       JMP.C.SET выполнила зацикливание при сброшенном C).
;5142      ;   11. Сейчас, когда "C" сброшен, JMP.C.SET выгружает стек и выполняется следую-
;5143      ;       щая инструкция. Если JMP.C.SET опять выполнила зацикливание, ошибка обна-
;5144      ;       руживается в шаге 10.
;5145      ;   12. Если JMP.C.SET проходит, проверяется бит C. Если "C" не сброшен, сообща-
;5146      ;       ется об ошибке (инструкция JMP.C.SET не выполнила зацикливания).
```

```
;5147      ;   13. Меняется номер ошибки и выполняется RETURN. Если выгрузка выполняется,
;5148      ;       RETURN выполняет переход к ячейке+1 второй инструкции JSR и тест за-
;5149      ;       вершается.
;5150      ;
;5151      ;   ОШИБКИ:
;5152      ;
;5153      ;   ошибка 1 - инструкция JMP.Z.CLR не исполнила зацикливания.
;5154      ;   ошибка 2 - JMP.Z.CLR зацикливает при установленном бите Z.
;5155      ;   ошибка 3 - инструкция JMP.Z.CLR не исполнила выгрузки стека, когда был пе-
;5156      ;              реход без зацикливания.
;5157      ;   ошибка 4 - JMP.Z.CLR выгружает стек при зацикливании.
;5158      ;   ошибка 5 - инструкция JMP.C.SET не исполнила зацикливания.
;5159      ;   ошибка 6 - JMP.C.SET зацикливает при сброшенном "C".
;5160      ;   ошибка 7 - инструкция JMP.C.SET не выполнила выгрузки стека, когда был пе-
;5161      ;              реход без зацикливания.
;5162      ;   ошибка 8 - JMP.C.SET выгружает стек при зацикливании.
;5163      ;
;5164      ;   НАЛАДКА:
;5165      ;
;5166      ;      ОШИБКА 1 - Эта ошибка показывает, что сигнал ENABLE RTS L не становится
;5167      ;   низким, когда выполняется JMP.Z CLR при сброшенном бите Z. Скорее всего, подо-
;5168      ;   зреваемой является сама пил УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были ис-
;5169      ;   пользованы раньше. Входы могут проверяться следующим образом: CSR 4-0 должны
;5170      ;   быть равны 11(H) и ножка 3 должна иметь низкий уровень. Выход на ножке 17 дол-
;5171      ;   жен быть высоким. Эти входы должны проверяться во время микрослова JMP.Z.CLR,
;5172      ;   когда бит Z сброшен (первое выполнение JMP.Z.CLR).
;5173      ;
;5174      ;      ОШИБКА 2 - Эта ошибка показывает, что сигнал ENABLE RTS L не становится вы-
;5175      ;   соким, когда выполняется JMP.Z.CLR при установленном бите Z. Скорее всего, по-
;5176      ;   дозреваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были ис-
;5177      ;   пользованы раньше. Входы могут проверяться при желании следующим образом:
;5178      ;   CSR 4-0 должны быть равны 11(H) и ножка 3 должна иметь высокий уровень. Вы-
;5179      ;   ход на ножке 17 должен быть низким. Эти входы должны проверять во время мик-
;5180      ;   рослова JMP.Z.CLR при установленном бите Z (второе выполнение JMP.Z.CLR).
;5181      ;
;5182      ;      ОШИБКА 3 - Эта ошибка показывает, что сигнал ENABLE SP L не становится низ-
;5183      ;   ким, когда JMP.Z.CLR выполняется при установленном бите Z. Скорее всего, подоз-
;5184      ;   реваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были
;5185      ;   использованы раньше. Входы можно проверить при желании следующим образом:
;5186      ;   CSR 4-0 должны быть равны 11(H), а ножка 3 должна иметь низкий уровень.
;5187      ;   Выход ENABLE SP L на ножке 15 должен иметь низкий уровень. Эти входы должны
;5188      ;   проверяться во время микрослова JMP.Z.CLR при установленном бите Z (второе
;5189      ;   выполнение JMP.Z.CLR).
;5190      ;
;5191      ;      ОШИБКА 4 - Эта ошибка показывает, что сигнал ENABLE SP L не становится вы-
;5192      ;   соким, когда JMP.Z.CLR выполняется при сброшенном бите Z. Скорее всего, подо-
;5193      ;   зреваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были ис-
;5194      ;   пользованы раньше. Входы можно проверять при желании следующим образом: CSR 4-0
;5195      ;   должны быть равны 11(H), А ножка 3 должна иметь низкий уровень. Выход ENABLE SP L
;5196      ;   на ножке 15 должен иметь высокий уровень. Эти входы должны проверяться во время
;5197      ;   микрослова JMP.Z.CLR, когда бит Z сброшен (первое выполнение JMP.Z.CLR).
;5198      ;
;5199      ;      ОШИБКА 5 - Эта ошибка показывает, что сигнал ENABLE RTS L не становится низ-
;5200      ;   ким, когда JMP.C.SET выполняется при установленном бите C. Скорее всего, подозре-
;5201      ;   ваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были использо-
```

```
                ;5202      ;    ваны раньше. Входы можно проверять при желании следующим образом: CSR 4-0 должны
                ;5203      ;    быть равны 13(H), а ножка 3 должна иметь низкий уровень. Выход на ножке 17 дол-
                ;5204      ;    жен иметь высокий уровень. Эти входы должны проверяться во время микрослова
                ;5205      ;    JMP.C.SET, когда бит C установлен (первое выполнение JMP.C.SET).
                ;5206      ;
                ;5207      ;       ОШИБКА 6 - Эта ошибка показывает, что сигнал ENABLE RTS L не становится
                ;5208      ;    высоким, когда выполняется JMP.C.SET при сброшенном бите C. Скорее всего, по-
                ;5209      ;    дозреваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были
                ;5210      ;    использованы раньше. Входы можно проверять при желании следующим образом:
                ;5211      ;    CSR 4-0 должны быть равны 13(H), а ножка 3 должна иметь высокий уровень.
                ;5212      ;    Выход на ножке 17 должен быть низким. Эти входы должны проверяться во время
                ;5213      ;    микрослова JMP.C.SET при сброшенном бите C (второе выполнение JMP.C.SET).
                ;5214      ;
                ;5215      ;       ОШИБКА 7 - Эта ошибка показывает, что сигнал ENABLE SP L не становится
                ;5216      ;    низким, когда выполняется JMP.C.SET при сброшенном бите C. Скорее всего, подо-
                ;5217      ;    зреваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были ис-
                ;5218      ;    пользованы раньше. Входы можно проверить при желании следующим образом: CSR 4-0
                ;5219      ;    должны быть равны 13(H), а ножка 3 должна иметь высокий уровень. Выход ENABLE SP L
                ;5220      ;    на ножке 15 должен быть низким. Эти входы должны проверяться во время микрослова
                ;5221      ;    JMP.C.SET при сброшенном бите "C" (второе выполнение JMP.C.SET).
                ;5222      ;
                ;5223      ;       ОШИБКА 8 - Эта ошибка показывает, что сигнал ENABLE SP L не становится высо-
                ;5224      ;    ким, когда выполняется JMP.C.SET при установленном бите C. Скорее всего, подо-
                ;5225      ;    зреваемой является сама ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Все входы этой ПМЛ были
                ;5226      ;    использованы раньше. Входы можно проверять при желании следующим образом:
                ;5227      ;    CSR 4-0 должны быть равны 13(H), а ножка 3 должна иметь низкий уровень. Выход
                ;5228      ;    ENABLE SP L на ножке 15 должнен быть высоким. Эти входы должны проверяться
                ;5229      ;    во время микрослова JMP.C.SET при бите "C" установленном (первое выполнение
                ;5230      ;    JMP.C.SET).
                ;5231      T.3:
U 156F, 865E,15 ;5232              MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;5233                                                       ; состояния
U 1570, 3E80,15 ;5234              MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
                ;5235                                                       ; 15 указывает начало теста для консольного процессора
U 1571, 10E0,15 ;5236              MISC [SET.CP.ATTN]                       ; выдача сигнала CPU ATTN для конс.процессора
                ;5237      WAIT.T3.0:
U 1572, 0957,24 ;5238              JMP [WAIT.T3.0]                          ; зацикливание ожидания ответа конс.процессора
U 1573, E58A,15 ;5239              CLR LS[ERROR.MASK]                       ; очистка маски ошибки
U 1574, 65A0,15 ;5240              CLR LS[PREVIOUS.ERROR]                   ; очистка предыдущего номера ошибки
U 1575, 2F80,15 ;5241              CLR WR[0]                                ;
U 1576, 2040,15 ;5242              INC WR[0]                                ; установка 1 в WR0
U 1577, BE82,15 ;5243              MOV WR[0] TO LS[ERROR.NUMBER]            ; установка номера ошибки для первой ошибки
U 1578, A3C0,15 ;5244              ROL WR[0]                                ; установка 2 в WR0
U 1579, 3E8C,15 ;5245              MOV WR[0] TO LS[MODULE.NUM]              ; установка кода модуля для модуля DAP
U 157A, C76E,15 ;5246              BIS LS[BIT23] TO WR[0]                   ;
U 157B, 3E80,15 ;5247              MOV WR[0] TO LS[CONTROL.STATUS]           ; загрузка слова управления и состояния
                ;5248      T3.BEGIN:
U 157C, 2F80,15 ;5249              CLR WR[0]                                ;
U 157D, 3FFC,15 ;5250              MOV WR[0] TO LS[ALU.CC]                   ; сброс кодов условий АЛУ
U 157E, 0958,6C ;5251              JSR [SUB.T3.A]                           ; переход на первую из трех подпрограмм
U 157F, 3644,15 ;5252              MOV LS[#4] TO WR[0]                      ;
U 1580, BE82,15 ;5253              MOV WR[0] TO LS[ERROR.NUMBER]            ; номер ошибки = 4
U 1581, 2F80,15 ;5254              CLR WR[0]                                ; установка умышленно ошибочных данных в WR0 и WR1
U 1582, 369E,95 ;5255              MOV LS[ONES] TO WR[1]                    ; для индикации ошибки зацикливания
U 1583, 0869,3C ;5256              JSR [CHECK.RESULT]                       ;
```

```
U 1584, 8957,C4 ;5257            JMP [T3.BEGIN]                          ; зацикливание при ошибке, если разрешено
U 1585, 0A71,04 ;5258            JMP [BEGIN1.T3]                         ; продолжение теста
                ;5259    SUB.T3.A:
U 1586, 8958,8C ;5260            JSR [SUB.T3.B]                          ; переход ко второй вложенной подпрограмме
U 1587, 0A71,04 ;5261            JMP [BEGIN1.T3]                         ; продолжение теста
                ;5262    SUB.T3.B:
U 1588, 0959,2C ;5263            JSR [SUB.T3.C]                          ; переход на последнюю вложенную подпрограмму
U 1589, 0959,C1 ;5264            JMP.IF[NEQ] TO [T3.D]                   ; переход, если Z сброшен
U 158A, 2F80,15 ;5265            CLR WR[0]                               ; установка умышленно ошибочных данных в WR0 и WR1
U 158B, 369E,95 ;5266            MOV LS[ONES] TO WR[1]                   ; для индикации ошибки зацикливания
U 158C, 0869,3C ;5267            JSR [CHECK.RESULT]                      ;
U 158D, 8957,C4 ;5268            JMP [T3.BEGIN]                          ; зацикливание при ошибке, если разрешено
U 158E, 3682,15 ;5269            MOV LS[ERROR.NUMBER] TO WR[0]           ; выборка номера ошибки
                ;5270            XOR LS[#2] WITH WR[0] TO Q,             ;
U 158F, CD42,35 ;5271            DT(LONG)&SET.ALU.CC                     ; номер ошибки = 2?
U 1590, 895A,09 ;5272            JMP.IF[EQL] TO [T3.E]                   ; переход, если да
U 1591, 5B00,14 ;5273            RETURN                                  ;
                ;5274    SUB.T3.C:
U 1592, 5B00,11 ;5275            SCTL/JMP.Z.CLR                          ;
U 1593, 895A,09 ;5276            JMP.IF[EQL] TO [T3.E]                   ; переход, если Z установлен
U 1594, 2F80,15 ;5277            CLR WR[0]                               ;
U 1595, 2040,15 ;5278            INC WR[0]                               ;
U 1596, BE82,15 ;5279            MOV WR[0] TO LS[ERROR.NUMBER]           ; номер ошибки = 1
U 1597, 2F80,15 ;5280            CLR WR[0]                               ; установка умышленно ошибочных данных в WR0 и WR1
U 1598, 369E,95 ;5281            MOV LS[ONES] TO WR[1]                   ; для индикации ошибки зацикливания
U 1599, 0869,3C ;5282            JSR [CHECK.RESULT]                      ;
U 159A, 8957,C4 ;5283            JMP [T3.BEGIN]                          ; зацикливание при ошибке, если разрешено
U 159B, 095A,04 ;5284            JMP [T3.E]                              ;
                ;5285    T3.D:
U 159C, 8870,0C ;5286            JSR [INC.EN]                            ; увеличение номера ошибки
U 159D, 3644,15 ;5287            MOV LS[BIT2] TO WR[0]                   ;
U 159E, 3FFC,15 ;5288            MOV WR[0] TO LS[ALU.CC]                 ; установка бита Z
U 159F, 8959,24 ;5289            JMP [SUB.T3.C]                          ;
                ;5290    T3.E:
U 15A0, 8870,0C ;5291            JSR [INC.EN]                            ; увеличение номера ошибки
U 15A1, 5B00,14 ;5292            RETURN                                  ;
                ;5293    2710:
                ;5294   BEGIN1.T3:
U 2710, 3644,15 ;5295                MOV LS[#4] TO WR[0]                 ; установка номера ошибки = 5
U 2711, 2040,15 ;5296                INC WR[0]
U 2712, BE82,15 ;5297                MOV WR[0] TO LS[ERROR.NUMBER]
                ;5298   T3.BEGIN1:
U 2713, B69E,15 ;5299                MOV LS[ONES] TO WR[0]               ;
U 2714, 3FFC,15 ;5300                MOV WR[0] TO LS[ALU.CC]             ; установка всех кодов условий
U 2715, 0A71,DC ;5301                JSR [SUB.T3.A1]
U 2716, B646,15 ;5302                MOV LS[#8] TO WR[0]
U 2717, BE82,15 ;5303                MOV WR[0] TO LS[ERROR.NUMBER]       ; номер ошибки = 8
U 2718, 2F80,15 ;5304                CLR WR[0]                           ; установка умышленно
U 2719, 369E,95 ;5305                MOV LS[ONES] TO WR[1]               ; ошибочных данных
U 271A, 0869,3C ;5306                JSR [CHECK.RESULT]
U 271B, 0A71,34 ;5307                JMP [T3.BEGIN1]
U 271C, 895A,24 ;5308                JMP [END.T3]
                ;5309
                ;5310    SUB.T3.A1:
U 271D, 8A71,FC ;5311                JSR [SUB.T3.B1]
```

```
U 271E, 895A,24 ;5312                JMP [END.T3]
                ;5313    SUB.T3.B1:
U 271F, 8A72,AC ;5314                JSR [SUB.T3.C1]
U 2720, 8A73,43 ;5315                JMP.IF[QEQU] TO [T3.D1]          ; переход, если "C" установлен
U 2721, 2F80,15 ;5316                CLR WR[0]                        ; установка умышленно
U 2722, 369E,95 ;5317                MOV LS[ONES] TO WR[1]            ; ошибочных данных
U 2723, 0869,3C ;5318                JSR [CHECK.RESULT]
U 2724, 0A71,34 ;5319                JMP [T3.BEGIN1]
U 2725, 3644,15 ;5320                MOV LS[#4] TO WR[0]
U 2726, 4742,15 ;5321                BIS LS[BIT1] TO WR[0]
                ;5322                XOR LS[ERROR.NUMBER] WITH WR[0] TO Q,
U 2727, CD82,35 ;5323                DT(LONG)&SET.ALU.CC              ; номер ошибки = 6?
U 2728, 8A73,89 ;5324                JMP.IF[EQL] TO [T3.E1]           ; переход, если да
U 2729, 5B00,14 ;5325                RETURN
                ;5326
                ;5327    SUB.T3.C1:
U 272A, DB00,13 ;5328                SCTL/JMP.C.SET
U 272B, 0A73,88 ;5329                JMP.IF[LSSU] TO [T3.E1]          ; переход, если бит C сброшен
U 272C, 3644,15 ;5330                MOV LS[#4] TO WR[0]
U 272D, C740,15 ;5331                BIS LS[BIT0] TO WR[0]
U 272E, BE82,15 ;5332                MOV WR[0] TO LS[ERROR.NUMBER]    ; номер ошибки=5
U 272F, 2F80,15 ;5333                CLR WR[0]                        ; установка умышленно
U 2730, 369E,95 ;5334                MOV LS[ONES] TO WR[1]            ; ошибочных данных
U 2731, 0869,3C ;5335                JSR [CHECK.RESULT]
U 2732, 0A71,34 ;5336                JMP [T3.BEGIN1]
U 2733, 0A73,84 ;5337                JMP [T3.E1]
                ;5338
                ;5339    T3.D1:
U 2734, 8870,0C ;5340                JSR [INC.EN]
U 2735, 2F80,15 ;5341                CLR WR[0]
U 2736, 3FFC,15 ;5342                MOV WR[0] TO LS[ALU.CC]          ; сброс бита C
U 2737, 0A72,A4 ;5343                JMP [SUB.T3.C1]
                ;5344
                ;5345    T3.E1:
U 2738, 8870,0C ;5346                JSR [INC.EN]
U 2739, 5B00,14 ;5347                RETURN
                ;5348    15A2:
                ;5349    END.T3:
                ;5350
```

```
;5351     .PAGE "ТЕСТ 4 - обьединение по *ИЛИ* OS и NAD 0-4 (модуль DAP)"
;5352      ;
;5353      ; ОПИСАНИЕ ТЕСТА:
;5354      ;
;5355      ;   Этот тест проверяет схемы обьединения по "ИЛИ" OS 0 H по OS 4 H с пятью младши-
;5356      ;   ми битами адреса перехода во время инструкции JUMP при установленном CSR 18.
;5357      ;   Тест загружает OS данными, содержащими все 0 или все 1, и выполняет JSR с
;5358      ;   младшими пятью битами адреса перехода (CSR4 по CSR8) всеми установленными или
;5359      ;   всеми сброшеными. Инструкция JSR выполняет переход к блоку из 32 микрослов,
;5360      ;   содержащих инструкции INC WR0. После этого блока инструкций INC следует инст-
;5361      ;   рукция RETURN. Ошибка обнаруживается при проверке WR0 после возврата. WR0
;5362      ;   должен быть 20(H), если переход выполняется со всеми нулями OS и всеми нуля-
;5363      ;   ми в CSR4 по CSR8. Другие три набора данных выполняют переход к последнему сло-
;5364      ;   ву в блоке и WR0 должен содержать 1.
;5365      ;
;5366      ; ПРЕДПОЛОЖЕНИЯ:
;5367      ;
;5368      ;   Предполагается, что предыдущие тесты выполнены успешно.
;5369      ;
;5370      ; ШАГИ ТЕСТА:
;5371      ;
;5372      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;5373      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной па-
;5374      ;      мяти.
;5375      ;   2. Сброс WR0 и загрузка регистра OS всеми нулями.
;5376      ;   3. Выполнение инструкции JSR с младшими пятью битами адреса перехода = 0. JSR
;5377      ;      выполняет переход на блок из 32 инструкций INC с последующей инструкцией
;5378      ;      RETURN.
;5379      ;   4. После инструкции RETURN проверяется наличие 20(H) в WR0.
;5380      ;   5. WR0 сбрасывается и выполняется инструкция JSR с единицами в пяти младших
;5381      ;      битах адреса перехода.
;5382      ;   6. После возврата проверяется наличие 1(H) в WR0.
;5383      ;   7. Регистр OS загружается всеми единицами, сбрасывается WR0 и повторяется
;5384      ;      шаг 3. Проверяется наличие 1(H) в WR0.
;5385      ;   8. Повторяются шаги 5 и 6 с регистром OS, содержащим все единицы.
;5386      ;
;5387      ; ОШИБКИ:
;5388      ;
;5389      ;   ошибка 1 - неправильный переход с установленным CSR 18 (обьединение по
;5390      ;              "ИЛИ" регистра OS и адреса перехода) при нулях в регистре OS
;5391      ;              и в CSR4 по CSR8.
;5392      ;   ошибка 2 - неправильный переход с установленным CSR18 (обьединение по
;5393      ;              "ИЛИ" регистра OS и адреса перехода) при нулях в регистре OS
;5394      ;              и единицах в CSR4 по CSR8.
;5395      ;   ошибка 3 - неправильный переход с установленным CSR 18 (обьединение по
;5396      ;              "ИЛИ" регистра OS и адреса перехода) при нулях в регистре OS
;5397      ;              и единицах и нулях в CSR4 по CSR8.
;5398      ;   ошибка 4 - неправильный переход с установленным CSR 18 (обьединение по
;5399      ;              "ИЛИ" регистра OS и адреса перехода при единицах в регистре OS
;5400      ;              и единицах в CSR4 по CSR8.
;5401      ;
;5402      ; НАЛАДКА:
;5403      ;
;5404      ;      ОШИБКА 1 - Эта ошибка показывает неисправность, связанную с ПМЛ МУЛЬТИПЛЕК-
;5405      ;   СОР CSR.OS или входами этой ПМЛ. Необходимо проверить во время инструкции JSR
```

```
                ;5406     ;   входы ПМЛ МУЛЬТИПЛЕКСОР CSR.OS - вход CSR18 Н должен быть высоким, а все ос-
                ;5407     ;   тальные входы низкими. Если входы правильные, может быть неисправной ПМЛ
                ;5408     ;   МУЛЬТИПЛЕКСОР CSR OS (все выходы должны быть нулевыми). Печатаемые получен-
                ;5409     ;   ные данные в сообщении об ошибке показывают, сколько инструкций INC было
                ;5410     ;   выполненно и по ним можно определить, в какое место в блоке из 32 микрослов
                ;5411     ;   был выполнен переход. Например, если полученные данные = 1E, тогда переход был
                ;5412     ;   выполнен к третьему слову в блоке из 32 слов.
                ;5413     ;
                ;5414     ;      ОШИБКА 2-4 - См. ошибку 1. Входы ПМЛ МУЛЬТИПЛЕКСОР CSR.OS даны в общем
                ;5415     ;   описании ошибок. Выходами для всех трех ошибок должны быть все единицы.
                ;5416     ;
                ;5417     T.4:
U 15A2, B65E,15 ;5418           MOV LS[BEGIN.TEST] TO WR[0]             ; установка бита 15 в WR0 для слова управления и
                ;5419                                                   ; состояния
U 15A3, 3EB0,15 ;5420           MOV WR[0] TO LS[CONTROL.STATUS]         ; установка бита 15 в слове управления и состояния. Бит
                ;5421                                                   ; 15 указываает начало теста для консольного процессора
U 15A4, 10E0,15 ;5422           MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для консольного процессора
                ;5423     WAIT.T4.0:
U 15A5, 095A,54 ;5424           JMP [WAIT.T4.0]                         ; зацикливание ожидания ответа консольного процессора
U 15A6, E58A,15 ;5425           CLR LS[ERROR.MASK]                      ; очистка маски ошибки
U 15A7, 65A0,15 ;5426           CLR LS[PREVIOUS.ERROR]                  ; очистка предыдущего номера ошибки
U 15A8, 2FB0,15 ;5427           CLR WR[0]                               ;
U 15A9, 2040,15 ;5428           INC WR[0]                               ; установка 1 в WR0
U 15AA, BE82,15 ;5429           MOV WR[0] TO LS[ERROR.NUMBER]           ; установка номера ошибки для первой ошибки
U 15AB, A3C0,15 ;5430           ROL WR[0]                               ; установка 2 в WR0
U 15AC, 3EBC,15 ;5431           MOV WR[0] TO LS[MODULE.NUM]             ; установка кода модуля для молуля DAP
U 15AD, B664,15 ;5432           MOV LS[BIT18] TO WR[0]                  ;
U 15AE, 3E80,15 ;5433           MOV WR[0] TO LS[CONTROL.STATUS]         ; загрузка слова управления и состояния
                ;5434     LOOP.T4.1:
U 15AF, 2FB0,15 ;5435           CLR WR[0]                               ; WR0 = 0
U 15B0, 3EF8,15 ;5436           MOV WR[0] TO LS[OS]                     ; OS = 0
U 15B1, 8D5E,0C ;5437           JSR.OS [T4.SUB.1]                       ; вызов подпрограммы
U 15B2, 364A,95 ;5438           MOV LS[#20] TO WR[1]                    ; ожидаемые данные = 00000020
U 15B3, 0869,3C ;5439           JSR [CHECK.RESULT]                      ; проверка WR0
U 15B4, 095A,F4 ;5440           JMP [LOOP.T4.1]                         ; зацикливание при ошибке, если разрешено
U 15B5, 8870,0C ;5441           JSR [INC.EN]                            ; увеличение номера ошибки до 2
                ;5442     LOOP.T4.2:
U 15B6, 2FB0,15 ;5443           CLR WR[0]                               ;
U 15B7, 0D5F,FC ;5444           JSR.OS [T4.SUB.2]                       ; вызов подпрограммы
U 15B8, 3640,95 ;5445           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 15B9, 0869,3C ;5446           JSR [CHECK.RESULT]                      ; проверка WR0
U 15BA, 895B,64 ;5447           JMP [LOOP.T4.2]                         ; зацикливание при ошибке, если разрешено
U 15BB, 8870,0C ;5448           JSR [INC.EN]                            ; увеличение номера ошибки до 3
                ;5449     LOOP.T4.3:
U 15BC, 2FB0,15 ;5450           CLR WR[0]                               ;
U 15BD, 369E,95 ;5451           MOV LS[ONES] TO WR[1]                   ;
U 15BE, BEF8,95 ;5452           MOV WR[1] TO LS[OS]                     ; установка всех битов в регистре OS
U 15BF, 8D5E,0C ;5453           JSR.OS [T4.SUB.1]                       ; вызов подпрограммы
U 15C0, 3640,95 ;5454           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 15C1, 0869,3C ;5455           JSR [CHECK.RESULT]                      ;
U 15C2, 895B,C4 ;5456           JMP [LOOP.T4.3]                         ; зацикливание при ошибке, если разрешено
U 15C3, 8870,0C ;5457           JSR [INC.EN]                            ; увеличение номера ошибки до 4
                ;5458     LOOP.T4.4:
U 15C4, 2FB0,15 ;5459           CLR WR[0]                               ;
U 15C5, 0D5F,FC ;5460           JSR.OS [T4.SUB.2]                       ; вызов подпрограммы
```

```
U 15C6, 3640,95 ;5461            MOV LS[#1] TO WR[1]                    ; ожидаемые данные = 1
U 15C7, 0B69,3C ;5462            JSR [CHECK.RESULT]                     ;
U 15C8, B95C,44 ;5463            JMP [LOOP.T4.4]                        ;
U 15C9, 8960,14 ;5464            JMP [END.T4]                           ;
                ;5465    15E0:
                ;5466    T4.SUB.1:
U 15E0, 2040,15 ;5467            INC WR[0]                              ; блок из 32 увеличений
U 15E1, 2040,15 ;5468            INC WR[0]                              ;
U 15E2, 2040,15 ;5469            INC WR[0]                              ;
U 15E3, 2040,15 ;5470            INC WR[0]                              ;
U 15E4, 2040,15 ;5471            INC WR[0]                              ;
U 15E5, 2040,15 ;5472            INC WR[0]                              ;
U 15E6, 2040,15 ;5473            INC WR[0]                              ;
U 15E7, 2040,15 ;5474            INC WR[0]                              ;
U 15E8, 2040,15 ;5475            INC WR[0]                              ;
U 15E9, 2040,15 ;5476            INC WR[0]                              ;
U 15EA, 2040,15 ;5477            INC WR[0]                              ;
U 15EB, 2040,15 ;5478            INC WR[0]                              ;
U 15EC, 2040,15 ;5479            INC WR[0]                              ;
U 15ED, 2040,15 ;5480            INC WR[0]                              ;
U 15EE, 2040,15 ;5481            INC WR[0]                              ;
U 15EF, 2040,15 ;5482            INC WR[0]                              ;
U 15F0, 2040,15 ;5483            INC WR[0]                              ;
U 15F1, 2040,15 ;5484            INC WR[0]                              ;
U 15F2, 2040,15 ;5485            INC WR[0]                              ;
U 15F3, 2040,15 ;5486            INC WR[0]                              ;
U 15F4, 2040,15 ;5487            INC WR[0]
U 15F5, 2040,15 ;5488            INC WR[0]
U 15F6, 2040,15 ;5489            INC WR[0]                              ;
U 15F7, 2040,15 ;5490            INC WR[0]                              ;
U 15F8, 2040,15 ;5491            INC WR[0]                              ;
U 15F9, 2040,15 ;5492            INC WR[0]                              ;
U 15FA, 2040,15 ;5493            INC WR[0]                              ;
U 15FB, 2040,15 ;5494            INC WR[0]                              ;
U 15FC, 2040,15 ;5495            INC WR[0]                              ;
U 15FD, 2040,15 ;5496            INC WR[0]                              ;
U 15FE, 2040,15 ;5497            INC WR[0]                              ;
                ;5498    T4.SUB.2:
U 15FF, 2040,15 ;5499            INC WR[0]                              ;
U 1600, 5B00,14 ;5500            RETURN                                 ;
                ;5501    END.T4:
```

```
;5502      .PAGE "ТЕСТ 5 - тест CONS HALT, пропуска и перехода при прерываниях (модуль DAP)"
;5503      ;
;5504      ; ОПИСАНИЕ ТЕСТА
;5505      ;
;5506      ;   Этот тест проверяет сигнал CONS HALT, генерируемый консольным процессором,
;5507      ;   и схемы пропуска, связанные с прерыванием, генерируемым сигналом CONS HALT.
;5508      ;   Также проверяет код идентификации на шинах D от D02 до D05. Выдается сигнал
;5509      ;   CPU ATTN для указания консольному процессору, что должен генерироваться
;5510      ;   сигнал CONS HALT, когда это необходимо. Тест выполняется следующим образом:
;5511      ;   выдается CPU ATTN с установленными битами 16 и 17 в слове управления и
;5512      ;   состояния, указывающими, что консольный процессор должен выдавать CONS HALT.
;5513      ;   Этот сигнал запоминается и передается на ПМЛ УПР.ПРЕРЫВАНИЕМ. До тех пор,
;5514      ;   пока HALT не маскируется инструкцией MISC, ПМЛ выставляет сигнал IRQ OUT L,
;5515      ;   который проходит через схему "ИЛИ" и запоминается как сигнал INTERR REQ H.
;5516      ;   Этот сигнал поступает на 8-входовый мультиплексор в схемах пропуска микро-
;5517      ;   инструкции. Для проверки этого сигнала используется инструкция, выполня-
;5518      ;   ющая пропуск, если нет прерывания. Код идентификации читается с использо-
;5519      ;   ванием инструкции MOVE при адресе местной памяти = 7E. Считывание осущест-
;5520      ;   вляется при низком уровне сигнала EN STAT REG L на выходе ПМЛ УПР.ПРЕРЫВА-
;5521      ;   НИЕМ. Тест повторяется со снятым сигналом CONS HALT для проверки пропуска
;5522      ;   и код идентификации проверяется на 1111 (не имеется ожидающего прерывания).
;5523      ;   Наконец, проверяется JMP при выставленном прерывании и при невыставленном
;5524      ;   прерывании. ПРИМЕЧАНИЕ: После того, как консоль возвращает управление
;5525      ;   микрокоду, для полной буферизации до проверки пропуска должны выдаваться
;5526      ;   3 микрослова.
;5527      ;
;5528      ; ПРЕДПОЛОЖЕНИЯ:
;5529      ;
;5530      ;   Предполагается, что схемы пропуска для других пропусков были проверены
;5531      ;   раньше.
;5532      ;
;5533      ; ШАГИ ТЕСТА:
;5534      ;
;5535      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;5536      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;5537      ;      памяти.
;5538      ;   2. Выдача сигнала CPU ATTN с установленными битами 16 и 17 в слове
;5539      ;      управления и состояния для генерации CONS HALT L.
;5540      ;   3. Загрузка 1F(H) в IPL для блокировки любых других прерываний (занимает 2
;5541      ;      микрослова). Выполнение двух инструкций NOP, затем выполнение инструкции
;5542      ;      MOVE с SCTL=NO.INTERRUPT и LS=7E для чтения кода идентификации и провер-
;5543      ;      ки на отсутствие пропуска.
;5544      ;   4. Проверка кода идентификации на 0 на шине D D05.
;5545      ;   5. Выдача сигнала CPU ATTN со сброшенными битами 16 и 17 в слове управления
;5546      ;      и состояния для снятия сигнала CONS HALT L.
;5547      ;   6. Выполнение трех инструкций NOP, затем выполнение MOVE с SCTL = NO.INTERRUPT
;5548      ;      и LS=7E для чтения кода идентификации и проверки на пропуск.
;5549      ;   7. Проверка кода идентификации на 1 на шине D D05.
;5550      ;   8. Выдача инструкции JMP с JCTL = NO.INTERRUPT и проверка на переход.
;5551      ;   9. Повторение шага 2, выполнение трех инструкций NOP и затем повторение шага
;5552      ;      8 для проверки на отсутствие перехода.
;5553      ;
;5554      ; ОШИБКИ:
;5555      ;
;5556      ;   ошибка 1 - SCTL = пропуск, если нет прерывания, выполняет пропуск при
```

```
                    ;5557      ;                выставленном CONS HALT.
                    ;5558      ;    ошибка 2 - выставленный CONS HALT не сформировал кода идентификации 0000.
                    ;5559      ;    ошибка 3 - инструкция с SCTL = пропуск, если нет прерывания, не выполнила
                    ;5560      ;               пропуска при снятом CONS HALT.
                    ;5561      ;    ошибка 4 - снятие CONS HALT не привело к формированию кода идентификации
                    ;5562      ;               1111
                    ;5563      ;    ошибка 5 - инструкция с JCTL = переход, если нет прерывания, не выполнили
                    ;5564      ;               перехода при снятом CONS HALT.
                    ;5565      ;    ошибка 6 - инструкция с JCTL = переход, если нет прерывания, не выполнилА
                    ;5566      ;               перехода при установленном CONS HALT.
                    ;5567      ;
                    ;5568      ; НАЛАДКА:
                    ;5569      ;
                    ;5570      ;    ОШИБКА 1 - Эту ошибку может вызвать ряд причин. Прежде всего, необходимо про-
                    ;5571      ;               верить сигнал IRQ OUT LOW на ножке 19 ПМЛ УПР.ПРЕРЫВАНИЕМ. Он дол-
                    ;5572      ;               жен быть низким во время инструкции MOVE с SCTL=NO.INTERRUPT. Если
                    ;5573      ;               нет, необходимо проверить входы CONSOLE HALT L (ножка 2) на низкий
                    ;5574      ;               уровень и MASK L (ножка 18) на высокий уровень. Сигнал CONSOLE
                    ;5575      ;               HALT L можно проследить в обратном направлении через 8-миразрядный
                    ;5576      ;               буфер, выходы которого подаются на ПМЛ, до 8-миразрядного буфера,
                    ;5577      ;               который загружается из консоли. Если сигнал CONS HALT L имеет высо-
                    ;5578      ;               кий уровень на первом буфере, необходимо проверить входы в то время,
                    ;5579      ;               когда консоль заполняет этот буфер. Если сигнал IRQ OUT L правильный,
                    ;5580      ;               необходимо проверить высокий уровень выхода двухвходовой схемы "НЕ
                    ;5581      ;               ИЛИ" и выход восмибитового буфера INTERR REQ H. Если этот сигнал
                    ;5582      ;               правильный, необходимо проверить тот же сигнал на входе 8-входового
                    ;5583      ;               мультиплексора схемы управления пропуском. Во время инструкции с
                    ;5584      ;               SCTL=NO.INTERRUPT выход на ножке 6 этого мультиплексора должен быть
                    ;5585      ;               низким, если 8-входовой мультиплексор работает исправно. Если этот
                    ;5586      ;               сигнал правильный, вероятнее всего, неисправна ПМЛ.
                    ;5587      ;    ОШИБКА 2 - Если это первая ошибка, причиной может быть или ПМЛ УПР.ПРЕРЫВАНИЕМ
                    ;5588      ;               или сигнал EN STAT REG L. Проверяется низкий уровень сигнала EN STAT
                    ;5589      ;               REG L на ПМЛ УПР.ПРЕРЫВАНИЕМ во время инструкции MOVE. Если это так,
                    ;5590      ;               сама ПМЛ неисправна. Другие входы этой ПМЛ должны быть: низкий уро-
                    ;5591      ;               вень для CONSOLE HALT L (ножка 2) и высокий уровень для MASK L (ножка
                    ;5592      ;               18). Если сигнал EN STAT REG L высокий, проверяется выход ПМЛ УПР.РЕ-
                    ;5593      ;               ГИСТРАМИ. Если здесь высокий уровень, по-видимому, неисправна эта ПМЛ.
                    ;5594      ;    ОШИБКА 3 - То же, что и при ошибке 1, за исключением того, что сигнал INTERR
                    ;5595      ;               REQ H должен быть низким, а сигналы, связанные с CONSOLE HALT, не выс-
                    ;5596      ;               тавлены.
                    ;5597      ;    ОШИБКА 4 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;5598      ;    ОШИБКА 5 - Подозревается ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ.
                    ;5599      ;    ОШИБКА 6 - Подозревается ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ.
                    ;5600      T.5:
U 1601, B65E,15     ;5601          MOV LS[BEGIN.TEST] TO WR[0]                  ; установка бита 15 в WR0 для слова управления и
                    ;5602                                                       ; состояния
U 1602, 3E80,15     ;5603          MOV WR[0] TO LS[CONTROL.STATUS]               ; установка бита 15 в слове управления и состояния. Бит
                    ;5604                                                       ; 15 указывает начало теста для конс.процессора
U 1603, 10E0,15     ;5605          MISC [SET.CP.ATTN]                           ; выдача сигнала CPU ATTN для конс.процессора
                    ;5606      WAIT.T5.0:
U 1604, 8960,44     ;5607          JMP [WAIT.T5.0]                              ; зацикливание для ожидания ответа конс.процессора
U 1605, B64A,15     ;5608          MOV LS[BIT5] TO WR[0]                        ;
U 1606, F88A,15     ;5609          MCOM WR[0] TO LS[ERROR.MASK]                  ; маска ошибки = FFFFFFDF
U 1607, 65A0,15     ;5610          CLR LS[PREVIOUS.ERROR]                       ; очистка предыдущего номера ошибки
U 1608, 3642,15     ;5611          MOV LS[CPU] TO WR[0]                          ; установка 2 в WR0
```

```
U 1609, 3EBC,15 ;5612            MOV WR[0] TO LS[MODULE.NUM]          ; установка кода модуля для модуля DAP
                ;5613    LOOP.T5.1:
U 160A, 0870,8C ;5614            JSR [ASSERT.NA]                      ;
U 160B, B640,15 ;5615            MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 160C, BEB2,15 ;5616            MOV WR[0] TO LS[ERROR.NUMBER]        ; установка номера ошибки для первой ошибки
U 160D, 3660,15 ;5617            MOV LS[INTERUPT.EN] TO WR[0]         ;
U 160E, C762,15 ;5618            BIS LS[CONSOLE.HALT] TO WR[0]        ;
U 160F, 3EB0,15 ;5619            MOV WR[0] TO LS[CONTROL.STATUS]      ; подготовка к установке сигнала CONSOLE HALT
U 1610, 10E0,15 ;5620            MISC [SET.CP.ATTN]                   ; вызов консольного процессора
                ;5621    WAIT.T5.1:
U 1611, 0961,14 ;5622            JMP [WAIT.T5.1]                      ; зацикливание до ответа конс.процессора
U 1612, B724,15 ;5623            MOV LS[HI.IPL] TO WR[0]              ; установка битов 16-20 в WR0
U 1613, BFFE,15 ;5624            MOV WR[0] TO LS[PSL.HW]              ; IPL = 1F
U 1614, DB00,15 ;5625            NOP                                  ;
U 1615, DB00,15 ;5626            NOP                                  ; выдержка времени до получения результата действия
U 1616, 36FD,95 ;5627            MOV LS[INTERRUPT.VEC] TO WR[3]       ; чтение кода идентификации
U 1617, DB00,07 ;5628            SKIP.IF[NO.INTERRUPT]                ; проверка на пропуск
U 1618, 0961,D4 ;5629            JMP [T5.2]                           ; переход к следующей части
U 1619, 2F80,15 ;5630            CLR WR[0]                            ; пропуск был: ошибка!
U 161A, 369E,95 ;5631            MOV LS[ONES] TO WR[1]                ; установка ошибочных данных
U 161B, 0869,3C ;5632            JSR [CHECK.RESULT]                   ; вызов подпрограммы обработки ошибки
U 161C, 0960,A4 ;5633            JMP [LOOP.T5.1]                      ; зацикливание при ошибке, если разрешено
                ;5634    T5.2:
U 161D, 8870,0C ;5635            JSR [INC.EN]                         ; увеличение номера ошибки до 2
U 161E, 0870,BC ;5636            JSR [DEASSERT.NA]                    ;
U 161F, A006,15 ;5637            MOV WR[3] TO WR[0]                   ; чтение кода идентификации
U 1620, 2FB2,95 ;5638            CLR WR[1]                            ; ожидаемые данные = 0
U 1621, 0869,3C ;5639            JSR [CHECK.RESULT]                   ; проверка кода идентификации
U 1622, 0960,A4 ;5640            JMP [LOOP.T5.1]                      ; зацикливание при ошибке, если разрешено
                ;5641    LOOP.T5.3:
U 1623, 0870,8C ;5642            JSR [ASSERT.NA]                      ;
U 1624, 36C0,15 ;5643            MOV LS[#3(H)] TO WR[0]               ; установка 3 в WR0
U 1625, BEB2,15 ;5644            MOV WR[0] TO LS[ERROR.NUMBER]        ; номер ошибки = 3
U 1626, 3660,15 ;5645            MOV LS[INTERUPT.EN] TO WR[0]         ; сброс прерывания
U 1627, 3EB0,15 ;5646            MOV WR[0] TO LS[CONTROL.STATUS]      ; в слове управления и состояния бит 16 установлен
U 1628, 10E0,15 ;5647            MISC [SET.CP.ATTN]                   ; вызов консольного процессора
                ;5648    WAIT.T5.3:
U 1629, 8962,94 ;5649            JMP [WAIT.T5.3]                      ; ожидание ответа конс.процессора
U 162A, DB00,15 ;5650            NOP                                  ;
U 162B, DB00,15 ;5651            NOP                                  ;
U 162C, DB00,15 ;5652            NOP                                  ; выдержка времени
U 162D, 36FD,95 ;5653            MOV LS[INTERRUPT.VEC] TO WR[3]       ; чтение кода идентификации
U 162E, DB00,07 ;5654            SKIP.IF[NO.INTERRUPT]                ; проверка пропуска
U 162F, 5B00,1E ;5655            SKIP                                 ;
U 1630, 0963,54 ;5656            JMP [T5.4]                           ; переход, если пропуск произошел
U 1631, 2F80,15 ;5657            CLR WR[0]                            ; пропуска нет: ошибка!
U 1632, 369E,95 ;5658            MOV LS[ONES] TO WR[1]                ; установка ошибочных данных
U 1633, 0869,3C ;5659            JSR [CHECK.RESULT]                   ; вызов подпрограммы обработки ошибки
U 1634, 8962,34 ;5660            JMP [LOOP.T5.3]                      ; зацикливание при ошибке, если разрешено
                ;5661    T5.4:
U 1635, 8870,0C ;5662            JSR [INC.EN]                         ; увеличение номера ошибки до 4
U 1636, 0870,BC ;5663            JSR [DEASSERT.NA]
U 1637, A006,15 ;5664            MOV WR[3] TO WR[0]                   ; чтение кода идентификации
U 1638, 364A,95 ;5665            MOV LS[BIT5] TO WR[1]                ; ожидаемые данные = установленный бит 5
U 1639, 0869,3C ;5666            JSR [CHECK.RESULT]                   ; проверка кода идентификации
```

ТЕСТ 5 — тест CONS HALT, пропуска и перехода при прерываниях (модуль DAP)

```
U 163A, 8962,34 ;5667            JMP [LOOP.T5.3]                    ; зацикливание при ошибке, если разрешено
U 163B, 8870,0C ;5668            JSR [INC.EN]                       ; увеличение номера ошибки до 5
                ;5669    LOOP.T5.5:
U 163C, 0870,BC ;5670            JSR [ASSERT.NA]                    ;
U 163D, 0964,27 ;5671            JMP.IF[NO.INTERRUPT] TO [T5.OK.5]  ;
U 163E, 2F80,15 ;5672            CLR WR[0]                          ; не было перехода: ошибка!
U 163F, 369E,95 ;5673            MOV LS[ONES] TO WR[1]              ; установка ошибочных данных
U 1640, 0869,3C ;5674            JSR [CHECK.RESULT]                 ; вызов подпрограммы обработки ошибки
U 1641, 0963,C4 ;5675            JMP [LOOP.T5.5]                    ; зацикливание при ошибке, если разрешено
                ;5676    T5.OK.5:
U 1642, 3660,15 ;5677            MOV LS[INTERUPT.EN] TO WR[0]       ;
U 1643, C762,15 ;5678            BIS LS[CONSOLE.HALT] TO WR[0]      ; установка сигнала CONSOLE HALT
U 1644, 3E80,15 ;5679            MOV WR[0] TO LS[CONTROL.STATUS]    ; в слове управления и состояния установлены биты 16 и 17
U 1645, 10E0,15 ;5680            MISC [SET.CP.ATTN]                 ; вызов консольного процессора
                ;5681    WAIT.T5.5:
U 1646, 8964,64 ;5682            JMP [WAIT.T5.5]                    ; зацикливание до ответа конс.процессора
U 1647, 8870,0C ;5683            JSR [INC.EN]                       ; увеличение номера ошибки до 6
                ;5684    LOOP.T5.6:
U 1648, 8964,A7 ;5685            JMP.IF[NO.INTERRUPT] TO [T5.ERR.6] ;
U 1649, 0964,E4 ;5686            JMP [CLEANUP.T5]                   ; все выполнено
                ;5687    T5.ERR.6:
U 164A, 2F80.15 ;5688            CLR WR[0]                          ; переход был: ошибка!
U 164B, B69E,15 ;5689            MOV LS[ONES] TO WR[0]              ; установка ошибочных данных
U 164C, 0869,3C ;5690            JSR [CHECK.RESULT]                 ; вызов подпрограммы обработки ошибки
U 164D, 0964.84 ;5691            JMP [LOOP.T5.6]                    ; зацикливание при ошибке, если разрешено
                ;5692    CLEANUP.T5:
U 164E, 3660,15 ;5693            MOV LS[INTERUPT.EN] TO WR[0]       ;
U 164F, 3E80,15 ;5694            MOV WR[0] TO LS[CONTROL.STATUS]    ;
U 1650, 10E0,15 ;5695            MISC [SET.CP.ATTN]                 ; сброс сигналов
                ;5696    WAIT.T5.6:
U 1651, 8965,14 ;5697            JMP [WAIT.T5.6]                    ; ожидание ответа конс.процессора
                ;5698    END.T5:
```

```
;5699      .PAGE "ТЕСТ 6 - тест ПМЛ УПР.ПРЕРЫВАНИЕМ, IPL и BUS IRQ (модуль DAP)"
;5700      ;
;5701      ; ОПИСАНИЕ ТЕСТА:
;5702      ;
;5703      ;       Этот тест выставляет поочередно все сигналы прерываний и проверяет, что
;5704      ;    для них разрешается выставить INTERR REQ, если IPL (уровень приоритета
;5705      ;    прерываний) находится на правильном уровне или ниже. Сигналы прерываний
;5706      ;    PWR FAIL INT L, INTRVL TIM INT L и CONS ATTN выставляются путем выдачи
;5707      ;    сигнала CPU ATTN для консольного процессора при установленных соответству-
;5708      ;    ющих битах в слове управления и состояния. Консольный процессор должен ус-
;5709      ;    тановить соответствующие сигналы и возвратить управление микрокоду. Сигна-
;5710      ;    лы BUS IRQ (уровни с 4 по 7) управляются путем выдачи инструкции MISC пос-
;5711      ;    ле загрузки регистра OS с 0 по 3. Биты IPL загружаются из шины.Y инструк-
;5712      ;    цией MODE при LS=FF. Это выполняется стробированием буфера PSL сигналом
;5713      ;    LOAD PSL L из ПМЛ УПР.РЕГИСТРАМИ. Тест возбуждает одну линию прерывания и
;5714      ;    запускает счет копии IPL по битам, начиная от 1F (H) до точки, в которой
;5715      ;    прерывание разрешается. (В это время должно проверяться отсутствие прерыва-
;5716      ;    ния инструкцией пропуска, если нет прерывания). Когда код IPL достигает зна-
;5717      ;    чения, которое разрешает данному прерыванию возбуждать INTERR REQ, инструкция
;5718      ;    пропуска, если нет прерывания, не выполняет пропуска. В это время проверяется
;5719      ;    правильность значения кода идентификации и продолжается счет копии IPL вниз
;5720      ;    до 0, с той же проверкой, как и раньше. Примечание: после каждого вызова
;5721      ;    консольного процессора должны выдаваться 3 микрослова для полной буферизации.
;5722      ;    2 микрослова должны выдаваться после изменения любой шины BR для разрешения
;5723      ;    пропуска, и 1 микрослово выдается для проверки кода идентификации.
;5724      ;
;5725      ; ПРЕДПОЛОЖЕНИЯ:
;5726      ;
;5727      ;    Предполагается, что предыдущие тесты прерываний, использующие CONSOLE HALT,
;5728      ;    успешно выполнены.
;5729      ;
;5730      ; ШАГИ ТЕСТА:
;5731      ;
;5732      ;       1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти (для
;5733      ;          распечатки ошибок) и очистка предыдущего номера ошибки в местной памяти.
;5734      ;       2. Выдача CPU ATTN с установленными битами 16 и 18 в слове управления и сос-
;5735      ;          тояния для генерации PWR FAIL INT L. Консольный процессор должен возвратить
;5736      ;          управление следующему микрослову.
;5737      ;       3. Загрузка в WR2 значения 1F 00 00(H) (1F на шинах BUS Y D29 по BUS Y D16) и
;5738      ;          в WR3 значения 1D 00 00(H). 1D является значением IPL, которое разрешает пре-
;5739      ;          рывание.
;5740      ;       4. Выполнение MOVE с LS = FF для загрузки в IPL PSL битов данных из WR2
;5741      ;          (начальное значение 1F(H)).
;5742      ;       5. Выполнение сравнения WR2 с WR3, пропуск, если WR2 не равен WR3. Переход к
;5743      ;          шагу 9, если WR2 равен WR3.
;5744      ;       6. Выполнение NOP для стробирования буфера INTERR REQ H.
;5745      ;       7. Выполнение NOP с SCTL = пропуск, если нет прерывания и проверка на пропуск.
;5746      ;       8. Вычитание 1 0000(H) из WR2 для уменьшения значения IPL на 1 и переход к
;5747      ;          шагу 4.
;5748      ;       9. Выполнение NOP с SCTL = пропуск, если нет прерывания.
;5749      ;          Проверка отсутствия пропуска.
;5750      ;       10. Выполнение MOVE из LS 7E(H) в WR0 для загрузки кода идентификации в
;5751      ;           WR0, загрузка WR1 ожидаемыми данными в битах 5-2.
;5752      ;       11. Проверка кода идентификации, затем проверка WR2 на нуль. Если нуль, пе-
;5753      ;           реход к шагу 14, в противном случае продолжение.
```

```
;5754       ;    12. Вычитание 1 0000(H) из WR2 для уменьшения значения IPL на 1.
;5755       ;    13. Выполнение MOVE с LS = FF из WR2 для загрузки данными битов IPL PSL.
;5756       ;    14. Выдача CPU ATTN с установленными битами 16 и 19 в слове управления и со-
;5757       ;        стояния для генерации INTRVL TIM INT L. Консольный процессор должен возв-
;5758       ;        ратить управление следующему микрослову.
;5759       ;    15. Загрузка в WR2 значения 1F0000(H) (1F на шинах BUS Y D20 по BUS Y D16)
;5760       ;        и в WR3 значения 170000(H). 17 является первым значением IPL, которое разре-
;5761       ;        шает прерывание. Повторение шагов с 4 по 13 посредством JSR (шаги с 4 по 13
;5762       ;        являются подпрограммой). Возврат после выполнения к следующему шагу.
;5763       ;    16. Выдача CPU ATTN с установленными битами 16 и 20 в слове управления и со-
;5764       ;        стояния для генерации CONS ATTN L. Консольный процессор должен возвратить
;5765       ;        управление следующему микрослову.
;5766       ;    17. Загрузка в WR2 значения 1F0000(H) (1F на шинах BUS Y D20 по BUS Y D16) и
;5767       ;        в WR3 значения 130000(H). 13 является первым значением IPL, которое разреша-
;5768       ;        ет прерывание. Повторение шагов с 4 по 13 посредством JSR (шаги с 4 по 13
;5769       ;        являются подпрограммой). Возврат после выполнения к следующему шагу.
;5770       ;    18. Выдача CPU ATTN с установленным битом 16 в слове управления и состояния
;5771       ;        для снятия прерываний из консоли.
;5772       ;    19. Загрузка в WR2 значения 1F0000(H) (1F на шинах BUS Y D20 по BUS Y D16)
;5773       ;        и в WR3 значения 160000(H). 16 является первым значением IPL, которое разре-
;5774       ;        шает прерывание для BR7. Повторение шагов с 4 по 13 посредством JSR (шаги с
;5775       ;        4 по 13 являются подпрограммой) со следующими дополнениями: шаг 4A - загруз-
;5776       ;        ка регистра OS требуемым значением BUS IRQ в битах 3-0. Выполнение инструк-
;5777       ;        ции MISC для передачи OS в регистр BUS IRQ. Шаг 9A - загрузка регистра OS
;5778       ;        требуемым значением BUS IRQ в битах 3-0. Выполнение инструкции MISC для пе-
;5779       ;        редачи OS в регистр BUS IRQ. выполнение NOP. Шаг 13 - исключается NOP и "ПЕ-
;5780       ;        РЕХОД К ШАГУ 10" заменяется "ПЕРЕХОДОМ К ШАГУ 9A". Возврат после выполнения
;5781       ;        к следующему шагу.
;5782       ;    20. Повторение шага 19 с данными = 15 в WR3 и проверка BR6.
;5783       ;    21. Повторение шага 19 с данными = 14 в WR3 и проверка BR5.
;5784       ;    22. Повторение шага 19 с данными = 13 в WR3 и проверка BR4.
;5785       ;
;5786       ; ОШИБКИ:
;5787       ;
;5788       ;    Другие данные: уровень приоритета прерывания.
;5789       ;
;5790       ;    ошибка 1 - сигнал PWR FAIL INT вызывает прерывание раньше, чем значение IPL
;5791       ;               стало достаточно низким для разрешения этого прерывания.
;5792       ;    ошибка 2 - сигнал PWR FAIL INT не вызвал прерывания, когда значение IPL было
;5793       ;               корректное для его разрешения.
;5794       ;    ошибка 3 - сигнал PWR FAIL INT сформировал некорректный код идентификации.
;5795       ;               Должен быть 000(B).
;5796       ;    ошибка 4 - сигнал INTRVL TIM INT вызвал прерывание раньше, чем значение IPL
;5797       ;               стало достаточно низким для этого прерывания. Полученные данные в
;5798       ;               распечатке представляют собой IPL во время прерывания, ожидаемые
;5799       ;               данные являются теми значениями IPL, которые должны быть до разре-
;5800       ;               шения прерывания.
;5801       ;    ошибка 5 - INTRVL TIM INT не вызвал прерывания, когда значение IPL было
;5802       ;               корректное для разрешения прерывания.
;5803       ;    ошибка 6 - сигнал INTRVL TIM INT сформировал некорректный код идентификации.
;5804       ;               Должен быть 001(B).
;5805       ;    ошибка 7 - сигнал CONS ATTN вызвал прерывание раньше, чем значение IPL стало
;5806       ;               достаточно низким для этого прерывания. Полученные данные в распе-
;5807       ;               чатке представляют собой IPL во время прерывания, ожидаемые данные
;5808       ;               являются теми значениями IPL, которые должны быть до разрешения пре-
```

```
;5809       ;              рывания.
;5810       ;    ошибка 8 - сигнал CONS ATTN не вызвал прерывания, когда значение IPL было
;5811       ;               корректное для его разрешения.
;5812       ;    ошибка 9 - сигнал CONS ATTN сформировал некорректный код идентификации.
;5813       ;               Должен быть 101(B).
;5814       ;    ошибка A - BUS IRQ7 вызывает прерывание раньше, чем значение IPL стало доста-
;5815       ;               точно низким для разрешения прерывания. полученные данные в распе-
;5816       ;               чатке представляют собой IPL во время прерывания, ожидаемые данные
;5817       ;               являются теми значениями IPL, которые должны быть до разрешения
;5818       ;               прерывания.
;5819       ;    ошибка B - сигнал BUS IRQ7 не вызвал прерывания, когда значение IPL было
;5820       ;               корректное для его разрешения.
;5821       ;    ошибка C - сигнал BUS IRQ7 сформировал некорректный код идентификации.
;5822       ;               должен быть 0101(B).
;5823       ;    ошибка D - сигнал BUS IRQ6 вызвал прерывание раньше, чем значение IPL стало
;5824       ;               достаточно низким для разрешения прерывания. Полученные данные в
;5825       ;               распечатке представляют собой IPL во время прерывания, ожидаемые
;5826       ;               данные являются теми значениями IPL, которые должны быть до раз-
;5827       ;               решения прерывания.
;5828       ;    ошибка E - сигнал BUS IRQ6 не вызвал прерывания, когда значение IPL было
;5829       ;               корректное для его разрешения.
;5830       ;    ошибка F - сигнал BUS IRQ6 сформировал некорректный код идентификации. Дол-
;5831       ;               жен быть 011(B).
;5832       ;    ошибка 10 - сигнал BUS IRQ5 вызвал прерывание раньше, чем значение IPL стало
;5833       ;                достаточно низким для его разрешения. Полученные данные в распе-
;5834       ;                чатке представляют собой IPL во время прерывания, ожидаемые данные
;5835       ;                являются теми значениями IPL, которые должны быть до разрешения
;5836       ;                прерывания.
;5837       ;    ошибка 11 - сигнал BUS IRQ5 не вызвал прерывания, когда значение IPL было
;5838       ;                корректное для его разрешения.
;5839       ;    ошибка 12 - сигнал BUS IRQ5 сформировал некорректный код идентификации.
;5840       ;                Должно быть 100(B).
;5841       ;    ошибка 13 - сигнал BUS IRQ4 вызвал прерывание раньше, чем значение IPL стало
;5842       ;                достаточно низким для его разрешения. Полученные данные в распе-
;5843       ;                чатке представляют собой IPL во время прерывания, ожидаемые данные
;5844       ;                являются теми значениями IPL, которые должны быть до разрешения
;5845       ;                прерывания.
;5846       ;    ошибка 14 - сигнал BUS IRQ4 не вызвал прерывания когда, значение IPL было
;5847       ;                корректное для его разрешения.
;5848       ;    ошибка 15 - сигнал BUS IRQ4 сформировал некорректный код идентификации.
;5849       ;                Должен быть 110(B).
;5850       ;
;5851       ; НАЛАДКА:
;5852       ;
;5853       ;    ПРИМЕЧАНИЕ: Если имеются какие-либо устройства общей шины, возможно, что
;5854       ;    они могут быть причиной возникновения прерывания на общей шине. Любые устрой-
;5855       ;    ства, имеющиеся на общей шине, должны быть убраны, если подозревается, что
;5856       ;    они вызывают неуспешное выполнение этого теста.
;5857       ;
;5858       ;    ОШИБКА 1 - Эта ошибка может появиться, если выход ПМЛ УПР.ПРЕРЫВАНИЕМ IRQ OUT
;5859       ;    L (ножка 19) имеет низкий уровень. Необходимо проверить входы IPL на значе-
;5860       ;    ния 1E(H) или 1F(H) во время NOP с SCTL = пропуск, если нет прерывания. Если
;5861       ;    они правильные, то вероятно, что ПМЛ УПР.ПРЕРЫВАНИЕМ неисправна. Если зна-
;5862       ;    чение IPL некорректное, необходимо проверить выходы 8-мибитового буфера, ко-
;5863       ;    торый запоминает биты PSL (IPL, текущий режим и T-TRAP). Если они некоррект-
```

```
;5864       ;   ные, необходимо проверить входы этого буфера во время инструкции MOVE, кото-
;5865       ;   рая загружает IPL. Вход стробирования (ножка 11) формируется выходом LOAD PSL
;5866       ;   L микросхемы ПМЛ УПР.РЕГИСТРАМИ и сигналом CLOCK REGS H.
;5867       ;
;5868       ;   ОШИБКА 2 - Эта ошибка появляется, если ПМЛ УПР.ПРЕРЫВАНИЕМ не выводит низко-
;5869       ;   го уровня на выход IRQ OUT L (ножка 19). Необходимо проверить входы этой ПМЛ,
;5870       ;   как и при ошибке 1. Если они правильные, необходимо проверить сигналы с IRQ2
;5871       ;   по IRQ0 (ножка 9,12 и 13 соответственно) на значение 7. Если оно правильное,
;5872       ;   то вероятно, что ПМЛ неисправна. Если эти входы некорректные, необходимо
;5873       ;   проверить низкий уровень на информационном входе 7 и входе выборки микросхе-
;5874       ;   мы декодирования приоритета. Микросхема декодирования приоритета неисправна,
;5875       ;   если эти входы правильные. В противном случае необходимо проследить в обрат-
;5876       ;   ном направлении цепи сигнала PWR FAIL INT L.
;5877       ;
;5878       ;   ОШИБКА 3 - Если это первая ошибка, наиболее вероятной причиной ошибки явля-
;5879       ;   ется ПМЛ УПР.ПРЕРЫВАНИЕМ.
;5880       ;
;5881       ;   ОШИБКА 4 - См. ошибку 1. Отличается только тем, что значение IPL должно быть
;5882       ;   18(H) или больше.
;5883       ;
;5884       ;   ОШИБКА 5 - См. ошибку 2. Отличается только тем, что значение IPL должно быть
;5885       ;   18(H) или больше и IRQ2 по IRQ0 равны 6. Прерывание вызывает сигнал
;5886       ;   INTRVL TIM INT.
;5887       ;
;5888       ;   ОШИБКА 6 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
;5889       ;
;5890       ;   ОШИБКА 7 - См. ошибку 1. Отличается только тем, что значение IPL должно быть
;5891       ;   14(H) или больше.
;5892       ;
;5893       ;   ОШИБКА 8 - См. ошибку 2. Отличается только тем, что значение IPL должно быть
;5894       ;   14(H) или больше и IRQ2 по IRQ0 равны 2. Прерывание вызывает сигнал CONS
;5895       ;   ATTN.
;5896       ;
;5897       ;   ОШИБКА 9 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
;5898       ;
;5899       ;   ОШИБКА A - См. ошибку 1. Отличается только тем, что значение IPL должно быть
;5900       ;   17(H) или больше.
;5901       ;
;5902       ;   ОШИБКА B - См. ошибку 2. Отличается только тем, что значение IPL должно быть
;5903       ;   17(H) или больше и IRQ2 по IRQ0 равны 5. Прерывание вызывает сигнал BR7.
;5904       ;
;5905       ;   ОШИБКА C - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
;5906       ;
;5907       ;   ОШИБКА D - См. ошибку 1. Отличается только тем, что значение IPL должно быть
;5908       ;   6;(H) или больше.
;5909       ;
;5910       ;   ОШИБКА E - См. ошибку 2. Отличается только тем, что значение IPL должно быть
;5911       ;   16(H) или больше и IRQ2 по IRQ0 равны 4. Прерывание вызывает сигнал BR6.
;5912       ;
;5913       ;   ОШИБКА F - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
;5914       ;
;5915       ;   ОШИБКА 10 - См. ошибку 1. Отличается только тем, что значение IPL должно
;5916       ;   быть 15(H) или больше.
;5917       ;
;5918       ;   ОШИБКА 11 - См. ошибку 2. Отличается только тем, что значение IPL должно
```

```
                    ;5919       ;   быть 15(H) или больше и IRQ2 по IRQ0 равны 3. Прерывание вызывает сигнал
                    ;5920       ;   BR5.
                    ;5921       ;
                    ;5922       ;   ОШИБКА 12 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;5923       ;
                    ;5924       ;   ОШИБКА 13 - См. ошибку 1. Отличается только тем, что значение IPL должно быть
                    ;5925       ;   14(H) или  больше.
                    ;5926       ;
                    ;5927       ;   ОШИБКА 14 - См. ошибку 2. Отличается только тем, что значение IPL должно
                    ;5928       ;   быть 14(H) или больше и IRQ2 по IRQ0 равны 1. Прерывание вызывает сигнал
                    ;5929       ;   BR4.
                    ;5930       ;
                    ;5931       ;   ОШИБКА 15 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;5932       ;
                    ;5933   T.6:
U 1652, 8A5E,15     ;5934           MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;5935                                                ; состояния
U 1653, 3EB0,15     ;5936           MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления и состояния. Бит
                    ;5937                                                ; 15 указывает начало теста для консольного процессора
U 1654, 10E0,15     ;5938           MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для консольного процессора
                    ;5939   WAIT.T6.0:
U 1655, 0965,54     ;5940           JMP [WAIT.T6.0]                      ; зацикливание для ожидания ответа консольного
                    ;5941                                                ; процессора
U 1656, 5F44,15     ;5942           MCOM LS[BIT2] TO WR[0]               ;
U 1657, 4546,15     ;5943           BIC LS[BIT3] TO WR[0]                ;
U 1658, C548,15     ;5944           BIC LS[BIT4] TO WR[0]                ;
U 1659, 454A,15     ;5945           BIC LS[BIT5] TO WR[0]                ;
U 165A, 3E6A,15     ;5946           MOV WR[0] TO LS[ERROR.MASK]           ; маска ошибки = FFFFFF03
U 165B, 65A0,15     ;5947           CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 165C, 3642,15     ;5948           MOV LS[CPU] TO WR[0]                 ; установка 2 в WR0
U 165D, 3E8C,15     ;5949           MOV WR[0] TO LS[MODULE.NUM]           ; установка кода модуля для модуля DAP
U 165E, 8724,15     ;5950           MOV LS[HI.IPL] TO WR[0]               ; установка битов 16-20 в WR0
U 165F, 3E10,15     ;5951           MOV WR[0] TO LS[T8]                   ; T8 = 1F 00 00
U 1660, 3660,15     ;5952           MOV LS[INTERUPT.EN] TO WR[0]          ;
U 1661, C764,15     ;5953           BIS LS[PWR.FAIL] TO WR[0]             ;
U 1662, 3EB0,15     ;5954           MOV WR[0] TO LS[CONTROL.STATUS]       ; установка слова управления для генерации
                    ;5955                                                ; PWR FAIL INT L
U 1663, 10E0,15     ;5956           MISC [SET.CP.ATTN]                   ; вызов консольного процессора
                    ;5957   WAIT.T6.1:
U 1664, 8966,44     ;5958           JMP [WAIT.T6.1]                      ; зацикливание до ответа консольного процессора
U 1665, 3611,15     ;5959           MOV LS[T8] TO WR[2]                   ; WR2 = 1F 00 00
U 1666, B611,95     ;5960           MOV LS[T8] TO WR[3]                   ;
U 1667, 4563,95     ;5961           BIC LS[BIT17] TO WR[3]                ; WR3 = 1D 00 00
U 1668, B64A,15     ;5962           MOV LS[BIT5] TO WR[0]                 ;
U 1669, BE12,15     ;5963           MOV WR[0] TO LS[T9]                   ; код идентификации = 100000(B)
                    ;5964   LOOP.T6.1:
U 166A, B640,15     ;5965           MOV LS[#1] TO WR[0]                   ;
U 166B, BEB2,15     ;5966           MOV WR[0] TO LS[ERROR.NUMBER]         ; номер ошибки = 1
U 166C, 096C,EC     ;5967           JSR [SUB.T6]                         ;
U 166D, 0966,A4     ;5968           JMP [LOOP.T6.1]                      ; зацикливание при ошибке, если разрешено
U 166E, 3660,15     ;5969           MOV LS[INTERUPT.EN] TO WR[0]          ;
U 166F, 4766,15     ;5970           BIS LS[INTERVAL.TIM] TO WR[0]         ;
U 1670, 3EB0,15     ;5971           MOV WR[0] TO LS[CONTROL.STATUS]       ; установка слова управления и состояния для генерации
                    ;5972                                                ; INTRVL TIM INT L
U 1671, 10E0,15     ;5973           MISC [SET.CP.ATTN]                   ; вызов консольного процессора
```

```
                    ;5974    WAIT.T6.4:
U 1672,  0967,24    ;5975            JMP [WAIT.T6.4]                    ; зацикливание до ответа консольного процессора
U 1673,  3611,15    ;5976            MOV LS[T8] TO WR[2]                 ; 1F 00 00
U 1674,  B611,95    ;5977            MOV LS[T8] TO WR[3]                 ;
U 1675,  C567,95    ;5978            BIC LS[BIT19] TO WR[3]              ; WR3 = 17 00 00
U 1676,  3644,15    ;5979            MOV LS[BIT2] TO WR[0]               ;
U 1677,  C74A,15    ;5980            BIS LS[BIT5] TO WR[0]               ;
U 1678,  BE12,15    ;5981            MOV WR[0] TO LS[T9]                 ; Код идентификации = 100100(B)
                    ;5982    LOOP.T6.4:
U 1679,  3644,15    ;5983            MOV LS[#4] TO WR[0]                 ;
U 167A,  BE82,15    ;5984            MOV WR[0] TO LS[ERROR.NUMBER]       ; номер ошибки = 4
U 167B,  096C,EC    ;5985            JSR [SUB.T6]                       ;
U 167C,  8967,94    ;5986            JMP [LOOP.T6.4]                    ; зацикливание при ошибке, если разрешено
U 167D,  3660,15    ;5987            MOV LS[INTERUPT.EN] TO WR[0]        ;
U 167E,  C768,15    ;5988            BIS LS[CONSOLE.ATTN] TO WR[0]       ;
U 167F,  3E80,15    ;5989            MOV WR[0] TO LS[CONTROL.STATUS]     ; установка слова управления и состояния для генерации
                    ;5990                                               ; CONS ATTN L
U 1680,  10E0,15    ;5991            MISC [SET.CP.ATTN]                 ; вызов консольного процессора
                    ;5992    WAIT.T6.7:
U 1681,  0968,14    ;5993            JMP [WAIT.T6.7]                    ; зацикливание до ответа консольного процессора
U 1682,  3611,15    ;5994            MOV LS[T8] TO WR[2]                 ; WR2 = 1F 00 00
U 1683,  B611,95    ;5995            MOV LS[T8] TO WR[3]                 ;
U 1684,  C567,95    ;5996            BIC LS[BIT19] TO WR[3]              ;
U 1685,  4565,95    ;5997            BIC LS[BIT18] TO WR[3]              ; WR3 = 13 00 00
U 1686,  3644,15    ;5998            MOV LS[BIT2] TO WR[0]               ;
U 1687,  4748,15    ;5999            BIS LS[BIT4] TO WR[0]               ;
U 1688,  C74A,15    ;6000            BIS LS[BIT5] TO WR[0]               ;
U 1689,  BE12,15    ;6001            MOV WR[0] TO LS[T9]                 ; Код идентификации = 110100(B)
                    ;6002    LOOP.T6.7:
U 168A,  3646,15    ;6003            MOV LS[#8] TO WR[0]                 ;
U 168B,  C140,15    ;6004            SUB LS[#1] FROM WR[0]               ;
U 168C,  BE82,15    ;6005            MOV WR[0] TO LS[ERROR.NUMBER]       ; номер ошибки = 7
U 168D,  096C,EC    ;6006            JSR [SUB.T6]                       ;
U 168E,  8968,A4    ;6007            JMP [LOOP.T6.7]                    ; зацикливание при ошибке, если разрешено
U 168F,  3660,15    ;6008            MOV LS[INTERUPT.EN] TO WR[0]        ;
U 1690,  3E80,15    ;6009            MOV WR[0] TO LS[CONTROL.STATUS]     ; установка слова управления и состояния для снятия
                    ;6010                                               ; прерываний
U 1691,  8870,FC    ;6011            JSR [ASSERT.OTHER]                 ; будет использоваться поле "ДРУГИЕ ДАННЫЕ"
U 1692,  10E0,15    ;6012            MISC [SET.CP.ATTN]                 ; вызов консольного процессора
                    ;6013    WAIT.T6.A:
U 1693,  0969,34    ;6014            JMP [WAIT.T6.A]                    ; зацикливание до ответа консольного процессора
U 1694,  3611,15    ;6015            MOV LS[T8] TO WR[2]                 ; 1F 00 00
U 1695,  B611,95    ;6016            MOV LS[T8] TO WR[3]                 ;
U 1696,  C561,95    ;6017            BIC LS[BIT16] TO WR[3]              ;
U 1697,  C567,95    ;6018            BIC LS[BIT19] TO WR[3]              ; WR3 = 16 00 00
U 1698,  B646,15    ;6019            MOV LS[BIT3] TO WR[0]               ;
U 1699,  C74A,15    ;6020            BIS LS[BIT5] TO WR[0]               ;
U 169A,  BE12,15    ;6021            MOV WR[0] TO LS[T9]                 ; Код идентификации = 101000(B)
U 169B,  B646,15    ;6022            MOV LS[BIT3] TO WR[0]               ;
U 169C,  3E16,15    ;6023            MOV WR[0] TO LS[T11]                ; значение BUS IRQ = 7
                    ;6024    LOOP.T6.A:
U 169D,  B646,15    ;6025            MOV LS[#8] TO WR[0]                 ;
U 169E,  4742,15    ;6026            BIS LS[#2] TO WR[0]                 ;
U 169F,  BE82,15    ;6027            MOV WR[0] TO LS[ERROR.NUMBER]       ; номер ошибки = A
U 16A0,  896F,5C    ;6028            JSR [SUB2.T6]                      ;
```

```
U 16A1, 8969,D4 ;6029            JMP [LOOP.T6.A]                    ; зацикливание при ошибке, если разрешено
U 16A2, 3611,15 ;6030            MOV LS[T8] TO WR[2]                ; WR2 = 1F 00 00
U 16A3, B611,95 ;6031            MOV LS[T8] TO WR[3]                ;
U 16A4, 4563,95 ;6032            BIC LS[BIT17] TO WR[3]             ;
U 16A5, C567,95 ;6033            BIC LS[BIT19] TO WR[3]             ; WR3 = 15 00 00
U 16A6, B646,15 ;6034            MOV LS[BIT3] TO WR[0]              ;
U 16A7, 4744,15 ;6035            BIS LS[BIT2] TO WR[0]              ;
U 16A8, C74A,15 ;6036            BIS LS[BIT5] TO WR[0]              ;
U 16A9, BE12,15 ;6037            MOV WR[0] TO LS[T9]                ; код идентификации = 101100(B)
U 16AA, 3644,15 ;6038            MOV LS[BIT2] TO WR[0]              ;
U 16AB, 3E16,15 ;6039            MOV WR[0] TO LS[T11]               ; значение BUS IRQ = 6
                ;6040    LOOP.T6.D:
U 16AC, B646,15 ;6041            MOV LS[#8] TO WR[0]
U 16AD, 4744,15 ;6042            BIS LS[#4] TO WR[0]
U 16AE, 2040,15 ;6043            INC WR[0]                          ;
U 16AF, BE82,15 ;6044            MOV WR[0] TO LS[ERROR.NUMBER]      ; номер ошибки = D
U 16B0, 896F,5C ;6045            JSR [SUB2.T6]                      ;
U 16B1, 096A,C4 ;6046            JMP [LOOP.T6.D]                    ; зацикливание при ошибке, если разрешено
U 16B2, 3611,15 ;6047            MOV LS[T8] TO WR[2]                ; WR2 = 1F 00 00
U 16B3, B669,95 ;6048            MOV LS[#100000] TO WR[3]           ; загрузка 100000 в WR3
U 16B4, C765,95 ;6049            BIS LS[#40000] TO WR[3]            ; WR3 = 14 00 00
U 16B5, 3648,15 ;6050            MOV LS[BIT4] TO WR[0]              ;
U 16B6, C74A,15 ;6051            BIS LS[BIT5] TO WR[0]              ;
U 16B7, BE12,15 ;6052            MOV WR[0] TO LS[T9]                ; Код идентификации = 110000(B)
U 16B8, 3642,15 ;6053            MOV LS[BIT1] TO WR[0]              ;
U 16B9, 3E16,15 ;6054            MOV WR[0] TO LS[T11]               ; значение BUS IRQ = 5
                ;6055    LOOP.T6.10:
U 16BA, 3648,15 ;6056            MOV LS[#10] TO WR[0]               ;
U 16BB, BE82,15 ;6057            MOV WR[0] TO LS[ERROR.NUMBER]      ; номер ошибки = 10
U 16BC, 896F,5C ;6058            JSR [SUB2.T6]                      ;
U 16BD, 896B,A4 ;6059            JMP [LOOP.T6.10]                   ; зацикливание при ошибке, если разрешено
U 16BE, 3611,15 ;6060            MOV LS[T8] TO WR[2]                ; WR2 = 1F 00 00
U 16BF, B611,95 ;6061            MOV LS[T8] TO WR[3]                ;
U 16C0, 4565,95 ;6062            BIC LS[BIT18] TO WR[3]             ;
U 16C1, C567,95 ;6063            BIC LS[BIT19] TO WR[3]             ; WR3 = 13 00 00
U 16C2, 3648,15 ;6064            MOV LS[BIT4] TO WR[0]              ;
U 16C3, C746,15 ;6065            BIS LS[BIT3] TO WR[0]              ;
U 16C4, C74A,15 ;6066            BIS LS[BIT5] TO WR[0]              ;
U 16C5, BE12,15 ;6067            MOV WR[0] TO LS[T9]                ; код идентификации = 111000(B)
U 16C6, B640,15 ;6068            MOV LS[BIT0] TO WR[0]              ;
U 16C7, 3E16,15 ,6069            MOV WR[0] TO LS[T11]               ; значение BUS IRQ = 4
                ;6070    LOOP.T6.13:
U 16C8, 3648,15 ;6071            MOV LS[#10] TO WR[0]               ;
U 16C9, C0C0,15 ;6072            ADD LS[#3(H)] TO WR[0]             ;
U 16CA, BE82,15 ;6073            MOV WR[0] TO LS[ERROR.NUMBER]      ; номер ошибки = 13
U 16CB, 896F,5C ;6074            JSR [SUB2.T6]                      ;
U 16CC, 896C,84 ;6075            JMP [LOOP.T6.13]                   ; зацикливание при ошибке, если разрешено
U 16CD, 8972,14 ;6076            JMP [END.T6]                       ; все выполнено
                ;6077    SUB.T6:
                ;6078    SUBT6.1:
U 16CE, 8870,FC ;6079            JSR [ASSERT.OTHER]                 ; будет использоваться поле "ДРУГИЕ ДАННЫЕ"
U 16CF, 0870,8C ;6080            JSR [ASSERT.NA]                    ;
U 16D0, 3E89,15 ;6081            MOV WR[2] TO LS[ADDRESS.DATA]      ; WR2 будет показан в поле "ДРУГИЕ ДАННЫЕ"
U 16D1, 3FFF,15 ;6082            MOV WR[2] TO LS[PSL.HW]            ; загрузка IPL из WR2
                ;6083            XOR WR[2] WITH WR[3] TO Q,         ;
```

```
U 16D2, AB85,B5 ;6084            DT(LONG)&SET.ALU.CC                    ; IPL понижен до разрешения прерывания?
U 16D3, 896D,D9 ;6085            JMP.IF[EQL] TO [SUBT6.2]               ; переход, если да
U 16D4, DB00,07 ;6086            SKIP.IF[NO.INTERRUPT]                  ; проверка пропуском
U 16D5, 096D,74 ;6087            JMP [SUBT6.ERROR.1]                    ; сообщение об ошибке - имеется прерывание
U 16D6, 096D,B4 ;6088            JMP [SUBT6.NOERROR.1]                  ;
                ;6089    SUBT6.ERROR.1:
U 16D7, 2FB0,15 ;6090            CLR WR[0]                              ;
U 16D8, 369E,95 ;6091            MOV LS[ONES] TO WR[1]                  ; установка ошибочных данных для
                ;6092                                                   ; индикации ошибки
U 16D9, 0869,3C ;6093            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 16DA, 5B00,14 ;6094            RETURN                                 ; зацикливание при ошибке, если разрешено
                ,6095    SUBT6.NOERROR.1:
U 16DB, C161,15 ;6096            SUB LS[#10000] FROM WR[2]              ; WR2 = WR2-10000 (следующее пониженное значение IPL)
U 16DC, 896C,E4 ;6097            JMP [SUBT6.1]                          ; возврат и попытка исполнения при следующем IPL
                ;6098    SUBT6.2:
U 16DD, 8870,0C ;6099            JSR [INC.EN]                           ; увеличение номера ошибки
U 16DE, 0870,8C ;6100            JSR [ASSERT.NA]                        ;
U 16DF, DB00,07 ;6101            SKIP.IF[NO.INTERRUPT]                  ; проверка на отсутствие пропуска
U 16E0, 896E,54 ;6102            JMP [SUBT6.3]                          ; обход сообщения об ошибке
U 16E1, 2FB0,15 ;6103            CLR WR[0]                              ;
U 16E2, 369E,95 ;6104            MOV LS[ONES] TO WR[1]                  ; установка ошибочных данных для
                ;6105                                                   ; индикации ошибки
U 16E3, 0869,3C ;6106            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 16E4, 5B00,14 ;6107            RETURN                                 ; зацикливание при ошибке, если разрешено
                ;6108    SUBT6.3:
U 16E5, 8870,0C ;6109            JSR [INC.EN]                           ; увеличение номера ошибки
U 16E6, 0870,8C ;6110            JSR [DEASSERT.NA]                      ;
U 16E7, 3E89,15 ;6111            MOV WR[2] TO LS[ADDRESS.DATA]          ;
U 16E8, 36FC,15 ;6112            MOV LS[INTERRUPT.VEC] TO WR[0]         ; чтение кода идентификации
U 16E9, B612,95 ;6113            MOV LS[T9] TO WR[1]                    ; выборка ожидаемых данных
U 16EA, 0869,3C ;6114            JSR [CHECK.RESULT]                     ; проверка правильности кода идентификации
U 16EB, 5B00,14 ,6115            RETURN                                 ; цикл при ошибке, если разрешено
                ,6116            TST WR[2],                             ;
U 16EC, 2005,35 ;6117            DT(LONG)&SET.ALU.CC                    ; WR2 = 0?
U 16ED, 096F,49 ;6118            JMP.IF[EQL] TO [SUBT6.END]             ; переход, если да
U 16EE, C161,15 ;6119            SUB LS[#10000] FROM WR[2]              ; WR2 = WR2 - 10000
U 16EF, 3E89,15 ;6120            MOV WR[2] TO LS[ADDRESS.DATA]          ; WR2 будет показан в поле "ДРУГИЕ ДАННЫЕ"
U 16F0, 3FFF,15 ;6121            MOV WR[2] TO LS[PSL.HW]                ; загрузка в PSL битов IPL
U 16F1, 0870,4C ;6122            JSR [DEC.EN]                           ;
U 16F2, 0870,4C ;6123            JSR [DEC.EN]                           ;
U 16F3, 096D,D4 ;6124            JMP [SUBT6.2]                          ; переход на проверку с пониженным значением IPL
                ;6125    SUBT6.END:
U 16F4, DB00,16 ,6126            RETURN+1
                ,6127    SUB2.T6:
                ,6128    SUB2T6.1:
U 16F5, 0870,8C ;6129            JSR [ASSERT.NA]                        ;
U 16F6, 3E89,15 ;6130            MOV WR[2] TO LS[ADDRESS.DATA]          ; WR2 будет показан в поле "ДРУГИЕ ДАННЫЕ"
U 16F7, 3FFF,15 ;6131            MOV WR[2] TO LS[PSL.HW]                ; загрузка IPL из WR2
U 16F8, B616,15 ,6132            MOV LS[T11] TO WR[0]                   ;
U 16F9, 3EF8,15 ;6133            MOV WR[0] TO LS[OS]                    ;
                ,6134            XOR WR[2] WITH WR[3] TO Q,             ;
U 16FA, AB85,B5 ;6135            DT(LONG)&SET.ALU.CC                    ; IPL понижен до разрешения прерывания?
U 16FB, 8970,79 ;6136            JMP.IF[EQL] TO [SUB2T6.2]              ; переход, если да
U 16FC, 10F8,15 ;6137            MISC2 [READ.ACC.UPC]                   ; загрузка BUS IRQ из регистра OS
U 16FD, DB00,15 ;6138            NOP                                    ;
```

```
U 16FE, D800,07 ;6139            SKIP.IF[NO.INTERRUPT]                  ; проверка на пропуск
U 16FF, 0970,14 ;6140            JMP [SUB2T6.ERROR.1]                   ; сообщение об ошибке - имеется прерывание
U 1700, 8970,54 ;6141            JMP [SUB2T6.NOERROR.1]                 ;
                ;6142    SUB2T6.ERROR.1:
U 1701, 2F80,15 ;6143            CLR WR[0]                              ;
U 1702, 369E,95 ;6144            MOV LS[ONES] TO WR[1]                  ; установка ошибочных данных для
                ;6145                                                   ; индикации ошибки
U 1703, 0869,3C ;6146            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 1704, 5B00,14 ;6147            RETURN                                 ; зацикливание при ошибке, если разрешено
                ;6148    SUB2T6.NOERROR.1:
U 1705, C161,15 ;6149            SUB LS[#10000] FROM WR[2]              ; WR2 = WR2 - 10000 (следующее пониженное значение IPL)
U 1706, 096F,54 ;6150            JMP [SUB2T6.1]                         ; возврат и попытка исполнения при следующем IPL
                ;6151    SUB2T6.2:
U 1707, 8870,0C ;6152            JSR [INC.EN]                           ; увеличение номера ошибки
U 1708, 0870,8C ;6153            JSR [ASSERT.NA]                        ;
U 1709, 10FE,15 ;6154            MISC2 [READ.ACC.UPC]                   ; загрузка BUS IRQ из регистра OS
U 170A, D800,15 ;6155            NOP                                    ;
U 170B, 36F0,95 ;6156            MOV LS[INTERRUPT.VEC] TO WR[3]         ; чтение кода идентификации
U 170C, D800,07 ;6157            SKIP.IF[NO.INTERRUPT]                  ; проверка на отсутствие пропуска
U 170D, 8971,24 ;6158            JMP [SUB2T6.NOERROR.2]                 ; обход сообщения об ошибке
U 170E, 2F80,15 ;6159            CLR WR[0]                              ;
U 170F, 369E,95 ;6160            MOV LS[ONES] TO WR[1]                  ; установка ошибочных данных для
                ;6161                                                   ; индикации ошибки
U 1710, 0869,3C ;6162            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 1711, 5B00,14 ;6163            RETURN                                 ; зацикливание при ошибке, если разрешено
                ;6164    SUB2T6.NOERROR.2:
U 1712, 8870,0C ;6165            JSR [INC.EN]                           ; увеличение номера ошибки
                ;6166    SUB2T6.3:
U 1713, 0870,8C ;6167            JSR [DEASSERT.NA]                      ;
U 1714, A006,15 ;6168            MOV WR[3] TO WR[0]                     ; чтение кода идентификации
U 1715, B612,95 ;6169            MOV LS[T9] TO WR[1]                    ; выборка ожидаемых данных
U 1716, 0869,3C ;6170            JSR [CHECK.RESULT]                     ; проверка правильности кода идентификации
U 1717, 5B00,14 ;6171            RETURN                                 ; зацикливание при ошибке, если разрешено
                ;6172            TST WR[2],                             ;
U 1718, 2005,35 ;6173            DT(LONG)&SET.ALU.CC                    ; WR2 = 0?
U 1719, 8972,09 ;6174            JMP.IF[EQL] TO [SUB2T6.END]            ; переход, если да
U 171A, C161,15 ;6175            SUB LS[#10000] FROM WR[2]              ; WR2 = WR2 - 10000
U 171B, 3E89,15 ;6176            MOV WR[2] TO LS[ADDRESS.DATA]          ; WR2 будет показан в поле "ДРУГИЕ ДАННЫЕ"
U 171C, 3FFF,15 ;6177            MOV WR[2] TO LS[PSL.HW]                ; загрузка в PSL битов IPL
U 171D, 0870,4C ;6178            JSR [DEC.EN]                           ;
U 171E, 0870,4C ;6179            JSR [DEC.EN]                           ;
U 171F, 0970,74 ;6180            JMP [SUB2T6.2]                         ; переход на проверку при пониженном IPL
                ;6181    SUB2T6.END:
U 1720, D800,16 ;6182            RETURN+1                               ;
                ;6183    END.T6
```

```
;6184      .PAGE "ТЕСТ 7 - тест микросхемы декодирования приоритета (модуль DAP)"
;6185      ;
;6186      ; ОПИСАНИЕ ТЕСТА:
;6187      ;
;6188      ;     Этот тест проверяет микросхему декодирования приоритета, выходы которой по-
;6189      ;   ступают на ПМЛ УПР.ПРЕРЫВАНИЕМ. Микросхема декодирования приоритета проверя-
;6190      ;   ется возбуждением всех сигналов прерывания и контролем того, что проходит
;6191      ;   сигнал наивысшего приоритета путем проверки кода идентификации. Затем пре-
;6192      ;   рывание наивысшего приоритета запрещается, и тест повторяется до тех пор, по-
;6193      ;   ка остается только сигнал прерывания наинизшего приоритета. IPL во время
;6194      ;   этого теста устанавливается на 0. Прерывание наинизшего приоритета (BR4) от-
;6195      ;   дельно не проверяется, так как оно уже проверялось в предыдущем тесте.
;6196      ;
;6197      ; ПРЕДПОЛОЖЕНИЯ:
;6198      ;
;6199      ;   Предполагается, что предыдущие тесты прерываний выполнены успешно.
;6200      ;
;6201      ; ШАГИ ТЕСТА:
;6202      ;
;6203      ;   1) Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;6204      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной па-
 6205      ;      мяти.
 6206      ;   2) Выдача сигнала CPU ATTN со словом управления, установленным для генерации
 6207      ;      сигналов PWR FAIL INT, INTRVL TIM INT и CONS ATTN.
 6208      ;   3) Установка IPL на 0.
 6209      ;   4) Загрузка в WR1 кода идентификации прерывания PWR FAIL (000(B)).
 6210      ;   5) Загрузка OS 3-0 единицами и выполнение инструкции MISC для загрузки
 6211      ;      BUS IRQ 7-4 единицами.
 6212      ;   6) Выполнение двух инструкций NOP для загрузки буфера IRQ и ПМЛ УПР.ПРЕРЫ-
 6213      ;      ВАНИЕМ.
 6214      ;   7) Выполнение MOVE с LS=7E в WR0 для загрузки кода идентификации.
 6215      ;   8) Выдача сигнала CPU ATTN со словом управления для снятия PWR FAIL INT
 6216      ;      при оставленных двух других прерываниях.
 6217      ;   9) Загрузка в WR1 кода идентификации прерывания INTRVL TIM INT (001(B))
 6218      ;      и повторение шагов с 5 по 7.
 6219      ;  10) Выдача сигнала CPU ATTN со словом управления для снятия INTRVL TIM INT
 6220      ;      при оставленных других прерываниях.
 6221      ;  11) Загрузка в WR1 кода идентификации прерывания BR7 (010(B)) и повторение
;6222      ;      шагов с 5 по 7.
 6223      ;  12) Загрузка в WR1 кода идентификации прерывания BR6 (011(B)).
 6224      ;  13) Загрузка OS 3-0 данными 0111(B) и выполнение инструкции MISC для загрузки
 6225      ;      BUS IRQ 7-4 данными 0111(B). Повторение шагов 6 и 7.
;6226      ;  14) Загрузка в WR1 кода идентификации прерывания BR5 (100(B)).
;6227      ;  15) Загрузка OS 3-0 данными 0011(B) и выполнение инструкции MISC для загрузки
 6228      ;      BUS IRQ 7-4 данными 0011(B). повторение шагов 6 и 7.
;6229      ;  16) Загрузка в WR1 кода идентификации прерывания CONS ATTN (101(B)).
;6230      ;  17) Загрузка OS 3-0 данными 0001(B) и выполнение инструкции MISC для загрузки
;6231      ;      BUS IRQ 7-4 данными 0001(B). Повторение шагов 6 и 7.
;6232      ;
;6233      ; ОШИБКИ:
;6234      ;
;6235      ;   ошибка 1 - PWR FAIL INT не запрещает прерываний более низкого уровня.
;6236      ;   ошибка 2 - INTRVL TIM INT не запрещает прерываний более низкого уровня.
;6237      ;   ошибка 3 - BR7 (BUS IRQ 7) не запрещает прерываний более низкого уровня.
;6238      ;   ошибка 4 - BR6 (BUS IRQ 6) не запрещает прерываний более низкого уровня.
```

```
                ;6239       ;   ошибка 5 - BR5 (BUS IRQ 5) не запрещает прерываний более низкого уровня.
                ;6240       ;   ошибка 6 - CONS ATTN не запрещает прерывания более низкого уровня.
                ;6241       ;
                ;6242       ; НАЛАДКА:
                ;6243       ;
                ;6244       ;   ПРИМЕЧАНИЕ: Если имеются какие-либо устройства на общей шине, возможно, что
                ;6245       ;   они могут быть причиной вызова прерывания BR на общей шине. Любые устройства,
                ;6246       ;   имеющиеся на общей шине, должны быть отключены, если подозревается, что они
                ;6247       ;   вызывают неуспешное выполнение этого теста.
                ;6248       ;
                ;6249       ;   ОШИБКИ С 1 ПО 6 - Наиболее вероятной причиной всех этих ошибок является ми-
                ;6250       ;   кросхема декодирования приоритета. Двоичное значение на выходных ножках 6,
                ;6251       ;   7 и 9 этого элемента должно быть равно наивысшему двоичному значению входа.
                ;6252       ;   Например, для ошибки 1, выход должен быть 7, для ошибки 2, выход должен
                ;6253       ;   быть 6 и т.д.
                ;6254       ;
                ;6255       T.7:
U 1721, B65E,15 ;6256           MOV LS[BEGIN.TEST] TO WR[0]             ; установка бита 15 в WR0 для слова управления и
                ;6257                                                   ; состояния
U 1722, 3E80,15 ;6258           MOV WR[0] TO LS[CONTROL.STATUS]         ; установка бита 15 в слове управления и состояния. Бит
                ;6259                                                   ; 15 указывает начало теста для консольного процессора
U 1723, 10E0,15 ;6260           MISC [SET.CP.ATTN]                      ; выдача CPU ATTN для консольного процессора
                ;6261       WAIT.T7.0:
U 1724, B972,44 ;6262           JMP [WAIT.T7.0]                         ; зацикливание для ожидания ответа консольного
                ;6263                                                   ; процессора
U 1725, 5F44,15 ;6264           MCOM LS[BIT2] TO WR[0]                  ;
U 1726, 4546,15 ;6265           BIC LS[BIT3] TO WR[0]                   ;
U 1727, C548,15 ;6266           BIC LS[BIT4] TO WR[0]                   ;
U 1728, 454A,15 ;6267           BIC LS[BIT5] TO WR[0]                   ;
U 1729, 3E8A,15 ;6268           MOV WR[0] TO LS[ERROR.MASK]             ; маска ошибки = FFFFFFD3
U 172A, 65A0,15 ;6269           CLR LS[PREVIOUS.ERROR]                  ; очистка предыдущего номера ошибки
U 172B, B640,15 ;6270           MOV LS[#1] TO WR[0]                     ; установка 1 в WR0
U 172C, BE82,15 ;6271           MOV WR[0] TO LS[ERROR.NUMBER]           ; установка номера ошибки для первой ошибки
U 172D, A3C0,15 ;6272           ROL WR[0]                               ; установка 2 в WR0
U 172E, 3E8C,15 ;6273           MOV WR[0] TO LS[MODULE.NUM]             ; установка кода модуля для модуля DAP
U 172F, 3660,15 ;6274           MOV LS[INTERUPT.EN] TO WR[0]            ;
U 1730, C764,15 ;6275           BIS LS[PWR.FAIL] TO WR[0]               ; бит для генерации PWR FAIL INT
U 1731, 4766,15 ;6276           BIS LS[INTERVAL.TIM] TO WR[0]           ; бит для генерации INTRVL TIM INT
U 1732, C768,15 ;6277           BIS LS[CONSOLE.ATTN] TO WR[0]           ; бит для генерации CONS ATTN
U 1733, 3E80,15 ;6278           MOV WR[0] TO LS[CONTROL.STATUS]         ;
U 1734, 10E0,15 ;6279           MISC [SET.CP.ATTN]                      ;
                ;6280       WAIT.T7.1:
U 1735, B973,54 ;6281           JMP [WAIT.T7.1]                         ; генерация прерываний
U 1736, 2F80,15 ;6282           CLR WR[0]                               ; 0 в WR0
U 1737, BFFE,15 ;6283           MOV WR[0] TO LS[PSL.HW]                 ; установка IPL на 0
                ;6284       LOOP.T7.1:
U 1738, 364A,95 ;6285           MOV LS[BIT5] TO WR[1]                   ; ожидаемый код идентификации = 100000(B)
U 1739, 3648,15 ;6286           MOV LS[#10] TO WR[0]                    ;
U 173A, C140,15 ;6287           SUB LS[#1] FROM WR[0]                   ;
U 173B, 3EF8,15 ;6288           MOV WR[0] TO LS[OS]                     ; OS = F
U 173C, 10F8,15 ;6289           MISC2 [READ.ACC.UPC]                    ; загрузка BUS IRQ 7-4
U 173D, DB00,15 ;6290           NOP                                     ;
U 173E, 36FC,15 ;6291           MOV LS[INTERRUPT.VEC] TO WR[0]          ; чтение кода идентификации
U 173F, 0869,3C ;6292           JSR [CHECK.RESULT]                      ; проверка кода идентификации
U 1740, 0973,84 ;6293           JMP [LOOP.T7.1]                         ;
```

```
U 1741, 8870,0C ;6294          JSR [INC.EN]                          ; увеличение номера ошибки до 2
U 1742, 3660,15 ;6295          MOV LS[INTERUPT.EN] TO WR[0]          ;
U 1743, 4766,15 ;6296          BIS LS[INTERVAL.TIM] TO WR[0]         ; бит для генерации INTRVL TIM INT
U 1744, C768,15 ;6297          BIS LS[CONSOLE.ATTN] TO WR[0]         ; бит для генерации CONS ATTN
U 1745, 3E80,15 ;6298          MOV WR[0] TO LS[CONTROL.STATUS]       ;
U 1746, 10E0,15 ;6299          MISC [SET.CP.ATTN]                    ;
                ;6300    WAIT.T7.2:
U 1747, 8974,74 ;6301          JMP [WAIT.T7.2]                       ; генерация прерываний
                ;6302    LOOP.T7.2:
U 1748, 364A,95 ;6303          MOV LS[BIT5] TO WR[1]                 ;
U 1749, C744,95 ;6304          BIS LS[BIT2] TO WR[1]                 ; ожидаемый код идентификации = 100100(B)
U 174A, 3648,15 ;6305          MOV LS[#10] TO WR[0]                  ;
U 174B, C140,15 ;6306          SUB LS[#1] FROM WR[0]                 ;
U 174C, 3EF8,15 ;6307          MOV WR[0] TO LS[OS]                   ; OS = F
U 174D, 10F8,15 ;6308          MISC2 [READ.ACC.UPC]                  ; загрузка BUS IRQ 7-4
U 174E, DB00,15 ;6309          NOP                                   ;
U 174F, 36FC,15 ;6310          MOV LS[INTERRUPT.VEC] TO WR[0]        ; чтение кода идентификации
U 1750, 0869,3C ;6311          JSR [CHECK.RESULT]                    ; проверка кода идентификации
U 1751, 8974,84 ;6312          JMP [LOOP.T7.2]                       ;
U 1752, 8870,0C ;6313          JSR [INC.EN]                          ; увеличение номера ошибки до 3
U 1753, 3660,15 ;6314          MOV LS[INTERUPT.EN] TO WR[0]          ;
U 1754, C768,15 ;6315          BIS LS[CONSOLE.ATTN] TO WR[0]         ; бит для генерации CONS ATTN
U 1755, 3E80,15 ;6316          MOV WR[0] TO LS[CONTROL.STATUS]       ;
U 1756, 10E0,15 ;6317          MISC [SET.CP.ATTN]                    ;
                ;6318    WAIT.T7.3:
U 1757, 0975,74 ;6319          JMP [WAIT.T7.3]                       ; генерация прерываний
                ;6320    LOOP.T7.3:
U 1758, 3646,95 ;6321          MOV LS[BIT3] TO WR[1]                 ;
U 1759, 474A,95 ;6322          BIS LS[BIT5] TO WR[1]                 ; ожидаемый код идентификации = 101000(B)
U 175A, 3648,15 ;6323          MOV LS[#10] TO WR[0]                  ;
U 175B, C140,15 ;6324          SUB LS[#1] FROM WR[0]                 ;
U 175C, 3EF8,15 ;6325          MOV WR[0] TO LS[OS]                   ; OS = F
U 175D, 10F8,15 ;6326          MISC2 [READ.ACC.UPC]                  ; загрузка BUS IRQ 7-4
U 175E, DB00,15 ;6327          NOP                                   ;
U 175F, 36FC,15 ;6328          MOV LS[INTERRUPT.VEC] TO WR[0]        ; чтение кода идентификации
U 1760, 0869,3C ;6329          JSR [CHECK.RESULT]                    ; проверка кода идентификации
U 1761, 0975,84 ;6330          JMP [LOOP.T7.3]                       ;
U 1762, 8870,0C ;6331          JSR [INC.EN]                          ; увеличение номера ошибки до 4
                ;6332    LOOP.T7.4:
U 1763, 3646,95 ;6333          MOV LS[BIT3] TO WR[1]                 ;
U 1764, 474A,95 ;6334          BIS LS[BIT5] TO WR[1]                 ;
U 1765, C744,95 ;6335          BIS LS[BIT2] TO WR[1]                 ; ожидаемый код идентификации = 101100(B)
U 1766, 8646,15 ;6336          MOV LS[#8] TO WR[0]                   ;
U 1767, C140,15 ;6337          SUB LS[#1] FROM WR[0]                 ;
U 1768, 3EF8,15 ;6338          MOV WR[0] TO LS[OS]                   ; OS = 7
U 1769, 10F8,15 ;6339          MISC2 [READ.ACC.UPC]                  ; загрузка BUS IRQ 7-4
U 176A, DB00,15 ;6340          NOP                                   ;
U 176B, 36FC,15 ;6341          MOV LS[INTERRUPT.VEC] TO WR[0]        ; чтение кода идентификации
U 176C, 0869,3C ;6342          JSR [CHECK.RESULT]                    ; проверка кода идентификации
U 176D, 8976,34 ;6343          JMP [LOOP.T7.4]                       ;
U 176E, 8870,0C ;6344          JSR [INC.EN]                          ; увеличение номера ошибки до 5
                ;6345    LOOP.T7.5:
U 176F, 8648,95 ;6346          MOV LS[BIT4] TO WR[1]                 ;
U 1770, 474A,95 ;6347          BIS LS[BIT5] TO WR[1]                 ; ожидаемый код идентификации = 110000(B)
U 1771, 36C0,15 ;6348          MOV LS[#3(H)] TO WR[0]                ;
```

```
U 1772,  3EF8,15 ;6349            MOV WR[0] TO LS[OS]                    ; OS = 3
U 1773,  10F8,15 ;6350            MISC2 [READ.ACC.UPC]                   ; загрузка BUS IRQ 7-4
U 1774,  DB00,15 ;6351            NOP                                    ;
U 1775,  36FC,15 ;6352            MOV LS[INTERRUPT.VEC] TO WR[0]         ; чтение кода идентификации
U 1776,  0869,3C ;6353            JSR [CHECK.RESULT]                     ; проверка кода идентификации
U 1777,  8976,F4 ;6354            JMP [LOOP.T7.5]                        ;
U 1778,  8870,0C ;6355            JSR [INC.EN]                           ; увеличение номера ошибки до 6
                 ;6356    LOOP.T7.6:
U 1779,  8648,95 ;6357            MOV LS[BIT4] TO WR[1]                  ;
U 177A,  474A,95 ;6358            BIS LS[BIT5] TO WR[1]                  ;
U 177B,  C744,95 ;6359            BIS LS[BIT2] TO WR[1]                  ; ожидаемый код идентификации = 110100(B)
U 177C,  8640,15 ;6360            MOV LS[#1] TO WR[0]                    ;
U 177D,  3EF8,15 ;6361            MOV WR[0] TO LS[OS]                    ; OS = 1
U 177E,  10F8,15 ;6362            MISC2 [READ.ACC.UPC]                   ; загрузка BUS IRQ 7-4
U 177F,  DB00,15 ;6363            NOP                                    ;
U 1780,  36FC,15 ;6364            MOV LS[INTERRUPT.VEC] TO WR[0]         ; чтение кода идентификации
U 1781,  0869,3C ;6365            JSR [CHECK.RESULT]                     ; проверка кода идентификации
U 1782,  0977,94 ;6366            JMP [LOOP.T7.6]                        ;
U 1783,  3660,15 ;6367            MOV LS[INTERUPT.EN] TO WR[0]           ;
U 1784,  3E80,15 ;6368            MOV WR[0] TO LS[CONTROL.STATUS]        ; слово управления и состояния = бит 16 установлен
U 1785,  10E0,15 ;6369            MISC [SET.CP.ATTN]                     ; вызов консольного процессора
                 ;6370    WAIT.T7.CLRINT:
U 1786,  0978,64 ;6371            JMP [WAIT.T7.CLRINT]                   ; сброс всех прерываний
                 ;6372    END.T7:
```

```
;6373     .PAGE "ТЕСТ 8 - тест прерывания по биту T и маскирования (модуль DAP)"
;6374     ;
;6375     ; ОПИСАНИЕ ТЕСТА:
;6376     ;
;6377     ;       Тест прерывания по биту T состоит из разрешения прерывания по биту T
;6378     ;    и проверки формирования правильного кода идентификации при всех единицах
;6379     ;    в IPL, пока нет других ожидающих прерываний. Также проверяется пропуск,
;6380     ;    если нет прерывания. Прерывание по биту T формируется установкой бита 4
;6381     ;    или бита 30 и загрузкой буфера PSL (буфер также выдает биты IPL и биты
;6382     ;    текущего режима). Проверяются все четыре комбинации битов 30 и 4.
;6383     ;       Затем, при IPL равном нулю, одновременно задаются прерывания BR4, CONS
;6384     ;    ATTN, CONS HALT и BR6 и проверяется код идентификации, чтобы убедиться, что
;6385     ;    эти прерывания подавляют прерывание по биту T. Тест маскирования состоит из
;6386     ;    установки значения 1F (H) в IPL и разрешения обоих сигналов прерывания по
;6387     ;    биту T и CONS HALT (IPL не запрещает этих прерываний). Тогда выполняется
;6388     ;    инструкция MISC, которая устанавливает на ножке 9 дешифратора сигнал маски-
;6389     ;    рования прерываний HALT и по биту T, поступающий на входную ножку 1B ПМЛ
;6390     ;    УПР.ПРЕРЫВАНИЕМ. Выполняется инструкция NOP с пропуском, если нет прерыва-
;6391     ;    ния. Эта инструкция должна выполнить пропуск, так как прерывание запрещает-
;6392     ;    ся маскированием для этого цикла. Затем выполняется другая инструкция NOP с
;6393     ;    пропуском, если нет прерывания. На этот раз пропуск не должен выполнятся,
;6394     ;    так как маскирование действует только один цикл.
;6395     ;
;6396     ; ПРЕДПОЛОЖЕНИЯ:
;6397     ;
;6398     ;    Предполагается, что все предыдущие тесты прерываний выполнены успешно.
;6399     ;
;6400     ; ШАГИ ТЕСТА:
;6401     ;
;6402     ;    1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;6403     ;       (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;6404     ;       памяти.
;6405          2. Загрузка 40 1F 00 10(H) в WR0 и выполнение инструкции MOVE из WR0 в LS FF.
;6406     ;       Этим загружается IPL значением 1F, а также устанавливается бит T (биты 30
;6407     ;       и 4 шины Y установлены).
;6408     ;    3. Загрузка 1111(B) в биты WR1 с 5 по 2 (код идентификации).
;6409     ;    4. Выполнение MOVE в WR0 из LS 7E(H) (регистр состояния) и проверка кода
;6410     ;       идентификации.
;6411     ;    5. Выполнение инструкции NOP с пропуском, если нет прерывания и проверка на
;6412     ;       отсутствие пропуска.
;6413     ;    6. Загрузка 00 1F 00 10(H) в WR0 и выполнение MOVE из WR0 в LS FF. Этим заг-
;6414     ;       ружается IPL значением 1F, а также устанавливается бит T (бит 4 шины Y
;6415     ;       установлен).
;6416     ;    7. Загрузка 1111(B) в WR1 с 5 по 2 (код идентификации) и повторение шага 4.
;6417     ;    8. Загрузка 40 1F 00 00(H) в WR0 и выполнение MOVE из WR0 в LS FF. Этим заг-
;6418     ;       ружается IPL значением 1F, а также устанавливается бит TP (бит 30 шины
;6419     ;       Y установлен).
;6420     ;    9. Загрузка 1111(B) в биты WR1 с 5 до 2 (код идентификации) и повторение
;6421     ;       шага 4.
;6422     ;    10. Загрузка 00 1F 00 00(H) в WR0 и выполнение MOVE из WR0 в LS FF. Этим
;6423     ;        загружается IPL значением 1F, но прерывание по биту T не устанавлива-
;6424     ;        ется (биты шины Y 30 и 4 сброшены).
;6425     ;    11. Загрузка 1111(B) в биты WR1 с 5 по 2 (код идентификации при отсутствии
;6426     ;        прерывания) и повторение шага 4.
;6427     ;    12. Загрузка IPL нулями и разрешение прерывания по биту T. Затем возбуждение
```

```
;6428      ;        BR4 посредством регистра OS.
;6429      ;    13. Загрузка 1110 в WR1 (код идентификации для BR4) и повторение шага 4.
;6430      ;    14. Возбуждение BR6; загрузка 0111 в WR1 (код идентификации для BR6) и
;6431      ;        повторение шага 4.
;6432      ;    15. Выполнение CPU ATTN с установленными битами 16 и 20 в слове управления
;6433      ;        и состояния для разрешения CONS ATTN.
;6434      ;    16. Загрузка 1101 в WR1 (код идентификации для CONS ATTN), выполнение NOP
;6435      ;        и повторение шага 4.
;6436      ;    17. Выполнение CPU ATTN с установленными битами 16 и 17 в слове управления
;6437      ;        и состояния для разрешения CONS HALT.
;6438      ;    18. Загрузка 0111 в WR1 (код идентификации для CONS HALT), выполнение NOP
;6439      ;        и повторение шага 4.
;6440      ;    19. Выполнение CPU ATTN с установленными битами 16 и 17 в слове управления
;6441      ;        и состояния для разрешения CONS HALT.
;6442      ;    20. Загрузка всех единиц в WR1 для получения ошибки, если должна вызываться
;6443      ;        подпрограмма проверки результата. Загрузка 40 1F 00 00(H) в WR0 и выпол-
;6444      ;        нение MOVE из WR0 в LS FF. Этим загружается IPL значением 1F, а также ус-
;6445          танавливается прерывание по биту (бит 30 шины Y установлен).
;6446          21. Выполнение инструкции MISC для маскирования прерываний по биту T и HALT.
;6447          22. Выполнение инструкции NOP с пропуском, если нет прерывания. Инструкция
;6448          должна выполнить пропуск.
;6449          23. Выполнение другой инструкции NOP с пропуском, если нет прерывания. На
;6450          этот раз инструкция не должна выполнять пропуска.
;6451          24. Выполнение инструкции BIS LS[6] TO WR[0]. Этим на входе дешифратора
;6452      ;        устанавливаются те же биты, как и при инструкции MISC, за исключением
;6453      ;        того, что сигнал MISC INSTR н должен быть низким.
;6454      ;    25. Выполнение другой инструкции NOP с пропуском, если нет прерываний. Опять
;6455      ;        инструкция не должна выполнять пропуска.
;6456      ;
;6457      ; ОШИБКИ:
;6458      ;
;6459      ;    ошибка 1 - прерывание по биту T формирует некорректный код идентификации
;6460      ;               при установленных битах шины Y 30 и 4.
;6461      ;    ошибка 2 - бит T не вызывает прерывания.
;6462      ;    ошибка 3 - прерывание по биту T формирует некорректный код идентификации
;6463      ;               при установленном бите 4 на шине Y .
;6464      ;    ошибка 4 - прерывание по биту T формирует некорректный код идентификации
;6465      ;               при установленном бите 30 на шине Y.
;6466      ;    ошибка 5 - прерывание по биту T не запрещается при сброшенных битах 30 и 4
;6467      ;               на шине Y.
;6468      ;    ошибка 6 - бит T не вызывает прерывания.
;6469      ;    ошибка 7 - прерывание по биту T не подавлено прерыванием BR4.
;6470      ;    ошибка 8 - прерывание по биту T не подавлено прерыванием BR6.
;6471      ;    ошибка 9 - прерывание по биту T не подавлено прерыванием CONS ATTN.
;6472      ;    ошибка A - прерывание по биту T не подавлено прерыванием CONS HALT.
;6473      ;    ошибка B - функция MASK HALT AND T-BIT не исполнила маскирования прерывания.
;6474      ;    ошибка C - функция MASK HALT AND T-BIT не снимается в следующем цикле.
;6475      ;    ошибка D - функция MASK HALT AND T-BIT выполняется для инструкции, отличной
;6476      ;               от MISC.
;6477
;6478      ; НАЛАДКА:
;6479      ;
;6480      ;    ОШИБКА 1 - Эта ошибка показывает, что или прерывание по биту T не было выс-
;6481      ;    тавлено или неисправна ПМЛ УПР.ПРЕРЫВАНИЕМ. Прежде всего необходимо прове-
;6482      ;    верить сигнал T-TRAP на входе ПМЛ УПР.ПРЕРЫВАНИЕМ во время инструкции MOVE,
```

```
                    ;6483      ;    которая загружает код идентификации в WR0. Сигнал должен быть высоким. Если
                    ;6484      ;    этот вход правильный, по-видимому, неисправна ПМЛ. Если этот сигнал низкий,
                    ;6485      ;    необходимо проверить непосредственно выход из буфера PSL, который выдает этот
                    ;6486      ;    сигнал. Если этот сигнал неправильный, необходимо проверить вход T.OR.TP H во
                    ;6487      ;    время инструкции, которая загружает буфер (MOVE в LS FF). Вход должен иметь
                    ;6488      ;    высокий уровень во время этой инструкции. Если так, по-видимому, буфер неис-
                    ;6489      ;    правен. Если T.OR.TP H низкий, необходимо проверить входы ПМЛ СДВИГ.ДАННЫХ
                    ;6490      ;    BUS Y D 30 H и BUS Y D 04 H (должны быть высокие уровни). Если один из них
                    ;6491      ;    имеет высокий уровень, на выходе T.OR.TP H должен быть также высокий уровень.
                    ;6492      ;    Если нет, подозревается ПМЛ СДВИГ.ДАННЫХ.
                    ;6493      ;
                    ;6494      ;    ОШИБКА 2 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6495      ;
                    ;6496      ;    ОШИБКА 3 - Подозревается ПМЛ СДВИГ.ДАННЫХ или вход BUS Y D 04 H. Этот вход
                    ;6497      ;    должен иметь высокий уровень во время инструкции, которая загружает буфер
                    ;6498      ;    PSL. Если вход правильный, по-видимому, неисправна ПМЛ СДВИГ.ДАННЫХ.
                    ;6499      ;
                    ;6500      ;    ОШИБКА 4 - Подозревается ПМЛ СДВИГ.ДАННЫХ или вход BUS Y D 30 H. Этот вход
                    ;6501      ;    должен иметь высокий уровень во время инструкции, которая загружает буфер
                    ;6502      ;    PSL. Если вход правильный, по-видимому, неисправна ПМЛ СДВИГ.ДАННЫХ.
                    ;6503      ;
                    ;6504      ;    ОШИБКА 5 - Подозревается ПМЛ СДВИГ.ДАННЫХ или входы BUS Y D 04 H или BUS Y
                    ;6505      ;    D 30 H. Эти входы должны быть низкими во время инструкции, которая загружа-
                    ;6506      ;    ет буфер PSL. Если эти входы правильные, по-видимому, ПМЛ СДВИГ.ДАННЫХ
                    ;6507      ;    неисправна.
                    ;6508      ;
                    ;6509      ;    ОШИБКА 6 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6510      ;    ОШИБКА 7 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6511      ;    ОШИБКА 8 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6512      ;    ОШИБКА 9 - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6513      ;    ОШИБКА A - Подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ.
                    ;6514
                    ;6515           ОШИБКА B - Необходимо проверить низкий уровень сигнала MASK L на входе ПМЛ
                    ;6516      ;    УПР.ПРЕРЫВАНИЕМ во время инструкции MISC, Которая устанавливает маскирова-
                    ;6517      ;    ние. Если сигнал правильный, подозревается ПМЛ УПР.ПРЕРЫВАНИЕМ. В противном
                    ;6518      ;    случае подозревается дешифратор, который выдает сигнал MASK L.
                    ;6519      ;
                    ;6520      ;    ОШИБКА C - Или сигнал MASK L остается низким, даже если инструкция MISC не
                    ;6521      ;    выполняется, или существует проблема временных соотношений.
                    ;6522      ;
                    ;6523      ;    ОШИБКА D - Необходимо проверить сигнал MISC INSTR H на входе дешифратора,
                    ;6524      ;    который выдает сигнал маскирования прерываний по биту T и HALT. Если сигнал
                    ;6525      ;    низкий во время инструкции BIS LS[6] TO WR[0], по-видимому, дешифратор неис-
                    ;6526      ;    правен.
                    ;6527      ;
                    ;6528      T.8:
U 1787, B65E,15     ;6529           MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;6530                                                ; состояния
U 1788, 3E80,15     ;6531           MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления состояния. Бит 15
                    ;6532                                                ; указывает начало теста для консольного процессора
U 1789, 10E0,15     ;6533           MISC [SET.CP.ATTN]                   ; выдача CPU ATTN для консольного процессора
                    ;6534      WAIT.T8.0:
U 178A, 0978,A4     ;6535           JMP [WAIT.T8.0]                      ; зацикливание для ожидания ответа консольного
                    ;6536                                                ; процессора
U 178B, 3660,15     ;6537           MOV LS[BIT16] TO WR[0]                ; сброс всех прерываний
```

```
;  ENKCB.MIC                ТЕСТ 8 - тест прерывания по биту T и маскирования (модуль DAP)

U 178C, 3E80,15 ;6538              MOV WR[0] TO LS[CONTROL.STATUS]          ;
U 178D, 10E0,15 ;6539              MISC [SET.CP.ATTN]                      ; выдача CPU ATTN для консольного процессора
                ;6540      WAIT.T8.1:
U 178E, 8978,E4 ;6541              JMP [WAIT.T8.1]                         ; зацикливание для ожидания ответа консольного
                ;6542                                                      ; процессора
U 178F, 5F44,15 ;6543              MCOM LS[BIT2] TO WR[0]                  ;
U 1790, 4546,15 ;6544              BIC LS[BIT3] TO WR[0]                   ;
U 1791, C548,15 ;6545              BIC LS[BIT4] TO WR[0]                   ;
U 1792, 454A,15 ;6546              BIC LS[BIT5] TO WR[0]                   ;
U 1793, 3EBA,15 ;6547              MOV WR[0] TO LS[ERROR.MASK]             ; маска ошибки = FFFFFFC3
U 1794, 65A0,15 ;6548              CLR LS[PREVIOUS.ERROR]                  ; очистка предыдущего номера ошибки
U 1795, B640,15 ;6549              MOV LS[#1] TO WR[0]                     ; установка 1 в WR0
U 1796, 3EB2,15 ;6550              MOV WR[0] TO LS[ERROR.NUMBER]           ; установка номера ошибки для первой ошибки
U 1797, A3C0,15 ;6551              ROL WR[0]                               ; установка 2 в WR0
U 1798, 3E8C,15 ;6552              MOV WR[0] TO LS[MODULE.NUM]             ; установка кода модуля для модуля DAP
                ;6553      LOOP.T8.1:
U 1799, B724,15 ;6554              MOV LS[HI.IPL] TO WR[0]                 ;
U 179A, C77C,15 ;6555              BIS LS[BIT30] TO WR[0]                  ;
U 179B, 4748,15 ;6556              BIS LS[BIT4] TO WR[0]                   ; WR0 = 40 1F 00 10
U 179C, BFFE,15 ;6557              MOV WR[0] TO LS[PSL.HW]                 ; загрузка IPL и установка T-TRAP
U 179D, 369E,95 ;6558              MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = 1111(B) в битах 2-5
U 179E, 36FC,15 ;6559              MOV LS[INTERRUPT.VEC] TO WR[0]          ; чтение кода идентификации
U 179F, 0869,3C ;6560              JSR [CHECK.RESULT]                      ; проверка кода идентификации
U 17A0, 8979,94 ;6561              JMP [LOOP.T8.1]                         ; зацикливание при ошибке, если разрешено
U 17A1, 8870,0C ;6562              JSR [INC.EN]                            ; увеличение номера ошибки до 2
U 17A2, 0870,8C ;6563              JSR [ASSERT.NA]                         ; установка N/A (не применимо) для полей ожидаемых и
                ;6564                                                      ; полученных данных
                ;6565      LOOP.T8.2:
U 17A3, DB00,07 ;6566              SKIP.IF[NO.INTERRUPT]                   ; не должно быть пропуска
U 17A4, 897A,94 ;6567              JMP [T8.OK.2]                           ; обход вызова для обработки ошибки
U 17A5, 2F80,15 ;6568              CLR WR[0]                               ;
U 17A6, 369E,95 ;6569              MOV LS[ONES] TO WR[1]                   ; установка неправильных данных
U 17A7, 0869,3C ;6570              JSR [CHECK.RESULT]                      ; сообщение об ошибке
U 17A8, 897A,34 ;6571              JMP [LOOP.T8.2]                         ; зацикливание при ошибке, если разрешено
                ;6572      T8.OK.2:
U 17A9, 8870,0C ;6573              JSR [INC.EN]                            ; увеличение номера ошибки до 3
                ;6574      LOOP.T8.3:
U 17AA, B724,15 ;6575              MOV LS[HI.IPL] TO WR[0]                 ;
U 17AB, 4748,15 ;6576              BIS LS[BIT4] TO WR[0]                   ; WR0 = 00 1F 00 10
U 17AC, BFFE,15 ;6577              MOV WR[0] TO LS[PSL.HW]                 ; загрузка IPL и установка T-TRAP
U 17AD, DB00,07 ;6578              SKIP.IF[NO.INTERRUPT]                   ; не должно быть пропуска
U 17AE, 0978,34 ;6579              JMP [T8.OK.3]                           ; обход вызова для обработки ошибки
U 17AF, 2F80,15 ;6580              CLR WR[0]                               ;
U 17B0, 369E,95 ;6581              MOV LS[ONES] TO WR[1]                   ; установка неправильных данных
U 17B1, 0869,3C ;6582              JSR [CHECK.RESULT]                      ; сообщение об ошибке
U 17B2, 897A,A4 ;6583              JMP [LOOP.T8.3]                         ; зацикливание при ошибке, если разрешено
                ;6584      T8.OK.3:
U 17B3, 8870,0C ;6585              JSR [INC.EN]                            ; увеличение номера ошибки до 4
                ;6586      LOOP.T8.4:
U 17B4, B724,15 ;6587              MOV LS[HI.IPL] TO WR[0]                 ;
U 17B5, C77C,15 ;6588              BIS LS[BIT30] TO WR[0]                  ; WR0 = 40 1F 00 00
U 17B6, BFFE,15 ;6589              MOV WR[0] TO LS[PSL.HW]                 ; загрузка IPL и установка T-TRAP
U 17B7, DB00,07 ;6590              SKIP.IF[NO.INTERRUPT]                   ; не должно быть пропуска
U 17B8, 8978,D4 ;6591              JMP [T8.OK.4]                           ; обход вызова для обработки ошибки
U 17B9, 2F80,15 ;6592              CLR WR[0]                               ;
```

```
U 17BA, 369E,95 ;6593            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 17BB, 0869,3C ;6594            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 17BC, 897B,44 ;6595            JMP [LOOP.T8.4]                        ; зацикливание при ошибке, если разрешено
                ;6596    T8.OK.4:
U 17BD, 8870,0C ;6597            JSR [INC.EN]                           ; увеличение номера ошибки до 5
                ;6598    LOOP.T8.5:
U 17BE, B724,15 ;6599            MOV LS[HI.IPL] TO WR[0]                ; WR0 = 00 1F 00 00
U 17BF, BFFE,15 ;6600            MOV WR[0] TO LS[PSL.HW]                ; загрузка IPL
U 17C0, DB00,07 ;6601            SKIP.IF[NO.INTERRUPT]                  ; должен быть пропуск
U 17C1, 5B00,1E ;6602            SKIP                                   ;
U 17C2, 097C,74 ;6603            JMP [T8.OK.5]                          ; обход вызова для обработки ошибки
U 17C3, 2F80,15 ;6604            CLR WR[0]                              ;
U 17C4, 369E,95 ;6605            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 17C5, 0869,3C ;6606            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 17C6, 897B,E4 ;6607            JMP [LOOP.T8.5]                        ; зацикливание при ошибке, если разрешено
                ;6608    T8.OK.5:
U 17C7, 8870,0C ;6609            JSR [INC.EN]                           ; увеличение номера ошибки до 6
                ;6610    LOOP.T8.6:
U 17C8, 2F80,15 ;6611            CLR WR[0]                              ;
U 17C9, C77C,15 ;6612            BIS LS[BIT30] TO WR[0]                 ;
U 17CA, 4748,15 ;6613            BIS LS[BIT4] TO WR[0]                  ; WR0 = 40 00 00 10
U 17CB, BFFE,15 ;6614            MOV WR[0] TO LS[PSL.HW]                ; разрешение T-TRAP
U 17CC, B724,15 ;6615            MOV LS[HI.IPL] TO WR[0]                ;
U 17CD, 4748,15 ;6616            BIS LS[BIT4] TO WR[0]                  ;
U 17CE, 37FE,15 ;6617            MOV LS[PSL.HW] TO WR[0]                ; чтение вместо загрузки IPL и установки T-TRAP
U 17CF, DB00,07 ;6618            SKIP.IF[NO.INTERRUPT]                  ; не должно быть пропуска
U 17D0, 097D,54 ;6619            JMP [T8.OK.6]                          ; прерывание. Конец теста
U 17D1, 2F80,15 ;6620            CLR WR[0]                              ; прерывания не оказалось
U 17D2, 369E,95 ;6621            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 17D3, 0869,3C ;6622            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 17D4, 097C,84 ;6623            JMP [LOOP.T8.6]                        ; зацикливание при ошибке, если разрешено
                ;6624    T8.OK.6:
U 17D5, 8870,0C ;6625            JSR [INC.EN]                           ; увеличение номера ошибки до 7
U 17D6, 0870,BC ;6626            JSR [DEASSERT.NA]                      ; восстановление поля ожидаемых и полученных данных
                ;6627    LOOP.T8.7:
U 17D7, 3648,15 ;6628            MOV LS[BIT4] TO WR[0]                  ;
U 17D8, BFFE,15 ;6629            MOV WR[0] TO LS[PSL.HW]                ; сброс IPL и разрешение T-TRAP
U 17D9, B640,15 ;6630            MOV LS[BIT0] TO WR[0]                  ;
U 17DA, 3EF8,15 ;6631            MOV WR[0] TO LS[OS]                    ; OS = бит 0
U 17DB, 10F8,15 ;6632            MISC2 [READ.ACC.UPC]                   ; установка BR4
U 17DC, DB00,15 ;6633            NOP                                    ;
U 17DD, 36FC,15 ;6634            MOV LS[INTERRUPT.VEC] TO WR[0]         ; чтение кода идентификации
U 17DE, 364A,95 ;6635            MOV LS[BIT5] TO WR[1]                  ;
U 17DF, C748,95 ;6636            BIS LS[BIT4] TO WR[1]                  ;
U 17E0, 4746,95 ;6637            BIS LS[BIT3] TO WR[1]                  ; ожидаемые данные = 1110(B) в битах 2-5
U 17E1, 0869,3C ;6638            JSR [CHECK.RESULT]                     ; проверка кода идентификации
U 17E2, 897D,74 ;6639            JMP [LOOP.T8.7]                        ; зацикливание при ошибке, если разрешено
U 17E3, 8870,0C ;6640            JSR [INC.EN]                           ; увеличение номера ошибки до 8
                ;6641    LOOP.T8.8:
U 17E4, 3648,15 ;6642            MOV LS[BIT4] TO WR[0]                  ;
U 17E5, BFFE,15 ;6643            MOV WR[0] TO LS[PSL.HW]                ; очистка IPL и разрешение T-TRAP
U 17E6, 3644,15 ;6644            MOV LS[BIT2] TO WR[0]                  ;
U 17E7, 3EF8,15 ;6645            MOV WR[0] TO LS[OS]                    ; OS = установленный бит 2
U 17E8, 10F8,15 ;6646            MISC2 [READ.ACC.UPC]                   ; установка BR6
U 17E9, DB00,15 ;6647            NOP                                    ;
```

```
U 17EA,  36FC,15 ;6648            MOV LS[INTERRUPT.VEC] TO WR[0]          ; чтение кода идентификации
U 17EB,  364A,95 ;6649            MOV LS[BIT5] TO WR[1]                   ;
U 17EC,  4746,95 ;6650            BIS LS[BIT3] TO WR[1]                   ;
U 17ED,  C744,95 ;6651            BIS LS[BIT2] TO WR[1]                   ; ожидаемые данные = 1011(B) в битах 2-5
U 17EE,  0869,3C ;6652            JSR [CHECK.RESULT]                      ; проверка кода идентификации
U 17EF,  897E,44 ;6653            JMP [LOOP.T8.B]                         ; зацикливание при ошибке, если разрешено
U 17F0,  8870,0C ;6654            JSR [INC.EN]                            ; увеличение номера ошибки до 9
                 ;6655    LOOP.T8.9:
U 17F1,  3660,15 ;6656            MOV LS[INTERUPT.EN] TO WR[0]            ;
U 17F2,  C768,15 ;6657            BIS LS[CONSOLE.ATTN] TO WR[0]           ;
U 17F3,  3E80,15 ;6658            MOV WR[0] TO LS[CONTROL.STATUS]         ; слово управления и состояния = бит 16, бит 20
U 17F4,  10E0,15 ;6659            MISC [SET.CP.ATTN]                      ; вызов консольного процессора
                 ;6660    WAIT.T8.9:
U 17F5,  897F,54 ;6661            JMP [WAIT.T8.9]                         ; установка CONS ATTN
U 17F6,  364A,95 ;6662            MOV LS[BIT5] TO WR[1]                   ;
U 17F7,  C748,95 ;6663            BIS LS[BIT4] TO WR[1]                   ;
U 17F8,  C744,95 ;6664            BIS LS[BIT2] TO WR[1]                   ; ожидаемые данные = 1101(B) в битах 2-5
U 17F9,  36FC,15 ;6665            MOV LS[INTERRUPT.VEC] TO WR[0]          ; чтение кода идентификации
U 17FA,  0869,3C ;6666            JSR [CHECK.RESULT]                      ; проверка кода идентификации
U 17FB,  097F,14 ;6667            JMP [LOOP.T8.9]                         ; зацикливание при ошибке, если разрешено
U 17FC,  8870,0C ;6668            JSR [INC.EN]                            ; увеличение номера ошибки до A
U 17FD,  3660,15 ;6669            MOV LS[INTERUPT.EN] TO WR[0]            ;
U 17FE,  C762,15 ;6670            BIS LS[CONSOLE.HALT] TO WR[0]           ;
U 17FF,  3E80,15 ;6671            MOV WR[0] TO LS[CONTROL.STATUS]         ; слово управления и состояния = биты 16 и 17
                 ;6672                                                    ; установлены
U 1800,  10E0,15 ;6673            MISC [SET.CP.ATTN]                      ; вызов консольного процессора
                 ;6674    WAIT.T8.A:
U 1801,  0980,14 ;6675            JMP [WAIT.T8.A]                         ; установка CONS HALT
                 ;6676    LOOP.T8.A:
U 1802,  8648,95 ;6677            MOV LS[BIT4] TO WR[1]                   ;
U 1803,  4746,95 ;6678            BIS LS[BIT3] TO WR[1]                   ;
U 1804,  C744,95 ;6679            BIS LS[BIT2] TO WR[1]                   ; ожидаемые данные = 0111(B) в битах 2-5
U 1805,  36FC,15 ;6680            MOV LS[INTERRUPT.VEC] TO WR[0]          ; чтение кода идентификации
U 1806,  0869,3C ;6681            JSR [CHECK.RESULT]                      ; проверка кода идентификации
U 1807,  0980,24 ;6682            JMP [LOOP.T8.A]                         ; зацикливание при ошибке, если разрешено
U 1808,  0870,8C ;6683            JSR [ASSERT.NA]                         ;
U 1809,  B724,15 ;6684            MOV LS[HI.IPL] TO WR[0]                 ;
U 180A,  C77C,15 ;6685            BIS LS[BIT30] TO WR[0]                  ; WR0 = 40 1F 00 00
U 180B,  BFFE,15 ;6686            MOV WR[0] TO LS[PSL.HW]                 ; загрузка IPL и разрешение T-TRAP
                 ;6687    LOOP.T8.B:
U 180C,  90FC,15 ;6688            MISC2 [MASK.HALT.AND.T.BIT]             ; маскирование прерываний HALT и по биту T
U 180D,  DB00,07 ;6689            SKIP.IF[NO.INTERRUPT]                   ; должен быть пропуск
U 180E,  8981,74 ;6690            JMP [T8.ERROR.B]                        ; переход на вызов для обработки ошибки
                 ;6691    T8.C:
U 180F,  DB00,07 ;6692            SKIP.IF[NO.INTERRUPT]                   ; не должно быть пропуска
U 1810,  5800,1E ;6693            SKIP                                    ;
U 1811,  0981,F4 ;6694            JMP [T8.ERROR.C]                        ; переход на вызов для обработки ошибки
                 ;6695    LOOP.T8.D:
U 1812,  D80C,15 ;6696            MOV LS[6] TO Q                          ; эта инструкция проверяет дешифратор MISC. Биты 18 и
                 ;6697                                                    ; 11-9 те же, что и для маскирования прерывания по биту T
U 1813,  DB00,07 ;6698            SKIP.IF[NO.INTERRUPT]                   ; не должно быть пропуска
U 1814,  5800,1E ;6699            SKIP                                    ;
U 1815,  8982,74 ;6700            JMP [T8.ERROR.D]                        ; переход на вызов для обработки ошибки
U 1816,  0982,F4 ;6701            JMP [END.T8]                            ;
                 ;6702    T8.ERROR.8:
```

```
U 1817,  B646,15 ;6703            MOV LS[#8] TO WR[0]                    ;
U 1818,  C0C0,15 ;6704            ADD LS[#3(H)] TO WR[0]                 ;
U 1819,  BE82,15 ;6705            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки = B
U 181A,  2F80,15 ;6706            CLR WR[0]                              ;
U 181B,  369E,95 ;6707            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 181C,  0869,3C ;6708            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 181D,  8980,C4 ;6709            JMP [LOOP.T8.B]                        ; зацикливание при ошибке, если разрешено
U 181E,  8980,F4 ;6710            JMP [T8.C]                             ; переход к следующей части теста
                 ;6711     T8.ERROR.C:
U 181F,  B646,15 ;6712            MOV LS[#8] TO WR[0]                    ;
U 1820,  C044,15 ;6713            ADD LS[#4] TO WR[0]                    ;
U 1821,  BE82,15 ;6714            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки = C
U 1822,  2F80,15 ;6715            CLR WR[0]                              ;
U 1823,  369E,95 ;6716            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 1824,  0869,3C ;6717            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 1825,  8980,C4 ;6718            JMP [LOOP.T8.B]                        ; зацикливание при ошибке, если разрешено
U 1826,  8981,24 ;6719            JMP [LOOP.T8.D]                        ; переход к выполнению части D теста
                 ;6720     T8.ERROR.D:
U 1827,  B646,15 ;6721            MOV LS[#8] TO WR[0]                    ;
U 1828,  C044,15 ;6722            ADD LS[#4] TO WR[0]                    ;
U 1829,  4040,15 ;6723            ADD LS[#1] TO WR[0]                    ;
U 182A,  BE82,15 ;6724            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки = D
U 182B,  2F80,15 ;6725            CLR WR[0]                              ;
U 182C,  369E,95 ;6726            MOV LS[ONES] TO WR[1]                  ; установка неправильных данных
U 182D,  0869,3C ;6727            JSR [CHECK.RESULT]                     ; сообщение об ошибке
U 182E,  8981,24 ;6728            JMP [LOOP.T8.D]                        ; зацикливание при ошибке, если разрешено
                 ;6729     END.T8:
U 182F,  B724,15 ;6730            MOV LS[HI.IPL] TO WR[0]                ;
U 1830,  BFFE,15 ;6731            MOV WR[0] TO LS[PSL.HW]                ; установка 1F в IPL
U 1831,  3660,15 ;6732            MOV LS[INTERUPT.EN] TO WR[0]           ; установка бита 16 в WR0
U 1832,  3E80,15 ;6733            MOV WR[0] TO LS[CONTROL.STATUS]         ; установка слова управления и состояния для сброса всех
                 ;6734                                                    ; прерываний консольного процессора
U 1833,  10E0,15 ;6735            MISC [SET.CP.ATTN]                     ; вызов консольного процессора
                 ;6736     WAIT.T8.END:
U 1834,  0983,44 ;6737            JMP [WAIT.T8.END]                      ; снятие прерываний консольного процессора
```

```
;6738      .PAGE "ТЕСТ 9 - тест ПМЛ СДВИГ.ДАННЫХ при SI=ASH.WR (модуль DAP)"
;6739      ;
;6740      ; ОПИСАНИЕ ТЕСТА:
;6741      ;
;6742      ;   Тест проверяет ПМЛ СДВИГ.ДАННЫХ при SI=ASH.WR (SI=2). В этом тесте исполь-
;6743      ;   зуются микроинструкции ASHL.WR и ASHR.WR. Эти инструкции используют те же
;6744      ;   самые сигналы управления АЛУ как и ROR и ROL и были проверены раньше. ПМЛ
;6745      ;   СДВИГ.ДАННЫХ обеспечивает засылку 0 в бит 0 при инструкции ASHL и выполня-
;6746      ;   ет расширение знака при инструкции ASHR. Пути данных работают следующим об-
;6747      ;   разом: данные в WR0 загружаются из LS. Тогда выполняется ASHL WR[0] или ASHR
;6748      ;   WR[0]. В WR1 загружаются ожидаемые данные и проверяется результат. Использу-
;6749      ;   емыми тестовыми данными являются различные наборы единиц и нулей в битах 31
;6750      ;   и 0 рабочего регистра.
;6751      ;
;6752      ; ПРЕДПОЛОЖЕНИЯ:
;6753      ;
;6754      ;   Предполагается, что предыдущие тесты сдвига в АЛУ выполнены успешно:
;6755      ;
;6756      ; ШАГИ ТЕСТА:
;6757      ;
;6758      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;6759      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;6760      ;      памяти.
;6761      ;   2. Загрузка единиц в биты 31 и 0 WR0 и выполнение ASHL WR[0]. Проверка
;6762      ;      наличия 2(H) в WR[0].
;6763      ;   3. Загрузка 1 в бит 31 WR0 и выполнение ASHR WR[0]. Проверка наличия
;6764      ;      C0000000(H) в WR0 (расширение знака отрицательного числа).
;6765      ;   4. Загрузка 1 в бит 0 WR0 и выполнение ASHR WR[0]. Проверка наличия нуля
;6766      ;      (расширение знака положительного числа).
;6767      ;
;6768      ; ОШИБКИ:
;6769      ;
;6770      ;   ошибка 1 - ASHL работает неправильно с 1 в битах 31 и 0.
;6771      ;   ошибка 2 - ASHR работает неправильно с единицей в бите 31.
;6772      ;   ошибка 3 - ASHR работает неправильно с единицей в бите 0.
;6773      ;
;6774      ; НАЛАДКА:
;6775      ;
;6776      ;   ОШИБКА 1 - Эта ошибка скорее всего появляется из-за ПМЛ СДВИГ.ДАННЫХ. Во
;6777      ;   время инструкции ASHL необходимо проверить наличие 2(H) на входах CSR 13-11.
;6778      ;   сигналы DEST CTL 1 и DEST CTL 2 должны иметь высокий уровень. Все это необходимо
;6779      ;   для получения низкого уровня на выходе RAM SHF LSB H. Если входы правильные,
;6780      ;   и на выходе высокий уровень - ПМЛ неисправна.
;6781      ;
;6782      ;   ОШИБКА 2 - Эта ошибка скорее всего появляется из-за ПМЛ СДВИГ.ДАННЫХ или из-
;6783      ;   за входа N LONG H. Во время инструкции ASHR неоходимо проверить наличие 2(H)
;6784      ;   на входах CSR 13-11, низкий уровень на входе DEST CTL 1 H, высокий уровень
;6785      ;   на входе DEST CTL 2 H и высокий уровень на входе N LONG H. Выход RAM SHF MSB H
;6786      ;   должен иметь высокий уровень. Если входы правильные, подозревается ПМЛ СДВИГ.ДАННЫХ.
;6787      ;
;6788      ;   ОШИБКА 3 - Эта ошибка скорее всего появляется из-за ПМЛ СДВИГ.ДАННЫХ или
;6789      ;   из-за входа N LONG H. Во время инструкции ASHR необходимо проверить наличие
;6790      ;   2(H) на входах CSR 13-11, низкий уровень на входе DEST CTL 1 H, высокий уровень
;6791      ;   на входе DEST CTL 2 H и низкий уровень на входе N LONG H. Выход RAM SHF MSB H
;6792      ;   должен иметь низкий уровень. Если входы правильные, подозревается ПМЛ СДВИГ.ДАННЫХ.
```

```
                    ;6793     ;
                    ;6794     T.9:
U 1835,  B65E,15    ;6795               MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;6796                                                    ; состояния
U 1836,  3EB0,15    ;6797               MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления и состояния. Бит
                    ;6798                                                    ; 15 указывает начало теста для конс.процессора
U 1837,  10E0,15    ;6799               MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для конс.процессора
                    ;6800     WAIT.T9.0:
U 1838,  0983,84    ;6801               JMP [WAIT.T9.0]                      ; зацикливание для ожидания ответа конс.процесора
U 1839,  E58A,15    ;6802               CLR LS[ERROR.MASK]                   ; очистка маски ошибок
U 183A,  65A0,15    ;6803               CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 183B,  B640,15    ;6804               MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 183C,  8E82,15    ;6805               MOV WR[0] TO LS[ERROR.NUMBER]        ; установка номера ошибки для первой ошибки
U 183D,  A3C0,15    ;6806               ROL WR[0]                            ; установка 2 в WR0
U 183E,  3E8C,15    ;6807               MOV WR[0] TO LS[MODULE.NUM]           ; установка кода модуля для модуля DAP
                    ;6808     LOOP.T9.1:
U 183F,  367E,15    ;6809               MOV LS[BIT31] TO WR[0]               ;
U 1840,  C740,15    ;6810               BIS LS[BIT0] TO WR[0]                ; WR0=80 00 00 01
U 1841,  23D0,15    ;6811               ASHL WR[0]                           ; сдвиг влево
U 1842,  B642,95    ;6812               MOV LS[BIT1] TO WR[1]                ; ожидаемые данные =00 00 00 02
U 1843,  0869,3C    ;6813               JSR [CHECK.RESULT]                   ;
U 1844,  8983,F4    ;6814               JMP [LOOP.T9.1]                      ; зацикливание при ошибке, если разрешено
U 1845,  8870,0C    ;6815               JSR [INC.EN]                         ; увеличение номера ошибки до 2
                    ;6816     LOOP.T9.2:
U 1846,  367E,15    ;6817               MOV LS[BIT31] TO WR[0]               ; WR0=80 00 00 00
U 1847,  A350,15    ;6818               ASHR WR[0]                           ; сдвиг вправо
U 1848,  867E,95    ;6819               MOV LS[BIT31] TO WR[1]               ;
U 1849,  477C,95    ;6820               BIS LS[BIT30] TO WR[1]               ; ожидаемые данные=00 00 00 00
U 184A,  0869,3C    ;6821               JSR [CHECK.RESULT]                   ;
U 184B,  0984,64    ;6822               JMP [LOOP.T9.2]                      ; зацикливание при ошибке, если разрешено
U 184C,  8870,0C    ;6823               JSR [INC.EN]                         ; увеличние номера ошибки до 3
                    ;6824     LOOP.T9.3:
U 184D,  B640,15    ;6825               MOV LS[BIT0] TO WR[0]                ; WR0=00 00 00 01
U 184E,  A350,15    ;6826               ASHR WR[0]                           ; сдвиг вправо
U 184F,  2F82,95    ;6827               CLR WR[1]                            ; ожидаемые данные=00 00 00 00
U 1850,  0869,3C    ;6828               JSR [CHECK.RESULT]                   ;
U 1851,  8984,D4    ;6829               JMP [LOOP.T9.3]                      ; зацикливание при ошибке, если разрешено
                    ;6830     END.T9:
```

```
;6831      .PAGE "ТЕСТ А – тест ПМЛ СДВИГ.ДАННЫХ при SI=ASH.WR.Q (модуль DAP)"
;6832      ;
;6833      ; ОПИСАНИЕ ТЕСТА:
;6834      ;
;6835      ;   Тест проверяет ПМЛ СДВИГ.ДАННЫХ при SI=ASH.WR.Q (SI=3). В этом тесте ис-
;6836      ;   пользуются микроинструкции ASHLC WR.Q и ASHRC WR.Q. Эти инструкции исполь-
;6837      ;   зуют те же самые сигналы управления АЛУ, как и ROLC, и были проверены рань-
;6838      ;   ше. ПМЛ СДВИГ.ДАННЫХ обеспечивает засылку 0 в младший бит регистра Q (ре-
;6839      ;   гистр Q содержит низшие 32 бита 64-битового четырехкратного слова) и заг-
;6840      ;   рузку младшего бита внутреннего ЗУ из старшего бита регистра Q инструкцией
;6841      ;   ASHLC. При инструкции ASHRC ПМЛ СДВИГ.ДАННЫХ обеспечивает расширение знака
;6842      ;   и старший бит внутреннего ЗУ (дублирующего содержание старшего бита внут-
;6843      ;   реннего ЗУ до сдвига) и загрузку старшего бита регистра Q из младшего бита
;6844      ;   внутреннего ЗУ. Используемыми тестовыми данными являются наборы единиц и
;6845      ;   нулей в битах 31 и 0 рабочего регистра и регистра Q.
;6846      ;
;6847      ; ПРЕДПОЛОЖЕНИЯ:
;6848      ;
;6849      ;   Предполагается, что предыдущие тесты сдвига в АЛУ выполнены успешно.
;6850      ;
;6851      ; ШАГИ ТЕСТА:
;6852      ;
;6853      ;   1)  Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;6854      ;       (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;6855      ;       памяти.
;6856      ;   2)  Загрузка 1 в биты 31 и 0 WR0 и регистра Q, нулей в другие биты.
;6857      ;   3)  Выполнение ASHLC WR[0].Q и проверка наличия 3(H) в WR0.
;6858      ;   4)  Загрузка в WR0 из регистра Q и проверка наличия 2(H). Этим проверяется
;6859      ;       заполнение нулем младшего бита.
;6860      ;   5)  Загрузка нулей в WR0 и "1" в бит 31 регистра Q.
;6861      ;   6)  Выполнение ASHLC WR[0].Q и проверка наличия 1(H) в WR0.
;6862      ;   7)  Загрузка WR0 из регистра Q и проверка наличия 0.
;6863      ;   8)  Загрузка 1 в бит 31 WR0, нулей в другие биты и загрузка регистра Q все-
;6864      ;       ми нулями.
;6865      ;   9)  Выполнение ASHRC WR[0].Q и проверка наличия C0 00 00 00(H) в WR0. Этим
;6866      ;       проверяется расширение знака отрицательного числа.
;6867      ;   10) Загрузка WR0 из регистра Q и проверка наличия 0.
;6868      ;   11) Загрузка WR0 всеми нулями и загрузка 1 в бит 0 регистра Q.
;6869      ;   12) Выполнение ASHRC WR[0].Q и проверка наличия 0 в WR0. Этим проверяется
;6870      ;       расширение знака положительного числа.
;6871      ;   13) Загрузка WR0 из регистра Q и проверка наличия 0.
;6872      ;   14) Загрузка 1 в бит 0 WR0, нулей в другие биты и загрузка регистра Q всеми
;6873      ;       нулями.
;6874      ;   15) Выполнение ASHRC WR[0].Q и проверка 0 в WR0.
;6875      ;   16) Загрузка WR0 из регистра Q и проверка наличия 80 00 00 00. Этим
;6876      ;       проверяется загрузка старшего бита в регистр Q.
;6877      ;
;6878      ; ОШИБКИ:
;6879      ;
;6880      ;   ошибка 1 – ASHLC WR [0].Q выполняет сдвиг в WR0 неправильно. Тестовые данные
;6881      ;              до сдвига состоят из 1 в битах 31 и 0 в WR0 и регистре Q.
;6882      ;   ошибка 2 – ASHLC WR[0].Q выполняет сдвиг в регистре Q неправильно. Данные
;6883      ;              до сдвига были – 1 в битах 31 и 0 в WR0 и регистре Q.
;6884      ;   ошибка 3 – ASHLC WR[0].Q  выполняет сдвиг в WR0 неправильно. Тестовыми дан-
;6885      ;              ными до сдвига была 1 в бите 31 регистра Q и нули во всех других
```

```
;6886      ;              битах.
;6887      ;   ошибка 4 – ASHLC WR[0].Q выполняет сдвиг в регистре Q неправильно. Данные
;6888      ;              до сдвига были – 1 в бите 31 регистра Q и нули во всех других
;6889      ;              битах.
;6890      ;   ошибка 5 – ASHRC WR[0].0 выполняет сдвиг в WR0 неправильно. Данные до
;6891      ;              сдвига были – 1 в бите 31 WR0 и нули во всех других битах.
;6892      ;   ошибка 6 – ASHRC WR[0].Q выполняет сдвиг в регистре Q неправильно. Данные
;6893      ;              до сдвига были – 1 в бите 31 WR0 и нули во всех других битах.
;6894      ;   ошибка 7 – ASHRC WR[0].Q выполняет сдвиг в WR0 неправильно. Данные до сдвига
;6895      ;              были – 1 в бите 0 регистра Q и нули во всех других битах.
;6896      ;   ошибка 8 – ASHRC WR[0].Q выполняет сдвиг в регистре Q неправильно. Данные
;6897      ;              до сдвига были – 1 в бите 0 регистра Q и нули во всех других битах.
;6898      ;   ошибка 9 – ASHRC WR[0].Q выполняет сдвиг в WR0 неправильно. Данные до сдвига
;6899      ;              были – 1 в бите 0 WR0 и нули во всех других битах.
;6900      ;   ошибка A – ASHRC WR[0].Q выполняет сдвиг в регистре Q неправильно. Данные
;6901      ;              до сдвига были – 1 в бите 0 WR0 и нули во всех других битах.
;6902      ;
;6903      ; НАЛАДКА:
;6904      ;
;6905      ;   ОШИБКА 1 – Эта ошибка скорее всего появляется из–за ПМЛ СДВИГ.ДАННЫХ.
;6906      ;   Выход RAM SHF LSB H должен иметь высокий уровень во время микрослова сдвига.
;6907      ;   Входы ПМЛ СДВИГ.ДАННЫХ должны иметь высокий уровень на Q SHF MSB H и 3(H) на
;6908      ;   CSR 13–11. Если эти входы правильные, а на выходе имеется низкий уровень –
;6909      ;   ПМЛ неисправна.
;6910      ;
;6911      ;   ОШИБКА 2 – Эта ошибка скорее всего появляется  из–за ПМЛ СДВИГ.ДАННЫХ. Выход
;6912      ;   Q SHF LSB H должен иметь низкий уровень во время микрослова сдвига. Если
;6913      ;   нет, необходимо проверить наличие 3(H) на CSR 13–11. Только это необходимо
;6914      ;   для получения низкого уровня на выходе.
;6915      ;
;6916      ;   ОШИБКА 3 – То же, что и для ошибки 1.
;6917      ;
;6918      ;   ОШИБКА 4 – То же, что и для ошибки 2.
;6919      ;
;6920      ;   ОШИБКА 5 – Эта ошибка скорее всего появляется из–за ПМЛ СДВИГ.ДАННЫХ. Выход
;6921      ;   RAM SHF MSB H должен иметь высокий уровень во время микрослова сдвига. Если
;6922      ;   нет, необходимо проверить наличие 3(H) на входах CSR 13–11 и высокий уровень
;6923      ;   на входе N LONG H. ПМЛ дефектна, если входы правильные, а на выходе имеется
;6924      ;   низкий уровень.
;6925      ;
;6926      ;   ОШИБКА 6 – Эта ошибка, скорее всего, появляется из–за ПМЛ СДВИГ.ДАННЫХ.
;6927      ;   Выход Q SHF MSB H должен иметь низкий уровень во время микрослова сдвига.
;6928      ;   Если нет, необходимо проверить наличие 3(H) на входах CSR 13–11 и низкий
;6929      ;   уровень на входе RAM SHF LSB H. ПМЛ дефектна, если входы правильные, а на
;6930      ;   выходе высокий уровень.
;6931      ;
;6932      ;   ОШИБКА 7 – Эта ошибка, скорее всего, появляется из–за ПМЛ СДВИГ.ДАННЫХ. Выход
;6933      ;   RAM SHF MSB H должен иметь низкий уровень во время микрослова сдвига. Если
;6934      ;   нет, необходимо проверить наличие 3(H) на входах CSR 13–11 и низкий уровень
;6935      ;   на входе N LONG H. ПМЛ дефектна, если входы правильные, а на выходе имеется
;6936      ;   низкий уровень.
;6937      ;
;6938      ;   ОШИБКА 8 – То же, что и для ошибки 6.
;6939      ;
;6940      ;   ОШИБКА 9 – То же, что и для ошибки 7.
```

```
                    ;6941     ;
                    ;6942     ;   ОШИБКА А - Эта ошибка скорее всего появляется из-за ПМЛ СДВИГ.ДАННЫХ. Выход
                    ;6943     ;   Q SHF MSB H должен иметь высокий уровень во время микрослова сдвига. Если
                    ;6944     ;   нет, необходимо проверить наличие 3(H) на входах CSR 13-11 и высокий уровень
                    ;6945     ;   на входе RAM SHF LSB H. ПМЛ дефектна, если входы правильные, а на выходе
                    ;6946     ;   имеется высокий уровень.
                    ;6947     ;
                    ;6948     T.A:
U 1852,  B65E,15    ;6949           MOV LSIBEGIN.TEST] TO WRI0]                 ; установка бита 15 в WR0 для слова управления и
                    ;6950                                                       ; состояния
U 1853,  3E80,15    ;6951           MOV WRI0] TO LSICONTROL.STATUS]             ; установка бита 15 в слове управления и состояния. Бит
                    ;6952                                                       ; 15 указывает начало теста для консольного процессора
U 1854,  10E0,15    ;6953           MISC ISET.CP.ATTN]                          ; выдача сигнала CPU ATTN для консольного процессора
                    ;6954     WAIT.TA.0:
U 1855,  8985,54    ;6955           JMP IWAIT.TA.0]                             ; зацикливание для ожидания ответа от консольного
                    ;6956                                                       ; процессора
U 1856,  E58A,15    ;6957           CLR LSIERROR.MASK]                          ; сброс маски ошибки
U 1857,  65A0,15    ;6958           CLR LSIPREVIOUS.ERROR]                      ; сброс предыдущего номера ошибки
U 1858,  3642,15    ;6959           MOV LSICPU] TO WRI0]                        ; установка бита 1 в WR0
U 1859,  3E8C,15    ;6960           MOV WRI0] TO LSIMODULE.NUM]                 ; установка кода модуля для модуля DAP
                    ;6961     LOOP.TA.1:
U 185A,  B640,15    ;6962           MOV LSI#1] TO WRI0]                         ; установка 1 в WR0
U 185B,  BE82,15    ;6963           MOV WRI0] TO LSIERROR.NUMBER]               ; номер ошибки=1
U 185C,  367E,15    ;6964           MOV LSIBIT31] TO WRI0]                      ;
U 185D,  C740,15    ;6965           BIS LSIBIT0] TO WRI0]                       ; WR0=80 00 00 01
U 185E,  D87E,15    ;6966           MOV LSIBIT31] TO Q                          ;
U 185F,  D340,15    ;6967           BIS LSIBIT0] TO Q                           ; Q=80 00 00 01
U 1860,  2398,15    ;6968           ASHLC WRI0].Q                               ; сдвиг влево четырехкратного слова (WR0-Q)
U 1861,  7610,15    ;6969           MOV Q TO LSIT8]                             ; запоминание Q
U 1862,  86C0,95    ;6970           MOV LSI#3(H)] TO WRI1]                      ; ожидаемые данные в WR0=00 00 00 03
U 1863,  0869,3C    ;6971           JSR ICHECK.RESULT]                          ; проверка
U 1864,  8985,A4    ;6972           JMP ILOOP.TA.1]                             ; зацикливание при ошибке, если разрешено
U 1865,  8870,0C    ;6973           JSR IINC.EN]                                ; увеличение номера ошибки до 2
                    ;6974     TA.2:
U 1866,  B610,15    ;6975           MOV LSIT8] TO WRI0]                         ; выборка Q
U 1867,  B642,95    ;6976           MOV LSI#2] TO WRI1]                         ; ожидаемые данные в регистре Q=00 00 00 02
U 1868,  0869,3C    ;6977           JSR ICHECK.RESULT]                          ; проверка
U 1869,  8985,A4    ;6978           JMP ILOOP.TA.1]                             ; зацикливание при ошибке, если разрешено
                    ;6979     LOOP.TA.3:
U 186A,  36C0,15    ;6980           MOV LSI#3(H)] TO WRI0]                      ;
U 186B,  BE82,15    ;6981           MOV WRI0] TO LSIERROR.NUMBER]               ; номер ошибки=3
U 186C,  D87E,15    ;6982           MOV LSIBIT31] TO Q                          ; Q=80 00 00 00
U 186D,  2F80,15    ;6983           CLR WRI0]                                   ; WR0=00 00 00 00
U 186E,  2398,15    ;6984           ASHLC WRI0].Q                               ; сдвиг влево
U 186F,  7610,15    ;6985           MOV Q TO LSIT8]                             ; запоминание Q
U 1870,  3640,95    ;6986           MOV LSI#1] TO WRI1]                         ; ожидаемые данные (в WR0)=00 00 00 01
U 1871,  0869,3C    ;6987           JSR ICHECK.RESULT]                          ;
U 1872,  8986,A4    ;6988           JMP ILOOP.TA.3]                             ; зацикливание при ошибке, если разрешено
U 1873,  8870,0C    ;6989           JSR IINC.EN]                                ; увеличение номера ошибки до 4
                    ;6990     TA.4:
U 1874,  B610,15    ;6991           MOV LSIT8]  TO WRI0]                        ; выборка Q
U 1875,  2F82,95    ;6992           CLR WRI1]                                   ; ожидаемые данные (в регистре Q)=00 00 00 00
U 1876,  0869,3C    ;6993           JSR ICHECK.RESULT]                          ;
U 1877,  8986,A4    ;6994           JMP ILOOP.TA.3]                             ; зацикливание при ошибке, если разрешено
                    ;6995     LOOP.TA.5:
```

```
U 1878, 3644,15 ;6996            MOV LS[#4] TO WR[0]                    ;
U 1879, 2040,15 ;6997            INC WR[0]                              ;
U 187A, BE82,15 ;6998            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки=
U 187B, 367E,15 ;6999            MOV LS[BIT31] TO WR[0]                 ; WR0=80 00 00 00
U 187C, D89C,15 ;7000            MOV LS[ZERO] TO Q                      ; Q=00 00 00 00
U 187D, A318,15 ;7001            ASHRC WR[0].Q                          ; сдвиг вправо
U 187E, 7610,15 ;7002            MOV Q TO LS[T8]                        ; запоминание Q
U 187F, B67E,95 ;7003            MOV LS[BIT31] TO WR[1]                 ;
U 1880, 477C,95 ;7004            BIS LS[BIT30] TO WR[1]                 ; ожидаемые данные (в WR0)=C0 00 00 00
U 1881, 0869,3C ;7005            JSR [CHECK.RESULT]                     ;
U 1882, 8987,84 ;7006            JMP [LOOP.TA.5]                        ; зацикливание при ошибке, если разрешено
U 1883. 8870,0C ;7007            JSR [INC.EN]                           ; увеличение номера ошибки до 6
                ;7008    TA.6:
U 1884, B610,15 ;7009            MOV LS[T8] TO WR[0]                    ; выборка Q
U 1885, 2F82,95 ;7010            CLR WR[1]                              ; ожидаемые данные (в регистре Q)=00 00 00 00
U 1886, 0869,3C ;7011            JSR [CHECK.RESULT]                     ;
U 1887, 8987,84 ;7012            JMP [LOOP.TA.5]                        ; зацикливание при ошибке, если разрешено
                ;7013    LOOP.TA.7:
U 1888, B646,15 ;7014            MOV LS[#8] TO WR[0]                    ;
U 1889, 2100,15 ;7015            DEC WR[0]                              ;
U 188A, BE82,15 ;7016            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки=7
U 188B, 2F80,15 ;7017            CLR WR[0]                              ; WR0=00 00 00 00
U 188C, 5840,15 ;7018            MOV LS[#1] TO Q                        ; Q=00 00 00 01
U 188D, A318,15 ;7019            ASHRC WR[0].Q                          ; сдвиг вправо
U 188E, 7610,15 ;7020            MOV Q TO LS[T8]                        ; запоминание регистра Q
U 188F, 2F82,95 ;7021            CLR WR[1]                              ; ожидаемые данные (в WR0)=00 00 00 00
U 1890, 0869,3C ;7022            JSR [CHECK.RESULT]                     ;
U 1891, 8988,84 ;7023            JMP [LOOP.TA.7]                        ; зацикливание при ошибке, если разрешено
U 1892, 8870,0C ;7024            JSR [INC.EN]                           ; увеличение номера ошибки до 8
                ;7025    TA.8:
U 1893, B610,15 ;7026            MOV LS[T8] TO WR[0]                    ; выборка значения Q после временного хранения
U 1894, 2F82,95 ;7027            CLR WR[1]                              ; ожидаемые данные в регистре Q=00 00 00 00
U 1895, 0869,3C ;7028            JSR [CHECK.RESULT]                     ;
U 1896, 8988,84 ;7029            JMP [LOOP.TA.7]                        ; зацикливание при ошибке, если разрешено
                ;7030    LOOP.TA.9:
U 1897, 8646,15 ;7031            MOV LS[#8] TO WR[0]                    ;
U 1898, 2040,15 ;7032            INC WR[0]                              ;
U 1899, BE82,15 ;7033            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки=9
U 189A, B640,15 ;7034            MOV LS[#1] TO WR[0]                    ; WR0=00 00 00 01
U 189B, D89C,15 ;7035            MOV LS[ZERO] TO Q                      ; Q=00 00 00 00
U 189C, A318,15 ;7036            ASHRC WR[0].Q                          ; сдвиг вправо
U 189D, 7610,15 ;7037            MOV Q TO LS[T8]                        ; запоминание регистра Q
U 189E, 2F82,95 ;7038            CLR WR[1]                              ; ожидаемые данные (в WR0)=00 00 00 00
U 189F, 0869,3C ;7039            JSR [CHECK.RESULT]                     ;
U 18A0, 0989,74 ;7040            JMP [LOOP.TA.9]                        ; зацикливание при ошибке, если разрешено
U 18A1, 8870,0C ;7041            JSR [INC.EN]                           ; увеличение номера ошибки до A
                ;7042    TA.A:
U 18A2, B610,15 ;7043            MOV LS[T8] TO WR[0]                    ; получение запомненного значения Q
U 18A3, B67E,95 ;7044            MOV LS[BIT31] TO WR[1]                 ; ожидаемые данные (в регистре Q)= 80 00 00 00
U 18A4, 0869,3C ;7045            JSR [CHECK.RESULT]                     ;
U 18A5, 0989,74 ;7046            JMP [LOOP.TA.9]                        ; зацикливание при ошибке, если разрешено
                ;7047    END.TA:
```

```
                ;7048      .PAGE "ТЕСТ В - тест ПМЛ СДВИГ.ДАННЫХ при SI=SHF.WR.Q (модуль DAP)"
                ;7049      ;
                ;7050      ; ОПИСАНИЕ ТЕСТА:
                ;7051      ;
                ;7052      ;   Тест проверяет ПМЛ СДВИГ.ДАННЫХ при SI=SHF.WR.Q (SI=6). В этом тесте исполь-
                ;7053      ;   зуется микроинструкция SHRC WR.Q. Эта инструкция использует те же самые сиг-
                ;7054      ;   налы управления АЛУ, как ASHRC, и была проверена раньше. ПМЛ СДВИГ.ДАННЫХ
                ;7055      ;   обеспечивает засылку 0 в бит 31 рабочего регистра. Пути данных работают сле-
                ;7056      ;   дующим образом: WR0 загружается данными из LS. Тогда выполняется SHRC WR[0].Q.
                ;7057      ;   В WR1 загружаются ожидаемые данные и проверяется результат. Используемыми
                ;7058      ;   тестовыми данными являются наборы единиц и нулей в битах 31 и 0 WR.
                ;7059      ;
                ;7060      ; ПРЕДПОЛОЖЕНИЯ:
                ;7061      ;
                ;7062      ;   Предполагается, что предыдущие тесты сдвига в АЛУ выполнены успешно.
                ;7063      ;
                ;7064      ; ШАГИ ТЕСТА:
                ;7065      ;
                ;7066      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти
                ;7067      ;      (для распечатки ошибок) и очистка предыдущего номера ошибки в местной па-
                ;7068      ;      мяти.
                ;7069      ;   2. Загрузка 1 в бит 31 WR0 и 1 в биты 31 и 0 регистра Q и выполнение SHRC
                ;7070      ;      WR[0].Q. Проверка наличия 40 00 00 00(H) в WR0 и 40 00 00 00(H) в ре-
                ;7071      ;      гистре Q.
                ;7072      ;   3. Загрузка 1 в бит 0 WR0 и 0 в регистр Q. Выполнение SHRC WR[0].Q и провер-
                ;7073      ;      ка наличия нуля в WR0 и 80 00 00 00(H) в регистре Q.
                ;7074      ;
                ;7075      ; ОШИБКИ:
                ;7076      ;
                ;7077      ;   ошибка 1 - WR0 работает неправильно при SHRC WR[0].Q с 1 в бите 31 WR0 и
                ;7078      ;              1 в битах 31 и 0 регистра Q.
                ;7079      ;   ошибка 2 - регистр Q работает неправильно с вышеуказанными данными
                ;7080      ;   ошибка 3 - WR0 работает неправильно при SHRC WR.Q с 1 в бите 0 WR0 и нулями
                ;7081      ;              в регистре Q.
                ;7082      ;   ошибка 4 - регистр Q работает неправильно с вышеуказанными данными.
                ;7083      ;
                ;7084      ; НАЛАДКА:
                ;7085      ;
                ;7086      ;   ОШИБКА 1 - Эта ошибка скорее всего появляется из-за ПМЛ СДВИГ.ДАННЫХ.
                ;7087      ;   Необходимо проверить во время инструкции SHRC наличие 6(H) на входах
                ;7088      ;   CSR 13-11 этой ПМЛ. Выход RAM SHF MSB H должен иметь низкий уровень.
                ;7089      ;   Если нет, ПМЛ неисправна.
                ;7090      ;
                ;7091      ;   ОШИБКА 2 - То же, что и для ошибки 1, за исключением того, что выход
                ;7092      ;   Q SHF MSB H должен иметь низкий уровень.
                ;7093      ;
                ;7094      ;   ОШИБКА 3 - То же, что и для ошибки 1.
                ;7095      ;
                ;7096      ;   ОШИБКА 4 - То же, что и для ошибки 1, за исключением того, что выход
                ;7097      ;   Q SHF MSB H должен иметь высокий уровень.
                ;7098      ;
                ;7099      T.B:
U 18A6, B65E,15 ;7100              MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;7101                                                       ; состояния
U 18A7, 3E80,15 ;7102              MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
```

```
                ;7103
                                                ; 15 указывает начало теста для консольного процессора
U 18A8, 10E0,15 ;7104           MISC [SET.CP.ATTN]      ; выдача сигнала CPU ATTN для консольного процессора
                ;7105   WAIT.TB.0:
U 18A9, 898A,94 ;7106           JMP [WAIT.TB.0]         ; зацикливание для ожидания ответа консольного
                ;7107                                   ; процессора
U 18AA, E58A,15 ;7108           CLR LS[ERROR.MASK]      ; очистка маски ошибки
U 18AB, 65A0,15 ;7109           CLR LS[PREVIOUS.ERROR]  ; очистка предыдущего номера ошибки
U 18AC, 3642,15 ;7110           MOV LS[CPU] TO WR[0]    ; установка бита 1 в WR0
U 18AD, 3E8C,15 ;7111           MOV WR[0] TO LS[MODULE.NUM]     ; установка кода модуля для модуля DAP
                ;7112   LOOP.TB.1:
U 18AE, B640,15 ;7113           MOV LS[#1] TO WR[0]     ; установка 1 в WR0
U 18AF, BE82,15 ;7114           MOV WR[0] TO LS[ERROR.NUMBER]   ; установка номера ошибки=1
U 18B0, 367E,15 ;7115           MOV LS[BIT31] TO WR[0]  ; WR0=80 00 00 00
U 18B1, D87E,15 ;7116           MOV LS[BIT31] TO Q      ;
U 18B2, D340,15 ;7117           BIS LS[BIT0] TO Q       ; Q=80 00 00 01
U 18B3, A330,15 ;7118           SHRC WR[0].Q            ;
U 18B4, 7610,15 ;7119           MOV Q TO LS[TB]         ; запоминание регистра Q
U 18B5, 367C,95 ;7120           MOV LS[BIT30] TO WR[1]  ; ожидаемые данные (в WR0)=40 00 00 00
U 18B6, 0869,3C ;7121           JSR [CHECK.RESULT]      ;
U 18B7, 098A,E4 ;7122           JMP [LOOP.TB.1]         ; зацикливание при ошибке, если разрешено
U 18B8, 8870,0C ;7123           JSR [INC.EN]            ; увеличение номера ошибки до 2
                ;7124   TB.2:
U 18B9, B610,15 ;7125           MOV LS[TB] TO WR[0]     ; выборка значения Q после временного хранения
U 18BA, 367C,95 ;7126           MOV LS[BIT30] TO WR[1]  ; ожидаемые данные (в регистре Q)= 40 00 00 00
U 18BB, 0869,3C ;7127           JSR [CHECK.RESULT]      ;
U 18BC, 098A,E4 ;7128           JMP [LOOP.TB.1]         ; зацикливание при ошибке, если разрешено
                ;7129   LOOP.TB.3:
U 18BD, 36C0,15 ;7130           MOV LS[#3(H)] TO WR[0]  ;
U 18BE, BE82,15 ;7131           MOV WR[0] TO LS[ERROR.NUMBER]   ; установка номера ошибки=3
U 18BF, B640,15 ;7132           MOV LS[#1] TO WR[0]     ; WR0=00 00 00 01
U 18C0, D89C,15 ;7133           MOV LS[ZERO] TO Q       ; Q=00 00 00 00
U 18C1, A330,15 ;7134           SHRC WR[0].Q            ;
U 18C2, 7610,15 ;7135           MOV Q TO LS[TB]         ; запоминание регистра Q
U 18C3, 2F82,95 ;7136           CLR WR[1]               ; ожидаемые данные (в WR0)=00 00 00 00
U 18C4, 0869,3C ;7137           JSR [CHECK.RESULT]      ;
U 18C5, 8988,D4 ;7138           JMP [LOOP.TB.3]         ; зацикливание при ошибке, если разрешено
U 18C6, 8870,0C ;7139           JSR [INC.EN]            ; увеличение номера ошибки до 4
                ;7140   TB.4:
U 18C7, B610,15 ;7141           MOV LS[TB] TO WR[0]     ; выборка значения Q  после временного хранения
U 18C8, B67E,95 ;7142           MOV LS[BIT31] TO WR[1]  ; ожидаемые данные (в регистре Q)= 80 00 00 00
U 18C9, 0869,3C ;7143           JSR [CHECK.RESULT]      ;
U 18CA, 8988,D4 ;7144           JMP [LOOP.TB.3]         ; зацикливание при ошибке, если разрешено
                ;7145   END.TB:
```

```
;7146      .PAGE "ТЕСТ С - тест умножения со знаком (SI=MUL) (модуль DAP)"
;7147      ;
;7148      ; ОПИСАНИЕ ТЕСТА:
;7149      ;
;7150      ;   Этот тест проверяет логические схемы условного суммирования и сдвига. В тесте
;7151      ;   используется инструкция COND.ADD&SHIFT WR TO WR.Q. Эта инструкция использу-
;7152      ;   ется для апаратного умножения со знаком и без знака. Эта инструкция отлича-
;7153      ;   ется от нормального управления АЛУ тем, что режим SRC (источника) зависит от
;7154      ;   значения нулевого бита регистра Q до предыдущей операции сдвига. Сигналы уп-
;7155      ;   равления АЛУ во время условной инструкции следующие: SRC=MUL FUNC=R.PLUS.S
;7156      ;   DST=RSHF.RAM.Q CIN=NOCIN. Режим SRC будет или A.B, если регистр Q содержит 1
;7157      ;   в бите 0 до предыдущего сдвига, или 0.B, если регистр Q содержал 0 в бите 0
;7158      ;   до предыдущего сдвига. В этом тесте в поле SI установлено на 4. Этот тест прове-
;7159      ;   ряет операцию COND.ADD&SHIFT WR TO WR.Q с обоими условиями предыдущего сдви-
;7160      ;   га. В начале тест загружает два рабочих регистра и регистр Q данными и сдви-
;7161      ;   гает регистр Q вправо для загрузки MPLIER LSB H, который управляет режимом
;7162      ;   SRC для следующей инструкции. Тогда выполняется инструкция COND.ADD&SHIFT
;7163      ;   WR[2] TO WR[0].Q и проверяется результат. Тестовые данные подбираются так,
;7164      ;   чтобы проверить сдвиг и исключающее "ИЛИ" для битов N и V, которое поступа-
;7165      ;   ет в старший бит рабочего регистра. Другая инструкция COND.SUB&SHIFT
;7166      ;   WR FROM WR.Q выполняется также для проверки ПЗУ УПР.ФУНКЦИЕЙ АЛУ и ПМЛ УПР.
;7167      ;   ИСТ.ПРИЕМН.АЛУ. Сигналы управления этой инструкции следующие:
;7168      ;   SRC=MUL FUNC=S.MINUS.R DST=RSHF.RAM.Q CIN=CIN. Для проверки этой функции ис-
;7169      ;   пользуется только один набор, так как единственное отличие от условного сло-
;7170      ;   жения заключается в управлении функцией АЛУ.
;7171      ;
;7172      ; ПРЕДПОЛОЖЕНИЯ:
;7173      ;
;7174      ;   Предполагается, что предыдущие тесты ПМЛ СДВИГ. ДАННЫХ выполнены успешно.
;7175      ;
;7176      ; ШАГИ ТЕСТА:
;7177      ;
;7178      ;   1. Установка маски ошибки, номера ошибки и номера модуля в местной памяти (для
;7179      ;      распечатки ошибок) и очистка предыдущего номера ошибки в местной памяти.
;7180      ;   2. Загрузка 1 в WR2, WR0 и регистр Q. Загрузка всеми нулями WR3 и выполнение
;7181      ;      ASHRC WR[3].Q. этим MPLIER LSB H загружается единицей и остается 0 в регист-
;7182      ;      ре Q .
;7183      ;   3. Выполнение COND.ADD&SHIFT WR[2] TO WR[0].Q. Высокий уровень сигнала
;7184      ;      MPLIER LSB H вызовет выполнение A PLUS B (1 PLUS 1). Сдвиг засылает 1 в бит
;7185      ;      0 WR0. В регистре Q все еще 0. Бит 31 WR0 будет=0, так как N LONG и V LONG
;7186      ;      сброшены. Проверяются результаты для этих операндов.
;7187      ;   4. Загрузка 1 в WR2 и WR0, сброс регистра Q и выполнение ASHR WR[3].Q для
;7188      ;      загрузки 0 в MPLIER LSB H.
;7189      ;   5. Выполнение COND.ADD&SHIFT WR[2] TO WR[0].Q. Низкий уровень сигнала
;7190      ;      MPLIER LSB H вызовет выполнение 0 PLUS в (0 плюс 1). Сдвиг засылает 1 в бит
;7191      ;      31 регисра Q, а в WR0 будет 0. Бит 31 WR0 будет 0, так как N LONG и V LONG
;7192      ;      сброшены. Проверяются результаты для этих операндов.
;7193      ;   6. Повторение шага 2, за исключением того, что в WR2 загружается 80000001
;7194      ;      (отрицательное число).
;7195      ;   7. Выполнение COND.ADD&SHIFT WR[2] TO WR[0].Q. Высокий уровень сигнала
;7196      ;      MPLIER LSB H вызовет выпонение A PLUS B (80 00 00 01 плюс 1). Сдвиг засы-
;7197      ;      лает 1 в бит 0 и бит 30 WR0, а в регистре Q будет 0. Бит 31 WR0 будет 1,
;7198      ;      так как N LONG установлен и V LONG сброшен. Проверяются результаты для
;7199      ;      этих операндов.
;7200      ;   8. Повторение шага 2, за исключением того, что в WR2 и в WR0 загружается
```

```
;7201      ;       40 00 00 00.
;7202      ;    9. Выполнение COND.ADD&SHIFT WR[2] TO WR[0].Q. Высокий уровень сигнала
;7203      ;       MPLIER LSB H вызовет выпонение A PLUS B (40 00 00 00 плюс 40 00 00 00).
;7204      ;       Суммирование устанавливает бит 31 WR0, а сдвиг пересылает эту 1 в бит
;7205      ;       30 WR0. Бит 31 WR0 будет 0, так как N LONG и V LONG установлены. В ре-
;7206      ;       гистре Q будет 0. Проверка результатов для этих операндов.
;7207      ;    10.Повторение шага 2, за исключением того, что в WR2 и в WR0 загружается
;7208      ;       80 00 00 00.
;7209      ;    11.Выполнение COND.ADD&SHIFT WR[2] TO WR[0].Q. Высокий уровень сигнала
;7210      ;       MPLIER LSB H вызовет выполнение A PLUS B (80 00 00 00 плюс 80 00 00 00).
;7211      ;       Инструкция ADD сбросит WR0 и установит V LONG. Бит 31 WR0 будет=1, а в
;7212      ;       регистре Q будет 0. Бит 31 WR0 будет=1, так как N LONG сброшен и V LONG
;7213      ;       установлен. Проверка результатов для этих операндов.
;7214      ;    12.Повторение шага 2, за исключением того, что в WR2 загружается 2, а в WR0
;7215      ;       загружается 3.
;7216      ;    13.Выполнение COND.SUB&SHIFT WR[2] FROM WR[0].Q. Высокий уровень сигнала
;7217      ;       MPLIER LSB H вызовет выполнение B MINUS A (3 минус 2), за которым следует
;7218      ;       сдвиг вправо. Сдвиг засылает 1 в бит 31 регистра Q и оставляет сброшенным
;7219      ;       WR0. Бит 31 WR0 будет сброшен, так как N LONG и V LONG сброшены.
;7220      ;
;7221      ; ОШИБКИ:
;7222      ;
;7223      ;    ошибка 1 - WR0 работает неправильно при COND.ADD&SHIFT WR[2] TO WR[0].Q со
;7224      ;               следующими данными до сложения и сдвига: WR2=1,WR0=1,регистр Q=0,
;7225      ;               MPLIER=установлен, N LONG и V LONG сброшены после сложения.
;7226      ;    ошибка 2 - такая же, как и ошибка 1, только неправильно работает регистр Q.
;7227      ;    ошибка 3 - WR0 работат неправильно при COND.ADD&SHIFT WR[2] TO WR[0].Q со
;7228      ;               следующими данными до сложения и сдвига: WR2=1, WR0=1, регистр Q=0,
;7229      ;               MPLIER=сброшен, N LONG и V LONG оба сброшены после сложения.
;7230      ;    ошибка 4 - такая же, как и ошибка 3, только неправилно работает регистр Q.
;7231      ;    ошибка 5 - WR0 работает неправильно при COND.ADD&SHIFT WR[2] TO WR[0].Q со
;7232      ;               следующими данными до сложения и сдвига: WR2=80000000(H), WR0=1,
;7233      ;               регистр Q=0 MPLIER=установлен. N LONG должен быть установлен,
;7234      ;               а V LONG сброшен после сложения.
;7235      ;    ошибка 6 - такая же, как ошибка 5, только неправильно работает регистр Q.
;7236      ;    ошибка 7 - WR0 работает неправильно при COND.ADD&SHIFT WR[2] TO WR[0].Q со
;7237      ;               следующими данными до сложения и сдвига: WR2=40 00 00 00(H), WR0=
;7238      ;               40 00 00 00(H), регистр Q=0, MPLIER=установлен, N LONG и V LONG
;7239      ;               должны быть установлены после сложения.
;7240      ;    ошибка 8 - такая же, как и ошибка 7, только неправильно работает регистр Q.
;7241      ;    ошибка 9 - WR0 работает неправильно при COND.ADD&SHIFT WR[2] TO WR[0].Q со
;7242      ;               следующими данными до сложения и сдвига: WR2=80 00 00 00(H), WR0=
;7243      ;               80 00 00 00(H), регистр Q =0, MPLIER=установлен. N LONG должен быть
;7244      ;               сброшен, а V LONG установлен после сложения.
;7245      ;    ошибка A - такая же, как и ошибка 9, только неправильно работает регистр Q.
;7246      ;    ошибка B - WR0 работает неправильно при COND.SUB&SHIFT WR[2] FROM WR[0].Q
;7247      ;               со следующими данными до вычитания и сдвига: WR2=2, WR0=3, регистр
;7248      ;               Q=0, MPLIER=установлен, N LONG и V LONG должны быть сброшены после
;7249      ;               вычитания.
;7250      ;    ошибка C - такая же, как и ошибка B, только неправильно работает регистр Q.
;7251      ;
;7252      ; НАЛАДКА:
;7253      ;
;7254      ;    ОШИБКА 1 - Эта ошибка может быть в результате нескольких неисправностей.
;7255      ;       Прежде всего, необходимо проверить линии управления АЛУ во время инструкции
```

```
;7256      ;       COND.ADD&SHIFT WR[2] TO WR[0].Q. Источники (SRC) должны быть A.B. За значе-
;7257      ;       ниями сигналов управления необходимо обратиться к листингу содержимого управ-
;7258      ;       ляющей памяти путей данных и к описанию этого теста. Если линии управления
;7259      ;       неправильные, проверяется наличие 010001000(B) (биты CSR 22-14) на входах
;7260      ;       ПЗУ УПР.ФУНКЦИЕЙ АЛУ и ПМЛ УПР.ИСТ.ПРИЕМН.АЛУ, а также высокий уро-
;7261      ;       вень сигнала MPLIER LSB H на входе ПМЛ УПР.ИСТ.ПРИЕМН.АЛУ. Если эти
;7262      ;       входы правильные, подозревается ПЗУ УПР.ФУНКЦИЕЙ АЛУ или ПМЛ УПР.ИСТ.
;7263      ;       ПРИЕМН.АЛУ в зависимости от того, который выход неправильный. Если сигнал
;7264      ;       MPLIER LSB H имеет низкий уровень, необходимо проверить сигнал QSHF LSB H на
;7265      ;       входе ПМЛ АЛУ N,Z во время инструкции ASHRC WR[3].Q. На этом входе должен по-
;7266      ;       явиться высокий уровень. Сигнал Q SHF LSB H уже был проверен, но связь к этой
;7267      ;       ПМЛ может быть оборвана. Если вход правильный, а на выходе имеется низкий
;7268      ;       уровень, подозревается ПМЛ АЛУ N,Z. Если линии управления правильные, ошибка
;7269      ;       может быть в бите 31 WR0. Необходимо проверить низкий уровень сигнала RAM
;7270      ;       SHF MSB H на выходе ПМЛ СДВИГ.ДАННЫХ во время инструкции COND.ADD&SHIFT .
;7271      ;       WR[2] TO WR[0].Q. Если он неправильный, необходимо проверить низкий уровень
;7272      ;       сигналов N LONG и V LONG и наличие 4(H) на CSR 13-11. Если входы правильные,
;7273      ;       подозревается ПМЛ СДВИГ.ДАННЫХ.
;7274      ;
;7275      ;   ОШИБКА 2 - Эта ошибка указывает на неправильную работу узла регистра Q при
;7276      ;       выполнении инструкции COND.ADD&SHIFT WR[2] TO WR[0].Q. Прежде всего необ-
;7277      ;       ходимо проверить линии управления АЛУ, как и при ошибке 1. Если они правиль-
;7278      ;       ные, возможно, что неисправна ПМЛ СДВИГ.ДАННЫХ. Необходимо проверить наличие
;7279      ;       4(H) для CSR 13-11 на входах этой ПМЛ во время инструкции условного сложения
;7280      ;       Если они правильные, повидимому, ПМЛ дефектна.
;7281      ;
;7282      ;   ОШИБКА 3 - То же, что и при ошибке 1, со со следующими исключениями: MPLIER
;7283      ;       LSB H должен иметь низкий уровень, чтобы режим источника был 0.B. Сигнал Q
;7284      ;       SHF LSB H должен иметь низкий уровень на входе ПМЛ АЛУ N,Z, что приводит к
;7285      ;       формированию MPLIER LSB H низким уровнем.
;7286      ;
;7287      ;   ОШИБКА 4 - То же, что и при ошибке 2, со следующими исключениями: сигнал
;7288      ;       MPLIER LSB H должен иметь низкий уровень, чтобы режим источника был 0.B.
;7289      ;
;7290      ;   ОШИБКА 5 - То же, что и при ошибке 1, со следующими исключениями: сигнал RAM
;7291      ;       SHF MSB H должен иметь высокий уровень, и сигнал N LONG должен иметь высокий
;7292      ;       уровень на входе ПМЛ СДВИГ.ДАННЫХ.
;7293      ;
;7294      ;   ОШИБКА 6 - То же, что и при ошибке 2.
;7295      ;   ОШИБКА 7 - То же, что и при ошибке 1, со следующими исключениями: силналы N
;7296      ;       LONG и V LONG должны иметь высокий уровень на входе ПМЛ СДВИГ.ДАННЫХ. Сигнал
;7297      ;       RAM SHF MSB H должен иметь низкий уровень, как и при ошибке 1.
;7298      ;
;7299      ;   ОШИБКА 8 - То же, что и при ошибке 2.
;7300      ;   ОШИБКА 9 - То же, что и при ошибке 1, со следующими исключениями: сигналы RAM
;7301      ;       SHF LSB H и V LONG H должны иметь высокий уровень на входе ПМЛ СДВИГ.ДАННЫХ.
;7302      ;
;7303      ;   ОШИБКА A - То же, что и при ошибке 2.
;7304      ;   ОШИБКА B - Эта ошибка, скорее всего, показывает неисправную работу ПЗУ УПР.
;7305      ;       ФУНКЦИЕЙ АЛУ или ПМЛ УПР.ИСТ.ПРИЕМН.АЛУ. Входы CSR 22-14 должны быть
;7306      ;       0 1000 1000. Для информации о правильных значениях линий управления при
;7307      ;       COND.SUB&SHIFT WR[2] FROM WR[0].Q необходимо обратиться к описанию линий
;7308      ;       управления АЛУ и к описанию этого теста.
;7309      ;
;7310      ;   ОШИБКА C - То же, что и при ошибке B.
```

```
                    ;7311     ;
                    ;7312     T.C:
U 18CB,  B65E,15    ;7313               MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;7314                                                    ; состояния
U 18CC,  3E80,15    ;7315               MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления и состояния. Бит
                    ;7316                                                    ; 15 указывает начало теста для консольного процессора
U 18CD,  10E0,15    ;7317               MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для консольного процессора
                    ;7318     WAIT.TC.0:
U 18CE,  098C,E4    ;7319               JMP [WAIT.TC.0]                      ; зацикливание для ожидания ответа от консольного
                    ;7320                                                    ; процессора
U 18CF,  E58A,15    ;7321               CLR LS[ERROR.MASK]                   ; очистка маски ошибки
U 18D0,  65A0,15    ;7322               CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 18D1,  3642,15    ;7323               MOV LS[CPU] TO WR[0]                 ; загрузка 1 в WR0
U 18D2,  3E8C,15    ;7324               MOV WR[0] TO LS[MODULE.NUM]           ; установка кода модуля для модуля DAP
                    ;7325     LOOP.TC.1:
U 18D3,  8640,15    ;7326               MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 18D4,  8EB2,15    ;7327               MOV WR[0] TO LS[ERROR.NUMBER]         ; номер ошибки = 1
U 18D5,  8640,15    ;7328               MOV LS[#1] TO WR[0]                  ; WR0=1
U 18D6,  2001,15    ;7329               MOV WR[0] TO WR[2]                   ; WR2=1
U 18D7,  AAC0,15    ;7330               MOV WR[0] TO Q                       ; Q=1
U 18D8,  2F87,95    ;7331               CLR WR[3]                            ; WR3=0
U 18D9,  A319,95    ;7332               ASHRC WR[3].Q                        ; загрузка 0 в MPLIER LSB H и регистр Q
U 18DA,  2224,15    ;7333               COND.ADD&SHIFT WR[2] TO WR[0].Q
U 18DB,  7610,15    ;7334               MOV Q TO LS[T8]                      ; запоминание регистра Q
U 18DC,  3640,95    ;7335               MOV LS[#1] TO WR[1]                  ; ожидаемые данные (в WR0)=1
U 18DD,  0869,3C    ;7336               JSR [CHECK.RESULT]                   ;
U 18DE,  098D,34    ;7337               JMP [LOOP.TC.1]                      ; зацикливание при ошибке, если разрешено
U 18DF,  8870,0C    ;7338               JSR [INC.EN]                         ; увеличение номера ошибки до 2
                    ;7339     TC.2:
U 18E0,  8610,15    ;7340               MOV LS[T8] TO WR[0]                  ; выборка Q
U 18E1,  2F82,95    ;7341               CLR WR[1]                            ; ожидаемые данные (в регистре Q)=0
U 18E2,  0869,3C    ;7342               JSR [CHECK.RESULT]                   ;
U 18E3,  098D,34    ;7343               JMP [LOOP.TC.1]                      ; зацикливание при ошибке, если разрешено
                    ;7344     LOOP.TC.3:
U 18E4,  36C0,15    ;7345               MOV LS[#3(H)] TO WR[0]               ; загрузка 3 в WR0
U 18E5,  8EB2,15    ;7346               MOV WR[0] TO LS[ERROR.NUMBER]         ; установка номера ошибки=3
U 18E6,  8640,15    ;7347               MOV LS[#1] TO WR[0]                  ; WR0=1
U 18E7,  2001,15    ;7348               MOV WR[0] TO WR[2]                   ; WR2=1
U 18E8,  A880,15    ;7349               CLR Q                                ; Q=0
U 18E9,  A319,95    ;7350               ASHRC WR[3].Q                        ; загрузка MPLIER LSB H
U 18EA,  2224,15    ;7351               COND.ADD&SHIFT WR[2] TO WR[0].Q       ;
U 18EB,  7610,15    ;7352               MOV Q TO LS[T8]                      ; запоминание регистра Q
U 18EC,  2F82,95    ;7353               CLR WR[1]                            ; ожидаемые данные (в WR0)=0
U 18ED,  0869,3C    ;7354               JSR [CHECK.RESULT]                   ;
U 18EE,  898E,44    ;7355               JMP [LOOP.TC.3]                      ; зацикливание при ошибке, если разрешено
U 18EF,  8870,0C    ;7356               JSR [INC.EN]                         ; увеличение номера ошибки до 4
                    ;7357     TC.4:
U 18F0,  8610,15    ;7358               MOV LS[T8] TO WR[0]
U 18F1,  867E,95    ;7359               MOV LS[BIT31] TO WR[1]                ; ожидаемые данные (в регистре Q) = 80 00 00 00
U 18F2,  0869,3C    ;7360               JSR [CHECK.RESULT]
U 18F3,  898E,44    ;7361               JMP [LOOP.TC.3]                      ; зацикливание при ошибке, если разрешено
                    ;7362     LOOP.TC.5:
U 18F4,  3644,15    ;7363               MOV LS[#4] TO WR[0]                  ;
U 18F5,  2040,15    ;7364               INC WR[0]                            ;
U 18F6,  8EB2,15    ;7365               MOV WR[0] TO LS[ERROR.NUMBER]         ; установка номера ошибки=5
```

```
U 18F7, B640,15 ;7366            MOV LS[#1] TO WR[0]                    ; WR0=1
U 18F8, 2001,15 ;7367            MOV WR[0] TO WR[2]                     ;
U 18F9, C77F,15 ;7368            BIS LS[BIT31] TO WR[2]                 ; WR2=80 00 00 01
U 18FA, AAC0,15 ;7369            MOV WR[0] TO Q                         ; Q=1
U 18FB, A319,95 ;7370            ASHRC WR[03].Q                         ; загрузка MPLIER LSB H
U 18FC, 2224,15 ;7371            COND.ADD&SHIFT WR[2] TO WR[0].Q        ;
U 18FD, 7610,15 ;7372            MOV Q TO LS[T8]                        ; запоминание регистра Q
U 18FE, B67E,95 ;7373            MOV LS[BIT31] TO WR[1]                 ;
U 18FF, 477C,95 ;7374            BIS LS[BIT30] TO WR[1]                 ;
U 1900, 2042,95 ;7375            INC WR[1]                              ; ожидаемые данные (в WR0)=C0 00 00 01
U 1901, 0869,3C ;7376            JSR [CHECK.RESULT]                     ;
U 1902, 098F,44 ;7377            JMP [LOOP.TC.5]                        ; зацикливание при ошибке, если разрешено
U 1903, 8870,0C ;7378            JSR [INC.EN]                           ; увеличение номера ошибки до 6
                ;7379    TC.6:
U 1904, B610,15 ;7380            MOV LS[T8] TO WR[0]                    ; выборка регистра Q
U 1905, 2F82,95 ;7381            CLR WR[1]                              ; ожидаемые данные (в регистре Q)=0
U 1906, 0869,3C ;7382            JSR [CHECK.RESULT]                     ;
U 1907, 098F,44 ;7383            JMP [LOOP.TC.5]                        ; зацикливание при ошибке, если разрешено
                ;7384    LOOP.TC.7:
U 1908, B646,15 ;7385            MOV LS[#8] TO WR[0]                    ;
U 1909, C140,15 ;7386            SUB LS[#1] FROM WR[0]                  ;
U 190A, BE82,15 ;7387            MOV WR[0] TO LS[ERROR.NUMBER]          ; установка номера ошибки=7
U 190B, B67C,15 ;7388            MOV LS[BIT30] TO WR[0]                 ; WR0=40 00 00 00
U 190C, 2001,15 ;7389            MOV WR[0] TO WR[2]                     ; WR2=40 00 00 00
U 190D, 5840,15 ;7390            MOV LS[#1] TO Q                        ; Q=1
U 190E, A319,95 ;7391            ASHRC WR[3].Q                          ; загрузка MPLIER LSB H
U 190F, 2224,15 ;7392            COND.ADD&SHIFT WR[2] TO WR[0].Q        ;
U 1910, 7610,15 ;7393            MOV Q TO LS[T8]                        ; запоминание регистра Q
U 1911, 367C,95 ;7394            MOV LS[BIT30] TO WR[1]                 ; ожидаемые данные (в WR0)=40 00 00 00
U 1912, 0869,3C ;7395            JSR [CHECK.RESULT]                     ;
U 1913, 8990,84 ;7396            JMP [LOOP.TC.7]                        ; зацикливание при ошибке, если разрешено
U 1914, 8870,0C ;7397            JSR [INC.EN]                           ; увеличение номера ошибки до 8
                ;7398    TC.8:
U 1915, B610,15 ;7399            MOV LS[T8] TO WR[0]                    ; выборка регистра Q
U 1916, 2F82,95 ;7400            CLR WR[1]                              ; ожидаемые данные (в регистре Q)= 0
U 1917, 0869,3C ;7401            JSR [CHECK.RESULT]                     ;
U 1918, 8990,84 ;7402            JMP [LOOP.TC.7]                        ; зацикливание при ошибке, если разрешено
                ;7403    LOOP.TC.9:
U 1919, B646,15 ;7404            MOV LS[#8] TO WR[0]                    ;
U 191A, 2040,15 ;7405            INC WR[0]                              ;
U 191B, BE82,15 ;7406            MOV WR[0] TO LS[ERROR.NUMBER]          ; установка номера ошибки=9
U 191C, 367E,15 ;7407            MOV LS[BIT31] TO WR[0]                 ; WR0=80 00 00 00
U 191D, 2001,15 ;7408            MOV WR[0] TO WR[2]                     ; WR2=80 00 00 00
U 191E, 5840,15 ;7409            MOV LS[#1] TO Q                        ; Q=1
U 191F, A319,95 ;7410            ASHRC WR[3].Q                          ; загрузка MPLIER LSB H
U 1920, 2224,15 ;7411            COND.ADD&SHIFT WR[2] TO WR[0].Q        ;
U 1921, 7610,15 ;7412            MOV Q TO LS[T8]                        ; запоминание регистра Q
U 1922, B67E,95 ;7413            MOV LS[BIT31] TO WR[1]                 ; ожидаемые данные (в WR0)=80 00 00 00
U 1923, 0869,3C ;7414            JSR [CHECK.RESULT]                     ;
U 1924, 8991,94 ;7415            JMP [LOOP.TC.9]                        ; зацикливание при ошибке, если разрешено
U 1925, 8870,0C ;7416            JSR [INC.EN]                           ; увеличение номера ошибки до A
                ;7417    TC.A:
U 1926, B610,15 ;7418            MOV LS[T8] TO WR[0]                    ; выборка регистра Q
U 1927, 2F82,95 ;7419            CLR WR[1]                              ; ожидаемые данные (в регистре Q)=0
U 1928, 0869,3C ;7420            JSR [CHECK.RESULT]                     ;
```

```
U 1929, 8991,94 ;7421            JMP [LOOP.TC.9]                          ; зацикливание при ошибке, если разрешено
                ;7422    LOOP.TC.B:
U 192A, 8646,15 ;7423            MOV LS[#8] TO WR[0]                      ;
U 192B, C0C0,15 ;7424            ADD LS[#3(H)] TO WR[0]                   ;
U 192C, BE82,15 ;7425            MOV WR[0] TO LS[ERROR.NUMBER]            ; установка номера ошибки=B
U 192D, B643,15 ;7426            MOV LS[#2] TO WR[2]                      ; WR2=2
U 192E, 36C0,15 ;7427            MOV LS[#3(H)] TO WR[0]                   ; WR0=3
U 192F, 5840,15 ;7428            MOV LS[#1] TO Q                          ; Q=1
U 1930, A319,95 ;7429            ASHRC WR[3].Q                            ; загрузка MPLIER LSB H
U 1931, A264,15 ;7430            COND.SUB&SHIFT WR[2] FROM WR[0].Q        ;
U 1932, 7610,15 ;7431            MOV Q TO LS[T8]                          ; запоминание регистра Q
U 1933, 2F82,95 ;7432            CLR WR[1]                                ; ожидаемые данные (в WR0)=80 00 00 00
U 1934, 0869,3C ;7433            JSR [CHECK.RESULT]                       ;
U 1935, 8992,A4 ;7434            JMP [LOOP.TC.B]                          ; зацикливание при ошибке, если разрешено
U 1936, 8870,0C ;7435            JSR [INC.EN]                             ; увеличение номера ошибки до C
                ;7436    TC.C:
U 1937, B610,15 ;7437            MOV LS[T8] TO WR[0]                      ; выборка регистра Q
U 1938, B67E,95 ;7438            MOV LS[BIT31] TO WR[1]                   ; ожидаемые данные (в регистре Q)=0
U 1939, 0869,3C ;7439            JSR [CHECK.RESULT]                       ;
U 193A, 8992,A4 ;7440            JMP [LOOP.TC.B]                          ; зацикливание при ошибке, если разрешено
                ;7441    END.TC:
```

```
;7442      .PAGE "ТЕСТ D - тест беззнакового умножения (SI=UMUL) (модуль DAP)"
;7443      ;
;7444      ; ОПИСАНИЕ ТЕСТА:
;7445      ;
;7446      ;   Этот тест проверяет инструкцию беззнакового умножения и ПМЛ СДВИГ.ДАННЫХ.
;7447      ;   В этом тесте используется инструкция UNSIGNED.MUL WR TO WR.Q. Эта инструкция
;7448      ;   использует такой же режим условного сложения и сдвига, как и при умножении
;7449      ;   со знаком, проверенном раньше. Режим источника (SRC) зависит от значения
;7450      ;   бита 0 регистра Q до предыдущей операции сдвига. Сигналы управления для АЛУ
;7451      ;   во время условной инструкции следующие: CSR=MUL, FUNC=R.PLUS.S, DST=RSHF.RAM.Q,
;7452      ;   CIN=NOCIN. Режим источника бывает A.B, если регистр Q содержит 1 в бите 0 до
;7453      ;   предыдущего сдвига, или 0.B, если регистр Q содержал 0 в бите 0 до предыду-
;7454      ;   щего сдвига. Этот тест в поле SI устанавливает значение 5. Так как условное
;7455      ;   сложение и сдвиг были проверены в предыдущем тесте, этот тест использует опе-
;7456      ;   ранды только для создания режима источника, равного A.B. Этот тест вначале
;7457      ;   загружает 2 рабочих регистра и регистр Q данными и сдвигает Q регистр вправо
;7458      ;   для загрузки MPLIER LSB H, который управляет режимом источника для следующей
;7459      ;   инструкции. Тогда выполняется инструкция USIGNED.MUL WR[2] TO WR[0].Q и прове-
;7460      ;   ряется результат. Тестовые данные подбираются такими, чтобы происходил сдвиг
;7461      ;   в старший бит регистра Q.
;7462      ;
;7463      ; ПРЕДПОЛОЖЕНИЯ:
;7464      ;
;7465      ;   Предполагается, что предыдущий тест умножения со знаком выполнен успешно.
;7466      ;
;7467      ; ШАГИ ТЕСТА:
;7468      ;
;7469      ;   1.Установка маски ошибок, номера ошибки и номера модуля в местной памяти (для
;7470      ;     распечатки ошибок) и сброс предыдущего номера ошибки в местной памяти.
;7471      ;   2.Загрузка 1 в WR2, WR0 и регистр Q. Загрузка всех нулей в WR3 и выполнение
;7472      ;     ASHRC WR[3] Q. Этим загружается 1 в MPLIER LSB H  и остается 0 в регистре Q.
;7473      ;   3.Выполнение UNSIGNED.MUL WR[2] TO WR[0].Q. Высокий уровень сигнала MPLIER LSB H
;7474      ;     вызовет выполнение A PLUS B (1 плюс 1). Сдвиг засылает 1 в бит 0 WR0, а ре-
;7475      ;     гистр Q все еще содержит 0. Бит 31 WR0 будет 0, так как CARRY 32 H имеет низ-
;7476      ;     кий уровень. Проверка результата для этих операндов.
;7477      ;   4.Повторение шага 2, за исключением того, что в WR2 загружается C0 00 00 01(H)
;7478      ;     и в WR0 - 40 00 00 00(H).
;7479      ;   5.Выполнение UNSIGNED.MUL WR[2] TO WR[0].Q. Высокий уровень сигнала MPLIER LSB H
;7480      ;     вызовет выполнение A PLUS B (C0 00 00 01 плюс 60 00 00 00). Сложение генериру-
;7481      ;     ет CARRY 32 H, так что бит 31 WR0 будет установлен. В регистре Q бит 31 тоже
;7482      ;     будет установлен сдвигом из WR0 бита 0. Проверка результатов для этих операндов.
;7483      ;
;7484      ; ОШИБКИ:
;7485      ;
;7486      ;   ошибка 1 - неправильно работает WR0 при UNSIGNED.MUL WR[2] TO WR[0].Q со
;7487      ;               следующими данными до сложения и сдвига: WR2=1, WR0=1, регистр
;7488      ;               Q=0, MPLIER=установлен. После сложения сигнал CARRY 32 H будет
;7489      ;               иметь низкий уровень.
;7490      ;   ошибка 2 - такая же, что и ошибка 1, только неправильно работает регистр Q.
;7491      ;   ошибка 3 - неправильно работает WR0 при UNSIGNED.MUL WR[2] TO WR[0].Q со
;7492      ;               следующими данными до сложения и сдвига: WR2=C0000001(H),
;7493      ;               WR0=40000000(H), регистр Q=0, MPLIER=установлен. Сигнал CARRY 32 H
;7494      ;               будет установлен после сложения.
;7495      ;   ошибка 4 - такая же, как и ошибка 3, только неправильно работает регистр Q.
;7496      ;
```

```
                    ;7497      ; НАЛАДКА:
                    ;7498      ;
                    ;7499      ;   ОШИБКА 1 - Это ошибка, по-видимому в бите 31 WR0. Необходимо проверить низкий
                    ;7500      ;   уровень сигнала RAM SHF MSB H на выходе ПМЛ СДВИГ.ДАННЫХ во время инструкции
                    ;7501      ;   UNSIGNED.MUL WR[2] TO WR[0].Q. Если он неправильный, необходимо проверить
                    ;7502      ;   низкий уровень сигнала CARRY 32 H и наличие 5(H) для битов CSR 13-11 на входах
                    ;7503      ;   этой ПМЛ. Если входы правильные, подозревается ПМЛ СДВИГ.ДАННЫХ.
                    ;7504      ;
                    ;7505      ;   ОШИБКА 2 - Эта ошибка указывает на неправильную работу узла регистра Q при
                    ;7506      ;   выполнении инструкции UNSIGNED.MUL WR[2] TO WR[0].Q. По-видимому, неправильно
                    ;7507      ;   работает ПМЛ СДВИГ.ДАННЫХ. Необходимо проверить наличие 5(H) на входах этой
                    ;7508      ;   ПМЛ для битов CSR 13-11 во время инструкции условного сложения. Если они
                    ;7509      ;   правильные, по-видимому, неисправна ПМЛ.
                    ;7510      ;
                    ;7511      ;   ОШИБКА 3 - То же, что и при ошибке 1, со следующими исключениями: сигнал
                    ;7512      ;   RAM SHF MSB H должен иметь высокий уровень на выходе ПМЛ СДВИГ.ДАННЫХ при
                    ;7513      ;   высоком уровне сигнала CARRY 32 H.
                    ;7514      ;
                    ;7515      ;   ОШИБКА 4 - То же, что и при ошибке 2.
                    ;7516      ;
                    ;7517      T.D:
U 193B,  B65E,15    ;7518            MOV LSI[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;7519                                                  ; состояния
U 193C,  3E80,15    ;7520            MOV WR[0] TO LSI[CONTROL.STATUS]      ; установка бита 15 в слове управления и состояния. Бит
                    ;7521                                                  ; 15 указывает начало теста для конс. процессора
U 193D,  10E0,15    ;7522            MISC [SET.CP.ATTN]                    ; выдача сигнала CPU ATTN для конс.процессора
                    ;7523      WAIT.TD.0:
U 193E,  8993,E4    ;7524            JMP [WAIT.TD.0]                       ; зацикливание для ожидания ответа КОНС.ПРОЦЕССОРА
U 193F,  E58A,15    ;7525            CLR LSI[ERROR.MASK]                   ; очистка маски ошибок
U 1940,  65A0,15    ;7526            CLR LSI[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 1941,  3642,15    ;7527            MOV LSI[CPU] TO WR[0]                 ; установка бита 1 в WR0
U 1942,  3E8C,15    ;7528            MOV WR[0] TO LSI[MODULE.NUM]          ; установка кода модуля для модуля DAP
                    ;7529      LOOP.TD.1:
U 1943,  B640,15    ;7530            MOV LSI[#1] TO WR[0]                  ; установка 1 в WR0
U 1944,  BE82,15    ;7531            MOV WR[0] TO LSI[ERROR.NUMBER]        ; установка номера ошибки = 1
U 1945,  B640,15    ;7532            MOV LSI[#1] TO WR[0]                  ; WR0 = 1
U 1946,  2001,15    ;7533            MOV WR[0] TO WR[2]                    ; WR2 = 1
U 1947,  AAC0,15    ;7534            MOV WR[0] TO Q                        ; Q = 1
U 1948,  2F87,95    ;7535            CLR WR[3]                             ; WR3 = 0
U 1949,  A319,95    ;7536            ASHRC WR[3].Q                         ; загрузка нулем MPLIER LSB H и регистра Q
U 194A,  A22C,15    ;7537            UNSIGNED.MUL WR[2] TO WR[0].Q         ;
U 194B,  7610,15    ;7538            MOV Q TO LSI[T8]                      ; запоминание регистра Q
U 194C,  3640,95    ;7539            MOV LSI[#1] TO WR[1]                  ; ожидаемые данные (в WR0) = 1
U 194D,  0869,3C    ;7540            JSR [CHECK.RESULT]                    ;
U 194E,  8994,34    ;7541            JMP [LOOP.TD.1]                       ; зацикливание при ошибке, если разрешено
U 194F,  8870,0C    ;7542            JSR [INC.EN]                          ; увеличение номера ошибки до 2
                    ;7543      TD.2:
U 1950,  B610,15    ;7544            MOV LSI[T8] TO WR[0]                  ; выборка регистра Q
U 1951,  2F82,95    ;7545            CLR WR[1]                             ; ожидаемые данные (в регистре Q) = 0
U 1952,  0869,3C    ;7546            JSR [CHECK.RESULT]                    ;
U 1953,  8994,34    ;7547            JMP [LOOP.TD.1]                       ; зацикливание при ошибке, если разрешено
                    ;7548      LOOP.TD.3:
U 1954,  36C0,15    ;7549            MOV LSI[#3(H)] TO WR[0]               ;
U 1955,  BE82,15    ;7550            MOV WR[0] TO LSI[ERROR.NUMBER]        ; установка номера ошибки = 3
U 1956,  B67C,15    ;7551            MOV LSI[BIT30] TO WR[0]               ; WR0 = 40 00 00 00
```

```
U 1957, 2001,15 ;7552            MOV WR[0] TO WR[2]                          ;
U 1958, C77F,15 ;7553            BIS LS[BIT31] TO WR[2]                      ;
U 1959, 2045,15 ;7554            INC WR[2]                                   ; WR2 = C0 00 00 01
U 195A, 5840,15 ;7555            MOV LS[#1] TO Q                             ; Q = 1
U 195B, 2F87,95 ;7556            CLR WR[3]                                   ; WR3 = 0
U 195C, A319,95 ;7557            ASHRC WR[3],Q                               ; загрузка нулем MPLIER LSB H и регистра Q
U 195D, A22C,15 ;7558            UNSIGNED.MUL WR[2] TO WR[0].Q               ;
U 195E, 7610,15 ;7559            MOV Q TO LS[TB]                             ; запоминание регистра Q
U 195F, B67E,95 ;7560            MOV LS[BIT31] TO WR[1]                      ; ожидаемые данные (в WR0) = 80 00 00 00
U 1960, 0869,3C ;7561            JSR [CHECK.RESULT]                          ;
U 1961, 8995,44 ;7562            JMP [LOOP.TD.3]                             ; зацикливание при ошибке, если разрешено
U 1962, 8870,00 ;7563            JSR [INC.EN]                                ; увеличение номера ошибки до 4
                ;7564    TD.4:
U 1963, B610,15 ;7565            MOV LS[TB] TO WR[0]                         ; выборка регистра Q
U 1964, B67E,95 ;7566            MOV LS[BIT31] TO WR[1]                      ; ожидаемые данные (в регистре Q) = 80 00 00 00
U 1965, 0869,3C ;7567            JSR [CHECK.RESULT]                          ;
U 1966, 8995,44 ;7568            JMP [LOOP.TD.3]                             ; зацикливание при ошибке, если разрешено
                ;7569    END TD:
```

```
;7570      .PAGE "ТЕСТ E - тест схем расширения знака (модуль DAP)"
;7571      ;
;7572      ;   ОПИСАНИЕ ТЕСТА:
;7573      ;
;7574      ;   Этот тест проверяет ПМЛ УПР.РАСШИРЕНИЕМ ЗНАКА, часть 2-входового мультиплек-
;7575      ;   сора и три 8-миразрядных буфера, которые выполняют расширение знака до байта
;7576      ;   или слова. Часть этой схемы может быть проверена только с данными памяти и бу-
;7577      ;   дет проверена микродиагностикой контроллера памяти. Тест загружает регистр OS
;7578      ;   тестовыми данными - нулями с 7-ым битом, установленным или сброшенным. Выполняет-
;7579      ;   ся инструкция MOVE формата LS[OS] TO WR[0] и проверяются 32 бита. Если бит 7
;7580      ;   регистра OS был 0, все биты должны быть 0. Если бит 7 регистра OS был 1, тогда
;7581      ;   биты 31-15 должны быть=1. Этим проверяются выходы 12 и 13 ПМЛ УПР.РАСШИРЕНИЕМ
;7582      ;   ЗНАКА, а также вход OS 7 2-входового мультиплексора и выходы на ножках 7 и 9.
;7583      ;
;7584      ;  ПРЕДПОЛОЖЕНИЯ:
;7585      ;
;7586      ;   Предполагается, что предыдущие тесты регистра OS выполнены успешно.
;7587      ;
;7588      ;  ШАГИ ТЕСТА:
;7589      ;
;7590      ;   1.Установка маски ошибки, номера ошибки и номера модуля в местной памяти
;7591      ;     (для распечатки ошибок) и очистка предыдущего номера ошибки в местной
;7592      ;     памяти.
;7593      ;   2.Загрузка регистра OS нулями. Выполнение MOVE LS[OS] TO WR[0]. Проверка
;7594      ;     результата на нуль.
;7595      ;   3.Загрузка 80(H) в регистр OS. Выполнение MOVE LS[OS] TO WR[0]. Проверка
;7596      ;     результата на FF FF FF 80(H).
;7597      ;
;7598      ;  ОШИБКИ:
;7599      ;
;7600      ;   ошибка 1 - схемы расширения знака работают неправильно с положительными данными.
;7601      ;   ошибка 2 - схемы расширения знака работают неправильно с отрицательными данными.
;7602      ;
;7603      ;  НАЛАДКА:
;7604      ;
;7605      ;   ОШИБКА 1 - Прежде всего, необходимо проверить разрешение на ножке 1 трех
;7606      ;   8-миразрядных буферов, выходы которых поступают на шину D во время инструкции
;7607      ;   MOVE LS[OS] TO WR[0]. Все три разрешения должны иметь низкий уровень. Если
;7608      ;   нет, необходимо проверить низкий уровень на ножках 12 и 13 ПМЛ УПР.РАСШИРЕ-
;7609      ;   НИЕМ ЗНАКА. Если не оба выхода имеют низкий уровень, необходимо проверить
;7610      ;   входы: высокий уровень на входах CSR 21 H, CSR 17 H и CSR 18 H и низкий уро-
;7611      ;   вень на входах SEL REGS L и DISABLE D-BUS H. Если они правильные, по-видимо-
;7612      ;   му, ПМЛ УПР.РАСШИРЕНИЕМ ЗНАКА неисправна. Если сигнал разрешения для буферов
;7613      ;   правильный, тогда необходимо проверить низкий уровень на ножках 7 и 9 2-вхо-
;7614      ;   дового мультиплексора. Если хотя бы один выход имеет высокий уровень, необхо-
;7615      ;   димо проверить низкий уровень на ножках 4,3,12,13,1 и 15 и высокий уровень на
;7616      ;   ножке 2. Если они правильные, мультиплексор неисправен. Наконец, если оба
;7617      ;   выхода имеют низкий уровень, необходимо проверить низкий уровень на входах
;7618      ;   буферов, особенно если только один бит работает неправильно. Подозревается
;7619      ;   буфер, если входы правильные (необходимо убедиться, что на ножке 11 каждого
;7620      ;   буфера имеется высокий уровень).
;7621      ;
;7622      ;   ОШИБКА 2 - То же, что и при ошибке 1, за исключением того, что ножки 4,3,12,
;7623      ;   13,9 и 7 2-входового мультиплексора имеют высокий уровень.
;7624      T.E:
```

```
U 1967, B65E,15 ;7625            MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                ;7626                                                 ; состояния
U 1968, 3E80,15 ;7627            MOV WR[0] TO LS[CONTROL.STATUS]      ; установка бита 15 в слове управления и состояния. Бит
                ;7628                                                 ; 15 указывает начало теста для консольного процессора
U 1969, 10E0,15 ;7629            MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для консольного процессора
                ;7630    WAIT.TE.0:
U 196A, 0996,A4 ;7631            JMP [WAIT.TE.0]                      ; зацикливание для ожидания ответа консольного
                ;7632                                                 ; процессора
U 196B, E58A,15 ;7633            CLR LS[ERROR.MASK]                   ; очистка маски ошибки
U 196C, 65A0,15 ;7634            CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 196D, B640,15 ;7635            MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 196E, BEB2,15 ;7636            MOV WR[0] TO LS[ERROR.NUMBER]        ; установка номера ошибки для первой ошибки
U 196F, A3C0,15 ;7637            ROL WR[0]                            ; установка 2 в WR0
U 1970, 3E8C,15 ;7638            MOV WR[0] TO LS[MODULE.NUM]          ; установка кода модуля для модуля DAP
                ;7639    LOOP.TE.1:
U 1971, E5F8,15 ;7640            CLR LS[OS]                           ; OS = 00 00 00 00
U 1972, B6F8,15 ;7641            MOV LS[OS] TO WR[0]                  ;
U 1973, 2F82,95 ;7642            CLR WR[1]                            ; ожидаемые данные = 00 00 00 00
U 1974, 0869,3C ;7643            JSR [CHECK.RESULT]                   ;
U 1975, 0997,14 ;7644            JMP [LOOP.TE.1]                      ; зацикливание при ошибке, если разрешено
U 1976, 8870,0C ;7645            JSR [INC.EN]                         ; увеличение номера ошибки до 2
                ;7646    LOOP.TE.2:
U 1977, 364E,15 ;7647            MOV LS[#80] TO WR[0]                 ;
U 1978, 3EF8,15 ;7648            MOV WR[0] TO LS[OS]                  ; OS = 00 00 00 80
U 1979, B6F8,15 ;7649            MOV LS[OS] TO WR[0]
U 197A, 362C,95 ;7650            MOV LS[#FFFFFF00] TO WR[1]
U 197B, C74E,95 ;7651            BIS LS[#80] TO WR[1]                   ожидаемые данные = FF FF FF 80
U 197C, 0869,3C ;7652            JSR [CHECK.RESULT]                   ;
U 197D, 0997,74 ;7653            JMP [LOOP.TE.2]                      ; зацикливание при ошибке, если разрешено
                ;7654    END TE
```

```
                ;7656      .PAGE "ТЕСТ F - тест ПЗУ УПР.ФУНКЦИЕЙ АЛУ и ПМЛ УПР.ИСТ.ПРИЕМН.АЛУ (модуль DAP)"
                ;7657      ;
                ;7658      ; ОПИСАНИЕ ТЕСТА:
                ;7659      ;
                ;7660      ;    Этот тест проверяет каждую ячейку ПЗУ УПР.ФУНКЦИЕЙ АЛУ и все возможные входы
                ;7661      ;    ПМЛ УПР.ИСТОЧНИКОМ-ПРИЕМНИКОМ. Подразумевается проверка того, что любой мик-
                ;7662      ;    рокод инструкции будет управлять АЛУ так, как ожидается, даже если данная ин-
                ;7663      ;    струкция не была проверена в предыдущих тестах. Некоторые инструкции, кото-
                ;7664      ;    рые уже проверены, в этом тесте не проверяются. Тест разделен на секции,
                ;7665      ;    соответствующие типу проверяемой инструкции, т.е. BASIC, MOVE, EXTENDED,
                ;7666      ;    MEM.REQ., MISC., JUMP и DECODE. В этом тесте память запрещена. Проверяются
                ;7667      ;    только режимы АЛУ. Нет смысла расширять это описание. Вместо этого, необхо-
                ;7668      ;    димо обращаться к комментариям листинга.
                ;7669      ;
                ;7670      ; ПРЕДПОЛОЖЕНИЯ:
                ;7671      ;
                ;7672      ;    Предполагается, что все предыдущие тесты выполнены успешно.
                ;7673      ;
                ;7674      ; ШАГИ ТЕСТА:
                ;7675      ;
                ;7676      ;    См. комментарий листинга.
                ;7677      ;
                ;7678      ; ОШИБКИ:
                ;7679      ;
                ;7680      ;    Другие данные: адрес ПЗУ УПР.ФУНКЦИЕЙ АЛУ.
                ;7681      ;
                ;7682      ;    ошибка 1 - JUMP микроинструкция изменила рабочий регистр
                ;7683      ;    ошибка 2 - JUMP.OS микроинструкция изменила рабочий регистр
                ;7684      ;    ошибка 3 - MISC микроинструкция изменила рабочий регистр
                ;7685      ;    ошибка 4 - PORT микроинструкция изменила рабочий регистр
                ;7686      ;    ошибка 5 - ошибка микроинструкции EXTENDED
                ;7687      ;    ошибка 6 - ошибка микроинструкции BASIC
                ;7688      ;    ошибка 7 - ошибка микроинструкции MOVE
                ;7689      ;
                ;7690      ; НАЛАДКА:
                ;7691      ;
                ;7692      ;    Любая ошибка, и все ошибки затрагивают или ПЗУ УПР.ФУНКЦИЕЙ АЛУ или ПМЛ
                ;7693      ;    УПР.ИСТ.ПРИЕМН.АЛУ. Подозреваемые выходы могут быть определены из
                ;7694      ;    листинга для линий управления АЛУ и режима (см. содержимое управляющей памяти
                ;7695      ;    путей данных). Таким образом, неисправность может быть определена до одной
                ;7696      ;    из двух микросхем, упомянутых выше.
                ;7697      ;
                ;7698     T.F:
U 197E, B65E,15 ;7699          MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                ;7700                                               ; состояния
U 197F, 3E80,15 ;7701          MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления и состояния.Бит
                ;7702                                               ; 15 указывает начало теста для консольного процессора
U 1980, 10E0,15 ;7703          MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для консольного процессора
                ;7704     WAIT.TF.0:
U 1981, 0998,14 ;7705          JMP [WAIT.TF.0]                      ; зацикливание для ожидания ответа консольного
                ;7706                                               ; процессора
U 1982, E58A,15 ;7707          CLR LS[ERROR.MASK]                   ; очистка маски ошибки
U 1983, 65A0,15 ;7708          CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 1984, B640,15 ;7709          MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 1985, BE82,15 ;7710          MOV WR[0] TO LS[ERROR.NUMBER]        ; установка номера ошибки для первой ошибки
```

```
U 1986, A3C0,15 ;7711            ROL WR[0]                              ; установка  в WR0
U 1987, 3E8C,15 ;7712            MOV WR[0] TO LS[MODULE.NUM]             ; установка кода модуля для модуля DAP
U 1988, 8870,FC ;7713            JSR [ASSERT.OTHER]                     ; будет использовано поле "ДРУГИЕ ДАННЫЕ"
                ;7714     ;
                ;7715     ; ТЕСТЫ ИНСТРУКЦИИ JUMP
                ;7716     ;
                ;7717    TF.1:
U 1989, B699,95 ;7718            MOV LS[ALTER.ODD] TO WR[3]              ; установка тестовых данных в WR3
U 198A, B64A,15 ;7719            MOV LS[#20] TO WR[0]                    ;
  198B, BE88,15 ;7720            MOV WR[0] TO LS[ADDRESS.DATA]           ; установка начального значения 20 поля "ДРУГИЕ ДАННЫЕ"
                ;7721    LOOP.TF.1.20:
U 198C, 083F,FC ;7722            JSR [RETURN.3FF]                       ; переход на определенную ячейку и возврат
U 198D, 8A6F,6C ;7723            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 198E, 8998,C4 ;7724            JMP [LOOP.TF.1.20]                     ; зацикливание при ошибке, если разрешено
U 198F, 9A70,1C ;7725            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7726     LOOP.TF.1.21:
U 1990, 8871,8C ;7727            JSR [RETURN.718]                       ; переход на определенную ячейку и возврат
U 1991, 8A6F,6C ;7728            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 1992, 0999,04 ;7729            JMP [LOOP.TF.1.21]                     ; зацикливание при ошибке, если разрешено
U 1993, 8A70,1C ;7730            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ,7731    LOOP.TF.1.22:
U 1994, 8881,8C ;7732            JSR [RETURN.818]                       ; переход на определенную ячейку и возврат
U 1995, 8A6F,6C ;7733            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 1996, 8999,44 ;7734            JMP [LOOP.TF.1.22]                     ; зацикливание при ошибке, если разрешено
U 1997, 8A70,1C ;7735            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7736    LOOP.TF.1.23:
U 1998, 0881,9C ;7737            JSR [RETURN.C18]                       ; переход на определенную ячейку и возврат
U 1999, 8A6F,6C ;7738            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 199A, 8999,B4 ;7739            JMP [LOOP.TF.1.23]                     ; зацикливание при ошибке, если разрешено
U 199B, 8A70,1C ;7740            JSR [ INC.OTHER]                       ; увеличение "ДРУГИХ" данных
                ,7741    LOOP.TF.1.24:
U 199C, 8901,8C ;7742            JSR [RETURN.1018]                      ; переход на определенную ячейку и возврат
U 199D, 8A6F,6C ;7743            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 199E, 0999,C4 ;7744            JMP [LOOP.TF.1.24]                     ; зацикливание при ошибке, если разрешено
U 199F, 8A70,1C ;7745            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7746    LOOP.TF.1.25:
U 19A0, 0941,8C ;7747            JSR [RETURN.1418]                      ; переход на определенную ячейку и возврат
U 19A1, 8A6F,6C ;7748            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 19A2, 099A,04 ;7749            JMP [LOOP.TF.1.25]                     ; зацикливание при ошибке, если разрешено
U 19A3, 8A70,1C ;7750            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7751    LOOP.TF.1.26:
U 19A4, 89A1,8C ;7752            JSR [RETURN.1A18]                      ; переход на определенную ячейку и возврат
U 19A5, 8A6F,6C ;7753            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 19A6, 899A,44 ;7754            JMP [LOOP.TF.1.26]                     ; зацикливание при ошибке, если разрешено
U 19A7, 8A70,1C ;7755            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7756    LOOP.TF.1.27:
U 19A8, 89C1,8C ;7757            JSR [RETURN.1C18]                      ; переход на определенную ячейку и возврат
U 19A9, 8A6F,6C ;7758            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 19AA, 899A,B4 ;7759            JMP [LOOP.TF.1.27]                     ; зацикливание при ошибке, если разрешено
U 19AB, 8A70,1C ;7760            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
                ;7761    LOOP.TF.1.28:
U 19AC, 8A01,8C ;7762            JSR [RETURN.2018]                      ; переход на определенную ячейку и возврат
U 19AD, 8A6F,6C ;7763            JSR [CHECK.WR3]                        ; проверка, что данные в WR3 остались правильными
U 19AE, 099A,C4 ;7764            JMP [LOOP.TF.1.28]                     ; зацикливание при ошибке, если разрешено
U 19AF, 8A70,1C ;7765            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных
```

```
                    ;7766    LOOP.TF.1.29:
U 19B0, 0A41,8C     ;7767            JSR [RETURN.2418]                  ; переход на определенную ячейку и возврат
U 19B1, 8A6F,6C     ;7768            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19B2, 899B,04     ;7769            JMP [LOOP.TF.1.29]                 ; зацикливание при ошибке, если разрешено
U 19B3, 8A70,1C     ;7770            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7771    LOOP.TF.1.2A:
U 19B4, 0AB1,8C     ;7772            JSR [RETURN.2818]                  ; переход на определенную ячейку и возврат
U 19B5, 8A6F,6C     ;7773            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19B6, 099B,44     ;7774            JMP [LOOP.TF.1.2A]                 ; зацикливание при ошибке, если разрешено
U 19B7, 8A70,1C     ;7775            JSR [INC.OTHER]                    ; увеличение других данных
                    ;7776    LOOP.TF.1.2B:
U 19B8, 8AC1,8C     ;7777            JSR [RETURN.2C18]                  ; переход на определенную ячейку и возврат
U 19B9, 8A6F,6C     ;7778            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19BA, 099B,84     ;7779            JMP [LOOP.TF.1.2B]                 ; зацикливание при ошибке, если разрешено
U 19BB, 8A70,1C     ;7780            JSR [INC.OTHER]                    ; увеличение других данных
                    ;7781    LOOP.TF.1.2C:
U 19BC, 0B01,8C     ;7782            JSR [RETURN.3018]                  ; переход на определенную ячейку и возврат
U 19BD, 8A6F,6C     ;7783            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19BE, 899B,C4     ;7784            JMP [LOOP.TF.1.2C]                 ; зацикливание при ошибке, если разрешено
U 19BF, 8A70,1C     ;7785            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7786    LOOP.TF.1.2D:
U 19C0, 8B41,8C     ;7787            JSR [RETURN.3418]                  ; переход на определенную ячейку и возврат
U 19C1, 8A6F,6C     ;7788            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались праильными
U 19C2, 099C,04     ;7789            JMP [LOOP.TF.1.2D]                 ; зацикливание при ошибке, если разрешено
U 19C3, 8A70,1C     ;7790            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7791    LOOP.TF.1.2E:
U 19C4, 8B81,8C     ;7792            JSR [RETURN.3818]                  ; переход на определенную ячейку и возврат
U 19C5, 8A6F,6C     ;7793            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19C6, 899C,44     ;7794            JMP [LOOP.TF.1.2E]                 ; зацикливание при ошибке, если разрешено
U 19C7, 8A70,1C     ;7795            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7796    LOOP.TF.1.2F:
U 19C8, 0BC1,8C     ;7797            JSR [RETURN.3C18]                  ; переход на определенную ячейку и возврат
U 19C9, 8A6F,6C     ;7798            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19CA, 899C,84     ;7799            JMP [LOOP.TF.1.2F]                 ; зацикливание при ошибке, если разрешено
                    ;7800    TF.2:
U 19CB, 8870,0C     ;7801            JSR [INC.EN]                       ; увеличение номера ошибки до 2
U 19CC, E5F8,15     ;7802            CLR LS[OS]                         ; обеспечение, что OS сброшен
U 19CD, B64A,15     ;7803            MOV LS[#20] TO WR[0]               ;
U 19CE, C048,15     ;7804            ADD LS[#10] TO WR[0]               ;
U 19CF, BE88,15     ;7805            MOV WR[0] TO LS[ADDRESS.DATA]      ; установка 30 для начала поля "ДРУГИХ" данных
                    ;7806    LOOP.TF.2.30:
U 19D0, 8C3F,FC     ;7807            JSR.OS [RETURN.3FF]                ; переход на определенную ячейку и возврат
U 19D1, 8A6F,6C     ;7808            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19D2, 899D,04     ;7809            JMP [LOOP.TF.2.30]                 ; зацикливание при ошибке, если разрешено
U 19D3, 8A70,1C     ;7810            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7811    LOOP.TF.2.31:
U 19D4, 0C71,8C     ;7812            JSR.OS [RETURN.718]                ; переход на определенную ячейку и возврат
U 19D5, 8A6F,6C     ;7813            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19D6, 099D,44     ;7814            JMP [LOOP.TF.2.31]                 ; зацикливание при ошибке, если разрешено
U 19D7, 8A70,1C     ;7815            JSR [INC.OTHER]                    ; увеличение "ДРУГИХ" данных
                    ;7816    LOOP.TF.2.32:
U 19D8, 0C81,8C     ;7817            JSR.OS [RETURN.818]                ; переход на определенную ячейку и возврат
U 19D9, 8A6F,6C     ;7818            JSR [CHECK.WR3]                    ; проверка, что данные в WR3 остались правильными
U 19DA, 099D,84     ;7819            JMP [LOOP.TF.2.32]                 ; зацикливание при ошибке, если разрешено
U 19DB, 8A70,1C     ;7820            JSR [INC.OTHER]                      увеличение "ДРУГИХ" данных
```

```
                    ;7821    LOOP.TF.2.33:
U 19DC, 8C81,9C     ;7822             JSR.OS [RETURN.C18]          ; переход на определенную ячейку и возврат
U 19DD, 8A6F,6C     ;7823             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19DE, 899D,C4     ;7824             JMP [LOOP.TF.2.33]           ; зацикливание при ошибке, если разрешено
U 19DF, 8A70,1C     ;7825             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7826    LOOP.TF.2.34:
U 19E0, 0D01,8C     ;7827             JSR.OS [RETURN.1018]         ; переход на определенную ячейку и возврат
U 19E1, 8A6F,6C     ;7828             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19E2, 899E,04     ;7829             JMP [LOOP.TF.2.34]           ; зацикливание при ошибке, если разрешено
U 19E3, 8A70,1C     ;7830             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7831    LOOP.TF.2.35:
U 19E4, 8D41,8C     ;7832             JSR.OS [RETURN.1418]         ; переход на определенную ячейку и возврат
U 19E5, 8A6F,6C     ;7833             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19E6, 099E,44     ;7834             JMP [LOOP.TF.2.35]           ; зацикливание при ошибке, если разрешено
U 19E7, 8A70,1C     ;7835             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ,7836    LOOP.TF.2.36:
U 19E8, 0DA1,8C     ;7837             JSR.OS [RETURN.1A18]         ; переход на определенную ячейку и возврат
U 19E9, 8A6F,6C     ;7838             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19EA, 099E,84     ;7839             JMP [LOOP.TF.2.36]           ; зацикливание при ошибке, если разрешено
U 19EB, 8A70,1C     ;7840             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7941    LOOP TF.2.37:
U 19EC, 0DC1,8C     ;7842             JSR.OS [RETURN.1C18]         ; переход на определенную ячейку и возврат
U 19ED. 8A6F,6C     ;7843             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19EE, 899E,C4     ;7844             JMP [LOOP.TF.2.37]           ; зацикливание при ошибке, если разрешено
U 19EF, 8A70,1C     ;7845             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7846    LOOP.TF.2.38:
U 19F0. 0E01,8C     ;7847             JSR.OS [RETURN.2018]         ; переход на определенную ячейку и возврат
U 19F1. 8A6F,6C     ;7848             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19F2, 099F,04     ;7849             JMP [LOOP.TF.2.38]           ; зацикливание при ошибке, если разрешено
U 19F3, 8A70,1C     ;7850             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7951    LOOP.TF 2.39:
U 19F4, 8E41,8C     ;7852             JSR.OS [RETURN.2418]         ; переход на определенную ячейку и возврат
U 19F5, 8A6F,6C     ;7853             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19F6, 899F,44     ;7854             JMP [LOOP.TF.2.39]           ; зацикливание при ошибке, если разрешено
U 19F7, 8A70,1C     ;7855             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7856    LOOP.TF.2.3A:
U 19F8, 8E81,8C     ;7857             JSR.OS [RETURN.2818]         ; переход на определенную ячейку и возврат
U 19F9, 8A6F,6C     ;7858             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19FA. 899F,84     ;7859             JMP [LOOP.TF.2.3A]           ; зацикливание при ошибке, если разрешено
U 19FB, 8A70,1C     ;7860             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7861    LOOP.TF.2.3B:
U 19FC, 0EC1,8C     ;7862             JSR.OS [RETURN.2C18]         ; переход на определенную ячейку и возврат
U 19FD, 8A6F,6C     ;7863             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 19FE, 099F,C4     ;7864             JMP [LOOP.TF.2.3B]           ; зацикливание при ошибке, если разрешено
U 19FF, 8A70,1C     ;7865             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7866    LOOP.TF.2.3C:
U 1A00, 8F01,8C     ;7867             JSR.OS [RETURN.3018]         ; переход на определенную ячейку и возврат
U 1A01, 8A6F,6C     ;7868             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 1A02, 09A0,04     ;7869             JMP [LOOP.TF.2.3C]           ; зацикливание при ошибке, если разрешено
U 1A03, 8A70,1C     ;7870             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
                    ;7871    LOOP.TF.2.3D:
U 1A04, 0F41,8C     ;7872             JSR.OS [RETURN.3418]         ; переход на определенную ячейку и возврат
U 1A05, 8A6F,6C     ;7873             JSR [CHECK.WR3]              ; проверка, что данные в WR3 остались правильными
U 1A06, 89A0,44     ;7874             JMP [LOOP.TF.2.3D]           ; зацикливание при ошибке, если разрешено
U 1A07, 8A70,1C     ;7875             JSR [INC.OTHER]              ; увеличение "ДРУГИХ" данных
```

```
                ;7876    LOOP.TF.2.3E:
U 1A08, 0FB1,8C ;7877            JSR.OS [RETURN.3818]                    ; переход на определенную ячейку и возврат
U 1A09, 8A6F,6C ;7878            JSR [CHECK.WR3]                         ; проверка, что данные в WR3 остались правильными
U 1A0A, 89A0,84 ;7879            JMP [LOOP.TF.2.3E]                      ; зацикливание при ошибке, если разрешено
U 1A0B, 8A70,1C ;7880            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных
                ;7881    LOOP.TF.2.3F:
U 1A0C, 8FC1,8C ;7882            JSR.OS [RETURN.3C18]                    ; переход на определенную ячейку и возврат
U 1A0D, 8A6F,6C ;7883            JSR [CHECK.WR3]                         ; проверка, что данные в WR3 остались правильными
U 1A0E, 09A0,C4 ;7884            JMP [LOOP.TF.2.3F]                      ; зацикливание при ошибке, если разрешено
U 1A0F, 89A1,A4 ;7885            JMP [L.1A1A]                            ;
                ;7886     ;
                ;7887     ; ИНСТРУКЦИЯ MISC
                ;7888     ;
                ;7889     ;
                ;7890    1A1A:
                ;7891    L.1A1A:
                ;7892    TF.3:
U 1A1A, 8870,0C ;7893            JSR [INC.EN]                            ; увеличение номера ошибки до 3
U 1A1B, B64C,15 ;7894            MOV LS[#40] TO WR[0]                    ;
U 1A1C, 2100,15 ;7895            DEC WR[0]                               ;
U 1A1D, BE88,15 ;7896            MOV WR[0] TO LS[ADDRESS.DATA]           ; установка начального значения 3F в поле "ДРУГИЕ
U 1A1E, 369A,15 ;7897            MOV LS[ALTER.EVEN] TO WR[0]
                ;7898                                                    ; ДАННЫЕ"
                ;7899    LOOP.TF.3.3F:
U 1A1F, 9030,15 ;7900            MISC [CLR.S0]                           ;
U 1A20, 8A6F,CC ;7901            JSR [CHECK.WR0]                         ; проверка, что инструкция MISC не меняет WR0
U 1A21, 89A1,F4 ;7902            JMP [LOOP.TF.3.3F]                      ; зацикливание при ошибке, если разрешено
U 1A22, 8A70,1C ;7903            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных
                ;7904    LOOP.TF.3.40:
U 1A23, 9050,15 ;7905            MISC [SET S0]                           ;
U 1A24, 8A6F,CC ;7906            JSR [CHECK.WR0]                         ; проверка, что инструкция MISC не меняет WR0
U 1A25, B9A2,34 ;7907            JMP [LOOP.TF.3.40]                      ; зацикливание при ошибке, если разрешено
U 1A26, 8A70,1C ;7908            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных
                ;7909    LOOP.TF.3.41:
U 1A27, 1080,15 ;7910            MISC [SET.ACC.I.O]                      ;
U 1A28, 8A6F,CC ;7911            JSR [CHECK.WR0]                         ; проверка, что инструкция MISC не меняет WR0
U 1A29, 09A2,74 ;7912            JMP [LOOP.TF.3.41]                      ; зацикливание при ошибке, если разрешено
U 1A2A, 8A70,1C ;7913            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных
                ;7914    LOOP.TF.3.42:
U 1A2B, 90F0,15 ;7915            MISC [NOP]                              ;
U 1A2C, 8A6F,CC ;7916            JSR [CHECK.WR0]                         ; проверка, что инструкция MISC не меняет WR0
U 1A2D, 09A2,B4 ;7917            JMP [LOOP.TF.3.42]                      ; зацикливание при ошибке, если разрешено
                ;7918    TF.4:
U 1A2E, 8870,0C ;7919            JSR [INC.EN]                            ; увеличение номера ошибки до 4
U 1A2F, B64C,15 ;7920            MOV LS[#40] TO WR[0]                    ;
U 1A30, 4748,15 ;7921            BIS LS[#10] TO WR[0]                    ;
U 1A31, 2100,15 ;7922            DEC WR[0]                               ;
U 1A32, BE88,15 ;7923            MOV WR[0] TO LS[ADDRESS.DATA]           ; установка начального значения 4F поля "ДРУГИЕ ДАННЫЕ"
U 1A33, 369A,15 ;7924            MOV LS[ALTER.EVEN] TO WR[0]             ;
                ;7925    LOOP.TF.4.4F:
U 1A34, 1400,15 ;7926            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/0,PORT.FUNC/READ ;
U 1A35, 8A6F,CC ;7927            JSR [CHECK.WR0]                         ; проверка, что инструкция PORT не меняет WR0
U 1A36, 89A3,44 ;7928            JMP [LOOP.TF.4.4F]                      ; зацикливание при ошибке, если разрешено
U 1A37, 8A70,1C ;7929            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных
                ;7930    LOOP.TF.4.50:
```

```
U 1A3B, 9440,15 ;7931            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/0,PORT.FUNC/CONTROL ;
U 1A39, 8A6F,CC ;7932            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A3A, 89A3,84 ;7933            JMP [LOOP.TF.4.50]                       ; зацикливание при ошибке, если разрешено
U 1A3B, 8A70,1C ;7934            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7935    LOOP.TF.4.51:
U 1A3C, 9480,15 ;7936            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/1,PORT.FUNC/READ ;
U 1A3D, 8A6F,CC ;7937            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A3E, 09A3,C4 ;7938            JMP [LOOP.TF.4.51]                       ; зацикливание при ошибке, если разрешено
U 1A3F, 8A70,1C ;7939            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7940    LOOP.TF.4.52:
U 1A40, 14C0,15 ;7941            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/1,PORT.FUNC/CONTROL ;
U 1A41, 8A6F,CC ;7942            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A42, 89A4,04 ;7943            JMP [LOOP.TF.4.52]                       ; зацикливание при ошибке, если разрешено
U 1A43, 8A70,1C ;7944            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7945    LOOP.TF.4.53:
U 1A44, 9500,15 ;7946            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/2,PORT.FUNC/READ ;
U 1A45, 8A6F,CC ;7947            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A46, 09A4,44 ;7948            JMP [LOOP.TF.4.53]                       ; зацикливание при ошибке, если разрешено
U 1A47, 8A70,1C ;7949            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7950    LOOP.TF.4.54:
U 1A48, 1540,15 ;7951            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/2,PORT.FUNC/CONTROL ;
U 1A49, 8A6F,CC ;7952            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A4A, 09A4,84 ;7953            JMP [LOOP.TF.4.54]                       ; зацикливание при ошибке, если разрешено
U 1A4B, 8A70,1C ;7954            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7955    LOOP.TF.4.55:
U 1A4C, 1580,15 ;7956            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/3,PORT.FUNC/READ ;
U 1A4D, 8A6F,CC ;7957            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A4E, 89A4,C4 ;7958            JMP [LOOP.TF.4.55]                       ; зацикливание при ошибке, если разрешено
U 1A4F, 8A70,1C ;7959            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7960    LOOP.TF.4.56:
U 1A50, 95C0,15 ;7961            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/3,PORT.FUNC/CONTROL ;
U 1A51, 8A6F,CC ;7962            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A52, 09A5,04 ;7963            JMP [LOOP.TF.4.56]                       ; зацикливание при ошибке, если разрешено
U 1A53, 8A70,1C ;7964            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7965    LOOP.TF.4.57:
U 1A54, 9600,15 ;7966            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/4,PORT.FUNC/READ ;
U 1A55, 8A6F,CC ;7967            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A56, 89A5,44 ;7968            JMP [LOOP.TF.4.57]                       ; зацикливание при ошибке, если разрешено
U 1A57, 8A70,1C ;7969            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7970    LOOP.TF.4.58:
U 1A58, 1640,15 ;7971            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/4,PORT.FUNC/CONTROL ;
U 1A59, 8A6F,CC ;7972            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A5A, 89A5,84 ;7973            JMP [LOOP.TF.4.58]                       ; зацикливание при ошибке, если разрешено
U 1A5B, 8A70,1C ;7974            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7975    LOOP.TF.4.59:
U 1A5C, 1680,15 ;7976            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/5,PORT.FUNC/READ ;
U 1A5D, 8A6F,CC ;7977            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A5E, 09A5,C4 ;7978            JMP [LOOP.TF.4.59]                       ; зацикливание при ошибке, если разрешено
U 1A5F, 8A70,1C ;7979            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7980    LOOP.TF.4.5A:
U 1A60, 96C0,15 ;7981            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/5,PORT.FUNC/CONTROL ;
U 1A61, 8A6F,CC ;7982            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A62, 09A6,04 ;7983            JMP [LOOP.TF.4.5A]                       ; зацикливание при ошибке, если разрешено
U 1A63, 8A70,1C ;7984            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7985    LOOP.TF.4.5B:
```

```
U 1A64, 1700,15 ;7986            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/6,PORT.FUNC/READ ;
U 1A65, 8A6F,CC ;7987            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A66, 89A6,44 ;7988            JMP [LOOP.TF.4.5B]                       ; зацикливание при ошибке, если разрешено
U 1A67, 8A70,1C ;7989            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7990    LOOP.TF.4.5C:
U 1A68, 9740,15 ;7991            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/6,PORT.FUNC/CONTROL ;
U 1A69, 8A6F,CC ;7992            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A6A, 89A6,84 ;7993            JMP [LOOP.TF.4.5C]                       ; зацикливание при ошибке, если разрешено
U 1A6B, 8A70,1C ;7994            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;7995    LOOP.TF.4.5D:
U 1A6C, 9780,15 ;7996            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/7,PORT.FUNC/READ ;
U 1A6D, 8A6F,CC ;7997            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A6E, 09A6,C4 ;7998            JMP [LOOP.TF.4.5D]                       ; зацикливание при ошибке, если разрешено
U 1A6F, 8A70,1C ;7999            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных
                ;8000    LOOP.TF.4.5E:
U 1A70, 17C0,15 ;8001            OPC2/MISC,MISC.PORT/PORT,R.W.FUNC/0,PORT.SELECT/7,PORT.FUNC/CONTROL ;
U 1A71, 8A6F,CC ;8002            JSR [CHECK.WR0]                          ; проверка, что инструкция PORT не меняет WR0
U 1A72, 89A7,04 ;8003            JMP [LOOP.TF.4.5E]                       ; зацикливание при ошибке, если разрешено
                ;8004     ;
                ;8005     ; ТЕСТЫ ИНСТРУКЦИЙ EXTENDED
                ;8006     ;
                ;8007    TF.5:
U 1A73, 8870,0C ;8008            JSR [INC.EN]                             ; увеличение номера ошибки до 5
U 1A74, 364E,15 ;8009            MOV LS[#80] TO WR[0]                     ;
U 1A75, 8E88,15 ;8010            MOV WR[0] TO LS[ADDRESS.DATA]            ; установка начального значения 80 поля "ДРУГИЕ ДАННЫЕ"
                ;8011    LOOP.TF.5.80:
U 1A76, 2F80,15 ;8012            CLR WR[0]                                ;
U 1A77, 3698,95 ;8013            MOV LS[ALTER.ODD] TO WR[1]               ; установка данных
U 1A78, 2002,15 ;8014            MOV WR[1] TO WR[0]                       ;
U 1A79, 0869,3C ;8015            JSR [CHECK.RESULT]                       ; проверка операции WR.PLUS.0.TO.WR
U 1A7A, 89A7,64 ;8016            JMP [LOOP.TF.5.80]                       ; зацикливание при ошибке, если разрешено
U 1A7B, 8A70,1C ;8017            JSR [INC.OTHER]                          ; предварительно проверенная операция WR.INC.WR
U 1A7C, 8A70,1C ;8018            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных до 82
                ;8019    LOOP.TF.5.82:
U 1A7D, 8640,15 ;8020            MOV LS[#1] TO WR[0]                      ; данные=1
U 1A7E, 369E,95 ;8021            MOV LS[ONES] TO WR[1]                    ; ожидаемые данные=-1
U 1A7F, 2080,15 ;8022            NEG WR[0]                                ;
U 1A80, 0869,3C ;8023            JSR [CHECK.RESULT]                       ; проверка операции WR.NEG.WR
U 1A81, 09A7,D4 ;8024            JMP [LOOP.TF.5.82]                       ; зацикливание при ошибке, если разрешено
U 1A82, 8A70,1C ;8025            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных до 83
                ;8026    LOOP.TF.5.83:
U 1A83, 8698,15 ;8027            MOV LS[ALTER.ODD] TO WR[0]               ; данные=AAAAAAAA
U 1A84, 869A,95 ;8028            MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые данные=55555555
U 1A85, A0C0,15 ;8029            COM WR[0]                                ;
U 1A86, 0869,3C ;8030            JSR [CHECK.RESULT]                       ; проверка операции WR.COM.WR
U 1A87, 89A8,34 ;8031            JMP [LOOP.TF.5.83]                       ; зацикливание при ошибке, если разрешено
U 1A88, 8A70,1C ;8032            JSR [INC.OTHER]                          ; предварительно проверенная операция WR.DEC.WR
U 1A89, 8A70,1C ;8033            JSR [INC.OTHER]                          ; Адрес 85 пустой
U 1A8A, 8A70,1C ;8034            JSR [INC.OTHER]                          ; увеличение "ДРУГИХ" данных до 86
                ;8035    LOOP.TF.5.86:
U 1A8B, 5898,15 ;8036            MOV LS[ALTER.ODD] TO Q                   ; данные = AAAAAAAA
U 1A8C, 3698,95 ;8037            MOV LS[ALTER.ODD] TO WR[1]               ; ожидаемые данные = AAAAAAAA
U 1A8D, A180,15 ;8038            MOV Q TO WR[0]                           ;
U 1A8E, 0869,3C ;8039            JSR [CHECK.RESULT]                       ; проверка операции MOV.Q.WR
U 1A8F, 09A8,B4 ;8040            JMP [LOOP.TF.5.86]                       ; зацикливание при ошибке, если разрешено
```

```
U 1A90, 8A70,1C ;8041            JSR [INC.OTHER]                        ; адрес 87 пустой
U 1A91, 8A70,1C ;8042            JSR [INC.OTHER]                        ; предварительно проверенная операция COND.ADD&SHIFT
U 1A92, 8A70,1C ;8043            JSR [INC.OTHER]                        ; предварительно проверенная операция COND.SUB&SHIFT
U 1A93, 8A70,1C ;8044            JSR [INC.OTHER]                        ; Адрес 8a пустой
U 1A94, 8A70,1C ;8045            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до 8B
                ;8046    LOOP.TF.5.88:
U 1A95, B640,15 ;8047            MOV LS[BIT0] TO WR[0]                  ; данные = 00000001
U 1A96, B644,95 ;8048            MOV LS[BIT2] TO WR[1]                  ; ожидаемые данные = 00000004
U 1A97, A2D0,15 ;8049            SHL2 WR[0]                             ;
U 1A98, 0869,3C ;8050            JSR [CHECK.RESULT]                     ; проверка операции SHL2
U 1A99, 09A9,54 ;8051            JMP [LOOP.TF.5.88]                     ; зацикливание при ошибке, если разрешено
U 1A9A, 8A70,1C ;8052            JSR [INC.OTHER]                        ; уже проверенная операция ASHRC
U 1A9B, 8A70,1C ;8053            JSR [INC.OTHER]                        ; уже проверенная операция ASHR
U 1A9C, 8A70,1C ;8054            JSR [INC.OTHER]                        ; уже проверенная операция ASHLC
U 1A9D, 8A70,1C ;8055            JSR [INC.OTHER]                        ; уже проверенная операция ASHL
U 1A9E, 8A70,1C ;8056            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до 90
                ;8057    LOOP.TF.5.90:
U 1A9F, DF46,15 ;8058            MCOM LS[BIT3] TO WR[0]                 ;
U 1AA0, 3E8A,15 ;8059            MOV WR[0] TO LS[ERROR.MASK]            ; маскирование всеми битами, кроме бита 3 (проверка бита N)
U 1AA1, 589E,15 ;8060            MOV LS[ONES] TO Q                      ; данные = -1
U 1AA2, 3646,95 ;8061            MOV LS[BIT3] TO WR[1]                  ; ожидаемые = установлен бит N АЛУ
                ,8062              TST Q,                               ;
U 1AA3, A400,35 ;8063            DT(LONG)&SET.ALU.CC                    ;
U 1AA4, 7610,15 ;8064            MOV Q TO LS[T8]                        ;
U 1AA5, B7FC,15 ;8065            MOV LS[ALU.CC] TO WR[0]                ;
U 1AA6, 0869,3C ;8066            JSR [CHECK.RESULT]                     ; проверка, что N установлен
U 1AA7, 09A9,F4 ;8067            JMP [LOOP.TF.5.90]                     ; зацикливание при ошибке, если разрешено
U 1AA8, E58A,15 ;8068            CLR LS[ERROR.MASK]                     ;
U 1AA9, 369E,95 ;8069            MOV LS[ONES] TO WR[1]                  ;
U 1AAA, B610,15 ;8070            MOV LS[T8] TO WR[0]                    ; восстановление Q
U 1AAB, 0869,3C ;8071            JSR [CHECK.RESULT]                     ; проверка, что регистр Q не записывался
U 1AAC, 09A9,F4 ;8072            JMP [LOOP.TF.5.90]                     ; зацикливание при ошибке, если разрешено
U 1AAD, 8A70,1C ;8073            JSR [INC.OTHER]                        ; адрес 91 пустой
U 1AAE, 8A70,1C ;8074            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до 92
                ;8075    LOOP.TF.5.92:
U 1AAF, 2F80,15 ;8076            CLR WR[0]                              ; WR0 = 0
U 1AB0, 5840,15 ;8077            MOV LS[#1] TO Q                        ; Q = 1
                ;8078            CMP WR[0]  WITH Q,                     ;
U 1AB1, 2480,35 ;8079            DT(LONG)&SET.ALU.CC                    ;
U 1AB2, 7610,15 ;8080            MOV Q TO LS[T8]                        ; сохранение Q
U 1AB3, 37FC,95 ;8081            MOV LS[ALU.CC] TO WR[1]                ;
U 1AB4, 3E12,95 ;8082            MOV WR[1] TO LS[T9]                    ;
U 1AB5, 2F82,95 ;8083            CLR WR[1]                              ;
U 1AB6, 0869,3C ;8084            JSR [CHECK.RESULT]                     ; проверка, что WR0 не записывался
U 1AB7, 09AA,F4 ;8085            JMP [LOOP.TF.5.92]                     ; зацикливание при ошибке, если разрешено
U 1AB8, DF46,15 ;8086            MCOM LS[BIT3] TO WR[0]                 ;
U 1AB9, 3E8A,15 ;8087            MOV WR[0] TO LS[ERROR.MASK]            ; маскирование всеми битами, кроме бита 3 (проверка бита N)
U 1ABA, 3646,95 ;8088            MOV LS[BIT3] TO WR[1]                  ;
U 1ABB, 3612,15 ;8089            MOV LS[T9]  TO WR[0]                   ;
U 1ABC, 0869,3C ;8090            JSR [CHECK.RESULT]                     ; проверка, что N установлен
U 1ABD, 09AA,F4 ;8091            JMP [LOOP.TF.5.92]                     ; зацикливание при ошибке, если разрешено
U 1ABE, E58A,15 ;8092            CLR LS[ERROR.MASK]                     ;
U 1ABF, B610,15 ;8093            MOV LS[T8] TO WR[0]                    ; восстановление Q
U 1AC0, 3640,95 ;8094            MOV LS[#1] TO WR[1]                    ; ожидаемые данные = 1
U 1AC1, 0869,3C ;8095            JSR [CHECK.RESULT]                     ; проверка, что регистр Q не записывался
```

```
U 1AC2, 09AA,F4 ;8096           JMP [LOOP.TF.5.92]                  ; зацикливание при ошибке, если разрешено
U 1AC3, 8A70,1C ;8097           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 93
                ;8098   LOOP.TF.5.93:
U 1AC4, AB80,15 ;8099           CLR Q                               ; Q = 0
U 1AC5, B640,15 ;8100           MOV LS[#1] TO WR[0]                 ; WR0 = 1
                ;8101           CMP Q WITH WR[0],                   ;
U 1AC6, A4C0,35 ;8102           DT(LONG)&SET.ALU.CC                 ;
U 1AC7, 37FC,95 ;8103           MOV LS[ALU.CC] TO WR[1]             ;
U 1AC8, 3E12,95 ;8104           MOV WR[1] TO LS[T9]                 ;
U 1AC9, 7610,15 ;8105           MOV Q TO LS[T8]                     ; сохранение Q
U 1ACA, 3640,95 ;8106           MOV LS[#1] TO WR[1]                 ;
U 1ACB, 0869,3C ;8107           JSR [CHECK.RESULT]                  ; проверка, что в WR0 не записывался
U 1ACC, 89AC,44 ;8108           JMP [LOOP.TF.5.93]                  ; зацикливание при ошибке, если разрешено
U 1ACD, DF46,15 ;8109           MCOM LS[BIT3] TO WR[0]              ;
U 1ACE, 3E8A,15 ;8110           MOV WR[0] TO LS[ERROR.MASK]         ; маскирование всеми битами, кроме бита 3 (проверка бита
U 1ACF, 3612,15 ;8111           MOV LS[T9] TO WR[0]                 ;
U 1AD0, 3646,95 ;8112           MOV LS[BIT3] TO WR[1]               ; ожидаемые данные = N установлен
U 1AD1, 0869,3C ;8113           JSR [CHECK.RESULT]                  ; проверка, что N установлен
U 1AD2, 89AC,44 ;8114           JMP [LOOP.TF.5.93]                  ; зацикливание при ошибке, если разрешено
U 1AD3, B610,15 ;8115           MOV LS[T8] TO WR[0]                 ; восстановление Q
U 1AD4, 2F82,95 ;8116           CLR WR[1]                           ; ожидаемые данные = 0
U 1AD5, E58A,15 ;8117           CLR LS[ERROR.MASK]                  ;
U 1AD6, 0869,3C ;8118           JSR [CHECK.RESULT]                  ; проверка, что регистр Q не записывался
U 1AD7, 89AC,44 ;8119           JMP [LOOP.TF.5.93]                  ; зацикливание при ошибке, если разрешено
U 1AD8, 8A70,1C ;8120           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 94
                ;8121   LOOP.TF.5.94:
U 1AD9, A880,15 ;8122           CLR Q                               ; данные = 0
U 1ADA, A500,15 ;8123           DEC Q                               ;
U 1ADB, A180,15 ;8124           MOV Q TO WR[0]                      ; полученные данные = 0
U 1ADC, 369E,95 ;8125           MOV LS[ONES] TO WR[1]               ; ожидаемые данные = -1
U 1ADD, 0869,3C ;8126           JSR [CHECK.RESULT]                  ; проверка операции DEC.Q
U 1ADE, 89AD,94 ;8127           JMP [LOOP.TF.5.94]                  ; зацикливание при ошибке, если разрешено
U 1ADF, 8A70,1C ;8128           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 95
                ;8129   LOOP.TF.5.95:
U 1AE0, 5840,15 ;8130           MOV  LS[#1] TO Q                    ; данные = 1
U 1AE1, 2540,15 ;8131           INC Q                               ;
U 1AE2, A180,15 ;8132           MOV  Q TO WR[0]                     ; полученные данные = Q
U 1AE3, B642,95 ;8133           MOV LS[#2] TO WR[1]                 ; ожидаемые данные = 2
U 1AE4, 0869,3C ;8134           JSR [CHECK.RESULT]                  ; проверка операции INC.Q
U 1AE5, 89AE,04 ;8135           JMP [LOOP.TF.5.95]                  ; зацикливание при ошибке, если разрешено
U 1AE6, 8A70,1C ;8136           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 96
                ;8137   LOOP.TF.5.96:
U 1AE7, 5840,15 ;8138           MOV LS[#1] TO Q                     ; данные = 1
U 1AE8, 2580,15 ;8139           NEG Q                               ;
U 1AE9, A180,15 ;8140           MOV Q TO WR[0]                      ; полученные данные = Q
U 1AEA, 369E,95 ;8141           MOV LS[ONES] TO WR[1]               ; ожидаемые данные = -1
U 1AEB, 0869,3C ;8142           JSR [CHECK.RESULT]                  ; проверка операции NEG.Q
U 1AEC, 09AE,74 ;8143           JMP [LOOP.TF.5.96]                  ; зацикливание при ошибке, если разрешено
U 1AED, 8A70,1C ;8144           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 97
                ;8145   LOOP.TF.5.97:
U 1AEE, 5898,15 ;8146           MOV  LS[ALTER.ODD] TO Q             ; данные = AAAAAAAA
U 1AEF, A5C0,15 ;8147           COM Q                               ;
U 1AF0, A180,15 ;8148           MOV Q TO WR[0]                      ;
U 1AF1, B69A,95 ;8149           MOV LS[ALTER.EVEN] TO WR[1]         ; ожидаемые данные = 55555555
U 1AF2, 0869,3C ;8150           JSR  [CHECK.RESULT]                 ; проверка операции COM.Q
```

```
U 1AF3, 09AE,E4 ;8151           JMP [LOOP.TF.5.97]                  ; зацикливание при ошибке, если разрешено
U 1AF4, 8A70,1C ;8152           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 98
                ;8153   LOOP.TF.5.98:
U 1AF5, 367E,15 ;8154           MOV LS[BIT31] TO WR[0]              ; WR0 = 80000000
U 1AF6, B69B,15 ;8155           MOV LS[ALTER.EVEN] TO WR[2]         ; WR2 = 55555555
U 1AF7, 58BE,15 ;8156           MOV LS[ONES] TO Q                   ; Q = -1
                ;8157             BIT WR[0] WITH WR[2],             ;
U 1AF8, A601,35 ;8158           DT(LONG)&SET.ALU.CC                 ;
U 1AF9, B67E,95 ;8159           MOV LS[BIT31] TO WR[1]              ;
U 1AFA, 0869,3C ;8160           JSR [CHECK.RESULT]                  ; проверка, что WR0 не записывался
U 1AFB, 09AF,54 ;8161           JMP [LOOP.TF.5.98]                  ; зацикливание при ошибке, если разрешено
U 1AFC, 2004,15 ;8162           MOV WR[2] TO WR[0]                  ;
U 1AFD, B69A,95 ;8163           MOV LS[ALTER.EVEN] TO WR[1]         ;
U 1AFE, 0869,3C ;8164           JSR [CHECK.RESULT]                  ; проверка, что WR2 не записывался
U 1AFF, 09AF,54 ;8165           JMP [LOOP.TF.5.98]                  ; зацикливание при ошибке, если разрешено
U 1B00, B7FC,15 ;8166           MOV LS[ALU.CC] TO WR[0]             ;
U 1B01, DF44,95 ;8167           MCOM LS[BIT2] TO WR[1]              ;
U 1B02, BE8A,95 ;8168           MOV WR[1] TO LS[ERROR.MASK]         ;
U 1B03, B644,95 ;8169           MOV LS[BIT2] TO WR[1]               ; ожидаемые данные = бит Z установлен
U 1B04, 0869,3C ;8170           JSR [CHECK.RESULT]                  ; проверка, что бит Z установлен
U 1B05, 09AF,54 ;8171           JMP [LOOP.TF.5.98]                  ; зацикливание при ошибке, если разрешено
U 1B06, 8A70,1C ;8172           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до 99
U 1B07, E58A,15 ;8173           CLR LS[ERROR.MASK]                  ;
                ;8174   LOOP.TF.5.99:
U 1B08, 2F85,15 ;8175           CLR WR[2]                           ; WR2 = 0
U 1B09, B640,15 ;8176           MOV LS[#1] TO WR[0]                 ; WR0 = 1
U 1B0A, 589E,15 ;8177           MOV LS[ONES] TO Q                   ; Q = -1
                ;8178           CMP WR[2] WITH WR[0],               ;
U 1B0B, 2644,35 ;8179           DT(LONG)&SET.ALU.CC                 ;
U 1B0C, 37FC,95 ;8180           MOV LS[ALU.CC] TO WR[1]             ;
U 1B0D, 3E12,95 ;8181           MOV WR[1] TO LS[T9]                 ;
U 1B0E, 3640,95 ;8182           MOV LS[#1] TO WR[1]                 ;
U 1B0F, 0869,3C ;8183           JSR [CHECK.RESULT]                  ; проверка, что WR0 не записывался
U 1B10, 09B0,84 ;8184           JMP [LOOP.TF.5.99]                  ; зацикливание при ошибке, если разрешено
U 1B11, 2004,15 ;8185           MOV WR[2] TO WR[0]                  ;
U 1B12, 2F82,95 ;8186           CLR WR[1]                           ;
U 1B13, 0869,3C ;8187           JSR [CHECK.RESULT]                  ; проверка, что WR2 не записывался
U 1B14, 09B0,84 ;8188           JMP [LOOP.TF.5.99]                  ; зацикливание при ошибке, если разрешено
U 1B15, DF46,15 ;8189           MCOM LS[BIT3] TO WR[0]              ;
U 1B16, 3E8A,15 ;8190           MOV WR[0] TO LS[ERROR.MASK]         ; маска всеми битами, кроме бита 3 (проверка бита N)
U 1B17, 3612,15 ;8191           MOV LS[T9] TO WR[0]                 ;
U 1B18, 3646,95 ;8192           MOV LS[BIT3] TO WR[1]               ; ожидаемые данные = бит N установлен
U 1B19, 0869,3C ;8193           JSR [CHECK.RESULT]                  ; проверка, что бит N установлен
U 1B1A, 09B0,84 ;8194           JMP [LOOP.TF.5.99]                  ; зацикливание при ошибке, если разрешено
U 1B1B, E58A,15 ;8195           CLR LS[ERROR.MASK]                  ;
U 1B1C, 8A70,1C ;8196           JSR [INC.OTHER]                     ; адрес 9A пустой
U 1B1D, 8A70,1C ;8197           JSR [INC.OTHER]                     ; предварительно проверена операция WR.PLUS.WR+1
U 1B1E, 8A70,1C ;8198           JSR [INC.OTHER]                     ; адрес 9C пустой
U 1B1F, 8A70,1C ;8199           JSR [INC.OTHER]                     ; адрес 9D пустой
U 1B20, 8A70,1C ;8200           JSR [INC.OTHER]                     ; адрес 9E пустой
U 1B21, 8A70,1C ;8201           JSR [INC.OTHER]                     ; Адрес 9F пустой
U 1B22, 8A70,1C ;8202           JSR [INC.OTHER]                     ; увеличение "ДРУГИХ" данных до A0
                ;8203   LOOP.TF.5.A0:
U 1B23, B69E,15 ;8204           MOV LS[ONES] TO WR[0]               ; WR0 = -1
U 1B24, 5840,15 ;8205           MOV LS[#1] TO Q                     ; Q = 1
```

```
U 1B25, 2800,15 ;8206            ADD WR[0] TO Q                          ;
U 1B26, A180,15 ;8207            MOV Q TO WR[0]                          ;
U 1B27, 2582,95 ;8208            CLR WR[1]                               ; ожидаемые данные = 0
U 1B28, 0869,3C ;8209            JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.Q
U 1B29, 0982,34 ;8210            JMP [LOOP.TF.5.A0]                      ; зацикливание при ошибке, если разрешено
U 1B2A, 8A70,1C ;8211            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A1
                ;8212    LOOP.TF.5.A1:
U 1B2B, 3642,15 ;8213            MOV LS[#2] TO WR[0]                     ; WR0 = 2
U 1B2C, 5840,15 ;8214            MOV LS[#1] TO Q                         ; Q = 1
U 1B2D, A840,15 ;8215            SUB WR[0] FROM Q                        ;
U 1B2E, A180,15 ;8216            MOV Q TO WR[0]                          ;
U 1B2F, 369E,95 ;8217            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 1B30, 0869,3C ;8218            JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.Q
U 1B31, 8982,B4 ;8219            JMP [LOOP.TF.5.A1]                      ; зацикливание при ошибке, если разрешено
U 1B32, 8A70,1C ;8220            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A2
                ;8221    LOOP.TF.5.A2:
U 1B33, B640,15 ;8222            MOV LS[BIT0] TO WR[0]                   ; WR0 = 1
U 1B34, D842,15 ;8223            MOV LS[#2] TO Q                         ; Q = 2
U 1B35, A880,15 ;8224            SUB Q FROM WR[0] TO Q                   ;
U 1B36, A180,15 ;8225            MOV Q TO WR[0]                          ; получение Q
U 1B37, 369E,95 ;8226            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 1B38, 0869,3C ;8227            JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.WR.Q
U 1B39, 8983,34 ;8228            JMP [LOOP.TF.5.A2]                      ; зацикливание при ошибке, если разрешено
U 1B3A, 8A70,1C ;8229            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A3
                ;8230    LOOP.TF.5.A3:
U 1B3B, B698,15 ;8231            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1B3C, 2040,15 ;8232            INC WR[0]                               ; WR0 = AAAAAAAB
U 1B3D, D89A,15 ;8233            MOV LS[ALTER.EVEN] TO Q                 ; Q = 55555555
U 1B3E, 28C0,15 ;8234            BIS WR[0] TO Q                          ;
U 1B3F, A180,15 ;8235            MOV Q TO WR[0]                          ;
U 1B40, 369E,95 ;8236            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1B41, 0869,3C ;8237            JSR [CHECK.RESULT]                      ; проверка операции WR.OR.Q
U 1B42, 0983,B4 ;8238            JMP [LOOP.TF.5.A3]                      ; зацикливание при ошибке, если разрешено
U 1B43, 8A70,1C ;8239            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A4
                ;8240    LOOP.TF.5.A4:
U 1B44, 3698,95 ;8241            MOV LS[ALTER.ODD] TO WR[1]              ; WR1 = AAAAAAAA
U 1B45, 589E,15 ;8242            MOV LS[ONES] TO Q                       ; Q = FFFFFFFF
U 1B46, 2902,15 ;8243            AND WR[1] TO Q                          ;
U 1B47, A180,15 ;8244            MOV Q TO WR[0]                          ;
U 1B48, 0869,3C ;8245            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.Q
U 1B49, 89B4,44 ;8246            JMP [LOOP.TF.5.A4]                      ; зацикливание при ошибке, если разрешено
U 1B4A, 8A70,1C ;8247            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A5
                ;8248    LOOP.TF.5.A5:
U 1B4B, B698,15 ;8249            MOV LS[ALTER.ODD] TO WR[0]              ; WR0 = AAAAAAAA
U 1B4C, 589E,15 ;8250            MOV LS[ONES] TO Q                       ; Q = FFFFFFFF
U 1B4D, 2940,15 ;8251            BIC WR[0] TO Q                          ;
U 1B4E, A180,15 ;8252            MOV Q TO WR[0]                          ;
U 1B4F, B69A,95 ;8253            MOV LS[ALTER.EVEN] TO WR[1]             ;
U 1B50, 0869,3C ;8254            JSR [CHECK.RESULT]                      ; проверка операции WR.MASK.Q
U 1B51, 8984,B4 ;8255            JMP [LOOP.TF.5.A5]                      ; зацикливание при ошибке, если разрешено
U 1B52, 8A70,1C ;8256            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A6
                ;8257    LOOP.TF.5.A6:
U 1B53, 369A,15 ;8258            MOV LS[ALTER.EVEN] TO WR[0]             ; WR0 = 55555555
U 1B54, 589E,15 ;8259            MOV LS[ONES] TO Q                       ; Q = FFFFFFFF
U 1B55, 2980,15 ;8260            XOR WR[0] TO Q
```

```
U 1B56, A180,15 ;8261            MOV Q TO WR[0]                          ;
U 1B57, 3698,95 ;8262            MOV LS[ALTER.ODD] TO WR[1]              ;
U 1B58, 0869,3C ;8263            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.Q
U 1B59, 8985,34 ;8264            JMP [LOOP.TF.5.A6]                      ; зацикливание при ошибке, если разрешено
U 1B5A, 8A70,1C ;8265            JSR [INC.OTHER]                         ; адрес A7 пустой
U 1B5B, 8A70,1C ;8266            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A8
                ;8267    LOOP.TF.5.A8:
U 1B5C, 369F,15 ;8268            MOV LS[ONES] TO WR[2]                   ; WR2 = -1
U 1B5D, B641,95 ;8269            MOV LS[BIT0] TO WR[3]                   ; WR3 = 1
U 1B5E, 5898,15 ;8270            MOV LS[ALTER.ODD] TO Q                  ;
U 1B5F, 2A05,95 ;8271            ADD WR[2] PLUS WR[3] TO Q               ;
U 1B60, A180,15 ;8272            MOV Q TO WR[0]                          ;
U 1B61, 2F82,95 ;8273            CLR WR[1]                               ; ожидаемые данные = 0
U 1B62, 0869,3C ;8274            JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS WR Q
U 1B63, 8985,C4 ;8275            JMP [LOOP.TF.5.A8]                      ; зацикливание при ошибке, если разрешено
U 1B64, 8A70,1C ;8276            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до A9
                ;8277    LOOP.TF.5.A9:
U 1B65, B643,15 ;8278            MOV LS[#2] TO WR[2]                     ; WR2 = 2
U 1B66, B641,95 ;8279            MOV LS[BIT0] TO WR[3]                   ; WR3 = 1
U 1B67, 5898,15 ;8280            MOV LS[ALTER.ODD] TO Q                  ;
U 1B68, AA45,95 ;8281            SUB WR[2] FROM WR[3] TO Q               ;
U 1B69, A180,15 ;8282            MOV Q TO WR[0]                          ;
U 1B6A, 369E,95 ;8283            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 1B6B, 0869,3C ;8284            JSR [CHECK.RESULT]                      ; проверка операции WR FROM WR Q
U 1B6C, 8986,54 ;8285            JMP [LOOP.TF.5.A9]                      ; зацикливание при ошибке, если разрешено
U 1B6D, 8A70,1C ;8286            JSR [INC.OTHER]                         ; адрес AA пустой
U 1B6E, 8A70,1C ;8287            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до AB
                ;8288    LOOP.TF.5.AB:
U 1B6F, 3699,15 ;8289            MOV LS[ALTER.ODD] TO WR[2]              ;
U 1B70, 2045,15 ;8290            INC WR[2]                               ; WR2 = AAAAAAAB
U 1B71, 3698,95 ;8291            MOV LS[ALTER.EVEN] TO WR[3]             ; WR3 = 55555555
U 1B72, D828,15 ;8292            MOV LS[#FFFF] TO Q                      ;
U 1B73, 2AC5,95 ;8293            BIS WR[2] WITH WR[3] TO Q               ;
U 1B74, A180,15 ;8294            MOV Q TO WR[0]                          ;
U 1B75, 369E,95 ;8295            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1B76, 0869,3C ;8296            JSR [CHECK.RESULT]                      ; проверка операции WR.OR.WR.Q
U 1B77, 8986,F4 ;8297            JMP [LOOP.TF.5.AB]                      ; зацикливание при ошибке, если разрешено
U 1B78, 8A70,1C ;8298            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до AC
                ;8299    LOOP.TF.5.AC:
U 1B79, 3699,15 ;8300            MOV LS[ALTER.ODD] TO WR[2]              ;
U 1B7A, 2045,15 ;8301            INC WR[2]                               ; WR2 = AAAAAAAB
U 1B7B, 3698,95 ;8302            MOV LS[ALTER.EVEN] TO WR[3]             ; WR3 = 55555555
U 1B7C, D828,15 ;8303            MOV LS[#FFFF] TO Q                      ;
U 1B7D, AB05,95 ;8304            AND WR[2] WITH WR[3] TO Q               ;
U 1B7E, A180,15 ;8305            MOV Q TO WR[0]                          ;
U 1B7F, 3640,95 ;8306            MOV LS[BIT0] TO WR[1]                   ; ожидаемые данные = 00000001
U 1B80, 0869,3C ;8307            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.WR.Q
U 1B81, 0987,94 ;8308            JMP [LOOP.TF.5.AC]                      ; зацикливание при ошибке, если разрешено
U 1B82, 8A70,1C ;8309            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до AD
                ;8310    LOOP.TF.5.AD:
U 1B83, 3699,15 ;8311            MOV LS[ALTER.ODD] TO WR[2]              ; WR2 = AAAAAAAA
U 1B84, B69F,95 ;8312            MOV LS[ONES] TO WR[3]                   ; WR3 = FFFFFFFF
U 1B85, D828,15 ;8313            MOV LS[#FFFF] TO Q                      ;
U 1B86, 2B45,95 ;8314            BIC WR[2] WITH WR[3] TO Q               ;
U 1B87, A180,15 ;8315            MOV Q TO WR[0]                          ;
```

```
U 1B88, B69A,95 ;8316            MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные = 55555555
U 1B89, 0869,3C ;8317            JSR [CHECK.RESULT]                     ; проверка операции WR.MASK.WR.Q
U 1B8A, 0988,34 ;8318            JMP [LOOP.TF.5.AD]                     ; зацикливание при ошибке, если разрешено
U 1B8B, 8A70,1C ;8319            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до AE
                ;8320    LOOP.TF.5.AE:
U 1B8C, B698,15 ;8321            MOV LS[ALTER.EVEN] TO WR[2]            ; WR2 = 55555555
U 1B8D, B69F,95 ;8322            MOV LS[ONES] TO WR[3]                  ; WR3 = FFFFFFFF
U 1B8E, D828,15 ;8323            MOV LS[#FFFF] TO Q                     ;
U 1B8F, 2885,95 ;8324            XOR WR[2] WITH WR[3] TO Q              ;
U 1B90, A180,15 ;8325            MOV Q TO WR[0]                         ;
U 1B91, 3698,95 ;8326            MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные = AAAAAAAA
U 1B92, 0869,3C ;8327            JSR [CHECK.RESULT]                     ; проверка операции WR.XOR.WR.Q
U 1B93, 0988,C4 ;8328            JMP [LOOP.TF.5.AE]                     ; зацикливание при ошибке, если разрешено
U 1B94, 8A70,1C ;8329            JSR [INC.OTHER]                        ; адрес AF пустой
U 1B95, 8A70,1C ;8330            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до B0
                ;8331    LOOP.TF.5.B0:
U 1B96, 589E,15 ;8332            MOV LS[ONES] TO Q                      ; Q = -1
U 1B97, B640,15 ;8333            MOV LS[BIT0] TO WR[0]                  ; WR0 = 1
U 1B98, AC00,15 ;8334            ADD Q TO WR[0]                         ;
U 1B99, 2F82,95 ;8335            CLR WR[1]                              ; Ожидаемые данные = 0
U 1B9A, 0869,3C ;8336            JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.WR
U 1B9B, 8989,64 ;8337            JMP [LOOP.TF.5.B0]                     ; зацикливание при ошибке, если разрешено
U 1B9C, 8A70,1C ;8338            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до B1
                ;8339    LOOP.TF.5.B1:
U 1B9D, B643,15 ;8340            MOV LS[#2] TO WR[2]                    ; WR2 = 2
U 1B9E, 5840,15 ;8341            MOV LS[BIT0] TO Q                      ; Q = 1
U 1B9F, AC44,15 ;8342            SUB WRA[2] FROM Q TO WRB[0]            ;
U 1BA0, 369E,95 ;8343            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1BA1, 0869,3C ;8344            JSR [CHECK.RESULT]                     ; проверка операции WR.FROM.Q.WR
U 1BA2, 09B9,D4 ;8345            JMP [LOOP.TF.5.B1]                     ; зацикливание при ошибке, если разрешено
U 1BA3, 8A70,1C ;8346            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до B2
                ;8347    LOOP.TF.5.B2:
U 1BA4, D842,15 ;8348            MOV LS[#2] TO Q                        ; Q = 2
U 1BA5, B640,15 ;8349            MOV LS[BIT0] TO WR[0]                  ; WR0 = 1
U 1BA6, 2C80,15 ;8350            SUB Q FROM WR[0]                       ;
U 1BA7, 369E,95 ;8351            MOV LS[ONES] TO WR[1]                  ; Ожидаемые данные = -1
U 1BA8, 0869,3C ;8352            JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.WR
U 1BA9, 098A,44 ;8353            JMP [LOOP.TF.5.B2]                     ; зацикливание при ошибке, если разрешено
U 1BAA, 8A70,1C ;8354            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до B3
                ;8355    LOOP.TF.5.B3:
U 1BAB, 5898,15 ;8356            MOV LS[ALTER.ODD] TO Q                 ;
U 1BAC, 2540,15 ;8357            INC Q                                  ; Q = AAAAAAAB
U 1BAD, 369A,15 ;8358            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0 = 55555555
U 1BAE, ACC0,15 ;8359            BIS Q TO WR[0]                         ;
U 1BAF, 369E,95 ;8360            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = FFFFFFFF
U 1BB0, 0869,3C ;8361            JSR [CHECK.RESULT]                     ; проверка операции Q.OR.WR
U 1BB1, 098A,B4 ;8362            JMP [LOOP.TF.5.B3]                     ; зацикливание при ошибке, если разрешено
U 1BB2, 8A70,1C ;8363            JSR [INC.OTHER]                        ; увеличение "ДРУГИХ" данных до B4
                ;8364    LOOP.TF.5.B4:
U 1BB3, 5898,15 ;8365            MOV LS[ALTER.ODD] TO Q                 ;
U 1BB4, 2540,15 ;8366            INC Q                                  ; Q = AAAAAAAB
U 1BB5, 369A,15 ;8367            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0 = 55555555
U 1BB6, 2D00,15 ;8368            AND Q TO WR[0]                         ;
U 1BB7, 3640,95 ;8369            MOV LS[BIT0] TO WR[1]                  ; ожидаемые данные = 00000001
U 1BB8, 0869,3C ;8370            JSR [CHECK.RESULT]                     ; проверка операции Q.AND.WR
```

```
U 1BB9, 098B,34 ;8371            JMP [LOOP.TF.5.B4]                    ; зацикливание при ошибке, если разрешено
U 1BBA, BA70,1C ;8372            JSR [INC.OTHER]                       ; адрес B5 пустой
U 1BBB, BA70,1C ;8373            JSR [INC.OTHER]                       ; увеличение "ДРУГИХ" данных до B6
                ;8374    LOOP.TF.5.B6:
U 1BBC, 5898,15 ;8375            MOV LS[ALTER.ODD] TO Q                ; Q = AAAAAAAA
U 1BBD, B69E,15 ;8376            MOV LS[ONES] TO WR[0]                 ; WR0 = FFFFFFFF
U 1BBE, AD80,15 ;8377            XOR Q TO WR[0]                        ;
U 1BBF, B69A,95 ;8378            MOV LS[ALTER.EVEN] TO WR[1]           ; ожидаемые данные = 55555555
U 1BC0, 0869,3C ;8379            JSR [CHECK.RESULT]                    ; проверка операции Q.XOR.WR
U 1BC1, 098B,C4 ;8380            JMP [LOOP.TF.5.B6]                    ; зацикливание при ошибке, если разрешено
U 1BC2, BA70,1C ;8381            JSR [INC.OTHER]                       ; увеличение "ДРУГИХ" данных до B7
                ;8382    LOOP.TF.5.B7:
U 1BC3, D87E,15 ;8383            MOV LS[BIT31] TO Q                    ; Q = 80000000
U 1BC4, 369A,15 ;8384            MOV LS[ALTER.EVEN] TO WR[0]           ; WR0 = 55555555
                ;8385            BIT Q WITH WR[0],                     ;
U 1BC5, ADC0,35 ;8386            DT(LONG)&SET.ALU.CC                   ;
U 1BC6, 7610,15 ;8387            MOV Q TO LS[T8]                       ; сохранение Q
U 1BC7, B69A,95 ;8388            MOV LS[ALTER.EVEN] TO WR[1]           ;
U 1BC8, 0869,3C ;8389            JSR [CHECK.RESULT]                    ; проверка, что WR0 не записывался
U 1BC9, 89BC,34 ;8390            JMP [LOOP.TF.5.B7]                    ; зацикливание при ошибке, если разрешено
U 1BCA, B7FC,15 ;8391            MOV LS[ALU.CC] TO WR[0]               ;
U 1BCB, DF44,95 ;8392            MCOM LS[BIT2] TO WR[1]                ;
U 1BCC, BE8A,95 ;8393            MOV WR[1] TO LS[ERROR.MASK]           ;
U 1BCD, B644,95 ;8394            MOV LS[BIT2] TO WR[1]                 ; ожидаемые данные = Z установлен
U 1BCE, 0869,3C ;8395            JSR [CHECK.RESULT]                    ; проверка, что Z установлен
U 1BCF, 89BC,34 ;8396            JMP [LOOP.TF.5.B7]                    ; зацикливание при ошибке, если разрешено
U 1BD0, E58A,15 ;8397            CLR LS[ERROR.MASK]                    ;
U 1BD1, B610,15 ;8398            MOV LS[T8] TO WR[0]                   ; восстановление Q
U 1BD2, B67E,95 ;8399            MOV LS[BIT31] TO WR[1]                ;
U 1BD3, 0869,3C ;8400            JSR [CHECK.RESULT]                    ; проверка, что регистр Q не записывался
U 1BD4, 89BC,34 ;8401            JMP [LOOP.TF.5.B7]                    ; зацикливание при ошибке, если разрешено
U 1BD5, BA70,1C ;8402            JSR [INC.OTHER]                       ; B8 пустой
U 1BD6, BA70,1C ;8403            JSR [INC.OTHER]                       ; B9 пустой
U 1BD7, BA70,1C ;8404            JSR [INC.OTHER]                       ; увеличение "ДРУГИХ" данных до BA
                ;8405    LOOP.TF.5.BA:
U 1BD8, 3642,15 ;8406            MOV LS[#2] TO WR[0]                   ; WR0 = 2
U 1BD9, 3641,15 ;8407            MOV LS[BIT0] TO WR[2]                 ; WR2 = 1
U 1BDA, D828,15 ;8408            MOV LS[#FFFF] TO Q                    ;
U 1BDB, 2E84,15 ;8409            SUB WRB[0] FROM WRA[2] TO WRB[0]      ;
U 1BDC, 369E,95 ;8410            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные = -1
U 1BDD, 0869,3C ;8411            JSR [CHECK.RESULT]                    ; проверка операции WR.FROM.WR.WR
U 1BDE, 89BD,84 ;8412            JMP [LOOP.TF.5.BA]                    ; зацикливание при ошибке, если разрешено
U 1BDF, BA70,1C ;8413            JSR [INC.OTHER]                       ; увеличение "ДРУГИХ" данных до BB
                ;8414    LOOP.TF.5.BB:
U 1BE0, 3699,15 ;8415            MOV LS[ALTER.ODD] TO WR[2]            ;
U 1BE1, 2045,15 ;8416            INC WR[2]                             ; WR2 = AAAAAAAB
U 1BE2, 369A,15 ;8417            MOV LS[ALTER.EVEN] TO WR[0]           ; WR0 = 55555555
U 1BE3, D828,15 ;8418            MOV LS[#FFFF] TO Q                    ;
U 1BE4, AEC4,15 ;8419            BIS WR[2] TO WR[0]                    ;
U 1BE5, 369E,95 ;8420            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные = FFFFFFFF
U 1BE6, 0869,3C ;8421            JSR [CHECK.RESULT]                    ; проверка операции WR.OR.WR
U 1BE7, 09BE,04 ;8422            JMP [LOOP.TF.5.BB]                    ; зацикливание при ошибке, если разрешено
U 1BE8, BA70,1C ;8423            JSR [INC.OTHER]                       ; предварительно проверенная операция WR.FROM WR-1
U 1BE9, BA70,1C ;8424            JSR [INC.OTHER]                       ; увеличение "ДРУГИХ" данных до BD
                ;8425    LOOP.TF.5.BD:
```

```
U 1BEA, 3699,15 ;8426            MOV LS[ALTER.ODD] TO WR[2]              ; WR2 = AAAAAAAA
U 1BEB, B69E,15 ;8427            MOV LS[ONES] TO WR[0]                   ; WR0 = FFFFFFFF
U 1BEC, D828,15 ;8428            MOV LS[#FFFF] TO Q                      ;
U 1BED, AF44,15 ;8429            BIC WR[2] TO WR[0]                      ;
U 1BEE, B69A,95 ;8430            MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 1BEF, 0869,3C ;8431            JSR [CHECK.RESULT]                      ; проверка операции WR.MASK.WR
U 1BF0, 09BE,A4 ;8432            JMP [LOOP.TF.5.BD]                      ; зацикливание при ошибке, если разрешено
U 1BF1, 8A70,1C ;8433            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до BE
                ;8434    LOOP.TF.5.BE:
U 1BF2, B698,15 ;8435            MOV LS[ALTER.EVEN] TO WR[2]             ; WR2 = 55555555
U 1BF3, B69E,15 ;8436            MOV LS[ONES] TO WR[0]                   ; WR0 = FFFFFFFF
U 1BF4, D828,15 ;8437            MOV LS[#FFFF] TO Q                      ;
U 1BF5, AF84,15 ;8438            XOR WR[2] TO WR[0]                      ;
U 1BF6, 3698,95 ;8439            MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 1BF7, 0869,3C ;8440            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.WR
U 1BF8, 09BF,24 ;8441            JMP [LOOP.TF.5.BE]                      ; зацикливание при ошибке, если разрешено
U 1BF9, 8A70,1C ;8442            JSR [INC.OTHER]                         ; увеличение "ДРУГИХ" данных до BF
                ;8443    LOOP.TF.5.BF.
U 1BFA, 3699,15 ;8444            MOV LS[ALTER.ODD] TO WR[2]              ;
U 1BFB, 2045,15 ;8445            INC WR[2]                               ; WR2 = AAAAAAAB
U 1BFC, 369A,15 ;8446            MOV LS[ALTER.EVEN] TO WR[0]             ; WR0 = 55555555
U 1BFD, D828,15 ;8447            MOV LS[#FFFF] TO Q                      ;
U 1BFE, 2FC4,15 ;8448            AND WR[2] TO WR[0]                      ;
U 1BFF, 3640,95 ;8449            MOV LS[BIT0] TO WR[1]                   ; ожидаемые данные = 00000001
U 1C00, 0869,3C ;8450            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.WR
U 1C01, B9BF,A4 ;8451            JMP [LOOP.TF.5.BF]                      ; зацикливание при ошибке, если разрешено
U 1C02, B9C1,A4 ;8452            JMP [TF.6]                              ;
                ;8453
                ;8454
                ;8455    ;           ТЕСТЫ ИНСТРУКЦИЙ BASIC
                ;8456    ;
                ;8457    ;         Проверяются в четырех группах, т.е. каждая инструкция
                ;8458    ;         проверяется в четырех разных областях местной памяти:
                ;8459    ;         0-1F, 20-3F, 40-5F, и 60-7F. Используются четыре текущие
                ;8460             ячейки: LS 8  -  T8
                ;8461                     LS 30 -  #26
                ;8462    ;                LS 53 -  #F
                ;8463    ;                LS 73 -  #3F
                ;8464    ,
                ;8465    ;
                ;8466
                ;8467    1C1A:
                ;8468    TF.6.
U 1C1A, 8870,0C ;8469            JSR [INC.EN]                            ; увеличение номера ошибки до 6
U 1C1B, 3660,15 ;8470            MOV LS[L30] TO WR[0]                    ; сохранение LS 30, 53 и 73
U 1C1C, B6A6,95 ;8471            MOV LS[L53] TO WR[1]                    ;
U 1C1D, 36E7,15 ;8472            MOV LS[L73] TO WR[2]                    ;
U 1C1E, BE12,15 ;8473            MOV WR[0] TO LS[T9]                     ;
U 1C1F, 3E14,95 ;8474            MOV WR[1] TO LS[T10]                    ;
U 1C20, BE17,15 ;8475            MOV WR[2] TO LS[T11]                    ;
U 1C21, 3650,15 ;8476            MOV LS[#100] TO WR[0]                   ;
U 1C22, BE88,15 ;8477            MOV WR[0] TO LS[ADDRESS.DATA]           ; установка 100 для начальной установки поля "ДРУГИЕ
                :8478                                                    ; ДАННЫЕ"
U 1C23, 5F28,15 ;8479            MCOM LS[#FFFF] TO WR[0]                 ;
U 1C24, BE18,15 ;8480            MOV WR[0] TO LS[T12]                    ; T12=FFFF0000 (маска для старшего слова)
```

```
                    ;8481    LOOP.TF.6.100:
U 1C25, A880,15     ;8482             CLR Q                          ;
U 1C26, 3642,15     ;8483             MOV LS[#2] TO WR[0]             ;
U 1C27, 3E10,15     ;8484             MOV WR[0] TO LS[T8]             ; LS=2
U 1C28, B69E,15     ;8485             MOV LS[ONES] TO WR[0]           ; WR0=-1
U 1C29, 4010,15     ;8486             ADD LS[T8] TO WR[0]             ;
U 1C2A, 3640,95     ;8487             MOV LS[#1] TO WR[1]             ; ожидаемые данные=1
U 1C2B, 0869,3C     ;8488             JSR [CHECK.RESULT]              ; проверка операции LS.PLUS.WR
U 1C2C, B9C2,54     ;8489             JMP [LOOP.TF.6.100]             ; зацикливание при ошибке, если разрешено
U 1C2D, BA70,1C     ;8490             JSR [INC.OTHER]                 ;
                    ;8491    LOOP.TF.6.101:
U 1C2E, A880,15     ;8492             CLR Q                          ;
U 1C2F, 3642,15     ;8493             MOV LS[#2] TO WR[0]             ;
U 1C30, BE60,15     ;8494             MOV WR[0] TO LS[L30]            ; LS=2
U 1C31, B69E,15     ;8495             MOV LS[ONES] TO WR[0]           ; WR0=-1
U 1C32, C060,15     ;8496             ADD LS[L30] TO WR[0]            ;
U 1C33, 3640,95     ;8497             MOV LS[#1] TO WR[1]             ; ожидаемые данные=1
U 1C34, 0869,3C     ;8498             JSR [CHECK.RESULT]              ; проверка операции LS.PLUS.WR
U 1C35, 09C2,E4     ;8499             JMP [LOOP.TF.6.101]             ; зацикливание при ошибке, если разрешено
U 1C36, BA70,1C     ;8500             JSR [INC.OTHER]                 ;
                    ;8501    LOOP.TF.6.102:
U 1C37, A880,15     ;8502             CLR Q                          ;
U 1C38, 3642,15     ;8503             MOV LS[#2] TO WR[0]             ;
U 1C39, BEA6,15     ;8504             MOV WR[0] TO LS[L53]            ; LS=2
U 1C3A, B69E,15     ;8505             MOV LS[ONES] TO WR[0]           ; WR0=-1
U 1C3B, C0A6,15     ;8506             ADD LS[L53] TO WR[0]            ;
U 1C3C, 3640,95     ;8507             MOV LS[#1] TO WR[1]             ; ожидаемые данные=1
U 1C3D, 0869,3C     ;8508             JSR [CHECK.RESULT]              ; проверка операции LS.PLUS.WR
U 1C3E, B9C3,74     ;8509             JMP [LOOP.TF.6.102]             ; зацикливание при ошибке, если разрешено
U 1C3F, BA70,1C     ;8510             JSR [INC.OTHER]                 ;
                    ;8511    LOOP.TF.6.103:
U 1C40, A880,15     ;8512             CLR Q                          ;
U 1C41, 3642,15     ;8513             MOV LS[#2] TO WR[0]             ;
U 1C42, 3EE6,15     ;8514             MOV WR[0] TO LS[L73]            ; LS=2
U 1C43, B69E,15     ;8515             MOV LS[ONES] TO WR[0]           ; WR0=-1
U 1C44, 40E6,15     ;8516             ADD LS[L73] TO WR[0]            ;
U 1C45, 3640,95     ;8517             MOV LS[#1] TO WR[1]             ; ожидаемые данные=1
U 1C46, 0869,3C     ;8518             JSR [CHECK.RESULT]              ; проверка операции LS.PLUS.WR
U 1C47, B9C4,04     ;8519             JMP [LOOP.TF.6.103]             ; зацикливание при ошибке, если разрешено
U 1C48, BA70,1C     ;8520             JSR [INC.OTHER]                 ;
                    ;8521    LOOP.TF.6.104:
U 1C49, A880,15     ;8522             CLR Q                          ;
U 1C4A, 3642,15     ;8523             MOV LS[#2] TO WR[0]             ;
U 1C4B, 3E10,15     ;8524             MOV WR[0] TO LS[T8]             ; LS=2
U 1C4C, B640,15     ;8525             MOV LS[#1] TO WR[0]             ; WR0=1
U 1C4D, C110,15     ;8526             SUB LS[T8] FROM WR[0]           ;
U 1C4E, 369E,95     ;8527             MOV LS[ONES] TO WR[1]           ; ожидаемые данные =-1
U 1C4F, 0869,3C     ;8528             JSR [CHECK.RESULT]              ; проверка операции LS.FROM.WR
U 1C50, B9C4,94     ;8529             JMP [LOOP.TF.6.104]             ; зацикливание при ошибке, если разрешено
U 1C51, BA70,1C     ;8530             JSR [INC.OTHER]                 ;
                    ;8531    LOOP.TF.6.105:
U 1C52, A880,15     ;8532             CLR Q                          ;
U 1C53, 3642,15     ;8533             MOV LS[#2] TO WR[0]             ;
U 1C54, BE60,15     ;8534             MOV WR[0] TO LS[L30]            ; LS=2
U 1C55, B640,15     ;8535             MOV LS[#1] TO WR[0]             ; WR0=1
```

```
U 1C56, 4160,15 ;8536            SUB LS[L30] FROM WR[0]                ;
U 1C57, 369E,95 ;8537            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные=-1
U 1C58, 0869,3C ;8538            JSR [CHECK.RESULT]                    ; проверка операции LS.FROM.WR
U 1C59, 89C5,24 ;8539            JMP [LOOP.TF.6.105]                   ; зацикливание при ошибке, если разрешено
U 1C5A, 8A70,1C ;8540            JSR [INC.OTHER]                       ;
                ;8541    LOOP.TF.6.106:
U 1C5B, AB80,15 ;8542            CLR Q                                 ;
U 1C5C, 3642,15 ;8543            MOV LS[#2] TO WR[0]                   ;
U 1C5D, BEA6,15 ;8544            MOV WR[0] TO LS[L53]                  ; LS=2
U 1C5E, B640,15 ;8545            MOV LS[#1] TO WR[0]                   ; WR0=1
U 1C5F, 41A6,15 ;8546            SUB LS[L53] FROM WR[0]                ;
U 1C60, 369E,95 ;8547            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные=-1
U 1C61, 0869,3C ;8548            JSR [CHECK.RESULT]                    ; проверка операции LS.FROM.WR
U 1C62, 89C5,B4 ;8549            JMP [LOOP.TF.6.106]                   ; зацикливание при ошибке, если разрешено
U 1C63, 8A70,1C ;8550            JSR [INC.OTHER]                       ;
                ;8551    LOOP.TF.6.107:
U 1C64, AB80,15 ;8552            CLR Q                                 ;
U 1C65, 3642,15 ;8553            MOV LS[#2] TO WR[0]                   ;
U 1C66, 3EE6,15 ;8554            MOV WR[0] TO LS[L73]                  ; LS=2
U 1C67, B640,15 ;8555            MOV LS[#1] TO WR[0]                   ; WR0=1
U 1C68, C1E6,15 ;8556            SUB LS[L73] FROM WR[0]                ;
U 1C69, 369E,95 ;8557            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные=-1
U 1C6A, 0869,3C ;8558            JSR [CHECK.RESULT]                    ; проверка операции LS.FROM.WR
U 1C6B, 89C6,44 ;8559            JMP [LOOP.TF.6.107]                   ; зацикливание при ошибке, если разрешено
U 1C6C, 8A70,1C ;8560            JSR [INC.OTHER]                       ;
                ;8561    LOOP.TF.6.108:
U 1C6D, AB80,15 ;8562            CLR Q                                 ;
J 1C6E, B698,15 ;8563            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1C6F, 3E10,15 ;8564            MOV WR[0] TO LS[T8]                   ; LS=AAAAAAAA
U 1C70, 3628,15 ;8565            MOV LS[#FFFF] TO WR[0]                ; WR0=0000FFFF
J 1C71, C210,15 ;8566            AND LS[T8] TO WR[0]                   ;
J 1C72, 3698,95 ;8567            MOV LS[ALTER.ODD] TO WR[1]             ;
J 1C73, 4518,95 ;8568            BIC LS[T12] TO WR[1]                  ; ожидаемые данные=0000AAAA
J 1C74, 0869,3C ;8569            JSR [CHECK.RESULT]                    ; проверка операции LS.AND.WR
J 1C75, 89C6,D4 ;8570            JMP [LOOP.TF.6.108]                   ; зацикливание при ошибке, если разрешено
J 1C76, 8A70,1C ;8571            JSR [INC.OTHER]                       ;
                ;8572    LOOP.TF.6.109:
J 1C77, AB80,15 ;8573            CLR Q                                 ;
J 1C78, B698,15 ;8574            MOV LS[ALTER.ODD] TO WR[0]             ;
J 1C79, BE60,15 ;8575            MOV WR[0] TO LS[L30]                  ; LS=AAAAAAAA
J 1C7A, 3628,15 ;8576            MOV LS[#FFFF] TO WR[0]                ; WR0=0000FFFF
J 1C7B, 4260,15 ;8577            AND LS[L30] TO WR[0]                  ;
J 1C7C, 3698,95 ;8578            MOV LS[ALTER.ODD] TO WR[1]             ;
J 1C7D, 4518,95 ;8579            BIC LS[T12] TO WR[1]                  ; ожидаемые данные=0000AAAA
J 1C7E, 0869,3C ;8580            JSR [CHECK.RESULT]                    ; проверка операции LS.AND.WR
J 1C7F, 09C7,74 ;8581            JMP [LOOP.TF.6.109]                   ; зацикливание при ошибке, если разрешено
  1C80, 8A70,1C ;8582            JSR [INC.OTHER]                       ;
                ;8583    LOOP.TF.6.10A:
  1C81, AB80,15 ;8584            CLR Q                                 ;
  1C82, B698,15 ;8585            MOV LS[ALTER.ODD] TO WR[0]             ;
  1C83, BEA6,15 ;8586            MOV WR[0] TO LS[L53]                  ; LS=AAAAAAAA
  1C84, 3628,15 ;8587            MOV LS[#FFFF] TO WR[0]                ; WR0=0000FFFF
  1C85, 42A6,15 ;8588            AND LS[L53] TO WR[0]                  ;
  1C86, 3698,95 ;8589            MOV LS[ALTER.ODD] TO WR[1]             ;
  1C87, 4518,95 ;8590            BIC LS[T12] TO WR[1]                  ; ожидаемые данные=0000AAAA
```

```
U 1C88, 0869,3C ;8591            JSR [CHECK.RESULT]                    ; проверка операции LS.AND.WR
U 1C89, 09C8,14 ;8592            JMP [LOOP.TF.6.10A]                   ; зацикливание при ошибке, если разрешено
U 1C8A, 8A70,1C ;8593            JSR [INC.OTHER]                       ;
                ;8594    LOOP.TF.6.10B:
U 1C8B, AB80,15 ;8595            CLR Q                                 ;
U 1C8C, B698,15 ;8596            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1C8D, 3EE6,15 ;8597            MOV WR[0] TO LS[L73]                   ; LS=AAAAAAAA
U 1C8E, 3628,15 ;8598            MOV LS[#FFFF] TO WR[0]                 ; WR0=0000FFFF
U 1C8F, C2E6,15 ;8599            AND LS[L73] TO WR[0]                   ;
U 1C90, 3698,95 ;8600            MOV LS[ALTER.ODD] TO WR[1]             ;
U 1C91, 4518,95 ;8601            BIC LS[T12] TO WR[1]                   ; ожидаемые данные=0000AAAA
U 1C92, 0869,3C ;8602            JSR [CHECK.RESULT]                    ; проверка операции LS.AND.WR
U 1C93, 09C8,B4 ;8603            JMP [LOOP.TF.6.10B]                   ; зацикливание при ошибке, если разрешено
U 1C94, 8A70,1C ;8604            JSR [INC.OTHER]                       ;
                ;8605    LOOP.TF.6.10C:
U 1C95, AB80,15 ;8606            CLR Q                                 ;
U 1C96, B698,15 ;8607            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1C97, 3E10,15 ;8608            MOV WR[0] TO LS[T8]                    ; LS=AAAAAAAA
U 1C98, 3628,15 ;8609            MOV LS[#FFFF] TO WR[0]                 ; WR0=0000FFFF
U 1C99, 4310,15 ;8610            XOR LS[T8] TO WR[0]                    ;
U 1C9A, 3722,95 ;8611            MOV LS[ALTER.MIX] TO WR[1]             ; ожидаемые данные=AAAA5555
U 1C9B, 0869,3C ;8612            JSR [CHECK.RESULT]                    ; проверка операции LS.XOR.WR
U 1C9C, 09C9,54 ;8613            JMP [LOOP.TF.6.10C]                   ; зацикливание при ошибке, если разрешено
U 1C9D, 8A70,1C ;8614            JSR [INC.OTHER]                       ;
                ;8615    LOOP.TF.6.10D:
U 1C9E, AB80,15 ;8616            CLR Q                                 ;
U 1C9F, B698,15 ;8617            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1CA0, 8E60,15 ;8618            MOV WR[0] TO LS[L30]                   ; LS=AAAAAAAA
U 1CA1, 3628,15 ;8619            MOV LS[#FFFF] TO WR[0]                 ; WR0=0000FFFF
U 1CA2, C360,15 ;8620            XOR LS[L30] TO WR[0]                   ;
U 1CA3, 3722,95 ;8621            MOV LS[ALTER.MIX] TO WR[1]             ; ожидаемые данные=AAAA5555
U 1CA4, 0869,3C ;8622            JSR [CHECK.RESULT]                    ; проверка операции LS.XOR.WR
U 1CA5, 89C9,E4 ;8623            JMP [LOOP.TF.6.10D]                   ; зацикливание при ошибке, если разрешено
U 1CA6, 8A70,1C ;8624            JSR [INC.OTHER]                       ;
                ;8625    LOOP.TF.6.10E:
U 1CA7, AB80,15 ;8626            CLR Q                                 ;
U 1CA8, B698,15 ;8627            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1CA9, 8EA6,15 ;8628            MOV WR[0] TO LS[L53]                   ; LS=AAAAAAAA
U 1CAA, 3628,15 ;8629            MOV LS[#FFFF] TO WR[0]                 ; WR0=0000FFFF
U 1CAB, C3A6,15 ;8630            XOR LS[L53] TO WR[0]                   ;
U 1CAC, 3722,95 ;8631            MOV LS[ALTER.MIX] TO WR[1]             ; ожидаемые данные=AAAA5555
U 1CAD, 0869,3C ;8632            JSR [CHECK.RESULT]                    ; проверка операции LS.XOR.WR
U 1CAE, 89CA,74 ;8633            JMP [LOOP.TF.6.10E]                   ; зацикливание при ошибке, если разрешено
U 1CAF, 8A70,1C ;8634            JSR [INC.OTHER]                       ;
                ;8635    LOOP.TF.6.10F:
U 1CB0, AB80,15 ;8636            CLR Q                                 ;
U 1CB1, B698,15 ;8637            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1CB2, 3EE6,15 ;8638            MOV WR[0] TO LS[L73]                   ; LS=AAAAAAAA
U 1CB3, 3628,15 ;8639            MOV LS[#FFFF] TO WR[0]                 ; WR0=0000FFFF
U 1CB4, 43E6,15 ;8640            XOR LS[L73] TO WR[0]                   ;
U 1CB5, 3722,95 ;8641            MOV LS[ALTER.MIX] TO WR[1]             ; ожидаемые данные=AAAA5555
U 1CB6, 0869,3C ;8642            JSR [CHECK.RESULT]                    ; проверка операции LS.XOR.WR
U 1CB7, 89CB,04 ;8643            JMP [LOOP.TF.6.10F]                   ; зацикливание при ошибке, если разрешено
U 1CB8, 8A70,1C ;8644            JSR [INC.OTHER]                       ;
                ;8645    LOOP.TF.6.110:
```

```
U 1CB9. AB80,15 ;8646           CLR Q                                   ;
U 1CBA. B640,15 ;8647           MOV LS[#1] TO WR[0]                     ;
U 1CBB. 3E10,15 ;8648           MOV WR[0] TO LS[T8]                     ; LS=1
U 1CBC. 3642,15 ;8649           MOV LS[#2] TO WR[0]                     ; WR0=2
U 1CBD. C410,15 ;8650           SUB (LS[T8] FROM WR[0])-1               ;
U 1CBE. 2F82,95 ;8651           CLR WR[1]                               ; ожидаемые данные=0
U 1CBF. 0869,3C ;8652           JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.WR.-1
U 1CC0. 89CB,94 ;8653           JMP [LOOP.TF.6.110]                     ; зацикливание при ошибке, если разрешено
U 1CC1. 8A70,1C ;8654           JSR [INC.OTHER]                         ;
                ;8655   LOOP.TF.6.111:
U 1CC2. AB80,15 ;8656           CLR Q                                   ;
U 1CC3. B640,15 ;8657           MOV LS[#1] TO WR[0]                     ;
U 1CC4. 8E60,15 ;8658           MOV WR[0] TO LS[L30]                    ; LS=1
U 1CC5. 3642,15 ;8659           MOV LS[#2] TO WR[0]                     ; WR0=2
U 1CC6. 4460,15 ;8660           SUB (LS[L30] FROM WR[0])-1              ;
U 1CC7. 2F82,95 ;8661           CLR WR[1]                               ; ожидаемые данные=
U 1CC8. 0869,3C ;8662           JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.WR.-1
U 1CC9. 89CC,24 ;8663           JMP [LOOP.TF.6.111]                     ; зацикливание при ошибке, если разрешено
U 1CCA. 8A70,1C ;8664           JSR [INC.OTHER]                         ;
                ;8665   LOOP.TF.6.112:
U 1CCB. AB80,15 ;8666           CLR Q                                   ;
U 1CCC. B640,15 ;8667           MOV LS[#1] TO WR[0]                     ;
U 1CCD. 8EA6,15 ;8668           MOV WR[0] TO LS[L53]                    ; LS=1
U 1CCE. 3642,15 ;8669           MOV LS[#2] TO WR[0]                     ; WR0=2
U 1CCF. 44A6,15 ;8670           SUB (LS[L53] FROM WR[0])-1              ;
U 1CD0. 2F82,95 ;8671           CLR WR[1]                               ; ожидаемые данные=0
U 1CD1. 0869,3C ;8672           JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.WR.-1
U 1CD2. 89CC,B4 ;8673           JMP [LOOP.TF.6.112]                     ; зацикливание при ошибке, если разрешено
U 1CD3. 8A70,1C ;8674           JSR [INC.OTHER]                         ;
                ;8675   LOOP.TF.6.113:
U 1CD4. AB80,15 ;8676           CLR Q                                   ;
U 1CD5. B640,15 ;8677           MOV LS[#1] TO WR[0]                     ;
U 1CD6. 3EE6,15 ;8678           MOV WR[0] TO LS[L73]                    ; LS=1
U 1CD7. 3642,15 ;8679           MOV LS[#2] TO WR[0]                     ; WR0=2
U 1CD8. C4E6,15 ;8680           SUB (LS[L73] FROM WR[0])-1              ;
U 1CD9. 2F82,95 ;8681           CLR WR[1]                               ; ожидаемые данные=0
U 1CDA. 0869,3C ;8682           JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.WR.-1
U 1CDB. 09CD,44 ;8683           JMP [LOOP.TF.6.113]                     ; зацикливание при ошибке, если разрешено
U 1CDC. 8A70,1C ;8684           JSR [INC.OTHER]                         ;
                .8685   LOOP.TF.6.114:
U 1CDD. AB80,15 ;8686           CLR Q                                   ;
U 1CDE. B698,15 ;8687           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1CDF. 3E10,15 ;8688           MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAA
U 1CE0. B69E,15 ;8689           MOV LS[ONES] TO WR[0]                   ; WR0=FFFFFFFF
U 1CE1. 4510,15 ;8690           BIC LS[T8] TO WR[0]                     ;
U 1CE2. B69A,95 ;8691           MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные=55555555
U 1CE3. 0869,3C ;8692           JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.WR
U 1CE4. 09CD,D4 ;8693           JMP [LOOP.TF.6.114]                     ; зацикливание при ошибке, если разрешено
U 1CE5. 8A70,1C ;8694           JSR [INC.OTHER]                         ;
                ;8695   LOOP.TF.6.115:
U 1CE6. AB80,15 ;8696           CLR Q                                   ;
U 1CE7. B698,15 ;8697           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1CE8. BE60,15 ;8698           MOV WR[0] TO LS[L30]                    ; LS=AAAAAAAA
U 1CE9. B69E,15 ;8699           MOV LS[ONES] TO WR[0]                   ; WR0=FFFFFFFF
U 1CEA. C560,15 ;8700           BIC LS[L30] TO WR[0]                    ;
```

```
U 1CEB, B69A,95 ;8701            MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные=55555555
U 1CEC, 0869,3C ;8702            JSR [CHECK.RESULT]                     ; проверка операции LS.MASK.WR
U 1CED, 89CE,64 ;8703            JMP [LOOP.TF.6.115]                    ; зацикливание при ошибке, если разрешено
U 1CEE, 8A70,1C ;8704            JSR [INC.OTHER]                        ;
                ;8705    LOOP.TF.6.116:
U 1CEF, A880,15 ;8706            CLR Q                                  ;
U 1CF0, B698,15 ;8707            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1CF1, BEA6,15 ;8708            MOV WR[0] TO LS[L53]                    ; LS=AAAAAAAA
U 1CF2, B69E,15 ;8709            MOV LS[ONES] TO WR[0]                   ; WR0=FFFFFFFF
U 1CF3, C5A6,15 ;8710            BIC LS[L53]  TO WR[0]                   ;
U 1CF4, B69A,95 ;8711            MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные=55555555
U 1CF5, 0869,3C ;8712            JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.WR
U 1CF6, 89CE,F4 ;8713            JMP [LOOP.TF.6.116]                     ; зацикливание при ошибке, если разрешено
U 1CF7, 8A70,1C ;8714            JSR [INC.OTHER]                         ;
                ;8715    LOOP.TF.6.117:
U 1CF8, A880,15 ;6716            CLR Q                                   ;
U 1CF9, B698,15 ;8717            MOV LS[ALTER.ODD] TO WR[0]               ;
U 1CFA, 3EE6,15 ;8718            MOV WR[0] TO LS[L73]                     ; LS=AAAAAAAA
U 1CFB, B69E,15 ;8719            MOV LS[ONES] TO WR[0]                    ; WR0=FFFFFFFF
U 1CFC, 45E6,15 ;8720            B1C LS[L73]  TO WR[0]                    ;
U 1CFD, B69A,95 ;8721            MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые данные=55555555
U 1CFE, 0869,3C ;8722            JSR [CHECK.RESULT]                       ; проверка операции LS.MASK.WR
U 1CFF, 89CF,84 ;8723            JMP [LOOP.TF.6.117]                      ; зацикливание при ошибке, если разрешено
U 1D00, 8A70,1C ;8724            JSR [INC.OTHER]                          ;
                ;8725    LOOP.TF.6.118:
U 1D01, AB80,15 ;8726            CLR Q                                    ;
U 1D02, B69E,15 ;8727            MOV LS[ONES] TO WR[0]                    ;
U 1D03, 3E10,15 ;8728            MOV WR[0] TO LS[T8]                      ; LS=-1
U 1D04, B640,15 ;8729            MOV LS[#1] TO WR[0]                      ; WR0=1
U 1D05, 4610,15 ;8730            ADD (LS[T8] TO WR[0])+1                  ;
U 1D06, 3640,95 ;8731            MOV LS[#1] TO WR[1]                      ; ожидаемые данные=1
U 1D07, 0869,3C ;8732            JSR [CHECK.RESULT]                       ; проверка операции LS.PLUS.WR+1
U 1D08, 09D0,14 ;8733            JMP [LOOP.TF.6.118]                      ; зацикливание при ошибке, если разрешено
U 1D09, 8A70,1C ;8734            JSR [INC.OTHER]                          ;
                ;8735    LOOP.TF.6.119:
U 1D0A, AB80,15 ;8736            CLR Q                                    ;
U 1D0B, B69E,15 ;8737            MOV LS[ONES] TO WR[0]                    ;
U 1D0C, BE60,15 ;8738            MOV WR[0] TO LS[L30]                     ; LS=-1
U 1D0D, B640,15 ;8739            MOV LS[#1] TO WR[0]                      ; WR0=1
U 1D0E, C660,15 ;8740            ADD (LS[L30] TO WR[0])+1                 ;
U 1D0F, 3640,95 ;8741            MOV LS[#1] TO WR[1]                      ; ожидаемые данные=1
U 1D10, 0869,3C ;8742            JSR [CHECK.RESULT]                       ; проверка операции LS.PLUS.WR+1
U 1D11, 89D0,A4 ;8743            JMP [LOOP.TF.6.119]                      ; зацикливание при ошибке, если разрешено
U 1D12, 8A70,1C ;8744            JSR [INC.OTHER]                          ;
                ;8745    LOOP.TF.6.11A:
U 1D13, AB80,15 ;8746            CLR Q                                    ;
U 1D14, B69E,15 ;8747            MOV LS[ONES] TO WR[0]                    ;
U 1D15, BEA6,15 ;8748            MOV WR[0] TO LS[L53]                     ; LS=-1
U 1D16, B640,15 ;8749            MOV LS[#1] TO WR[0]                      ; WR0=1
U 1D17, C6A6,15 ;8750            ADD (LS[L53] TO WR[0])+1                 ;
U 1D18, 3640,95 ;8751            MOV LS[#1] TO WR[1]                      ; ожидаемые данные=1
U 1D19, 0869,3C ;8752            JSR [CHECK.RESULT]                       ; проверка операции LS.PLUS.WR+1
U 1D1A, 09D1,34 ;8753            JMP [LOOP.TF.6.11A]                      ; зацикливание при ошибке, если разрешено
U 1D1B, 8A70,1C ;8754            JSR [INC.OTHER]                          ;
                ;8755    LOOP.TF.6.11B:
```

```
U 1D1C, AB80,15 ;8756            CLR Q                                  ;
U 1D1D, B69E,15 ;8757            MOV LS[ONES] TO WR[0]                  ;
U 1D1E, 3EE6,15 ;8758            MOV WR[0] TO LS[L73]                   ; LS=-1
U 1D1F, B640,15 ;8759            MOV LS[#1] TO WR[0]                    ; WR0=1
U 1D20, 46E6,15 ;8760            ADD (LS[L73]' TO WR[0])+1              ;
U 1D21, 3640,95 ;8761            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 1D22, 0869,3C ;8762            JSR [CHECK.RESULT]                     ; проверка операции LS.PLUS.WR+1
U 1D23, 09D1,C4 ;8763            JMP [LOOP.TF.6.11B]                    ; зацикливание при ошибке, если разрешено
U 1D24, 8A70,1C ;8764            JSR [INC.OTHER]                        ;
                ;8765    LOOP.TF.6.11C:
U 1D25, AB80,15 ;8766            CLR Q                                  ;
U 1D26, B698,15 ;8767            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1D27, 2040,15 ;8768            INC WR[0]                              ;
U 1D28, B698,15 ;8769            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1D29, 3E10,15 ;8770            MOV WR[0] TO LS[T8]                    ; LS=AAAAAAAB
U 1D2A, 369A,15 ;8771            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 1D2B, C710,15 ;8772            BIS LS[T8] TO WR[0]                    ;
U 1D2C, 369E,95 ;8773            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=FFFFFFFF
U 1D2D, 0869,3C ;8774            JSR [CHECK.RESULT]                     ; проверка операции LS.OR.WR
U 1D2E, 09D2,54 ;8775            JMP [LOOP.TF.6.11C]                    ; зацикливание при ошибке, если разрешено
U 1D2F, 8A70,1C ;8776            JSR [INC.OTHER]                        ;
                ;8777    LOOP.TF.6.11D:
U 1D30, AB80,15 ;8778            CLR Q                                  ;
U 1D31, B698,15 ;8779            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D32, 2040,15 ;8780            INC WR[0]                              ;
U 1D33, B698,15 ;8781            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D34, BE60,15 ;8782            MOV WR[0] TO LS[L30]                   ; LS=AAAAAAAB
U 1D35, 369A,15 ;8783            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 1D36, 4760,15 ;8784            BIS LS[L30] TO WR[0]                   ;
U 1D37, 369E,95 ;8785            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=FFFFFFFF
U 1D38, 0869,3C ;8786            JSR [CHECK.RESULT]                     ; проверка операции LS.OR.WR
U 1D39, 89D3,04 ;8787            JMP [LOOP.TF.6.11D]                    ; зацикливание при ошибке, если разрешено
U 1D3A, 8A70,1C ;8788            JSR [INC.OTHER]                        ;
                ;8789    LOOP.TF.6.11E:
U 1D3B, AB80,15 ;8790            CLR Q                                  ;
U 1D3C, B698,15 ;8791            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D3D, 2040,15 ;8792            INC WR[0]                              ;
U 1D3E, B698,15 ;8793            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D3F, BEA6,15 ;8794            MOV WR[0] TO LS[L53]                   ; LS=AAAAAAAB
U 1D40, 369A,15 ;8795            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 1D41, 47A6,15 ;8796            BIS LS[L53] TO WR[0]                   ;
U 1D42, 369E,95 ;8797            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=FFFFFFFF
U 1D43, 0869,3C ;8798            JSR [CHECK.RESULT]                     ; проверка операции LS.OR.WR
U 1D44, 09D3,B4 ;8799            JMP [LOOP.TF.6.11E]                    ; зацикливание при ошибке, если разрешено
U 1D45, 8A70,1C ;8800            JSR [INC.OTHER]                        ;
                ;8801    LOOP.TF.6.11F:
U 1D46, AB80,15 ;8802            CLR Q                                  ;
U 1D47, B698,15 ;8803            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D48, 2040,15 ;8804            INC WR[0]                              ;
U 1D49, B698,15 ;8805            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1D4A, 3EE6,15 ;8806            MOV WR[0] TO LS[L73]                   ; LS=AAAAAAAB
U 1D4B, 369A,15 ;8807            MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 1D4C, C7E6,15 ;8808            BIS LS[L73] TO WR[0]                   ;
U 1D4D, 369E,95 ;8809            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=FFFFFFFF
U 1D4E, 0869,3C ;8810            JSR [CHECK.RESULT]                     ; проверка операции LS.OR.WR
```

```
U 1D4F, 09D4,64 ;8811            JMP [LOOP.TF.6.11F]                ; зацикливание при ошибке, если разрешено
U 1D50, 8A70,1C ;8812            JSR [INC.OTHER]                    ;
                ;8813    LOOP.TF.6.120:
U 1D51, AB80,15 ;8814            CLR Q                              ;
U 1D52, 3642,15 ;8815            MOV LS[#2] TO WR[0]                ; WR0=2
U 1D53, 369F,15 ;8816            MOV LS[ONES] TO WR[2]              ;
U 1D54, BE11,15 ;8817            MOV WR[2] TO LS[T8]                ; LS=-1
U 1D55, C810,15 ;8818            ADD WR[0] PLUS LS[T8] TO Q         ;
U 1D56, A180,15 ;8819            MOV Q TO WR[0]                     ;
U 1D57, 3640,95 ;8820            MOV LS[#1] TO WR[1]                ; ожидаемые данные=1
U 1D58, 0869,3C ;8821            JSR [CHECK.RESULT]                 ; проверка операции WR.PLUS.LS.Q
U 1D59, 09D5,14 ;8822            JMP [LOOP.TF.6.120]                ; зацикливание при ошибке, если разрешено
U 1D5A, 8A70,1C ;8823            JSR [INC.OTHER]                    ;
                ;8824    LOOP.TF.6.121:
U 1D5B, AB80,15 ;8825            CLR Q                              ;
U 1D5C, 3642,15 ;8826            MOV LS[#2] TO WR[0]                ; WR0=2
U 1D5D, 369F,15 ;8827            MOV LS[ONES] TO WR[2]              ;
U 1D5E, 3E61,15 ;8828            MOV WR[2] TO LS[L30]               ; LS=-1
U 1D5F, 4860,15 ;8829            ADD WR[0] PLUS LS[L30] TO Q        ;
U 1D60, A180,15 ;8830            MOV Q TO WR[0]                     ;
U 1D61, 3640,95 ;8831            MOV LS[#1] TO WR[1]                ; ожидаемые данные=1
U 1D62, 0869,3C ;8832            JSR [CHECK.RESULT]                 ; проверка операции WR.PLUS.LS.Q
U 1D63, 09D5,B4 ;8833            JMP [LOOP.TF.6.121]                ; зацикливание при ошибке, если разрешено
U 1D64, 8A70,1C ;8834            JSR [INC.OTHER]                    ;
                ;8835    LOOP.TF.6.122:
U 1D65, AB80,15 ;8836            CLR Q                              ;
U 1D66, 3642,15 ;8837            MOV LS[#2] TO WR[0]                ; WR0=2
U 1D67, 369F,15 ;8838            MOV LS[ONES] TO WR[2]              ;
U 1D68, 3EA7,15 ;8839            MOV WR[2] TO LS[L53]               ; LS=-1
U 1D69, 48A6,15 ;8840            ADD WR[0] PLUS LS[L53] TO Q        ;
U 1D6A, A180,15 ;8841            MOV Q TO WR[0]                     ;
U 1D6B, 3640,95 ;8842            MOV LS[#1] TO WR[1]                ; ожидаемые данные=1
U 1D6C, 0869,3C ;8843            JSR [CHECK.RESULT]                 ; проверка операции WR.PLUS.LS.Q
U 1D6D, 89D6,54 ;8844            JMP [LOOP.TF.6.122]                ; зацикливание при ошибке, если разрешено
U 1D6E, 8A70,1C ;8845            JSR [INC.OTHER]                    ;
                ;8846    LOOP.TF.6.123:
U 1D6F, AB80,15 ;8847            CLR Q                              ;
U 1D70, 3642,15 ;8848            MOV LS[#2] TO WR[0]                ; WR0=2
U 1D71, 369F,15 ;8849            MOV LS[ONES] TO WR[2]              ;
U 1D72, BEE7,15 ;8850            MOV WR[2] TO LS[L73]               ; LS=-1
U 1D73, C8E6,15 ;8851            ADD WR[0] PLUS LS[L73] TO Q        ;
U 1D74, A180,15 ;8852            MOV Q TO WR[0]                     ;
U 1D75, 3640,95 ;8853            MOV LS[#1] TO WR[1]                ; ожидаемые данные=1
U 1D76, 0869,3C ;8854            JSR [CHECK.RESULT]                 ; проверка операции WR.PLUS.LS.Q
U 1D77, 89D6,F4 ;8855            JMP [LOOP.TF.6.123]                ; зацикливание при ошибке, если разрешено
U 1D78, 8A70,1C ;8856            JSR [INC.OTHER]                    ;
                ;8857    LOOP.TF.6.124:
U 1D79, AB80,15 ;8858            CLR Q                              ;
U 1D7A, 3642,15 ;8859            MOV LS[#2] TO WR[0]                ;
U 1D7B, 3E10,15 ;8860            MOV WR[0] TO LS[T8]                ; LS=2
U 1D7C, B640,15 ;8861            MOV LS[#1] TO WR[0]                ; WR0=1
U 1D7D, 4910,15 ;8862            SUB LS[T8] FROM WR[0] TO Q         ;
U 1D7E, A180,15 ;8863            MOV Q TO WR[0]                     ; выборка Q
U 1D7F, 369E,95 ;8864            MOV LS[ONES] TO WR[1]              ; ожидаемые данные=-1
U 1D80, 0869,3C ;8865            JSR [CHECK.RESULT]                 ; проверка операции LS.FROM.WR.Q
```

```
U 1D81, 09D7,94 ;8866            JMP [LOOP.TF.6.124]                    ; зацикливание при ошибке, если разрешено
U 1D82, 8A70,1C ;8867            JSR [INC.OTHER]                        ;
                ;8868    LOOP.TF.6.125:
U 1D83, AB80,15 ;8869            CLR Q                                  ;
U 1D84, 3642,15 ;8870            MOV LS[#2] TO WR[0]                    ;
U 1D85, BE60,15 ;8871            MOV WR[0] TO LS[L30]                   ; LS=2
U 1D86, B640,15 ;8872            MOV LS[#1] TO WR[0]                    ; WR0=1
U 1D87, C960,15 ;8873            SUB LS[L30] FROM WR[0] TO Q            ;
U 1D88, A180,15 ;8874            MOV Q TO WR[0]                         ; выборка Q
U 1D89, 369E,95 ;8875            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1D8A, 0869,3C ;8876            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.Q
U 1D8B, 09D8,34 ;8877            JMP [LOOP.TF.6.125]                    ; зацикливание при ошибке, если разрешено
U 1D8C, 8A70,1C ;8878            JSR [INC.OTHER]                        ;
                ;8879    LOOP.TF.6.126:
U 1D8D, AB80,15 ;8880            CLR Q                                  ;
U 1D8E, 3642,15 ;8881            MOV LS[#2] TO WR[0]                    ;
U 1D8F, BEA6,15 ;8882            MOV WR[0] TO LS[L53]                   ; LS=2
U 1D90, B640,15 ;8883            MOV LS[#1] TO WR[0]                    ; WR0=1
U 1D91, C9A6,15 ;8884            SUB LS[L53] FROM WR[0] TO Q            ;
U 1D92, A180,15 ;8885            MOV Q TO WR[0]                         ; выборка Q
U 1D93, 369E,95 ;8886            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1D94, 0869,3C ;8887            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.Q
U 1D95, 89D8,D4 ;8888            JMP [LOOP.TF.6.126]                    ; зацикливание при ошибке, если разрешено
U 1D96, 8A70,1C ;8889            JSR [INC.OTHER]                        ;
                ;8890    LOOP.TF.6.127:
U 1D97, AB80,15 ;8891            CLR Q                                  ;
U 1D98, 3642,15 ;8892            MOV LS[#2] TO WR[0]                    ;
U 1D99, 3EE6,15 ;8893            MOV WR[0] TO LS[L73]                   ; LS=2
U 1D9A, B640,15 ;8894            MOV LS[#1] TO WR[0]                    ; WR0=1
U 1D9B, 49E6,15 ;8895            SUB LS[L73] FROM WR[0] TO Q            ;
U 1D9C, A180,15 ;8896            MOV Q TO WR[0]                         ; выборка Q
U 1D9D, 369E,95 ;8897            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1D9E, 0869,3C ;8898            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.Q
U 1D9F, 09D9,74 ;8899            JMP [LOOP.TF.6.127]                    ; зацикливание при ошибке, если разрешено
U 1DA0, 8A70,1C ;8900            JSR [INC.OTHER]                        ;
                ;8901    LOOP.TF.6.128:
U 1DA1, AB80,15 ;8902            CLR Q                                  ;
U 1DA2, 3642,15 ;8903            MOV LS[#2] TO WR[0]                    ; WR0=2
U 1DA3, 3641,15 ;8904            MOV LS[#1] TO WR[2]                    ;
U 1DA4, BE11,15 ;8905            MOV WR[2] TO LS[T8]                    ; LS=1
U 1DA5, 4A10,15 ;8906            SUB WR[0] FROM LS[T8] TO Q             ;
U 1DA6, A180,15 ;8907            MOV Q TO WR[0]                         ; выборка Q
U 1DA7, 369E,95 ;8908            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1DA8, 0869,3C ;8909            JSR [CHECK.RESULT]                     ; проверка операции WR.FROM.LS.Q
U 1DA9, 09DA,14 ;8910            JMP [LOOP.TF.6.128]                    ; зацикливание при ошибке, если разрешено
U 1DAA, 8A70,1C ;8911            JSR [INC.OTHER]                        ;
                ;8912    LOOP.TF.6.129:
U 1DAB, AB80,15 ;8913            CLR Q                                  ;
U 1DAC, 3642,15 ;8914            MOV LS[#2] TO WR[0]                    ; WR0=2
U 1DAD, 3641,15 ;8915            MOV LS[#1] TO WR[2]                    ;
U 1DAE, 3E61,15 ;8916            MOV WR[2] TO LS[L30]                   ; LS=1
U 1DAF, CA60,15 ;8917            SUB WR[0] FROM LS[L30] TO Q            ;
U 1DB0, A180,15 ;8918            MOV Q TO WR[0]                         ; выборка Q
U 1DB1, 369E,95 ;8919            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1DB2, 0869,3C ;8920            JSR [CHECK.RESULT]                     ; проверка операции WR.FROM.LS.B
```

```
U 1DB3, 09DA,B4 ;8921            JMP [LOOP.TF.6.129]                    ; зацикливание при ошибке, если разрешено
U 1DB4, 8A70,1C ;8922            JSR [INC.OTHER]                        ;
                ;8923    LOOP.TF.6.12A:
U 1DB5, AB80,15 ;8924            CLR Q                                  ;
U 1DB6, 3642,15 ;8925            MOV LS[#2] TO WR[0]                    ; WR0=2
U 1DB7, 3641,15 ;8926            MOV LS[#1] TO WR[2]                    ;
U 1DB8, 3EA7,15 ;8927            MOV WR[2] TO LS[L53]                   ; LS=1
U 1DB9, CAA6,15 ;8928            SUB WR[0] FROM LS[L53] TO Q            ;
U 1DBA, A180,15 ;8929            MOV Q TO WR[0]                         ; выборка Q
U 1DBB, 369E,95 ;8930            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1DBC, 0869,3C ;8931            JSR [CHECK.RESULT]                     ; проверка операции WR.FROM:LS.Q
U 1DBD, 09DB,54 ;8932            JMP [LOOP.TF.6.12A]                    ; зацикливание при ошибке, если разрешено
U 1DBE, 8A70,1C ;8933            JSR [INC.OTHER]                        ;
                ;8934    LOOP.TF.6.12B:
U 1DBF, AB80,15 ;8935            CLR Q                                  ;
U 1DC0, 3642,15 ;8936            MOV LS[#2] TO WR[0]                    ; WR0=2
U 1DC1, 3641,15 ;8937            MOV LS[#1] TO WR[2]                    ;
U 1DC2, BEE7,15 ;8938            MOV WR[2] TO LS[L73]                   ; LS=1
U 1DC3, 4AE6,15 ;8939            SUB WR[0] FROM LS[L73] TO Q            ;
U 1DC4, A180,15 ;8940            MOV Q TO WR[0]                         ; выборка Q
U 1DC5, 369E,95 ;8941            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1DC6, 0869,3C ;8942            JSR [CHECK.RESULT]                     ; проверка операции WR.FROM.LS.Q
U 1DC7, 09D8,F4 ;8943            JMP [LOOP.TF.6.12B]                    ; зацикливание при ошибке, если разрешено
U 1DC8, 8A70,1C ;8944            JSR [INC.OTHER]                        ;
                ;8945    LOOP.TF.6.12C:
U 1DC9, AB80,15 ;8946            CLR Q                                  ;
U 1DCA, B698,15 ;8947            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1DCB, 2040,15 ;8948            INC WR[0]                              ; WR0=AAAAAAAB
U 1DCC, B698,15 ;8949            MOV LS[ALTER.EVEN] TO WR[2]            ;
U 1DCD, BE11,15 ;8950            MOV WR[2] TO LS[T8]                    ; LS=55555555
U 1DCE, CB10,15 ;8951            BIS WR[0] WITH LS[T8] TO Q             ;
U 1DCF, A180,15 ;8952            MOV Q TO WR[0]                         ; выборка Q
U 1DD0, 369E,95 ;8953            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = FFFFFFFF
U 1DD1, 0869,3C ;8954            JSR [CHECK.RESULT]                     ; проверка операции WR.OR.LS.Q
U 1DD2, 89DC,94 ;8955            JMP [LOOP.TF.6.12C]                    ; зацикливание при ошибке, если разрешено
U 1DD3, 8A70,1C ;8956            JSR.[INC.OTHER]                        ;
                ;8957    LOOP.TF.6.12D: -
U 1DD4, AB80,15 ;8958            CLR Q                                  ;
U 1DD5, B698,15 ;8959            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1DD6, 2040,15 ;8960            INC WR[0]                              ; WR0=AAAAAAAB
U 1DD7, B698,15 ;8961            MOV LS[ALTER.EVEN] TO WR[2]            ;
U 1DD8, 3E61,15 ;8962            MOV WR[2] TO LS[L30]                   ; LS=55555555
U 1DD9, 4B60,15 ;8963            BIS WR[0] WITH LS[L30] TO Q            ;
U 1DDA, A180,15 ;8964            MOV Q TO WR[0]                         ; выборка Q
U 1DDB, 369E,95 ;8965            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = FFFFFFFF
U 1DDC, 0869,3C ;8966            JSR [CHECK.RESULT]                     ; проверка операции WR.OR.LS.Q
U 1DDD, 89DD,44 ;8967            JMP [LOOP.TF.6.12D]                    ; зацикливание при ошибке, если разрешено
U 1DDE, 8A70,1C ;8968            JSR [INC.OTHER]                        ;
                ;8969    LOOP.TF.6.12E:
U 1DDF, AB80,15 ;8970            CLR Q                                  ;
U 1DE0, B698,15 ;8971            MOV LS[ALTER.ODD] TO WR[0]             ;
U 1DE1, 2040,15 ;8972            INC WR[0]                              ; WR0=AAAAAAAB
U 1DE2, B698,15 ;8973            MOV LS[ALTER.EVEN] TO WR[2]            ;
U 1DE3, 3EA7,15 ;8974            MOV WR[2] TO LS[L53]                   ; LS=55555555
U 1DE4, 4BA6,15 ;8975            BIS WR[0] WITH LS[L53] TO Q            ;
```

```
U 1DE5, A180,15 ;8976            MOV Q TO WR[0]                          ; выборка Q
U 1DE6, 369E,95 ;8977            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFF
U 1DE7, 0869,3C ;8978            JSR [CHECK.RESULT]                      ; проверка операции WR.OR.LS.
U 1DE8, 09DD,F4 ;8979            JMP [LOOP.TF.6.12E]                     ; зацикливание при ошибке, если разрешено
U 1DE9, 8A70,1C ;8980            JSR [INC.OTHER]                         ;
                ;8981    LOOP.TF.6.12F:
U 1DEA, A880,15 ;8982            CLR Q                                   ;
U 1DEB, B698,15 ;8983            MOV LS[ALTER.ODD] TO WR[0]               ;
U 1DEC, 2040,15 ;8984            INC WR[0]                               ; WR0=AAAAAAAB
U 1DED, B698,15 ;8985            MOV LS[ALTER.EVEN] TO WR[2]              ;
U 1DEE, BEE7,15 ;8986            MOV WR[2] TO LS[L73]                     ; LS=55555555
U 1DEF, CBE6,15 ;8987            BIS WR[0] WITH LS[L73] TO Q              ;
U 1DF0, A180,15 ;8988            MOV Q TO WR[0]                          ; выборка Q
U 1DF1, 369E,95 ;8989            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1DF2, 0869,3C ;8990            JSR [CHECK.RESULT]                      ; проверка операции WR.OR.LS.Q
U 1DF3, 09DE,A4 ;8991            JMP [LOOP.TF.6.12F]                     ; зацикливание при ошибке, если разрешено
U 1DF4, 8A70,1C ;8992            JSR [INC.OTHER]                         ;
                ;8993    LOOP.TF.6.130:
U 1DF5, A880,15 ;8994            CLR Q                                   ;
U 1DF6, B698,15 ;8995            MOV LS[ALTER.ODD] TO WR[0]               ; WR0=AAAAAAAA
U 1DF7, B629,15 ;8996            MOV LS[#FFFF] TO WR[2]                   ;
U 1DF9, BE11,15 ;8997            MOV WR[2] TO LS[T8]                      ; LS=0000FFFF
U 1DF9, 4C10,15 ;8998            AND WR[0] WITH LS[T8] TO Q               ;
U 1DFA, A180,15 ;8999            MOV Q TO WR[0]                          ; выборка Q
U 1DFB, 3698,95 ;9000            MOV LS[ALTER.ODD] TO WR[1]               ;
U 1DFC, 4518,95 ;9001            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 1DFD, 0869,3C ;9002            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.LS.Q
U 1DFE, B9DF,54 ;9003            JMP [LOOP.TF.6.130]                     ; зацикливание при ошибке, если разрешено
U 1DFF, 8A70,1C ;9004            JSR [INC.OTHER]                         ;
                ;9005    LOOP.TF.6.131:
U 1E00, A880,15 ;9006            CLR Q                                   ;
U 1E01, B698,15 ;9007            MOV LS[ALTER.ODD] TO WR[0]               ; WR0=AAAAAAAA
U 1E02, B629,15 ;9008            MOV LS[#FFFF] TO WR[2]                   ;
U 1E03, 3E61,15 ;9009            MOV WR[2] TO LS[L30]                     ; LS=0000FFFF
U 1E04, CC60,15 ;9010            AND WR[0] WITH LS[L30] TO Q              ;
U 1E05, A180,15 ;9011            MOV Q TO WR[0]                          ; выборка Q
U 1E06, 3698,95 ;9012            MOV LS[ALTER.ODD] TO WR[1]               ;
U 1E07, 4518,95 ;9013            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 1E08, 0869,3C ;9014            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.LS.Q
U 1E09, B9E0,04 ;9015            JMP [LOOP.TF.6.131]                     ; зацикливание при ошибке, если разрешено
U 1E0A, 8A70,1C ;9016            JSR [INC.OTHER]                         ;
                ;9017    LOOP.TF.6.132:
U 1E0B, A880,15 ;9018            CLR Q                                   ;
U 1E0C, B698,15 ;9019            MOV LS[ALTER.ODD] TO WR[0]               ; WR0=AAAAAAAA
U 1E0D, B629,15 ;9020            MOV LS[#FFFF] TO WR[2]                   ;
U 1E0E, 3EA7,15 ;9021            MOV WR[2] TO LS[L53]                     ; LS=0000FFFF
U 1E0F, CCA6,15 ;9022            AND WR[0] WITH LS[L53] TO Q              ;
U 1E10, A180,15 ;9023            MOV Q TO WR[0]                          ; выборка Q
U 1E11, 3698,95 ;9024            MOV LS[ALTER.ODD] TO WR[1]               ;
U 1E12, 4518,95 ;9025            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 1E13, 0869,3C ;9026            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.LS.Q
U 1E14, 09E0,B4 ;9027            JMP [LOOP.TF.6.132]                     ; зацикливание при ошибке, если разрешено
U 1E15, 8A70,1C ;9028            JSR [INC.OTHER]                         ;
                ;9029    LOOP.TF.6.133:
U 1E16, A880,15 ;9030            CLR Q
```

```
U 1E17, B698,15 ;9031            MOV LS[ALTER.ODD] TO WR[0]              ; WR0=AAAAAAAA
U 1E18, B629,15 ;9032            MOV LS[#FFFF] TO WR[2]                  ;
U 1E19, BEE7,15 ;9033            MOV WR[2] TO LS[L73]                    ; LS=0000FFFF
U 1E1A, 4CE6,15 ;9034            AND WR[0] WITH LS[L73] TO Q             ;
U 1E1B, A180,15 ;9035            MOV Q TO WR[0]                          ; выборка Q
U 1E1C, 3698,95 ;9036            MOV LS[ALTER.ODD] TO WR[1]              ;
U 1E1D, 4518,95 ;9037            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 1E1E, 0869,3C ;9038            JSR [CHECK.RESULT]                      ; проверка операции WR.AND.LS.Q
U 1E1F, 09E1,64 ;9039            JMP [LOOP.TF.6.133]                     ; зацикливание при ошибке, если разрешено
U 1E20, 8A70,1C ;9040            JSR [INC.OTHER]                         ;
                ;9041    LOOP.TF.6.134:
U 1E21, A880,15 ;9042            CLR Q                                   ;
U 1E22, B698,15 ;9043            MOV LS[ALTER.ODD] TO WR[0]              ; WR0=AAAAAAAA
U 1E23, B629,15 ;9044            MOV LS[#FFFF] TO WR[2]                  ;
U 1E24, BE11,15 ;9045            MOV WR[2] TO LS[T8]                     ; LS=0000FFFF
U 1E25, CD10,15 ;9046            XOR WR[0] WITH LS[T8] TO Q              ;
U 1E26, A180,15 ;9047            MOV Q TO WR[0]                          ; выборка Q
U 1E27, 3722,95 ;9048            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1E28, 0869,3C ;9049            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.LS.Q
U 1E29, B9E2,14 ;9050            JMP [LOOP.TF.6.134]                     ; зацикливание при ошибке, если разрешено
U 1E2A, 8A70,1C ;9051            JSR [INC.OTHER]                         ;
                ;9052    LOOP.TF.6.135:
U 1E2B, A880,15 ;9053            CLR Q                                   ;
U 1E2C, B698,15 ;9054            MOV LS[ALTER.ODD] TO WR[0]              ; WR0=AAAAAAAA
U 1E2D, B629,15 ;9055            MOV LS[#FFFF] TO WR[2]                  ;
U 1E2E, 3E61,15 ;9056            MOV WR[2] TO LS[L30]                    ; LS=0000FFFF
U 1E2F, 4D60,15 ;9057            XOR WR[0] WITH LS[L30] TO Q             ;
U 1E30, A180,15 ;9058            MOV Q TO WR[0]                          ; выборка Q
U 1E31, 3722,95 ;9059            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1E32, 0869,3C ;9060            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.LS.Q
U 1E33, B9E2,B4 ;9061            JMP [LOOP.TF.6.135]                     ; зацикливание при ошибке, если разрешено
U 1E34, 8A70,1C ;9062            JSR [INC.OTHER]                         ;
                ;9063    LOOP.TF.6.136:
U 1E35, A880,15 ;9064            CLR Q                                   ;
U 1E36, B698,15 ;9065            MOV LS[ALTER.ODD] TO WR[0]              ; WR0=AAAAAAAA
U 1E37, B629,15 ;9066            MOV LS[#FFFF] TO WR[2]                  ;
U 1E38, 3EA7,15 ;9067            MOV WR[2] TO LS[L53]                    ; LS=0000FFFF
U 1E39, 4DA6,15 ;9068            XOR WR[0] WITH LS[L53] TO Q             ;
U 1E3A, A180,15 ;9069            MOV Q TO WR[0]                          ; выборка Q
U 1E3B, 3722,95 ;9070            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1E3C, 0869,3C ;9071            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.LS.QB
U 1E3D, B9E3,54 ;9072            JMP [LOOP.TF.6.136]                     ; зацикливание при ошибке, если разрешено
U 1E3E, 8A70,1C ;9073            JSR [INC.OTHER]                         ;
                ;9074    LOOP.TF.6.137:
U 1E3F, A880,15 ;9075            CLR Q                                   ;
U 1E40, B698,15 ;9076            MOV LS[ALTER.ODD] TO WR[0]              ; WR0=AAAAAAAA
U 1E41, B629,15 ;9077            MOV LS[#FFFF] TO WR[2]                  ;
U 1E42, BEE7,15 ;9078            MOV WR[2] TO LS[L73]                    ; LS=0000FFFF
U 1E43, CDE6,15 ;9079            XOR WR[0] WITH LS[L73] TO Q             ;
U 1E44, A180,15 ;9080            MOV Q TO WR[0]                          ; выборка Q
U 1E45, 3722,95 ;9081            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1E46, 0869,3C ;9082            JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.LS.Q
U 1E47, B9E3,F4 ;9083            JMP [LOOP.TF.6.137]                     ; зацикливание при ошибке, если разрешено
U 1E48, 8A70,1C ;9084            JSR [INC.OTHER]                         ;
U 1E49, DF46,15 ;9085            MCOM LS[BIT3] TO WR[0]
```

```
U 1E4A, 3E8A,15 ;9086            MOV WR[0] TO LS[ERROR.MASK]            ; маскирование всеми битами, кроме бита 3 (проверка бита N)
                ;9087                                                   ; N)
                ;9088    LOOP.TF.6.138:
U 1E4B, AB80,15 ;9089            CLR Q                                  ;
U 1E4C, B69E,15 ;9090            MOV LS[ONES] TO WR[0]                  ;
U 1E4D, 3E10,15 ;9091            MOV WR[0] TO LS[T8]                    ; LS=-1
U 1E4E, B640,15 ;9092            MOV LS[#1] TO WR[0]                    ; WR0=1
                ;9093            CMP LS[T8] WITH WR[0],                 ;
U 1E4F, 4E10,35 ;9094            DT(LONG)&SET.ALU.CC                    ;
U 1E50, B7FC,15 ;9095            MOV LS[ALU.CC] TO WR[0]                ;
U 1E51, 3646,95 ;9096            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные =N установлен
U 1E52, 0869,3C ;9097            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.WR
U 1E53, B9E4,B4 ;9098            JMP [LOOP.TF.6.138]                    ; зацикливание при ошибке, если разрешено
U 1E54, BA70,1C ;9099            JSR [INC.OTHER]                        ;
                ;9100    LOOP.TF.6.139:
U 1E55, AB80,15 ;9101            CLR Q                                  ;
U 1E56, B69E,15 ;9102            MOV LS[ONES] TO WR[0]                  ;
U 1E57, BE60,15 ;9103            MOV WR[0] TO LS[L30]                   ; LS=-1
U 1E58, B640,15 ;9104            MOV LS[#1] TO WR[0]                    ; WR0=1
                ;9105            CMP LS[L30] WITH WR[0],                ;
U 1E59, CE60,35 ;9106            DT(LONG)&SET.ALU.CC                    ;
U 1E5A, B7FC,15 ;9107            MOV LS[ALU.CC] TO WR[0]                ;
U 1E5B, 3646,95 ;9108            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные =N установлен
U 1E5C, 0869,3C ;9109            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.WR
U 1E5D, B9E5,54 ;9110            JMP [LOOP.TF.6.139]                    ; зацикливание при ошибке, если разрешено
U 1E5E, BA70,1C ;9111            JSR [INC.OTHER]                        ;
                ;9112    LOOP.TF.6.13A:
U 1E5F, AB80,15 ;9113            CLR Q                                  ;
U 1E60, B69E,15 ;9114            MOV LS[ONES] TO WR[0]                  ;
U 1E61, BEA6,15 ;9115            MOV WR[0] TO LS[L53]                   ; LS=-1
U 1E62, B640,15 ;9116            MOV LS[#1] TO WR[0]                    ; WR0=1
                ;9117            CMP LS[L53] WITH WR[0],                ;
U 1E63, CEA6,35 ;9118            DT(LONG)&SET.ALU.CC                    ;
U 1E64, B7FC,15 ;9119            MOV LS[ALU.CC] TO WR[0]                ;
U 1E65, 3646,95 ;9120            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные =N установлен
U 1E66, 0869,3C ;9121            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.WR
U 1E67, B9E5,F4 ;9122            JMP [LOOP.TF.6.13A]                    ; зацикливание при ошибке, если разрешено
U 1E68, BA70,1C ;9123            JSR [INC.OTHER]                        ;
                ;9124    LOOP.TF.6.13B:
U 1E69, AB80,15 ;9125            CLR Q                                  ;
U 1E6A, B69E,15 ;9126            MOV LS[ONES] TO WR[0]                  ;
U 1E6B, 3EE6,15 ;9127            MOV WR[0] TO LS[L73]                   ; LS=-1
U 1E6C, B640,15 ;9128            MOV LS[#1] TO WR[0]                    ; WR0=1
                ;9129            CMP LS[L73] WITH WR[0],                ;
U 1E6D, 4EE6,35 ;9130            DT(LONG)&SET.ALU.CC                    ;
U 1E6E, B7FC,15 ;9131            MOV LS[ALU.CC] TO WR[0]                ;
U 1E6F, 3646,95 ;9132            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные =N установлен
U 1E70, 0869,3C ;9133            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.WR
U 1E71, B9E6,94 ;9134            JMP [LOOP.TF.6.13B]                    ; зацикливание при ошибке, если разрешено
U 1E72, BA70,1C ;9135            JSR [INC.OTHER]                        ;
                ;9136    LOOP.TF.6.13C:
U 1E73, AB80,15 ;9137            CLR Q                                  ;
U 1E74, B69E,15 ;9138            MOV LS[ONES] TO WR[0]                  ; WR0=-1
U 1E75, 3641,15 ;9139            MOV LS[#1] TO WR[2]                    ;
U 1E76, BE11,15 ;9140            MOV WR[2] TO LS[T8]                    ; LS=1
```

```
                    ;9141           CMP WR[0] WITH LS[T8],                  ;
U 1E77, CF10,35     ;9142           DT(LONG)&SET.ALU.CC                     ;
U 1E78, B7FC,15     ;9143           MOV LS[ALU.CC] TO WR[0]                 ;
U 1E79, 3646,95     ;9144           MOV LS[BIT3] TO WR[1]                   ;
U 1E7A, 0869,3C     ;9145           JSR [CHECK.RESULT]                      ; проверка операции WR.CMP.LS
U 1E7B, 09E7,34     ;9146           JMP [LOOP.TF.6.13C]                     ; зацикливание при ошибке, если разрешено
U 1E7C, 8A70,1C     ;9147           JSR [INC.OTHER]                         ;
                    ;9148    LOOP.TF.6.13D:
U 1E7D, AB80,15     ;9149           CLR Q                                   ;
U 1E7E, B69E,15     ;9150           MOV LS[ONES] TO WR[0]                   ; WR0=-1
U 1E7F, 3641,15     ;9151           MOV LS[#1] TO WR[2]                     ;
U 1E80, 3E61,15     ;9152           MOV WR[2] TO LS[L30]                    ; LS=1
                    ;9153           CMP WR[0] WITH LS[L30],                 ;
U 1E81, 4F60,35     ;9154           DT(LONG)&SET ALU.CC                     ;
U 1E82, B7FC,15     ;9155           MOV LS[ALU.CC] TO WR[0]
U 1E83, 3646,95     ;9156           MOV LS[BIT3] TO WR[1]
U 1E84, 0869,3C     ;9157           JSR [CHECK.RESULT]                      ; проверка операции WR.CMP.LS
U 1E85, 89E7,D4     ;9158           JMP [LOOP.TF.6.13D]                     ; зацикливание при ошибке, если разрешено
U 1E86, 8A70,1C     ;9159           JSR [INC.OTHER]                         ;
                    ;9160    LOOP.TF.6.13E:
U 1E87, AB80,15     ;9161           CLR Q                                   ;
U 1E88, B69E,15     ;9162           MOV LS[ONES] TO WR[0]                   ; WR0=-1
U 1E89, 3641,15     ;9163           MOV LS[#1] TO WR[2]                     ;
U 1E8A, 3EA7,15     ;9164           MOV WR[2] TO LS[L53]                    ; LS=1
                    ;9165           CMP WR[0] WITH LS[L53],                 ;
U 1E8B, 4FA6,35     ;9166           DT(LONG)&SET.ALU.CC                     ;
U 1E8C, B7FC,15     ;9167           MOV LS[ALU.CC] TO WR[0]                 ;
U 1E8D, 3646,95     ;9168           MOV LS[BIT3] TO WR[1]                   ;
U 1E8E, 0869,3C     ;9169           JSR [CHECK.RESULT]                      ; проверка операции WR.CMP.LS
U 1E8F, 89E8,74     ;9170           JMP [LOOP.TF.6.13E]                     ; зацикливание при ошибке, если разрешено
U 1E90, 8A70,1C     ;9171           JSR [INC.OTHER]                         ;
                    ;9172    LOOP.TF.6.13F:
U 1E91, AB80,15     ;9173           CLR Q                                   ;
U 1E92, B69E,15     ;9174           MOV LS[ONES] TO WR[0]                   ; WR0=-1
U 1E93, 3641,15     ;9175           MOV LS[#1] TO WR[2]                     ;
U 1E94, BEE7,15     ;9176           MOV WR[2] TO LS[L73]                    ; LS=1
                    ;9177           CMP WR[0] WITH LS[L73],                 ;
U 1E95, CFE6,35     ;9178           DT(LONG)&SET.ALU.CC                     ;
U 1E96, B7FC,15     ;9179           MOV LS[ALU.CC] TO WR[0]                 ;
U 1E97, 3646,95     ;9180           MOV LS[BIT3] TO WR[1]                   ;
U 1E98, 0869,3C     ;9181           JSR [CHECK.RESULT]                      ; проверка операции WR.CMP.LS
U 1E99, 09E9,14     ;9182           JMP [LOOP.TF.6.13F]                     ; зацикливание при ошибке, если разрешено
U 1E9A, 8A70,1C     ;9183           JSR [INC.OTHER]                         ;
U 1E9B, E58A,15     ;9184           CLR LS[ERROR.MASK]                      ;
                    ;9185    LOOP.TF.6.140:
U 1E9C, 3642,15     ;9186           MOV LS[#2] TO WR[0]                     ;
U 1E9D, 3E10,15     ;9187           MOV WR[0] TO LS[T8]                     ; LS=2
U 1E9E, 589E,15     ;9188           MOV LS[ONES] TO Q                       ; Q=-1
U 1E9F, D010,15     ;9189           ADD LS[T8] TO Q                         ;
U 1EA0, A180,15     ;9190           MOV Q TO WR[0]                          ;
U 1EA1, 3640,95     ;9191           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = -1
U 1EA2, 0869,3C     ;9192           JSR [CHECK.RESULT]                      ; проверка операции LS.PLUS.Q
U 1EA3, 89E9,C4     ;9193           JMP [LOOP.TF.6.140]                     ; зацикливание при ошибке, если разрешено
U 1EA4, 8A70,1C     ;9194           JSR [INC.OTHER]                         ;
                    ;9195    LOOP.TF.6.141:
```

```
U 1EA5, 3642,15 ;9196            MOV LS[#2] TO WR[0]                    ;
U 1EA6, BE60,15 ;9197            MOV WR[0] TO LS[L30]                   ; LS=2
U 1EA7, 589E,15 ;9198            MOV LS[ONES] TO Q                      ; Q=-1
U 1EA8, 5060,15 ;9199            ADD LS[L30] TO Q                       ;
U 1EA9, A180,15 ;9200            MOV Q TO WR[0]                         ;
U 1EAA, 3640,95 ;9201            MOV LS[#1] TO WR[1]                    ; ожидаемые данные = -1
U 1EAB, 0869,3C ;9202            JSR [CHECK.RESULT]                     ; проверка операции LS.PLUS.Q
U 1EAC, 89EA,54 ;9203            JMP [LOOP.TF.6.141]                    ; зацикливание при ошибке, если разрешено
U 1EAD, 8A70,1C ;9204            JSR [INC.OTHER]                        ;
                ;9205    LOOP.TF.6.142:
U 1EAE, 3642,15 ;9206            MOV LS[#2] TO WR[0]                    ;
U 1EAF, BEA6,15 ;9207            MOV WR[0] TO LS[L53]                   ; LS=2
U 1EB0, 589E,15 ;9208            MOV LS[ONES] TO Q                      ; Q=-1
U 1EB1, 50A6,15 ;9209            ADD LS[L53] TO Q                       ;
U 1EB2, A180,15 ;9210            MOV Q TO WR[0]                         ;
U 1EB3, 3640,95 ;9211            MOV LS[#1] TO WR[1]                    ; ожидаемые данные = -1
U 1EB4, 0869,3C ;9212            JSR [CHECK.RESULT]                     ; проверка операции LS.PLUS.Q
U 1EB5, 09EA,E4 ;9213            JMP [LOOP.TF.6.142]                    ; зацикливание при ошибке, если разрешено
U 1EB6, 8A70,1C ;9214            JSR [INC.OTHER]                        ;
                ;9215    LOOP.TF.6.143:
U 1EB7, 3642,15 ;9216            MOV LS[#2] TO WR[0]                    ;
U 1EB8, 3EE6,15 ;9217            MOV WR[0] TO LS[L73]                   ; LS=2
U 1EB9, 589E,15 ;9218            MOV LS[ONES] TO Q                      ; Q=-1
U 1EBA, D0E6,15 ;9219            ADD LS[L73] TO Q                       ;
U 1EBB, A180,15 ;9220            MOV Q TO WR[0]                         ;
U 1EBC, 3640,95 ;9221            MOV LS[#1] TO WR[1]                    ; ожидаемые данные = -1
U 1EBD, 0869,3C ;9222            JSR [CHECK.RESULT]                     ; проверка операции LS.PLUS.Q
U 1EBE, 89EB,74 ;9223            JMP [LOOP.TF.6.143]                    ; зацикливание при ошибке, если разрешено
U 1EBF, 8A70,1C ;9224            JSR [INC.OTHER]                        ;
                ;9225    LOOP.TF.6.144:
U 1EC0, 3642,15 ;9226            MOV LS[#2] TO WR[0]                    ;
U 1EC1, 3E10,15 ;9227            MOV WR[0] TO LS[T8]                    ; LS=2
U 1EC2, 5840,15 ;9228            MOV LS[#1] TO Q                        ; Q=1
U 1EC3, 5110,15 ;9229            SUB LS[T8] FROM Q                      ;
U 1EC4, A180,15 ;9230            MOV Q TO WR[0]                         ;
U 1EC5, 369E,95 ;9231            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1EC6, 0869,3C ;9232            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.Q
U 1EC7, 89EC,04 ;9233            JMP [LOOP.TF.6.144]                    ; зацикливание при ошибке, если разрешено
U 1EC8, 8A70,1C ;9234            JSR [INC.OTHER]                        ;
                ;9235    LOOP.TF.6.145:
U 1EC9, 3642,15 ;9236            MOV LS[#2] TO WR[0]                    ;
U 1ECA, BE60,15 ;9237            MOV WR[0] TO LS[L30]                   ; LS=2
U 1ECB, 5840,15 ;9238            MOV LS[#1] TO Q                        ; Q=1
U 1ECC, D160,15 ;9239            SUB LS[L30] FROM Q                     ;
U 1ECD, A180,15 ;9240            MOV Q TO WR[0]                         ;
U 1ECE, 369E,95 ;9241            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1ECF, 0869,3C ;9242            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.Q
U 1ED0, 89EC,94 ;9243            JMP [LOOP.TF.6.145]                    ; зацикливание при ошибке, если разрешено
U 1ED1, 8A70,1C ;9244            JSR [INC.OTHER]                        ;
                ;9245    LOOP.TF.6.146:
U 1ED2, 3642,15 ;9246            MOV LS[#2] TO WR[0]                    ;
U 1ED3, BEA6,15 ;9247            MOV WR[0] TO LS[L53]                   ; LS=2
U 1ED4, 5840,15 ;9248            MOV LS[#1] TO Q                        ; Q=1
U 1ED5, D1A6,15 ;9249            SUB LS[L53] FROM Q                     ;
U 1ED6, A180,15 ;9250            MOV Q TO WR[0]
```

```
U 1ED7, 369E,95 ;9251            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1ED8, 0869,3C ;9252            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.Q
U 1ED9, 89ED,24 ;9253            JMP [LOOP.TF.6.146]                    ; зацикливание при ошибке, если разрешено
U 1EDA, 8A70,1C ;9254            JSR [INC.OTHER]                        ;
                ;9255    LOOP.TF.6.147:
U 1EDB, 3642,15 ;9256            MOV LS[#2] TO WR[0]                    ;
U 1EDC, 3EE6,15 ;9257            MOV WR[0] TO LS[L73]                   ; LS=2
U 1EDD, 5840,15 ;9258            MOV LS[#1] TO Q                        ; Q=1
U 1EDE, 51E6,15 ;9259            SUB LS[L73] FROM Q                     ;
U 1EDF, A180,15 ;9260            MOV Q TO WR[0]                         ;
U 1EE0, 369E,95 ;9261            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1EE1, 0869,3C ;9262            JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.Q
U 1EE2, 89ED,84 ;9263            JMP [LOOP.TF.6.147]                    ; зацикливание при ошибке, если разрешено
U 1EE3, 8A70,1C ;9264            JSR [INC.OTHER]                        ;
                ;9265    LOOP.TF.6.148:
U 1EE4, D842,15 ;9266            MOV LS[#2] TO Q                        ; Q=2
U 1EE5, B640,15 ;9267            MOV LS[#1] TO WR[0]                    ;
U 1EE6, 3E10,15 ;9268            MOV WR[0] TO LS[T8]                    ; LS=1
U 1EE7, 5210,15 ;9269            SUB Q FROM LS[T8] TO Q                 ;
U 1EE8, A180,15 ;9270            MOV Q TO WR[0]                         ;
U 1EE9, 369E,95 ;9271            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1EEA, 0869,3C ;9272            JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.LS.Q
U 1EEB, 89EE,44 ;9273            JMP [LOOP.TF.6.148]                    ; зацикливание при ошибке, если разрешено
U 1EEC, 8A70,1C ;9274            JSR [INC.OTHER]                        ;
                ;9275    LOOP.TF.6.149:
U 1EED, D842,15 ;9276            MOV LS[#2] TO Q                        ; Q=2
U 1EEE, B640,15 ;9277            MOV LS[#1] TO WR[0]                    ;
U 1EEF, BE60,15 ;9278            MOV WR[0] TO LS[L30]                   ; LS=1
U 1EF0, D260,15 ;9279            SUB Q FROM LS[L30] TO Q                ;
U 1EF1, A180,15 ;9280            MOV Q TO WR[0]                         ;
U 1EF2, 369E,95 ;9281            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1EF3, 0869,3C ;9282            JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.LS.Q
U 1EF4, 89EE,D4 ;9283            JMP [LOOP.TF.6.149]                    ; зацикливание при ошибке, если разрешено
U 1EF5, 8A70,1C ;9284            JSR [INC.OTHER]                        ;
                ;9285    LOOP.TF.6.14A:
U 1EF6, D842,15 ;9286            MOV LS[#2] TO Q                        ; Q=2
U 1EF7, B640,15 ;9287            MOV LS[#1] TO WR[0]                    ; LS=1
U 1EF8, BEA6,15 ;9288            MOV WR[0] TO LS[L53]                   ;
U 1EF9, D2A6,15 ;9289            SUB Q FROM LS[L53] TO Q                ;
U 1EFA, A180,15 ;9290            MOV Q TO WR[0]                         ;
U 1EFB, 369E,95 ;9291            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1EFC, 0869,3C ;9292            JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.LS.Q
U 1EFD, 89EF,64 ;9293            JMP [LOOP.TF.6.14A]                    ; зацикливание при ошибке, если разрешено
U 1EFE, 8A70,1C ;9294            JSR [INC.OTHER]                        ;
                ;9295    LOOP.TF.6.14B:
U 1EFF, D842,15 ;9296            MOV LS[#2] TO Q                        ; Q=2
U 1F00, B640,15 ;9297            MOV LS[#1] TO WR[0]                    ; LS=1
U 1F01, 3EE6,15 ;9298            MOV WR[0] TO LS[L73]                   ;
U 1F02, 52E6,15 ;9299            SUB Q FROM LS[L73] TO Q                ;
U 1F03, A180,15 ;9300            MOV Q TO WR[0]                         ;
U 1F04, 369E,95 ;9301            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 1F05, 0869,3C ;9302            JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.LS.Q
U 1F06, 89EF,F4 ;9303            JMP [LOOP.TF.6.14B]                    ; зацикливание при ошибке, если разрешено
U 1F07, 8A70,1C ;9304            JSR [INC.OTHER]                        ;
                ;9305    LOOP.TF.6.14C:
```

```
U 1F08, B698,15 ;9306           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F09, 2040,15 ;9307           INC WR[0]                               ;
U 1F0A, 3E10,15 ;9308           MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAB
U 1F0B, D89A,15 ;9309           MOV LS[ALTER.EVEN] TO Q                 ; Q=55555555
U 1F0C, D310,15 ;9310           BIS LS[T8] TO Q                         ;
U 1F0D, A180,15 ;9311           MOV Q TO WR[0]                          ;
U 1F0E, 369E,95 ;9312           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1F0F, 0869,3C ;9313           JSR [CHECK.RESULT]                      ; проверка операции LS.OR.Q
U 1F10, 89F0,84 ;9314           JMP [LOOP.TF.6.14C]                     ; зацикливание при ошибке, если разрешено
U 1F11, 8A70,1C ;9315           JSR [INC.OTHER]                         ;
                ;9316    LOOP.TF.6.14D:
U 1F12, B698,15 ;9317           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F13, 2040,15 ;9318           INC WR[0]                               ;
U 1F14, BE60,15 ;9319           MOV WR[0] TO LS[L30]                    ; LS=AAAAAAAB
U 1F15, D89A,15 ;9320           MOV LS[ALTER.EVEN] TO Q                 ; Q=55555555
U 1F16, 5360,15 ;9321           BIS LS[L30] TO Q                        ;
U 1F17, A180,15 ;9322           MOV Q TO WR[0]                          ;
U 1F18, 369E,95 ;9323           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1F19, 0869,3C ;9324           JSR [CHECK.RESULT]                      ; проверка операции LS.OR.Q
U 1F1A, 09F1,24 ;9325           JMP [LOOP.TF.6.14D]                     ; зацикливание при ошибке, если разрешено
U 1F1B, 8A70,1C ;9326           JSR [INC.OTHER]                         ;
                ;9327    LOOP.TF.6.14E:
U 1F1C, B698,15 ;9328           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F1D, 2040,15 ;9329           INC WR[0]                               ;
U 1F1E, BEA6,15 ;9330           MOV WR[0] TO LS[L53]                    ; LS=AAAAAAAB
U 1F1F, D89A,15 ;9331           MOV LS[ALTER.EVEN] TO Q                 ; Q=55555555
U 1F20, 53A6,15 ;9332           BIS LS[L53] TO Q                        ;
U 1F21, A180,15 ;9333           MOV Q TO WR[0]                          ;
U 1F22, 369E,95 ;9334           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1F23, 0869,3C ;9335           JSR [CHECK.RESULT]                      ; проверка операции LS.OR.Q
U 1F24, 89F1,C4 ;9336           JMP [LOOP.TF.6.14E]                     ; зацикливание при ошибке, если разрешено
U 1F25, 8A70,1C ;9337           JSR [INC.OTHER]                         ;
                ;9338    LOOP.TF.6.14F:
U 1F26, B698,15 ;9339           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F27, 2040,15 ;9340           INC WR[0]                               ;
U 1F28, 3EE6,15 ;9341           MOV WR[0] TO LS[L73]                    ; LS=AAAAAAAB
U 1F29, D89A,15 ;9342           MOV LS[ALTER.EVEN] TO Q                 ; Q=55555555
U 1F2A, D3E6,15 ;9343           BIS LS[L73] TO Q                        ;
U 1F2B, A180,15 ;9344           MOV Q TO WR[0]                          ;
U 1F2C, 369E,95 ;9345           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 1F2D, 0869,3C ;9346           JSR [CHECK.RESULT]                      ; проверка операции LS.OR.Q
U 1F2E, 89F2,64 ;9347           JMP [LOOP.TF.6.14F]                     ; зацикливание при ошибке, если разрешено
U 1F2F, 8A70,1C ;9348           JSR [INC.OTHER]                         ;
                ;9349    LOOP.TF.6.150:
U 1F30, B698,15 ;9350           MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F31, 3E10,15 ;9351           MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAA
U 1F32, 589E,15 ;9352           MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
U 1F33, 5410,15 ;9353           BIC LS[T8] TO Q                         ;
U 1F34, A180,15 ;9354           MOV Q TO WR[0]                          ;
U 1F35, B69A,95 ;9355           MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 1F36, 0869,3C ;9356           JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.Q
U 1F37, 09F3,04 ;9357           JMP [LOOP.TF.6.150]                     ; зацикливание при ошибке, если разрешено
U 1F38, 8A70,1C ;9358           JSR [INC.OTHER]                         ;
                ;9359    LOOP.TF.6.151:
U 1F39, B698,15 ;9360           MOV LS[ALTER.ODD] TO WR[0]
```

```
U 1F3A, BE60,15 ;9361            MOV WR[0] TO LS[L30]                    ; LS=AAAAAAAA
U 1F3B, 589E,15 ;9362            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
U 1F3C, D460,15 ;9363            BIC LS[L30] TO Q                        ;
U 1F3D, A180,15 ;9364            MOV Q TO WR[0]                          ;
U 1F3E, B69A,95 ;9365            MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 1F3F, 0869,3C ;9366            JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.Q
U 1F40, 09F3,94 ;9367            JMP [LOOP.TF.6.151]                     ; зацикливание при ошибке, если разрешено
U 1F41, 8A70,1C ;9368            JSR [INC.OTHER]                         ;
                ;9369    LOOP.TF.6.152:
U 1F42, B698,15 ;9370            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F43, BEA6,15 ;9371            MOV WR[0] TO LS[L53]                    ; LS=AAAAAAAA
U 1F44, 589E,15 ;9372            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
U 1F45, D4A6,15 ;9373            BIC LS[L53] TO Q                        ;
U 1F46, A180,15 ;9374            MOV Q TO WR[0]                          ;
U 1F47, B69A,95 ;9375            MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 1F48, 0869,3C ;9376            JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.Q
U 1F49, 09F4,24 ;9377            JMP [LOOP.TF.6.152]                     ; зацикливание при ошибке, если разрешено
U 1F4A, 8A70,1C ;9378            JSR [INC.OTHER]                         ;
                ;9379    LOOP.TF.6.153:
U 1F4B, B698,15 ;9380            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F4C, 3EE6,15 ;9381            MOV WR[0] TO LS[L73]                    ; LS=AAAAAAAA
U 1F4D, 589E,15 ;9382            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
U 1F4E, 54E6,15 ;9383            BIC LS[L73] TO Q                        ;
U 1F4F, A180,15 ;9384            MOV Q TO WR[0]                          ;
U 1F50, B69A,95 ;9385            MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 1F51, 0869,3C ;9386            JSR [CHECK.RESULT]                      ; проверка операции LS.MASK.Q
U 1F52, 09F4,B4 ;9387            JMP [LOOP.TF.6.153]                     ; зацикливание при ошибке, если разрешено
U 1F53, 8A70,1C ;9388            JSR [INC.OTHER]                         ;
                ;9389    LOOP.TF.6.154:
U 1F54, B698,15 ;9390            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F55, 3E10,15 ;9391            MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAA
U 1F56, D828,15 ;9392            MOV LS[#FFFF] TO Q                      ; Q=0000FFFF
U 1F57, D510,15 ;9393            XOR LS[T8] TO Q                         ;
U 1F58, A180,15 ;9394            MOV Q TO WR[0]                          ;
U 1F59, 3722,95 ;9395            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1F5A, 0869,3C ;9396            JSR [CHECK.RESULT]                      ; проверка операции LS.XOR.Q
U 1F5B, 89F5,44 ;9397            JMP [LOOP.TF.6.154]                     ; зацикливание при ошибке, если разрешено
U 1F5C, 8A70,1C ;9398            JSR [INC.OTHER]                         ;
                ;9399    LOOP.TF.6.155:
U 1F5D, B698,15 ;9400            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F5E, BE60,15 ;9401            MOV WR[0] TO LS[L30]                    ; LS=AAAAAAAA
U 1F5F, D828,15 ;9402            MOV LS[#FFFF] TO Q                      ; Q=0000FFFF
U 1F60, 5560,15 ;9403            XOR LS[L30] TO Q                        ;
U 1F61, A180,15 ;9404            MOV Q TO WR[0]                          ;
U 1F62, 3722,95 ;9405            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 1F63, 0869,3C ;9406            JSR [CHECK.RESULT]                      ; проверка операции LS.XOR.Q
U 1F64, 89F5,D4 ;9407            JMP [LOOP.TF.6.155]                     ; зацикливание при ошибке, если разрешено
U 1F65, 8A70,1C ;9408            JSR [INC.OTHER]                         ;
                ;9409    LOOP.TF.6.156:
U 1F66, B698,15 ;9410            MOV LS[ALTER.ODD] TO WR[0]              ;
U 1F67, BEA6,15 ;9411            MOV WR[0] TO LS[L53]                    ; LS=AAAAAAAA
U 1F68, D828,15 ;9412            MOV LS[#FFFF] TO Q                      ; Q=0000FFFF
U 1F69, 55A6,15 ;9413            XOR LS[L53] TO Q                        ;
U 1F6A, A180,15 ;9414            MOV Q TO WR[0]                          ;
U 1F6B, 3722,95 ;9415            MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
```

```
U 1F6C, 0869,3C ;9416            JSR [CHECK.RESULT]                  ; проверка операции LS.XOR.Q
U 1F6D, 09F6,64 ;9417            JMP [LOOP.TF.6.156]                 ; зацикливание при ошибке, если разрешено
U 1F6E, 8A70,1C ;9418            JSR [INC.OTHER]                     ;
                ;9419    LOOP.TF.6.157:
U 1F6F, B698,15 ;9420            MOV LS[ALTER.ODD] TO WR[0]          ;
U 1F70, 3EE6,15 ;9421            MOV WR[0] TO LS[L73]                ; LS=AAAAAAAA
U 1F71, D828,15 ;9422            MOV LS[#FFFF] TO Q                  ; Q=0000FFFF
U 1F72, D5E6,15 ;9423            XOR LS[L73] TO Q                    ;
U 1F73, A180,15 ;9424            MOV Q TO WR[0]                      ;
U 1F74, 3722,95 ;9425            MOV LS[ALTER.MIX] TO WR[1]          ; ожидаемые данные = AAAA5555
U 1F75, 0869,3C ;9426            JSR [CHECK.RESULT]                  ; проверка операции LS.XOR.Q
U 1F76, 09F6,F4 ;9427            JMP [LOOP.TF.6.157]                 ; зацикливание при ошибке, если разрешено
U 1F77, 8A70,1C ;9428            JSR [INC.OTHER]                     ;
                ;9429    LOOP.TF.6.158:
U 1F78, B698,15 ;9430            MOV LS[ALTER.ODD] TO WR[0]          ;
U 1F79, 3E10,15 ;9431            MOV WR[0] TO LS[T8]                 ; LS=AAAAAAAA
U 1F7A, D828,15 ;9432            MOV LS[#FFFF] TO Q                  ; Q=0000FFFF
U 1F7B, D610,15 ;9433            AND LS[T8] TO Q                     ;
U 1F7C, A180,15 ;9434            MOV Q TO WR[0]                      ;
U 1F7D, 3698,95 ;9435            MOV LS[ALTER.ODD] TO WR[1]          ;
U 1F7E, 4518,95 ;9436            BIC LS[T12] TO WR[1]                ; ожидаемые данные = 0000AAAA
U 1F7F, 0869,3C ;9437            JSR [CHECK.RESULT]                  ; проверка операции LS.AND.Q
U 1F80, 09F7,84 ;9438            JMP [LOOP.TF.6.158]                 ; зацикливание при ошибке, если разрешено
U 1F81, 8A70,1C ;9439            JSR [INC.OTHER]                     ;
                ;9440    LOOP.TF.6.159:
U 1F82, B698,15 ;9441            MOV LS[ALTER.ODD] TO WR[0]          ;
U 1F83, 8E60,15 ;9442            MOV WR[0] TO LS[L30]                ; LS=AAAAAAAA
U 1F84, D828,15 ;9443            MOV LS[#FFFF] TO Q                  ; Q=0000FFFF
U 1F85, 5660,15 ;9444            AND LS[L30] TO Q                    ;
U 1F86, A180,15 ;9445            MOV Q TO WR[0]                      ;
U 1F87, 3698,95 ;9446            MOV LS[ALTER.ODD] TO WR[1]          ;
U 1F88, 4518,95 ;9447            BIC LS[T12] TO WR[1]                ; ожидаемые данные = 0000AAAA
U 1F89, 0869,3C ;9448            JSR [CHECK.RESULT]                  ; проверка операции LS.AND.Q
U 1F8A, 09F8,24 ;9449            JMP [LOOP.TF.6.159]                 ; зацикливание при ошибке, если разрешено
U 1F8B, 8A70,1C ;9450            JSR [INC.OTHER]                     ;
U 1F8C, 8A00,24 ;9451            JMP [LOOP.TF.6.15A]                 ;
                ;9452    2002:
                ;9453    LOOP.TF.6.15A:
U 2002, B698,15 ;9454            MOV LS[ALTER.ODD] TO WR[0]          ;
U 2003, 8EA6,15 ;9455            MOV WR[0] TO LS[L53]                ; LS=AAAAAAAA
U 2004, D828,15 ;9456            MOV LS[#FFFF] TO Q                  ; Q=0000FFFF
U 2005, 56A6,15 ;9457            AND LS[L53] TO Q                    ;
U 2006, A180,15 ;9458            MOV Q TO WR[0]                      ;
U 2007, 3698,95 ;9459            MOV LS[ALTER.ODD] TO WR[1]          ;
U 2008, 4518,95 ;9460            BIC LS[T12] TO WR[1]                ; ожидаемые данные = 0000AAAA
U 2009, 0869,3C ;9461            JSR [CHECK.RESULT]                  ; проверка операции LS.AND.Q
U 200A, 8A00,24 ;9462            JMP [LOOP.TF.6.15A]                 ; зацикливание при ошибке, если разрешено
U 200B, 8A70,1C ;9463            JSR [INC.OTHER]                     ;
U 200C, 8A01,A4 ;9464            JMP [LOOP.TF.6.15B]                 ;
                ;9465    201A:
                ;9466    LOOP.TF.6.15B:
U 201A, B698,15 ;9467            MOV LS[ALTER.ODD] TO WR[0]          ;
U 201B, 3EE6,15 ;9468            MOV WR[0] TO LS[L73]                ; LS=AAAAAAAA
U 201C, D828,15 ;9469            MOV LS[#FFFF] TO Q                  ; Q=0000FFFF
U 201D, D6E6,15 ;9470            AND LS[L73] TO Q                    ;
```

```
U 201E, A180,15 ;9471            MOV Q TO WR[0]                          ;
U 201F, 3698,95 ;9472            MOV LS[ALTER.ODD] TO WR[1]              ;
U 2020, 4518,95 ;9473            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 2021, 0869,3C ;9474            JSR [CHECK.RESULT]                      ; проверка операции LS.AND.Q
U 2022, 8A01,A4 ;9475            JMP [LOOP.TF.6.158]                     ; зацикливание при ошибке, если разрешено
U 2023, 8A70,1C ;9476            JSR [INC.OTHER]                         ;
U 2024, DF46,15 ;9477            MCOM LS[BIT3] TO WR[0]                  ;
U 2025, 3E8A,15 ;9478            MOV WR[0] TO LS[ERROR.MASK]             ; маскирование всеми битами, кроме бита 3 (проверка бита N)
                ;9479    LOOP.TF.6.15C:
U 2026, 367E,15 ;9480            MOV LS[BIT31] TO WR[0]                  ;
U 2027, 3E10,15 ;9481            MOV WR[0] TO LS[T8]                     ; LS=80000000
U 2028, 589E,15 ;9482            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
                ;9483              BIT Q WITH LS[T8],                    ;
U 2029, D710,35 ;9484            DT(LONG)&SET.ALU.CC                     ;
U 202A, B7FC,15 ;9485            MOV LS[ALU.CC] TO WR[0]                 ;
U 202B, 3646,95 ;9486            MOV LS[BIT3] TO WR[1]                   ; ожидаемые данные = N установлен
U 202C, 0869,3C ;9487            JSR [CHECK.RESULT]                      ; проверка операции LS.BIT.Q
U 202D, 8A02,64 ;9488            JMP [LOOP.TF.6.15C]                     ; зацикливание при ошибке, если разрешено
U 202E, 8A70,1C ;9489            JSR [INC.OTHER]                         ;
                ;9490    LOOP.TF.6.15D:
U 202F, 367E,15 ;9491            MOV LS[BIT31] TO WR[0]                  ;
U 2030, BE60,15 ;9492            MOV WR[0] TO LS[L30]                    ; LS=80000000
U 2031, 589E,15 ;9493            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
                ;9494              BIT Q WITH LS[L30],                   ;
U 2032, 5760,35 ;9495            DT(LONG)&SET.ALU.CC                     ;
U 2033, B7FC,15 ;9496            MOV LS[ALU.CC] TO WR[0]                 ;
U 2034, 3646,95 ;9497            MOV LS[BIT3] TO WR[1]                   ; ожидаемые данные = N установлен
U 2035, 0869,3C ;9498            JSR [CHECK.RESULT]                      ; проверка операции LS.BIT.Q
U 2036, 8A02,F4 ;9499            JMP [LOOP.TF.6.15D]                     ; зацикливание при ошибке, если разрешено
U 2037, 8A70,1C ;9500            JSR [INC.OTHER]                         ;
                ;9501    LOOP.TF.6.15E:
U 2038, 367E,15 ;9502            MOV LS[BIT31] TO WR[0]                  ;
U 2039, BEA6,15 ;9503            MOV WR[0] TO LS[L53]                    ; LS=80000000
U 203A, 589E,15 ;9504            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
                ;9505              BIT Q WITH LS[L53],                   ;
U 203B, 57A6,35 ;9506            DT(LONG)&SET.ALU.CC                     ;
U 203C, B7FC,15 ;9507            MOV LS[ALU.CC] TO WR[0]                 ;
U 203D, 3646,95 ;9508            MOV LS[BIT3] TO WR[1]                   ; ожидаемые данные = N установлен
U 203E, 0869,3C ;9509            JSR [CHECK.RESULT]                      ; проверка операции LS.BIT.Q
U 203F, 8A03,84 ;9510            JMP [LOOP.TF.6.15E]                     ; зацикливание при ошибке, если разрешено
U 2040, 8A70,1C ;9511            JSR [INC.OTHER]                         ;
                ;9512    LOOP.TF.6.15F:
U 2041, 367E,15 ;9513            MOV LS[BIT31] TO WR[0]                  ;
U 2042, 3EE6,15 ;9514            MOV WR[0] TO LS[L73]                    ; LS=80000000
U 2043, 589E,15 ;9515            MOV LS[ONES] TO Q                       ; Q=FFFFFFFF
                ;9516              BIT Q WITH LS[L73],                   ;
U 2044, D7E6,35 ;9517            DT(LONG)&SET.ALU.CC                     ;
U 2045, B7FC,15 ;9518            MOV LS[ALU.CC] TO WR[0]                 ;
U 2046, 3646,95 ;9519            MOV LS[BIT3] TO WR[1]                   ; ожидаемые данные = N установлен
U 2047, 0869,3C ;9520            JSR [CHECK.RESULT]                      ; проверка операции LS.BIT.Q
U 2048, 0A04,14 ;9521            JMP [LOOP.TF.6.15F]                     ; зацикливание при ошибке, если разрешено
U 2049, 8A70,1C ;9522            JSR [INC.OTHER]                         ;
U 204A, E58A,15 ;9523            CLR LS[ERROR.MASK]                      ;
                ;9524    LOOP.TF.6.160:
U 204B, A880,15 ;9525            CLR Q                                   ;
```

```
U 204C, 369A,15 ;9526           MOV LS[ALTER.EVEN] TO WR[0]              ;
U 204D, 3E10,15 ;9527           MOV WR[0] TO LS[T8]                      ; LS=55555555
U 204E, 5810,15 ;9528           MOV LS[T8] TO Q                          ;
U 204F, A180,15 ;9529           MOV Q TO WR[0]                           ;
U 2050, B69A,95 ;9530           MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые даные = 55555555
U 2051, 0869,3C ;9531           JSR [CHECK.RESULT]                       ; проверка операции MOV.LS.Q
U 2052, 0A04,B4 ;9532           JMP [LOOP.TF.6.160]                      ; зацикливание при ошибке, если разрешено
U 2053, 8A70,1C ;9533           JSR [INC.OTHER]                          ;
                ;9534    LOOP.TF.6.161:
U 2054, AB80,15 ;9535           CLR Q                                    ;
U 2055, 369A,15 ;9536           MOV LS[ALTER.EVEN] TO WR[0]              ;
U 2056, BE60,15 ;9537           MOV WR[0] TO LS[L30]                     ; LS=55555555
U 2057, D860,15 ;9538           MOV LS[L30] TO Q                         ;
U 2058, A180,15 ;9539           MOV Q TO WR[0]                           ;
U 2059, B69A,95 ;9540           MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые данные = 55555555
U 205A, 0869,3C ;9541           JSR [CHECK.RESULT]                       ; проверка операции MOV.LS.Q
U 205B, 8A05,44 ;9542           JMP [LOOP.TF.6.161]                      ; зацикливание при ошибке, если разрешено
U 205C, 8A70,1C ;9543           JSR [INC.OTHER]                          ;
                ;9544    LOOP.TF.6.162:
U 205D, AB80,15 ;9545           CLR Q                                    ;
U 205E, 369A,15 ;9546           MOV LS[ALTER.EVEN] TO WR[0]              ;
U 205F, BEA6,15 ;9547           MOV WR[0] TO LS[L53]                     ; LS=55555555
U 2060, D8A6,15 ;9548           MOV LS[L53] TO Q                         ;
U 2061, A180,15 ;9549           MOV Q TO WR[0]                           ;
U 2062, B69A,95 ;9550           MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые данные=55555555
U 2063, 0869,3C ;9551           JSR [CHECK.RESULT]                       ; проверка операции MOV.LS.Q
U 2064, 8A05,D4 ;9552           JMP [LOOP.TF.6.162]                      ; зацикливание при ошибке, если разрешено
U 2065, 8A70,1C ;9553           JSR [INC.OTHER]                          ;
                ;9554    LOOP.TF.6.163:
U 2066, AB80,15 ;9555           CLR Q                                    ;
U 2067, 369A,15 ;9556           MOV LS[ALTER.EVEN] TO WR[0]              ;
U 2068, 3EE6,15 ;9557           MOV WR[0] TO LS[L73]                     ; LS=55555555
U 2069, 58E6,15 ;9558           MOV LS[L73] TO Q                         ;
U 206A, A180,15 ;9559           MOV Q TO WR[0]                           ;
U 206B, B69A,95 ;9560           MOV LS[ALTER.EVEN] TO WR[1]              ; ожидаемые данные=55555555
U 206C, 0869,3C ;9561           JSR [CHECK.RESULT]                       ; проверка операции MOV.LS.Q
U 206D, 0A06,64 ;9562           JMP [LOOP.TF.6.163]                      ; зацикливание при ошибке, если разрешено
U 206E, 8A70,1C ;9563           JSR [INC.OTHER]                          ;
U 206F, DF46,15 ;9564           MCOM LS[BIT3] TO WR[0]                   ;
U 2070, 3E8A,15 ;9565           MOV WR[0] TO LS[ERROR.MASK]              ; маскирование всеми битами кроме бита 3 (проверка бита N.
U 2071, 0A07,24 ;9566           JMP [LOOP.TF.6.164]                      ;
                ;9567    LOOP.TF.6.164:
U 2072, 367E,15 ;9568           MOV LS[BIT31] TO WR[0]                   ;
U 2073, 3E10,15 ;9569           MOV WR[0] TO LS[T8]                      ; LS=80000000
U 2074, B69E,15 ;9570           MOV LS[ONES] TO WR[0]                    ; WR0=FFFFFFFF
                ;9571           BIT LS[T8] WITH WR[0],                   ;
U 2075, 5910,35 ;9572           DT(LONG)&SET.ALU.CC                      ;
U 2076, B7FC,15 ;9573           MOV LS[ALU.CC] TO WR[0]                  ;
U 2077, 3646,95 ;9574           MOV LS[BIT3] TO WR[1]                    ; ожидаемые данные = N установлен
U 2078, 0869,3C ;9575           JSR [CHECK.RESULT]                       ; проверка операции WR.BIT.LS
U 2079, 0A07,24 ;9576           JMP [LOOP.TF.6.164]                      ; зацикливание при ошибке, если разрешено
U 207A, 8A70,1C ;9577           JSR [INC.OTHER]                          ;
                ;9578    LOOP.TF.6.165:
U 207B, 367E,15 ;9579           MOV LS[BIT31] TO WR[0]                   ;
U 207C, BE60,15 ;9580           MOV WR[0] TO LS[L30]                     ; LS=80000000
```

```
U 207D, B69E,15 ;9581            MOV LS[ONES] TO WR[0]                  ; WR=FFFFFFFF
                ;9582            BIT LS[L30] WITH WR[0],                ;
U 207E, D960,35 ;9583            DT(LONG)&SET.ALU.CC                    ;
U 207F, B7FC,15 ;9584            MOV LS[ALU.CC] TO WR[0]                ;
U 2080, 3646,95 ;9585            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 2081, 0869,3C ;9586            JSR [CHECK.RESULT]                     ; проверка операции WR.BIT.LS
U 2082, 0A07,B4 ;9587            JMP [LOOP.TF.6.165]                    ; зацикливание при ошибке, если разрешено
U 2083, 8A70,1C ;9588            JSR [INC.OTHER]                        ;
                ;9589    LOOP.TF.6.166:
U 2084, 367E,15 ;9590            MOV LS[BIT31] TO WR[0]                 ;
U 2085, BEA6,15 ;9591            MOV WR[0] TO LS[L53]                   ; LS=80000000
U 2086, B69E,15 ;9592            MOV LS[ONES] TO WR[0]                  ; WR=FFFFFFFF
                ;9593            BIT LS[L53] WITH WR[0],                ;
U 2087, D9A6,35 ;9594            DT(LONG)&SET.ALU.CC                    ;
U 2088, B7FC,15 ;9595            MOV LS[ALU.CC] TO WR[0]                ;
U 2089, 3646,95 ;9596            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 208A, 0869,3C ;9597            JSR [CHECK.RESULT]                     ; проверка операции WR.BIT.LS
U 208B, 0A08,44 ;9598            JMP [LOOP.TF.6.166]                    ; зацикливание при ошибке, если разрешено
U 208C, 8A70,1C ;9599            JSR [INC.OTHER]                        ;
                ;9600    LOOP.TF.6.167:
U 208D, 367E,15 ;9601            MOV LS[BIT31] TO WR[0]                 ;
U 208E, 3EE6,15 ;9602            MOV WR[0] TO LS[L73]                   ; LS=80000000
U 208F, B69E,15 ;9603            MOV LS[ONES] TO WR[0]                  ; WR=FFFFFFFF
                ;9604            BIT LS[L73] WITH WR[0],                ;
U 2090, 59E6,35 ;9605            DT(LONG)&SET.ALU.CC                    ;
U 2091, B7FC,15 ;9606            MOV LS[ALU.CC] TO WR[0]                ;
U 2092, 3646,95 ;9607            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 2093, 0869,3C ;9608            JSR [CHECK.RESULT]                     ; проверка операции  WR.BIT.LS
U 2094, 0A08,D4 ;9609            JMP [LOOP.TF.6.167]                    ; зацикливание при ошибке, если разрешено
U 2095, 8A70,1C ;9610            JSR [INC.OTHER]                        ;
                ;9611    LOOP.TF.6.168:
U 2096, B69E,15 ;9612            MOV LS[ONES] TO WR[0]                  ;
U 2097, 3E10,15 ;9613            MOV WR[0] TO LS[TB]                    ; LS=-1
U 2098, 5840,15 ;9614            MOV LS[BIT0] TO Q                      ; Q=1
                ;9615              CMP LS[TB] WITH Q,                   ;
U 2099, 5A10,35 ;9616            DT(LONG)&SET.ALU.CC                    ;
U 209A, B7FC,15 ;9617            MOV LS[ALU.CC] TO WR[0]                ;
U 209B, 3646,95 ;9618            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 209C, 0869,3C ;9619            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.Q
U 209D, 0A09,64 ;9620            JMP [LOOP.TF.6.168]                    ; зацикливание при ошибке, если разрешено
U 209E, 8A70,1C ;9621            JSR [INC.OTHER]                        ;
                ;9622    LOOP.TF.6.169:
U 209F, B69E,15 ;9623            MOV LS[ONES] TO WR[0]                  ;
U 20A0, BE60,15 ;9624            MOV WR[0] TO LS[L30]                   ; LS=-1
U 20A1, 5840,15 ;9625            MOV LS[BIT0] TO Q                      ; Q=1
                ;9626              CMP LS[L30] WITH Q,                  ;
U 20A2, DA60,35 ;9627            DT(LONG)&SET.ALU.CC                    ;
U 20A3, B7FC,15 ;9628            MOV LS[ALU.CC] TO WR[0]                ;
U 20A4, 3646,95 ;9629            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 20A5, 0869,3C ;9630            JSR [CHECK.RESULT]                     ; проверка операции LS.CMP.Q
U 20A6, 0A09,F4 ;9631            JMP [LOOP.TF.6.169]                    ; зацикливание при ошибке, если разрешено
U 20A7, 8A70,1C ;9632            JSR [INC.OTHER]                        ;
                ;9633    LOOP.TF.6.16A:
U 20A8, B69E,15 ;9634            MOV LS[ONES] TO WR[0]                  ;
U 20A9, BEA6,15 ;9635            MOV WR[0] TO LS[L53]                   ; LS=-1
```

```
U 20AA, 5840,15 ;9636            MOV LS[BIT0] TO Q                    ; Q=1
                ;9637              CMP LS[L53] WITH Q,                ;
U 20AB, DAA6,35 ;9638            DT(LONG)&SET.ALU.CC                  ;
U 20AC, B7FC,15 ;9639            MOV LS[ALU.CC] TO WR[0]              ;
U 20AD, 3646,95 ;9640            MOV LS[BIT3] TO WR[1]                ; ожидаемые данные = N установлен
U 20AE, 0869,3C ;9641            JSR [CHECK.RESULT]                   ; проверка операции LS.CMP.Q
U 20AF, 8A0A,84 ;9642            JMP [LOOP.TF.6.16A]                  ; зацикливание при ошибке, если разрешено
U 20B0, 8A70,1C ;9643            JSR [INC.OTHER]                      ;
                ;9644    LOOP.TF.6.16B:
U 20B1, B69E,15 ;9645            MOV LS[ONES] TO WR[0]                ;
U 20B2, 3EE6,15 ;9646            MOV WR[0] TO LS[L73]                 ; LS=-1
U 20B3, 5840,15 ;9647            MOV LS[BIT0] TO Q                    ; Q=1
                ;9648              CMP LS[L73] WITH Q,                ;
U 20B4, 5AE6,35 ;9649            DT(LONG)&SET.ALU.CC                  ;
U 20B5, B7FC,15 ;9650            MOV LS[ALU.CC] TO WR[0]              ;
U 20B6, 3646,95 ;9651            MOV LS[BIT3] TO WR[1]                ; ожидаемые данные = N установлен
U 20B7, 0869,3C ;9652            JSR [CHECK.RESULT]                   ; проверка операции LS.CMP.Q
U 20B8, 0A0B,14 ;9653            JMP [LOOP.TF.6.16B]                  ; зацикливание при ошибке, если разрешено
U 20B9, 8A70,1C ;9654            JSR [INC.OTHER]                      ;
                ;9655    LOOP.TF.6.16C:
U 20BA, B640,15 ;9656            MOV LS[#1] TO WR[0]                  ;
U 20BB, 3E10,15 ;9657            MOV WR[0] TO LS[T8]                  ; LS=1
U 20BC, 589E,15 ;9658            MOV LS[ONES] TO Q                    ; Q=-1
                ;9659            CMP Q WITH LS[T8],                   ;
U 20BD, DB10,35 ;9660            DT(LONG)&SET.ALU.CC                  ;
U 20BE, B7FC,15 ;9661            MOV LS[ALU.CC] TO WR[0]              ;
U 20BF, 3646,95 ;9662            MOV LS[BIT3] TO WR[1]                ; ожидаемые данные = N установлен
U 20C0, 0869,3C ;9663            JSR [CHECK.RESULT]                   ; проверка операции Q.CMP.LS
U 20C1, 8A0B,A4 ;9664            JMP [LOOP.TF.6.16C]                  ; зацикливание при ошибке, если разрешено
U 20C2, 8A70,1C ;9665            JSR [INC.OTHER]                      ;
                ;9666    LOOP.TF.6.16D:
U 20C3, B640,15 ;9667            MOV LS[#1] TO WR[0]                  ;
U 20C4, BE60,15 ;9668            MOV WR[0] TO LS[L30]                 ; LS=1
U 20C5, 589E,15 ;9669            MOV LS[ONES] TO Q                    ; Q=-1
                ;9670            CMP Q WITH LS[L30],                  ;
U 20C6, 5B60,35 ;9671            DT(LONG)&SET.ALU.CC                  ;
U 20C7, B7FC,15 ;9672            MOV LS[ALU.CC] TO WR[0]              ;
U 20C8, 3646,95 ;9673            MOV LS[BIT3] TO WR[1]                ; ожидаемые данные = N установлен
U 20C9, 0869,3C ;9674            JSR [CHECK.RESULT]                   ; проверка операции Q.CMP.LS
U 20CA, 0A0C,34 ;9675            JMP [LOOP.TF.6.16D]                  ; зацикливание при ошибке, если разрешено
U 20CB, 8A70,1C ;9676            JSR [INC.OTHER]                      ;
                ;9677    LOOP.TF.6.16E:
U 20CC, B640,15 ;9678            MOV LS[#1] TO WR[0]                  ;
U 20CD, BEA6,15 ;9679            MOV WR[0] TO LS[L53]                 ; LS=1
U 20CE, 589E,15 ;9680            MOV LS[ONES] TO Q                    ; Q=-1
                ;9681              CMP Q WITH LS[L53],                ;
U 20CF, 5BA6,35 ;9682            DT(LONG)&SET.ALU.CC                  ;
U 20D0, B7FC,15 ;9683            MOV LS[ALU.CC] TO WR[0]              ;
U 20D1, 3646,95 ;9684            MOV LS[BIT3] TO WR[1]                ; ожидаемые данные = N установлен
U 20D2, 0869,3C ;9685            JSR [CHECK.RESULT]                   ; проверка операции Q.CMP.LS
U 20D3, 0A0C,C4 ;9686            JMP [LOOP.TF.6.16E]                  ; зацикливание при ошибке, если разрешено
U 20D4, 8A70,1C ;9687            JSR [INC.OTHER]                      ;
                ;9688    LOOP.TF.6.16F:
U 20D5, B640,15 ;9689            MOV LS[#1] TO WR[0]                  ;
U 20D6, 3EE6,15 ;9690            MOV WR[0] TO LS[L73]                 ; LS=1
```

```
U 20D7, 589E,15 ;9691            MOV LS[ONES] TO Q                      ; Q=-1
                ;9692              CMP Q WITH LS[L73],                  ;
U 20D8, DBE6,35 ;9693            DT(LONG)&SET.ALU.CC                    ;
U 20D9, B7FC,15 ;9694            MOV LS[ALU.CC] TO WR[0]                ;
U 20DA, 3646,95 ;9695            MOV LS[BIT3] TO WR[1]                  ; ожидаемые данные = N установлен
U 20DB, 0869,3C ;9696            JSR [CHECK.RESULT]                     ; проверка операции Q.CMP.LS
U 20DC, 8A0D,54 ;9697            JMP [LOOP.TF.6.16F]                    ; зацикливание при ошибке, если разрешено
U 20DD, 8A70,1C ;9698            JSR [INC.OTHER]                        ;
U 20DE, E58A,15 ;9699            CLR LS[ERROR.MASK]                     ; восстановление маски
                ;9700    LOOP.TF.6.170:
U 20DF, 3642,15 ;9701            MOV LS[#2] TO WR[0]                    ;
U 20E0, 3E10,15 ;9702            MOV WR[0] TO LS[T8]                    ; LS=2
U 20E1, 589E,15 ;9703            MOV LS[ONES] TO Q                      ; Q=-1
U 20E2, DC10,15 ;9704            ADD Q PLUS LS[T8] TO WR[0]             ;
U 20E3, 3640,95 ;9705            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 20E4, 0869,3C ;9706            JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.LS.TO.WR
U 20E5, 8A0D,F4 ;9707            JMP [LOOP.TF.6.170]                    ; зацикливание при ошибке, если разрешено
U 20E6, 8A70,1C ;9708            JSR [INC.OTHER]                        ;
                ;9709    LOOP.TF.6.171:
U 20E7, 3642,15 ;9710            MOV LS[#2] TO WR[0]                    ;
U 20E8, BE60,15 ;9711            MOV WR[0] TO LS[L30]                   ; LS=2
U 20E9, 589E,15 ;9712            MOV LS[ONES] TO Q                      ; Q=-1
U 20EA, 5C60,15 ;9713            ADD Q PLUS LS[L30] TO WR[0]            ;
U 20EB, 3640,95 ;9714            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 20EC, 0869,3C ;9715            JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.LS.TO.WR
U 20ED, 0A0E,74 ;9716            JMP [LOOP.TF.6.171]                    ; зацикливание при ошибке, если разрешено
U 20EE, 8A70,1C ;9717            JSR [INC.OTHER]                        ;
                ;9718    LOOP.TF.6.172:
U 20EF, 3642,15 ;9719            MOV LS[#2] TO WR[0]                    ;
U 20F0, BEA6,15 ;9720            MOV WR[0] TO LS[L53]                   ; LS=2
U 20F1, 589E,15 ;9721            MOV LS[ONES] TO Q                      ; Q=-1
U 20F2, 5CA6,15 ;9722            ADD Q PLUS LS[L53] TO WR[0]            ;
U 20F3, 3640,95 ;9723            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 20F4, 0869,3C ;9724            JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.LS.TO.WR
U 20F5, 8A0E,F4 ;9725            JMP [LOOP.TF.6.172]                    ; зацикливание при ошибке, если разрешено
U 20F6, 8A70,1C ;9726            JSR [INC.OTHER]                        ;
                ;9727    LOOP.TF.6.173:
U 20F7, 3642,15 ;9728            MOV LS[#2] TO WR[0]                    ;
U 20F8, 3EE6,15 ;9729            MOV WR[0] TO LS[L73]                   ; LS=2
U 20F9, 589E,15 ;9730            MOV LS[ONES] TO Q                      ; Q=-1
U 20FA, DCE6,15 ;9731            ADD Q PLUS LS[L73] TO WR[0]            ;
U 20FB, 3640,95 ;9732            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 20FC, 0869,3C ;9733            JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.LS.TO.WR
U 20FD, 8A0F,74 ;9734            JMP [LOOP.TF.6.173]                    ; зацикливание при ошибке, если разрешено
U 20FE, 8A70,1C ;9735            JSR [INC.OTHER]                        ;
                ;9736    LOOP.TF.6.174:
U 20FF, 3642,15 ;9737            MOV LS[#2] TO WR[0]                    ; WR0=2
U 2100, 3641,15 ;9738            MOV LS[#1] TO WR[2]                    ;
U 2101, BE11,15 ;9739            MOV WR[2] TO LS[T8]                    ; LS=1
U 2102, 5D10,15 ;9740            SUB WR[0] FROM LS[T8] TO WR            ;
U 2103, 369E,95 ;9741            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2104, 0869,3C ;9742            JSR [CHECK.RESULT]                     ; проверка операции WRA.FROM.LS.WRB
U 2105, 0A0F,F4 ;9743            JMP [LOOP.TF.6.174]                    ; зацикливание при ошибке, если разрешено
U 2106, 8A70,1C ;9744            JSR [INC.OTHER]                        ;
                ;9745    LOOP.TF.6.175:
```

```
U 2107,  3642,15 ;9746            MOV LS[#2] TO WR[0]                    ; WR0=2
U 2108,  3641,15 ;9747            MOV LS[#1] TO WR[2]                    ;
U 2109,  3E61,15 ;9748            MOV WR[2] TO LS[L30]                   ; LS=1
U 210A,  DD60,15 ;9749            SUB WR[0] FROM LS[L30] TO WR           ;
U 210B,  369E,95 ;9750            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 210C,  0869,3C ;9751            JSR [CHECK.RESULT]                     ; проверка операции WRA.FROM.LS.WRB
U 210D,  0A10,74 ;9752            JMP [LOOP.TF.6.175]                    ; зацикливание при ошибке, если разрешено
U 210E,  8A70,1C ;9753            JSR [INC.OTHER]                        ;
                 ;9754    LOOP.TF.6.176:
U 210F,  3642,15 ;9755            MOV LS[#2] TO WR[0]                    ; WR0=2
U 2110,  3641,15 ;9756            MOV LS[#1] TO WR[2]                    ;
U 2111,  3EA7,15 ;9757            MOV WR[2] TO LS[L53]                   ; LS=1
U 2112,  DDA6,15 ;9758            SUB WR[0] FROM LS[L53] TO WR           ;
U 2113,  369E,95 ;9759            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2114,  0869,3C ;9760            JSR [CHECK.RESULT]                     ; проверка операции WRA.FROM.LS.WRB
U 2115,  8A10,F4 ;9761            JMP [LOOP.TF.6.176]                    ; зацикливание при ошибке, если разрешено
U 2116,  8A70,1C ;9762            JSR [INC.OTHER]                        ;
                 ;9763    LOOP.TF.6.177:
U 2117,  3642,15 ;9764            MOV LS[#2] TO WR[0]                    ; WR0=2
U 2118,  3641,15 ;9765            MOV LS[#1] TO WR[2]                    ;
U 2119,  BEE7,15 ;9766            MOV WR[2] TO LS[L73]                   ; LS=1
U 211A,  5DE6,15 ;9767            SUB WR[0] FROM LS[L73] TO WR           ;
U 211B,  369E,95 ;9768            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 211C,  0869,3C ;9769            JSR [CHECK.RESULT]                     ; проверка операции WRA.FROM.LS.WRB
U 211D,  8A11,74 ;9770            JMP [LOOP.TF.6.177]                    ; зацикливание при ошибке, если разрешено
U 211E,  8A70,1C ;9771            JSR [INC.OTHER]                        ;
                 ;9772    LOOP.TF.6.178:
U 211F,  B640,15 ;9773            MOV LS[#1] TO WR[0]                    ;
U 2120,  3E10,15 ;9774            MOV WR[0] TO LS[T8]                    ; LS=1
U 2121,  5E10,15 ;9775            MNEG LS[T8] TO WR[0]                   ;
U 2122,  369E,95 ;9776            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2123,  0869,3C ;9777            JSR [CHECK.RESULT]                     ; проверка операции LS.NEG.WR
U 2124,  0A11,F4 ;9778            JMP [LOOP.TF.6.178]                    ; зацикливание при ошибке, если разрешено
U 2125,  8A70,1C ;9779            JSR [INC.OTHER]                        ;
                 ;9780    LOOP.TF.6.179:
U 2126,  B640,15 ;9781            MOV LS[#1] TO WR[0]                    ;
U 2127,  BE60,15 ;9782            MOV WR[0] TO LS[L30]                   ; LS=1
U 2128,  DE60,15 ;9783            MNEG LS[L30] TO WR[0]                  ;
U 2129,  369E,95 ;9784            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 212A,  0869,3C ;9785            JSR [CHECK.RESULT]                     ; проверка операции LS.NEG.WR
U 212B,  0A12,64 ;9786            JMP [LOOP.TF.6.179]                    ; зацикливание при ошибке, если разрешено
U 212C,  8A70,1C ;9787            JSR [INC.OTHER]                        ;
                 ;9788    LOOP.TF.6.17A:
U 212D,  B640,15 ;9789            MOV LS[#1] TO WR[0]                    ;
U 212E,  BEA6,15 ;9790            MOV WR[0] TO LS[L53]                   ; LS=1
U 212F,  DEA6,15 ;9791            MNEG LS[L53] TO WR[0]                  ;
U 2130,  369E,95 ;9792            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2131,  0869,3C ;9793            JSR [CHECK.RESULT]                     ; проверка операции LS.NEG.WR
U 2132,  8A12,D4 ;9794            JMP [LOOP.TF.6.17A]                    ; зацикливание при ошибке, если разрешено
U 2133,  8A70,1C ;9795            JSR [INC.OTHER]                        ;
                 ;9796    LOOP.TF.6.17B:
U 2134,  B640,15 ;9797            MOV LS[#1] TO WR[0]                    ;
U 2135,  3EE6,15 ;9798            MOV WR[0] TO LS[L73]                   ; LS=1
U 2136,  5EE6,15 ;9799            MNEG LS[L73] TO WR[0]                  ;
U 2137,  369E,95 ;9800            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
```

```
U 2138, 0869,3C ;9801            JSR [CHECK.RESULT]                    ; проверка операции LS.NEG.WR
U 2139, 0A13,44 ;9802            JMP [LOOP.TF.6.17B]                   ; зацикливание при ошибке, если разрешено
U 213A, 8A70,1C ;9803            JSR [INC.OTHER]                       ;
                ;9804    LOOP.TF.6.17C:
U 213B, 369A,15 ;9805            MOV LS[ALTER.EVEN] TO WR[0]           ;
U 213C, 3E10,15 ;9806            MOV WR[0] TO LS[T8]                   ; LS=55555555
U 213D, DF10,15 ;9807            MCOM LS[T8] TO WR[0]
U 213E, 3698,95 ;9808            MOV LS[ALTER.ODD] TO WR[1]            ; ожидаемые данные=AAAAAAAA
U 213F, 0869,3C ;9809            JSR [CHECK.RESULT]                    ; проверка операции LS.COM.WR
U 2140, 0A13,B4 ;9810            JMP [LOOP.TF.6.17C]                   ; зацикливание при ошибке, если разрешено
U 2141, 8A70,1C ;9811            JSR [INC.OTHER]                       ;
                ;9812    LOOP.TF.6.17D:
U 2142, 369A,15 ;9813            MOV LS[ALTER.EVEN] TO WR[0]           ;
U 2143, 8E60,15 ;9814            MOV WR[0] TO LS[L30]                  ; LS=55555555
U 2144, 5F60,15 ;9815            MCOM LS[L30] TO WR[0]                 ;
U 2145, 3698,95 ;9816            MOV LS[ALTER.ODD] TO WR[1]            ; ожидаемые данные=AAAAAAAA
U 2146, 0869,3C ;9817            JSR [CHECK.RESULT]                    ; проверка операции LS.COM.WR
U 2147, 8A14,24 ;9818            JMP [LOOP.TF.6.17D]                   ; зацикливание при ошибке, если разрешено
U 2148, 8A70,1C ;9819            JSR [INC.OTHER]                       ;
                ;9820    LOOP.TF.6.17E:
U 2149, 369A,15 ;9821            MOV LS[ALTER.EVEN] TO WR[0]           ;
U 214A, BEA6,15 ;9822            MOV WR[0] TO LS[L53]                  ; LS=55555555
U 214B, 5FA6,15 ;9823            MCOM LS[L53] TO WR[0]                 ;
U 214C, 3698,95 ;9824            MOV LS[ALTER.ODD] TO WR[1]            ; ожидаемые данные=AAAAAAAA
U 214D, 0869,3C ;9825            JSR [CHECK.RESULT]                    ; проверка операции LS.COM.WR
U 214E, 0A14,94 ;9826            JMP [LOOP.TF.6.17E]                   ; зацикливание при ошибке, если разрешено
U 214F, 8A70,1C ;9827            JSR [INC.OTHER]                       ;
                ;9828    LOOP.TF.6.17F:
U 2150, 369A,15 ;9829            MOV LS[ALTER.EVEN] TO WR[0]           ;
U 2151, 3EE6,15 ;9830            MOV WR[0] TO LS[L73]                  ; LS=55555555
U 2152, DFE6,15 ;9831            MCOM LS[L73] TO WR[0]                 ;
U 2153, 3698,95 ;9832            MOV LS[ALTER.ODD] TO WR[1]            ; ожидаемые данные=AAAAAAAA
U 2154, 0869,3C ;9833            JSR [CHECK.RESULT]                    ; проверка операции LS.COM.WR
U 2155, 8A15,04 ;9834            JMP [LOOP.TF.6.17F]                   ; зацикливание при ошибке, если разрешено
U 2156, 8A70,1C ;9835            JSR [INC.OTHER]                       ;
                ;9836     ;
                ;9837     ; ТЕСТЫ ПРИЕМНИКА МЕСТНОЙ ПАМЯТИ
                ;9838     ;
                ;9839    LOOP.TF.6.180:
U 2157, 3642,15 ;9840            MOV LS[#2] TO WR[0]                   ;
U 2158, 3E10,15 ;9841            MOV WR[0] TO LS[T8]                   ; LS=2
U 2159, B69E,15 ;9842            MOV LS[ONES] TO WR[0]                 ; WR0=-1
                ;9843              ADD LS[T8] TO WR[0],                ;
U 215A, E010,15 ;9844            XCHG                                  ;
U 215B, 3640,95 ;9845            MOV LS[#1] TO WR[1]                   ; ожидаемые данные=1
U 215C, 0869,3C ;9846            JSR [CHECK.RESULT]                    ; проверка LS.PLUS.WR в операции LS.PLUS.WR.XCHG
U 215D, 0A15,74 ;9847            JMP [LOOP.TF.6.180]                   ; зацикливание при ошибке, если разрешено
U 215E, B610,15 ;9848            MOV LS[T8] TO WR[0]                   ;
U 215F, 369E,95 ;9849            MOV LS[ONES] TO WR[1]                 ; ожидаемые данные=1
U 2160, 0869,3C ;9850            JSR [CHECK.RESULT]                    ; проверка XCHG в операции LS.PLUS.WR.XCHG
U 2161, 0A15,74 ;9851            JMP [LOOP.TF.6.180]                   ; зацикливание при ошибке, если разрешено
U 2162, 8A70,1C ;9852            JSR [INC.OTHER]                       ;
                ;9853    LOOP.TF.6.181:
U 2163, 3642,15 ;9854            MOV LS[#2] TO WR[0]                   ;
U 2164, 8E60,15 ;9855            MOV WR[0] TO LS[L30]                  ; LS=2
```

```
U 2165, B69E,15 ;9856            MOV LS[ONES] TO WR[0]                  ; WR0=-1
                ;9857              ADD LS[L30] TO WR[0],                ;
U 2166, 6060,15 ;9858            XCHG                                   ;
U 2167, 3640,95 ;9859            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 2168, 0869,3C ;9860            JSR [CHECK.RESULT]                     ; проверка LS.PLUS.WR в операции LS.PLUS.WR.XCHG
U 2169, 8A16,34 ;9861            JMP [LOOP.TF.6.181]                    ; зацикливание при ошибке, если разрешено
U 216A, 3660,15 ;9862            MOV LS[L30] TO WR[0]                   ;
U 216B, 369E,95 ;9863            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=1
U 216C, 0869,3C ;9864            JSR [CHECK.RESULT]                     ; проверка XCHG в операции LS.PLUS.WR.XCHG
U 216D, 8A16,34 ;9865            JMP [LOOP.TF.6.181]                    ; зацикливание при ошибке, если разрешено
U 216E, 8A70,1C ;9866            JSR [INC.OTHER]                        ;
                ;9867    LOOP.TF.6.182:
U 216F, 3642,15 ;9868            MOV LS[#2] TO WR[0]                    ;
U 2170, BEA6,15 ;9869            MOV WR[0] TO LS[L53]                   ; LS=2
U 2171, B69E,15 ;9870            MOV LS[ONES] TO WR[0]                  ; WR0=-1
                ;9871              ADD LS[L53] TO WR[0],                ;
U 2172, 60A6,15 ;9872            XCHG                                   ;
U 2173, 3640,95 ;9873            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 2174, 0869,3C ;9874            JSR [CHECK.RESULT]                     ; проверка LS.PLUS.WR в операции LS.PLUS.WR.XCHG
U 2175, 8A16,F4 ;9875            JMP [LOOP.TF.6.182]                    ; зацикливание при ошибке, если разрешено
U 2176, 36A6,15 ;9876            MOV LS[L53] TO WR[0]                   ;
U 2177, 369E,95 ;9877            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=1
U 2178, 0869,3C ;9878            JSR [CHECK.RESULT]                     ; проверка XCHG в операции LS.PLUS.WR.XCHG
U 2179, 8A16,F4 ;9879            JMP [LOOP.TF.6.182]                    ; зацикливание при ошибке, если разрешено
U 217A, 8A70,1C ;9880            JSR [INC.OTHER]                        ;
                ;9881    LOOP.TF.6.183:
U 217B, 3642,15 ;9882            MOV LS[#2] TO WR[0]                    ;
U 217C, 3EE6,15 ;9883            MOV WR[0] TO LS[L73]                   ; LS=2
U 217D, B69E,15 ;9884            MOV LS[ONES] TO WR[0]                  ; WR0=-1
                ;9885              ADD LS[L73] TO WR[0],                ;
U 217E, E0E6,15 ;9886            XCHG                                   ;
U 217F, 3640,95 ;9887            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 2180, 0869,3C ;9888            JSR [CHECK.RESULT]                     ; проверка LS.PLUS.WR в операции LS.PLUS.WR.XCHG
U 2181, 8A17,84 ;9889            JMP [LOOP.TF.6.183]                    ; зацикливание при ошибке, если разрешено
U 2182, B6E6,15 ;9890            MOV LS[L73] TO WR[0]                   ;
U 2183, 369E,95 ;9891            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2184, 0869,3C ;9892            JSR [CHECK.RESULT]                     ; проверка XCHG в операции LS.PLUS.WR.XCHG
U 2185, 8A17,84 ;9893            JMP [LOOP.TF.6.183]                    ; зацикливание при ошибке, если разрешено
U 2186, 8A70,1C ;9894            JSR [INC.OTHER]                        ;
                ;9895    LOOP.TF.6.184:
U 2187, 3642,15 ;9896            MOV LS[#2] TO WR[0]                    ;
U 2188, 3E10,15 ;9897            MOV WR[0] TO LS[T8]                    ; LS=2
U 2189, B640,15 ;9898            MOV LS[#1] TO WR[0]                    ; WR0=1
                ;9899            SUB LS[T8] FROM WR[0],                 ;
U 218A, 6110,15 ;9900            XCHG                                   ;
U 218B, 369E,95 ;9901            MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 218C, 0869,3C ;9902            JSR [CHECK.RESULT]                     ; проверка LS.FROM.WR в операции LS.FROM.WR.XCHG
U 218D, 8A18,74 ;9903            JMP [LOOP.TF.6.184]                    ; зацикливание при ошибке, если разрешено
U 218E, B610,15 ;9904            MOV LS[T8] TO WR[0]                    ;
U 218F, 3640,95 ;9905            MOV LS[#1] TO WR[1]                    ; ожидаемые данные=1
U 2190, 0869,3C ;9906            JSR [CHECK.RESULT]                     ; проверка XCHG в операции LS.FROM.WR.XCHG
U 2191, 8A18,74 ;9907            JMP [LOOP.TF.6.184]                    ; зацикливание при ошибке, если разрешено
U 2192, 8A70,1C ;9908            JSR [INC.OTHER]                        ;
                ;9909    LOOP.TF.6.185:
U 2193, 3642,15 ;9910            MOV LS[#2] TO WR[0]                    ;
```

```
U 2194,  8E60,15 ;9911            MOV WR[0] TO LS[L30]                    ; LS=2
U 2195,  8640,15 ;9912            MOV LS[#1] TO WR[0]                     ; WR0=1
                 ;9913              SUB LS[L30] FROM WR[0],               ;
U 2196,  E160,15 ;9914            XCHG                                    ;
U 2197,  369E,95 ;9915            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные=-1
U 2198,  0869,3C ;9916            JSR [CHECK.RESULT]                      ; проверка LS.FROM.WR в операции LS.FROM.WR.XCHG
U 2199,  8A19,34 ;9917            JMP [LOOP.TF.6.185]                     ; зацикливание при ошибке, если разрешено
U 219A,  3660,15 ;9918            MOV LS[L30] TO WR[0]                    ;
U 219B,  3640,95 ;9919            MOV LS[#1] TO WR[1]                     ; ожидаемые данные=1
U 219C,  0869,3C ;9920            JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.FROM.WR.XCHG
U 219D,  8A19,34 ;9921            JMP [LOOP.TF.6.185]                     ; зацикливание при ошибке, если разрешено
U 219E,  8A70,1C ;9922            JSR [INC.OTHER]                         ;
                 ;9923    LOOP.TF.6.186:
U 219F,  3642,15 ;9924            MOV LS[#2] TO WR[0]                     ;
U 21A0,  8EA6,15 ;9925            MOV WR[0] TO LS[L53]                    ; LS=2
U 21A1,  8640,15 ;9926            MOV LS[#1] TO WR[0]                     ; WR0=1
                 ;9927            SUB LS[L53] FROM WR[0],                 ;
U 21A2,  E1A6,15 ;9928            XCHG                                    ;
U 21A3,  369E,95 ;9929            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные=-1
U 21A4,  0869,3C ;9930            JSR [CHECK.RESULT]                      ; проверка LS.FROM.WR в операции LS.FROM.WR.XCHG
U 21A5,  8A19,F4 ;9931            JMP [LOOP.TF.6.186]                     ; зацикливание при ошибке, если разрешено
U 21A6,  36A6,15 ;9932            MOV LS[L53] TO WR[0]                    ;
U 21A7,  3640,95 ;9933            MOV LS[#1] TO WR[1]                     ; ожидаемые данные=1
U 21A8,  0869,3C ;9934            JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.FROM.WR.XCHG
U 21A9,  8A19,F4 ;9935            JMP [LOOP.TF.6.186]                     ; зацикливание при ошибке, если разрешено
U 21AA,  8A70,1C ;9936            JSR [INC.OTHER]                         ;
                 ;9937    LOOP.TF.6.187:
U 21AB,  3642,15 ;9938            MOV LS[#2]  TO WR[0]                    ;
U 21AC,  3EE6,15 ;9939            MOV WR[0] TO LS[L73]                    ; LS=2
U 21AD,  8640,15 ;9940            MOV LS[#1] TO WR[0]                     ; WR0=1
                 ;9941            SUB LS[L73] FROM WR[0],                 ;
U 21AE,  61E6,15 ;9942            XCHG                                    ;
U 21AF,  369E,95 ;9943            MOV LS[ONES] TO WR[1]                   ; ожидаемые данные=-1
U 21B0,  0869,3C ;9944            JSR [CHECK.RESULT]                      ; проверка LS.FROM.WR в операции LS.FROM.WR.XCHG
U 21B1,  0A1A,B4 ;9945            JMP [LOOP.TF.6.187]                     ; зацикливание при ошибке, если разрешено
U 21B2,  B6E6,15 ;9946            MOV LS[L73] TO WR[0]                    ;
U 21B3,  3640,95 ;9947            MOV LS[#1] TO WR[1]                     ; ожидаемые данные=1
U 21B4,  0869,3C ;9948            JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.FROM.WR.XCHG
U 21B5,  0A1A,B4 ;9949            JMP [LOOP.TF.6.187]                     ; зацикливание при ошибке, если разрешено
U 21B6,  8A70,1C ;9950            JSR [INC.OTHER]                         ;
                 ;9951    LOOP.TF.6.188:
U 21B7,  B698,15 ;9952            MOV LS[ALTER.ODD] TO WR[0]              ;
U 21B8,  3E10,15 ;9953            MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAA
U 21B9,  3628,15 ;9954            MOV LS[#FFFF] TO WR[0]                  ; WR0=0000FFFF
                 ;9955            AND LS[T8] TO WR[0],                    ;
U 21BA,  6210,15 ;9956            XCHG                                    ;
U 21BB,  3698,95 ;9957            MOV LS[ALTER.ODD] TO WR[1]              ;
U 21BC,  4518,95 ;9958            BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 21BD,  0869,3C ;9959            JSR [CHECK.RESULT]                      ; проверка LS.AND.WR в операции LS.AND.WR.XCHG
U 21BE,  8A18,74 ;9960            JMP [LOOP.TF.6.188]                     ; зацикливание при ошибке, если разрешено
U 21BF,  B610,15 ;9961            MOV LS[T8] TO WR[0]                     ;
U 21C0,  B628,95 ;9962            MOV LS[#FFFF] TO WR[1]                  ; ожидаемые данные = 0000FFFF
U 21C1,  0869,3C ;9963            JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.AND.WR.XCHG
U 21C2,  8A18,74 ;9964            JMP [LOOP.TF.6.188]                     ; зацикливание при ошибке, если разрешено
U 21C3,  8A70,1C ;9965            JSR [INC.OTHER]                         ;
```

```
                    ;9966     LOOP.TF.6.189:
U 21C4,  B698,15    ;9967               MOV LS[ALTER.ODD] TO WR[0]                ;
U 21C5,  BE60,15    ;9968               MOV WR[0] TO LS[L30]                      ; LS=AAAAAAAA
U 21C6,  3628,15    ;9969               MOV LS[#FFFF] TO WR[0]                    ; WR0=0000FFFF
                    ;9970               AND LS[L30] TO WR[0],                     ;
U 21C7,  E260,15    ;9971               XCHG                                      ;
U 21C8,  3698,95    ;9972               MOV LS[ALTER.ODD] TO WR[1]                ;
U 21C9,  4518,95    ;9973               BIC LS[T12] TO WR[1]                      ; ожидаемые данные = 0000AAAA
U 21CA,  0869,3C    ;9974               JSR [CHECK.RESULT]                        ; проверка LS.AND.WR в операции LS.AND.WR.XCHG
U 21CB,  0A1C,44    ;9975               JMP [LOOP.TF.6.189]                       ; зацикливание при ошибке, если разрешено
U 21CC,  3660,15    ;9976               MOV LS[L30] TO WR[0]                      ;
U 21CD,  B628,95    ;9977               MOV LS[#FFFF] TO WR[1]                    ; ожидаемые данные = 0000FFFF
U 21CE,  0869,3C    ;9978               JSR [CHECK.RESULT]                        ; проверка XCHG в операции LS.AND.WR.XCHG
U 21CF,  0A1C,44    ;9979               JMP [LOOP.TF.6.189]                       ; зацикливание при ошибке, если разрешено
U 21D0,  8A70,1C    ;9980               JSR [INC.OTHER]                           ;
                    ;9981     LOOP.TF.6.18A:
U 21D1,  B698,15    ;9982               MOV LS[ALTER.ODD] TO WR[0]                ;
U 21D2,  BEA6,15    ;9983               MOV WR[0] TO LS[L53]                      ; LS=AAAAAAAA
U 21D3,  3628,15    ;9984               MOV LS[#FFFF] TO WR[0]                    ; WR0=0000FFFF
                    ;9985               AND LS[L53] TO WR[0],                     ;
U 21D4,  E2A6,15    ;9986               XCHG                                      ;
U 21D5,  3698,95    ;9987               MOV LS[ALTER.ODD] TO WR[1]                ;
U 21D6,  4518,95    ;9988               BIC LS[T12] TO WR[1]                      ; ожидаемые данные = 0000AAAA
U 21D7,  0869,3C    ;9989               JSR [CHECK.RESULT]                        ; проверка LS.AND.WR в операции LS.AND.WR.XCHG
U 21D8,  8A1D,14    ;9990               JMP [LOOP.TF.6.18A]                       ; зацикливание при ошибке, если разрешено
U 21D9,  36A6,15    ;9991               MOV LS[L53] TO WR[0]                      ;
U 21DA,  B628,95    ;9992               MOV LS[#FFFF] TO WR[1]                    ; ожидаемые данные = 0000FFFF
U 21DB,  0869,3C    ;9993               JSR [CHECK.RESULT]                        ; проверка XCHG в операции LS.AND.WR.XCHG
U 21DC,  8A1D,14    ;9994               JMP [LOOP.TF.6.18A]                       ;
U 21DD,  8A70,1C    ;9995               JSR [INC.OTHER]                           ;
                    ;9996     LOOP.TF.6.188:
U 21DE,  B698,15    ;9997               MOV LS[ALTER.ODD] TO WR[0]                ;
U 21DF,  3EE6,15    ;9998               MOV WR[0] TO LS[L73]                      ; LS=AAAAAAAA
U 21E0,  3628,15    ;9999               MOV LS[#FFFF] TO WR[0]                    ; WR0=0000FFFF
                    ;10000                AND LS[L73] TO WR[0],                   ;
U 21E1,  62E6,15    ;10001              XCHG                                      ;
U 21E2,  3698,95    ;10002              MOV LS[ALTER.ODD] TO WR[1]                ;
U 21E3,  4518,95    ;10003              BIC LS[T12] TO WR[1]                      ; ожидаемые данные = 0000AAAA
U 21E4,  0869,3C    ;10004              JSR [CHECK.RESULT]                        ; проверка LS.AND.WR в операции LS.AND.WR.XCHG
U 21E5,  8A1D,E4    ;10005              JMP [LOOP.TF.6.188]                       ; зацикливание при ошибке, если разрешено
U 21E6,  B6E6,15    ;10006              MOV LS[L73] TO WR[0]                      ;
U 21E7,  B628,95    ;10007              MOV LS[#FFFF] TO WR[1]                    ; ожидаемые данные = 0000FFFF
U 21E8,  0869,3C    ;10008              JSR [CHECK.RESULT]                        ; проверка XCHG в операции LS.AND.WR.XCHG
U 21E9,  8A1D,E4    ;10009              JMP [LOOP.TF.6.188]                       ; зацикливание при ошибке, если разрешено
U 21EA,  8A70,1C    ;10010              JSR [INC.OTHER]                           ;
                    ;10011    LOOP.TF.6.18C:
U 21EB,  B698,15    ;10012              MOV LS[ALTER.ODD] TO WR[0]                ;
U 21EC,  3E10,15    ;10013              MOV WR[0] TO LS[T8]                       ; LS=AAAAAAAA
U 21ED,  3628,15    ;10014              MOV LS[#FFFF] TO WR[0]                    ; WR0=0000FFFF
                    ;10015                XOR LS[T8] TO WR[0],                    ;
U 21EE,  E310,15    ;10016              XCHG                                      ;
U 21EF,  3722,95    ;10017              MOV LS[ALTER.MIX] TO WR[1]                ; ожидаемые данные = AAAA5555
U 21F0,  0869,3C    ;10018              JSR [CHECK.RESULT]                        ; проверка LS.XOR.WR в операции LS.XOR.WR.XCHG
U 21F1,  8A1E,B4    ;10019              JMP [LOOP.TF.6.18C]                       ; зацикливание при ошибке, если разрешено
U 21F2,  B610,15    ;10020              MOV LS[T8] TO WR[0]                       ;
```

```
U 21F3, B628,95 ;10021          MOV LS[#FFFF] TO WR[1]                  ; ожидаемые данные = 0000FFFF
U 21F4, 0869,3C ;10022          JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.XOR.WR.XCHG
U 21F5, BA1E,B4 ;10023          JMP [LOOP.TF.6.18C]                     ; зацикливание при ошибке, если разрешено
U 21F6, BA70,1C ;10024          JSR [INC.OTHER]                         ;
                ;10025  LOOP.TF.6.18D:
U 21F7, B698,15 ;10026          MOV LS[ALTER.ODD] TO WR[0]              ;
U 21F8, BE60,15 ;10027          MOV WR[0] TO LS[L30]                    ; LS=AAAAAAAA
U 21F9, 3628,15 ;10028          MOV LS[#FFFF] TO WR[0]                  ; WR0=0000FFFF
                ;10029            XOR LS[L30] TO WR[0],                 ;
U 21FA, 6360,15 ;10030          XCHG                                    ;
U 21FB, 3722,95 ;10031          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 21FC, 0869,3C ;10032          JSR [CHECK.RESULT]                      ; проверка LS.XOR.WR в операции LS.XOR.WR.XCHG
U 21FD, 0A1F,74 ;10033          JMP [LOOP.TF.6.18D]                     ; зацикливание при ошибке, если разрешено
U 21FE, 3660,15 ;10034          MOV LS[L30] TO WR[0]                    ;
U 21FF, B628,95 ;10035          MOV LS[#FFFF] TO WR[1]                  ; ожидаемые данные = 0000FFFF
U 2200, 0869,3C ;10036          JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.XOR.WR.XCHG
U 2201, 0A1F,74 ;10037          JMP [LOOP.TF.6.18D]                     ; зацикливание при ошибке, если разрешено
U 2202, BA70,1C ;10038          JSR [INC.OTHER]                         ;
                ;10039  LOOP.TF.6.18E:
U 2203, B698,15 ;10040          MOV LS[ALTER.ODD] TO WR[0]              ;
U 2204, BEA6,15 ;10041          MOV WR[0] TO LS[L53]                    ; LS=AAAAAAAA
U 2205, 3628,15 ;10042          MOV LS[#FFFF] TO WR[0]                  ; WR0=0000FFFF
                ;10043            XOR LS[L53] TO WR[0],                 ;
U 2206, 63A6,15 ;10044          XCHG                                    ;
U 2207, 3722,95 ;10045          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 2208, 0869,3C ;10046          JSR [CHECK.RESULT]                      ; проверка LS.XOR.WR в операции LS.XOR.WR.XCHG
U 2209, BA20,34 ;10047          JMP [LOOP.TF.6.18E]                     ; зацикливание при ошибке, если разрешено
U 220A, 36A6,15 ;10048          MOV LS[L53] TO WR[0]                    ;
U 220B, B628,95 ;10049          MOV LS[#FFFF] TO WR[1]                  ; ожидаемые данные = 0000FFFF
U 220C, 0869,3C ;10050          JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.XOR.WR.XCHG
U 220D, BA20,34 ;10051          JMP [LOOP.TF.6.18E]                     ; зацикливание при ошибке, если разрешено
U 220E, BA70,1C ;10052          JSR [INC.OTHER]                         ;
                ;10053  LOOP.TF.6.18F:
U 220F, B698,15 ;10054          MOV LS[ALTER.ODD] TO WR[0]              ;
U 2210, 3EE6,15 ;10055          MOV WR[0] TO LS[L73]                    ; LS=AAAAAAAA
U 2211, 3628,15 ;10056          MOV LS[#FFFF] TO WR[0]                  ; WR0=0000FFFF
                ;10057            XOR LS[L73] TO WR[0],                 ;
U 2212, E3E6,15 ;10058          XCHG                                    ;
U 2213, 3722,95 ;10059          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 2214, 0869,3C ;10060          JSR [CHECK.RESULT]                      ; проверка LS.XOR.WR в операции LS.XOR.WR.XCHG
U 2215, BA20,F4 ;10061          JMP [LOOP.TF.6.18F]                     ; зацикливание при ошибке, если разрешено
U 2216, B6E6,15 ;10062          MOV LS[L73] TO WR[0]                    ;
U 2217, B628,95 ;10063          MOV LS[#FFFF] TO WR[1]                  ; ожидаемые данные = 0000FFFF
U 2218, 0869,3C ;10064          JSR [CHECK.RESULT]                      ; проверка XCHG в операции LS.XOR.WR.XCHG
U 2219, BA20,F4 ;10065          JMP [LOOP.TF.6.18F]                     ; зацикливание при ошибке, если разрешено
U 221A, BA70,1C ;10066          JSR [INC.OTHER]                         ;
                ;10067  LOOP.TF.6.190:
U 221B, B698,15 ;10068          MOV LS[ALTER.ODD] TO WR[0]              ;
U 221C, 3E10,15 ;10069          MOV WR[0] TO LS[T8]                     ; LS=AAAAAAAA
U 221D, 369A,15 ;10070          MOV LS[ALTER.EVEN] TO WR[0]             ; WR0=55555555
U 221E, 6410,15 ;10071          SWAP LS[T8] WITH WR[0]                  ;
U 221F, 3698,95 ;10072          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные в WR=AAAAAAAA
U 2220, 0869,3C ;10073          JSR [CHECK.RESULT]                      ; проверка операции SWAP.LS.WR
U 2221, BA21,B4 ;10074          JMP [LOOP.TF.6.190]                     ; зацикливание при ошибке, если разрешено
U 2222, B610,15 ;10075          MOV LS[T8] TO WR[0]                     ;
```

```
U 2223, B69A,95 ;10076           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные в LS=55555555
U 2224, 0B69,3C ;10077           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 2225, BA21,B4 ;10078           JMP [LOOP.TF.6.190]                    ; зацикливание при ошибке, если разрешено
U 2226, BA70,1C ;10079           JSR [INC.OTHER]                        ;
                ;10080   LOOP.TF.6.191:
U 2227, B698,15 ;10081           MOV LS[ALTER.ODD] TO WR[0]             ;
U 2228, BF60,15 ;10082           MOV WR[0] TO LS[L30]                   ; LS=AAAAAAAA
U 2229, 369A,15 ;10083           MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 222A, E460,15 ;10084           SWAP LS[L30] WITH WR[0]                ;
U 222B, 3698,95 ;10085           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные в WR=AAAAAAAA
U 222C, 0B69,3C ;10086           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 222D, BA22,74 ;10087           JMP [LOOP.TF.6.191]                    ; зацикливание при ошибке, если разрешено
U 222E, 3660,15 ;10088           MOV LS[L30] TO WR[0]                   ;
U 222F, B69A,95 ;10089           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные в LS=55555555
U 2230, 0B69,3C ;10090           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 2231, BA22,74 ;10091           JMP [LOOP.TF.6.191]                    ; зацикливание при ошибке, если разрешено
U 2232, BA70,1C ;10092           JSR [INC.OTHER]                        ;
                ;10093   LOOP.TF.6.192:
U 2233, B698,15 ;10094           MOV LS[ALTER.ODD] TO WR[0]             ;
U 2234, BEA6,15 ;10095           MOV WR[0] TO LS[L53]                   ; LS=AAAAAAAA
U 2235, 369A,15 ;10096           MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 2236, E4A6,15 ;10097           SWAP LS[L53] WITH WR[0]                ;
U 2237, 3698,95 ;10098           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные в WR=AAAAAAAA
U 2238, 0B69,3C ;10099           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 2239, BA23,34 ;10100           JMP [LOOP.TF.6.192]                    ; зацикливание при ошибке, если разрешено
U 223A, 36A6,15 ;10101           MOV LS[L53] TO WR[0]                   ;
U 223B, B69A,95 ;10102           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные в LS=55555555
U 223C, 0B69,3C ;10103           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 223D, BA23,34 ;10104           JMP [LOOP.TF.6.192]                    ; зацикливание при ошибке, если разрешено
U 223E, BA70,1C ;10105           JSR [INC.OTHER]                        ;
                ;10106   LOOP.TF.6.193:
U 223F, B698,15 ;10107           MOV LS[ALTER.ODD] TO WR[0]             ;
U 2240, 3EE6,15 ;10108           MOV WR[0] TO LS[L73]                   ; LS=AAAAAAAA
U 2241, 369A,15 ;10109           MOV LS[ALTER.EVEN] TO WR[0]            ; WR0=55555555
U 2242, 64E6,15 ;10110           SWAP LS[L73] WITH WR[0]
U 2243, 3698,95 ;10111           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные в WR=AAAAAAAA
U 2244, 0B69,3C ;10112           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 2245, BA23,F4 ;10113           JMP [LOOP.TF.6.193]                    ; зацикливание при ошибке, если разрешено
U 2246, B6E6,15 ;10114           MOV LS[L73] TO WR[0]                   ;
U 2247, B69A,95 ;10115           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные в LS=55555555
U 2248, 0B69,3C ;10116           JSR [CHECK.RESULT]                     ; проверка операции SWAP.LS.WR
U 2249, BA23,F4 ;10117           JMP [LOOP.TF.6.193]                    ; зацикливание при ошибке, если разрешено
U 224A, BA70,1C ;10118           JSR [INC.OTHER]                        ;
                ;10119   LOOP.TF.6.194:
U 224B, 369A,15 ;10120           MOV LS[ALTER.EVEN] TO WR[0]            ;
U 224C, 3E10,15 ;10121           MOV WR[0] TO LS[T8]                    ; LS=55555555
U 224D, E510,15 ;10122           CLR LS[T8]                             ;
U 224E, B610,15 ;10123           MOV LS[T8] TO WR[0]                    ;
U 224F, 2FB2,95 ;10124           CLR WR[1]                              ; ожидаемые данные= 0
U 2250, 0B69,3C ;10125           JSR [CHECK.RESULT]                     ; проверка операции CLR.LS
U 2251, BA24,B4 ;10126           JMP [LOOP.TF.6.194]                    ; зацикливание при ошибке, если разрешено
U 2252, BA70,1C ;10127           JSR [INC.OTHER]                        ;
                ;10128   LOOP.TF.6.195:
U 2253, 369A,15 ;10129           MOV LS[ALTER.EVEN] TO WR[0]            ;
U 2254, BE60,15 ;10130           MOV WR[0] TO LS[L30]                   ; LS=55555555
```

```
U 2255, 6560,15 ;10131           CLR LS[L30]                          ;
U 2256, 3660,15 ;10132           MOV LS[L30] TO WR[0]                 ;
U 2257, 2F82,95 ;10133           CLR WR[1]                            ; ожидаемые данные= 0
U 2258, 0869,3C ;10134           JSR [CHECK.RESULT]                   ; проверка операции CLR.LS
U 2259, 8A25,34 ;10135           JMP [LOOP.TF.6.195]                  ; зацикливание при ошибке, если разрешено
U 225A, 8A70,1C ;10136           JSR [INC.OTHER]                      ;
                ;10137   LOOP.TF.6.196:
U 225B, 369A,15 ;10138           MOV LS[ALTER.EVEN] TO WR[0]          ;
U 225C, 8EA6,15 ;10139           MOV WR[0] TO LS[L53]                 ; LS=55555555
U 225D, 65A6,15 ;10140           CLR LS[L53]                          ;
U 225E, 36A6,15 ;10141           MOV LS[L53] TO WR[0]                 ;
U 225F, 2F82,95 ;10142           CLR WR[1]                            ; ожидаемые данные= 0
U 2260, 0869,3C ;10143           JSR [CHECK.RESULT]                   ; проверка операции CLR.LS
U 2261, 0A25,B4 ;10144           JMP [LOOP.TF.6.196]                  ; зацикливание при ошибке, если разрешено
U 2262, 8A70,1C ;10145           JSR [INC.OTHER]                      ;
                ;10146   LOOP.TF.6.197:
U 2263, 369A,15 ;10147           MOV LS[ALTER.EVEN] TO WR[0]          ;
U 2264, 3EE6,15 ;10148           MOV WR[0] TO LS[L73]                 ; LS=55555555
U 2265, E5E6,15 ;10149           CLR LS[73]                           ;
U 2266, B6E6,15 ;10150           MOV LS[L73] TO WR[0]                 ;
U 2267, 2F82,95 ;10151           CLR WR[1]                            ; ожидаемые данные= 0
U 2268, 0869,3C ;10152           JSR [CHECK.RESULT]                   ; проверка операции CLR.LS
U 2269, 8A26,34 ;10153           JMP [LOOP.TF.6.197]                  ; зацикливание при ошибке, если разрешено
U 226A, 8A70,1C ;10154           JSR [INC.OTHER]                      ;
                ;10155   LOOP.TF.6.198:
U 226B, 5898,15 ;10156           MOV LS[ALTER.ODD] TO Q               ; Q=AAAAAAAA
U 226C, 369A,15 ;10157           MOV LS[ALTER.EVEN] TO WR[0]          ; WR0=55555555
U 226D, E610,15 ;10158           MOV Q TO WR[0] XCHG TO LS[T8]        ;
U 226E, 3698,95 ;10159           MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные в WR=AAAAAAAA
U 226F, 0869,3C ;10160           JSR [CHECK.RESULT]                   ; проверка операции MOV.Q.WR&WR.LS
U 2270, 0A26,B4 ;10161           JMP [LOOP.TF.6.198]                  ; зацикливание при ошибке, если разрешено
U 2271, B610,15 ;10162           MOV LS[T8]  TO WR[0]                 ;
U 2272, B69A,95 ;10163           MOV LS[ALTER.EVEN] TO WR[1]          ; ожидаемые данные в LS=55555555
U 2273, 0869,3C ;10164           JSR [CHECK.RESULT]                   ; проверка операции MOV.Q.WR&WR.LS
U 2274, 0A26,B4 ;10165           JMP [LOOP.TF.6.198]                  ; зацикливание при ошибке, если разрешено
U 2275, 8A70,1C ;10166           JSR [INC.OTHER]                      ;
                ;10167   LOOP.TF.6.199:
U 2276, 5898,15 ;10168           MOV LS[ALTER.ODD] TO Q               ; Q=AAAAAAAA
U 2277, 369A,15 ;10169           MOV LS[ALTER.EVEN] TO WR[0]          ; WR0=55555555
U 2278, 6660,15 ;10170           MOV Q TO WR[0] XCHG TO LS[L30]       ;
U 2279, 3698,95 ;10171           MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные в WR=AAAAAAAA
U 227A, 0869,3C ;10172           JSR [CHECK.RESULT]                   ; проверка операции MOV.Q.WR&WR.LS
U 227B, 0A27,64 ;10173           JMP [LOOP.TF.6.199]                  ; зацикливание при ошибке, если разрешено
U 227C, 3660,15 ;10174           MOV LS[L30] TO WR[0]                 ;
U 227D, B69A,95 ;10175           MOV LS[ALTER.EVEN] TO WR[1]          ; ожидаемые данные в LS=55555555
U 227E, 0869,3C ;10176           JSR [CHECK.RESULT]                   ; проверка операции MOV.Q.WR&WR.LS
U 227F, 0A27,64 ;10177           JMP [LOOP.TF.6.199]                  ; зацикливание при ошибке, если разрешено
U 2280, 8A70,1C ;10178           JSR [INC.OTHER]                      ;
                ;10179   LOOP.TF.6.19A:
U 2281, 5898,15 ;10180           MOV LS[ALTER.ODD] TO Q               ; Q=AAAAAAAA
U 2282, 369A,15 ;10181           MOV LS[ALTER.EVEN] TO WR[0]          ; WR0=55555555
U 2283, 66A6,15 ;10182           MOV Q TO WR[0] XCHG TO LS[L53]       ;
U 2284, 3698,95 ;10183           MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные в WR=AAAAAAAA
U 2285, 0869,3C ;10184           JSR [CHECK.RESULT]                   ; проверка операции MOV.Q.WR&WR.LS
U 2286, 8A28,14 ;10185           JMP [LOOP.TF.6.19A]                  ; зацикливание при ошибке, если разрешено
```

```
U 2287, 36A6,15 ;10186           MOV LS[L53] TO WR[0]                    ;
U 2288, B69A,95 ;10187           MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные в LS=55555555
U 2289, 0869,3C ;10188           JSR [CHECK.RESULT]                      ; проверка операции MOV.Q.WR&WR.LS
U 228A, 8A28,14 ;10189           JMP [LOOP.TF.6.19A]                     ; зацикливание при ошибке, если разрешено
U 228B, 8A70,1C ;10190           JSR [INC.OTHER]                         ;
                ;10191   LOOP.TF.6.19B:
U 228C, 5898,15 ;10192           MOV LS[ALTER.ODD] TO Q                  ; Q=AAAAAAAA
U 228D, 369A,15 ;10193           MOV LS[ALTER.EVEN] TO WR[0]             ; WR0 = 55555555
U 228E, E6E6,15 ;10194           MOV .. TO WR[0] XCHG TO LS[L73]         ;
U 228F, 3698,95 ;10195           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные в WR = AAAAAAAA
U 2290, 0869,3C ;10196           JSR [CHECK.RESULT]                      ; проверка операции MOV.Q.WR&WR.LS
U 2291, 0A28,C4 ;10197           JMP [LOOP.TF.6.19B]                     ; зацикливание при ошибке, если разрешено
U 2292, B6E6,15 ;10198           MOV LS[L73] TO WR[0]                    ;
U 2293, B69A,95 ;10199           MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные в LS = 55555555
U 2294, 0869,3C ;10200           JSR [CHECK.RESULT]                      ; проверка операции MOV.Q.WR&WR.LS
U 2295, 0A28,C4 ;10201           JMP [LOOP.TF.6.19B]                     ; зацикливание при ошибке, если разрешено
U 2296, 8A70,1C ;10202           JSR [INC.OTHER]                         ;
                ;10203   LOOP.TF.6.19C:
U 2297, D842,15 ;10204           MOV LS[#2] TO Q                         ; Q = 2
U 2298, B69E,15 ;10205           MOV LS[ONES] TO WR[0]                   ; WR0 = -1
U 2299, 6710,15 ;10206           ADD Q TO WR[0] XCHG TO LS[T8]           ;
U 229A, 3640,95 ;10207           MOV LS[#1] TO WR[1]                     ; ожидаемые данные в WR = 1
U 229B, 0869,3C ;10208           JSR [CHECK.RESULT]                      ; проверка WR.PLUS.Q в операции WR.PLUS.Q.XCHG
U 229C, 0A29,74 ;10209           JMP [LOOP.TF.6.19C]                     ; зацикливание при ошибке, если разрешено
U 229D, B610,15 ;10210           MOV LS[T8] TO WR[0]                     ;
U 229E, 369E,95 ;10211           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные в LS = -1
U 229F, 0869,3C ;10212           JSR [CHECK.RESULT]                      ; проверка XCHG в операции WR.PLUS.Q.XCHG
U 22A0, 0A29,74 ;10213           JMP [LOOP.TF.6.19C]                     ; зацикливание при ошибке, если разрешено
U 22A1, 8A70,1C ;10214           JSR [INC.OTHER]                         ;
                ;10215   LOOP.TF.6.19D:
U 22A2, D842,15 ;10216           MOV LS[#2] TO Q                         ; Q = 2
U 22A3, B69E,15 ;10217           MOV LS[ONES] TO WR[0]                   ; WR0 = -1
U 22A4, E760,15 ;10218           ADD Q TO WR[0] XCHG TO LS[L30]          ;
U 22A5, 3640,95 ;10219           MOV LS[#1] TO WR[1]                     ; ожидаемые данные в WR = 1
U 22A6, 0869,3C ;10220           JSR [CHECK.RESULT]                      ; проверка WR.PLUS.Q в операции WR.PLUS.Q.XCHG
U 22A7, 0A2A,24 ;10221           JMP [LOOP.TF.6.19D]                     ; зацикливание при ошибке, если разрешено
U 22A8, 3660,15 ;10222           MOV LS[L30] TO WR[0]                    ;
U 22A9, 369E,95 ;10223           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные в LS = -1
U 22AA, 0869,3C ;10224           JSR [CHECK.RESULT]                      ; проверка XCHG в операции WR.PLUS.Q.XCHG
U 22AB, 0A2A,24 ;10225           JMP [LOOP.TF.6.19D]                     ; зацикливание при ошибке, если разрешено
U 22AC, 8A70,1C ;10226           JSR [INC.OTHER]                         ;
                ;10227   LOOP.TF.6.19E:
U 22AD, D842,15 ;10228           MOV LS[#2] TO Q                         ; Q = 2
U 22AE, B69E,15 ;10229           MOV LS[ONES] TO WR[0]                   ; WR0 = -1
U 22AF, E7A6,15 ;10230           ADD Q TO WR[0] XCHG TO LS[L53]          ;
U 22B0, 3640,95 ;10231           MOV LS[#1] TO WR[1]                     ; ожидаемые данные в WR = 1
U 22B1, 0869,3C ;10232           JSR [CHECK.RESULT]                      ; проверка WR.PLUS.Q в операции WR.PLUS.Q.XCHG
U 22B2, 0A2A,D4 ;10233           JMP [LOOP.TF.6.19E]                     ; зацикливание при ошибке, если разрешено
U 22B3, 36A6,15 ;10234           MOV LS[L53] TO WR[0]                    ;
U 22B4, 369E,95 ;10235           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные в LS = -1
U 22B5, 0869,3C ;10236           JSR [CHECK.RESULT]                      ; проверка XCHG в операции WR.PLUS.Q.XCHG
U 22B6, 0A2A,D4 ;10237           JMP [LOOP.TF.6.19E]                     ; зацикливание при ошибке, если разрешено
U 22B7, EA70,1C ;10238           JSR [INC.OTHER]                         ;
                ;10239   LOOP.TF.6.19F:
U 22B8, D842,15 ;10240           MOV LS[#2] TO Q                         ; Q = 2
```

```
U 22B9, B69E,15 ;10241          MOV LS[ONES] TO WR[0]                   ; WR0 = -1
U 22BA, 67E6,15 ;10242          ADD Q TO WR[0] XCHG TO LS[L73]          ;
U 22BB, 3640,95 ;10243          MOV LS[#1] TO WR[1]                     ; Ожидаемые данные в WR = 1
U 22BC, 0869,3C ;10244          JSR [CHECK.RESULT]                      ; проверка WR.PLUS.Q в операции WR.PLUS.Q.XCHG
U 22BD, 8A2B,84 ;10245          JMP [LOOP.TF.6.19F]                     ; зацикливание при ошибке, если разрешено
U 22BE, B6E6,15 ;10246          MOV LS[L73] TO WR[0]                    ;
U 22BF, 369E,95 ;10247          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные в LS = -1
U 22C0, 0869,3C ;10248          JSR [CHECK.RESULT]                      ; проверка XCHG в операции WR.PLUS.Q.XCHG
U 22C1, 8A2B,84 ;10249          JMP [LOOP.TF.6.19F]                     ; зацикливание при ошибке, если разрешено
U 22C2, 8A70,1C ;10250          JSR [INC.OTHER]                         ;
                ;10251  LOOP.TF.6.1A0:
U 22C3, B640,15 ;10252          MOV LS[#1] TO WR[0]                     ; WR0 = 1
U 22C4, B643,15 ;10253          MOV LS[#2] TO WR[2]                     ;
U 22C5, BE11,15 ;10254          MOV WR[2] TO LS[T8]                     ; LS = 2
U 22C6, 6810,15 ;10255          SUB (WR[0] FROM LS[T8])-1               ;
U 22C7, B610,15 ;10256          MOV LS[T8] TO WR[0]                     ;
U 22C8, 2F82,95 ;10257          CLR WR[1]                               ; Ожидаемые данные = 0
U 22C9, 0869,3C ;10258          JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.LS-1
U 22CA, 8A2C,34 ;10259          JMP [LOOP.TF.6.1A0]                     ; зацикливание при ошибке, если разрешено
U 22CB, 8A70,1C ;10260          JSR [INC.OTHER]                         ;
                ;10261  LOOP.TF.6.1A1:
U 22CC, B640,15 ;10262          MOV LS[#1] TO WR[0]                     ; WR0 = 1
U 22CD, B643,15 ;10263          MOV LS[#2] TO WR[2]                     ;
U 22CE, 3E61,15 ;10264          MOV WR[2] TO LS[L30]                    ; LS = 2
U 22CF, E860,15 ;10265          SUB (WR[0] FROM LS[L30])-1              ;
U 22D0, 3660,15 ;10266          MOV LS[L30] TO WR[0]                    ;
U 22D1, 2F82,95 ;10267          CLR WR[1]                               ; Ожидаемые данные = 0
U 22D2, 0869,3C ;10268          JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.LS-1
U 22D3, 8A2C,C4 ;10269          JMP [LOOP.TF.6.1A1]                     ; зацикливание при ошибке, если разрешено
U 22D4, 8A70,1C ;10270          JSR [INC.OTHER]                         ;
                ;10271  LOOP.TF.6.1A2:
U 22D5, B640,15 ;10272          MOV LS[#1] TO WR[0]                     ; WR0 = 1
U 22D6, B643,15 ;10273          MOV LS[#2] TO WR[2]                     ;
U 22D7, 3EA7,15 ;10274          MOV WR[2] TO LS[L53]                    ; LS = 2
U 22D8, E8A6,15 ;10275          SUB (WR[0] FROM LS[L53])-1              ;
U 22D9, 36A6,15 ;10276          MOV LS[L53] TO WR[0]                    ;
U 22DA, 2F82,95 ;10277          CLR WR[1]                               ; ожидаемые данные = 0
U 22DB, 0869,3C ;10278          JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.LS-1
U 22DC, 0A2D,54 ;10279          JMP [LOOP.TF.6.1A2]                     ; зацикливание при ошибке, если разрешено
U 22DD, 8A70,1C ;10280          JSR [INC.OTHER]                         ;
                ;10281  LOOP.TF.6.1A3:
U 22DE, B640,15 ;10282          MOV LS[#1] TO WR[0]                     ; WR0 = 1
U 22DF, B643,15 ;10283          MOV LS[#2] TO WR[2]                     ;
U 22E0, BEE7,15 ;10284          MOV WR[2] TO LS[L73]                    ; LS = 2
U 22E1, 68E6,15 ;10285          SUB (WR[0] FROM LS[L73])-1              ;
U 22E2, B6E6,15 ;10286          MOV LS[L73] TO WR[0]                    ;
U 22E3, 2F82,95 ;10287          CLR WR[1]                               ; Ожидаемые данные = 0
U 22E4, 0869,3C ;10288          JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.LS-1
U 22E5, 8A2D,E4 ;10289          JMP [LOOP.TF.6.1A3]                     ; зацикливание при ошибке, если разрешено
U 22E6, 8A70,1C ;10290          JSR [INC.OTHER]                         ;
                ;10291  LOOP.TF.6.1A4:
U 22E7, B643,15 ;10292          MOV LS[#2] TO WR[2]                     ;
U 22E8, BE11,15 ;10293          MOV WR[2] TO LS[T8]                     ; LS = 2
U 22E9, 3641,15 ;10294          MOV LS[#1] TO WR[2]                     ; WR2 = 1
U 22EA, 6911,15 ;10295          SUB LS[T8] FROM WR[2] TO LS             ;
```

```
U 22EB, B610,15 ;10296          MOV LS[TB] TO WR[0]                    ;
U 22EC, 369E,95 ;10297          MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 22ED, 0B69,3C ;10298          JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.LS
U 22EE, BA2E,74 ;10299          JMP [LOOP.TF.6.1A4]                    ; зацикливание при ошибке, если разрешено
U 22EF, BA70,1C ;10300          JSR [INC.OTHER]                        ;
                ;10301  LOOP.TF.6.1A5:
U 22F0, B643,15 ;10302          MOV LS[#2] TO WR[2]                    ;
U 22F1, 3E61,15 ;10303          MOV WR[2] TO LS[L30]                   ; LS = 2
U 22F2, 3641,15 ;10304          MOV LS[#1] TO WR[2]                    ; WR2 = 1
U 22F3, E961,15 ;10305          SUB LS[L30] FROM WR[2] TO LS           ;
U 22F4, 3660,15 ;10306          MOV LS[L30] TO WR[0]                   ;
U 22F5, 369E,95 ;10307          MOV LS[ONES] TO WR[1]                  ; Ожидаемые данные = -1
U 22F6, 0B69,3C ;10308          JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.LS
U 22F7, BA2F,04 ;10309          JMP [LOOP.TF.6.1A5]                    ; зацикливание при ошибке, если разрешено
U 22F8, BA70,1C ;10310          JSR [INC.OTHER]                        ;
                ;10311  LOOP.TF.6.1A6:
U 22F9, B643,15 ;10312          MOV LS[#2] TO WR[2]                    ;
U 22FA, 3EA7,15 ;10313          MOV WR[2] TO LS[L53]                   ; LS = 2
U 22FB, 3641,15 ;10314          MOV LS[#1] TO WR[2]                    ; WR2 = 1
U 22FC, E9A7,15 ;10315          SUB LS[L53] FROM WR[2] TO LS           ;
U 22FD, 36A6,15 ;10316          MOV LS[L53] TO WR[0]                   ;
U 22FE, 369E,95 ;10317          MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = -1
U 22FF, 0B69,3C ;10318          JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.LS
U 2300, BA2F,94 ;10319          JMP [LOOP.TF.6.1A6]                    ; зацикливание при ошибке, если разрешено
U 2301, BA70,1C ;10320          JSR [INC.OTHER]                        ;
                ;10321  LOOP.TF.6.1A7:
U 2302, B643,15 ;10322          MOV LS[#2] TO WR[2]                    ;
U 2303, BEE7,15 ;10323          MOV WR[2] TO LS[L73]                   ; LS = 2
U 2304, 3641,15 ;10324          MOV LS[#1] TO WR[2]                    ; WR2 = 1
U 2305, 69E7,15 ;10325          SUB LS[L73] FROM WR[2] TO LS           ;
U 2306, B6E6,15 ;10326          MOV LS[L73] TO WR[0]                   ;
U 2307, 369E,95 ;10327          MOV LS[ONES] TO WR[1]                  ; Ожидаемые данные = -1
U 2308, 0B69,3C ;10328          JSR [CHECK.RESULT]                     ; проверка операции LS.FROM.WR.LS
U 2309, BA30,24 ;10329          JMP [LOOP.TF.6.1A7]                    ; зацикливание при ошибке, если разрешено
U 230A, BA70,1C ;10330          JSR [INC.OTHER]                        ;
                ;10331  LOOP.TF.6.1A8:
U 230B, 369F,15 ;10332          MOV LS[ONES] TO WR[2]                  ; WR2 = -1
U 230C, B640,15 ;10333          MOV LS[#1] TO WR[0]                    ;
U 230D, 3E10,15 ;10334          MOV WR[0] TO LS[TB]                    ; LS = 1
U 230E, 6A11,15 ;10335          ADD (WR[2] TO LS[TB])+1                ;
U 230F, B610,15 ;10336          MOV LS[TB] TO WR[0]                    ;
U 2310, 3640,95 ;10337          MOV LS[#1] TO WR[1]                    ; ожидаемые данные = 1
U 2311, 0B69,3C ;10338          JSR [CHECK.RESULT]                     ; проверка операции WR.PLUS.LS+1
U 2312, BA30,B4 ;10339          JMP [LOOP.TF.6.1A8]                    ; зацикливание при ошибке, если разрешено
U 2313, BA70,1C ;10340          JSR [INC.OTHER]                        ;
                ;10341  LOOP.TF.6.1A9:
U 2314, 369F,15 ;10342          MOV LS[ONES] TO WR[2]                  ; WR2 = -1
U 2315, B640,15 ;10343          MOV LS[#1] TO WR[0]                    ;
U 2316, BE60,15 ;10344          MOV WR[0] TO LS[L30]                   ; LS = 1
U 2317, EA61,15 ;10345          ADD (WR[2] TO LS[L30])+1               ;
U 231B, 3660,15 ;10346          MOV LS[L30] TO WR[0]                   ;
U 2319, 3640,95 ;10347          MOV LS[#1] TO WR[1]                    ; ожидаемые данные = 1
U 231A, 0B69,3C ;10348          JSR [CHECK.RESULT]                     ; проверка операции WR.PLUS.LS+1
U 231B, BA31,44 ;10349          JMP [LOOP.TF.6.1A9]                    ; зацикливание при ошибке, если разрешено
U 231C, BA70,1C ;10350          JSR [INC.OTHER]                        ;
```

```
                    ;10351    LOOP.TF.6.1AA:
U 231D,  369F,15    ;10352              MOV LS[ONES] TO WR[2]                ; WR2 = -1
U 231E,  B640,15    ;10353              MOV LS[#1] TO WR[0]                  ;
U 231F,  BEA6,15    ;10354              MOV WR[0] TO LS[L53]                 ; LS = 1
U 2320,  EAA7,15    ;10355              ADD (WR[2] TO LS[L53])+1             ;
U 2321,  36A6,15    ;10356              MOV LS[L53] TO WR[0]                 ;
U 2322,  3640,95    ;10357              MOV LS[#1] TO WR[1]                  ; ожидаемые данные = 1
U 2323,  0869,3C    ;10358              JSR [CHECK.RESULT]                   ; проверка операции WR.PLUS.LS+1
U 2324,  0A31,D4    ;10359              JMP [LOOP.TF.6.1AA]                  ; зацикливание при ошибке, если разрешено
U 2325,  8A70,1C    ;10360              JSR [INC.OTHER]                      ;
                    ;10361    LOOP.TF.6.1AB:
U 2326,  369F,15    ;10362              MOV LS[ONES] TO WR[2]                ; WR2 = -1
U 2327,  B640,15    ;10363              MOV LS[#1] TO WR[0]                  ;
U 2328,  3EE6,15    ;10364              MOV WR[0] TO LS[L73]                 ; LS = 1
U 2329,  6AE7,15    ;10365              ADD (WR[2] TO LS[L73])+1             ;
U 232A,  B6E6,15    ;10366              MOV LS[L73] TO WR[0]                 ;
U 232B,  3640,95    ;10367              MOV LS[#1] TO WR[1]                  ; Ожидаемые данные = 1
U 232C,  0869,3C    ;10368              JSR [CHECK.RESULT]                   ; проверка операции WR.PLUS.LS+1
U 232D,  8A32,64    ;10369              JMP [LOOP.TF.6.1AB]                  ; зацикливание при ошибке, если разрешено
U 232E,  8A70,1C    ;10370              JSR [INC.OTHER]                      ;
                    ;10371    LOOP.TF.6.1AC:
U 232F,  B643,15    ;10372              MOV LS[#2] TO WR[2]                  ; WR2  = 2
U 2330,  B640,15    ;10373              MOV LS[#1] TO WR[0]                  ;
U 2331,  3E10,15    ;10374              MOV WR[0] TO LS[T8]                  ; LS = 1
U 2332,  EB11,15    ;10375              SUB WR[2] FROM LS[T8]                ;
U 2333,  B610,15    ;10376              MOV LS[T8] TO WR[0]                  ;
U 2334,  369E,95    ;10377              MOV LS[ONES] TO WR[1]                ; ожидаемые данные = -1
U 2335,  0869,3C    ;10378              JSR [CHECK.RESULT]                   ; проверка операции WR.FROM.LS
U 2336,  8A32,F4    ;10379              JMP [LOOP.TF.6.1AC]                  ; зацикливание при ошибке, если разрешено
U 2337,  8A70,1C    ;10380              JSR [INC.OTHER]                      ;
                    ;10381    LOOP.TF.6.1AD:
U 2338,  B643,15    ;10382              MOV LS[#2] TO WR[2]                  ; WR2 = 2
U 2339,  B640,15    ;10383              MOV LS[#1] TO WR[0]                  ;
U 233A,  BE60,15    ;10384              MOV WR[0] TO LS[L30]                 ; LS = 1
U 233B,  6B61,15    ;10385              SUB WR[2] FROM LS[L30]               ;
U 233C,  3660,15    ;10386              MOV LS[L30] TO WR[0]                 ;
U 233D,  369E,95    ;10387              MOV LS[ONES] TO WR[1]                ; Ожидаемые данные = -1
U 233E,  0869,3C    ;10388              JSR [CHECK.RESULT]                   ; проверка операции WR.FROM.LS
U 233F,  8A33,84    ;10389              JMP [LOOP.TF.6.1AD]                  ; зацикливание при ошибке, если разрешено
U 2340,  8A70,1C    ;10390              JSR [INC.OTHER]                      ;
                    ;10391    LOOP.TF.6.1AE:
U 2341,  B643,15    ;10392              MOV LS[#2] TO WR[2]                  ; WR2 = 2
U 2342,  B640,15    ;10393              MOV LS[#1] TO WR[0]                  ;
U 2343,  BEA6,15    ;10394              MOV WR[0] TO LS[L53]                 ; LS = 1
U 2344,  6BA7,15    ;10395              SUB WR[2] FROM LS[L53]               ;
U 2345,  36A6,15    ;10396              MOV LS[L53] TO WR[0]                 ;
U 2346,  369E,95    ;10397              MOV LS[ONES] TO WR[1]                ; ожидаемые данные = -1
U 2347,  0869,3C    ;10398              JSR [CHECK.RESULT]                   ; проверка операции WR.FROM.LS
U 2348,  0A34,14    ;10399              JMP [LOOP.TF.6.1AE]                  ; зацикливание при ошибке, если разрешено
U 2349,  8A70,1C    ;10400              JSR [INC.OTHER]                      ;
                    ;10401    LOOP.TF.6.1AF:
U 234A,  B643,15    ;10402              MOV LS[#2] TO WR[2]                  ; WR2 = 2
U 234B,  B640,15    ;10403              MOV LS[#1] TO WR[0]                  ;
U 234C,  3EE6,15    ;10404              MOV WR[0] TO LS[L73]                 ; LS = 1
U 234D,  EBE7,15    ;10405              SUB WR[2] FROM LS[L73]               ;
```

```
U 234E,  B6E6,15 ;10406           MOV LS[L73] TO WR[0]                    ;
U 234F,  369E,95 ;10407           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 2350,  0869,3C ;10408           JSR [CHECK.RESULT]                      ; проверка операции WR.FROM.LS
U 2351,  BA34,A4 ;10409           JMP [LOOP.TF.6.1AF]                     ; зацикливание при ошибке, если разрешено
U 2352,  8A70,1C ;10410           JSR [INC.OTHER]                         ;
                 ;10411   LOOP.TF.6.1B0:
U 2353,  8643,15 ;10412           MOV LS[#2] TO WR[2]                     ; WR2 = 2
U 2354,  B69E,15 ;10413           MOV LS[ONES] TO WR[0]                   ;
U 2355,  3E10,15 ;10414           MOV WR[0] TO LS[T8]                     ; LS = -1
U 2356,  6C11,15 ;10415           ADD WR[2] TO LS[T8]                     ;
U 2357,  B610,15 ;10416           MOV LS[T8] TO WR[0]                     ;
U 2358,  3640,95 ;10417           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 2359,  0869,3C ;10418           JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.LS
U 235A,  0A35,34 ;10419           JMP [LOOP.TF.6.1B0]                     ; зацикливание при ошибке, если разрешено
U 235B,  8A70,1C ;10420           JSR [INC.OTHER]                         ;
                 ;10421   LOOP.TF.6.1B1:
U 235C,  8643,15 ;10422           MOV LS[#2] TO WR[2]                     ; WR2 = 2
U 235D,  B69E,15 ;10423           MOV LS[ONES] TO WR[0]                   ;
U 235E,  BE60,15 ;10424           MOV WR[0] TO LS[L30]                    ; LS = -1
U 235F,  EC61,15 ;10425           ADD WR[2] TO LS[L30]                    ;
U 2360,  3660,15 ;10426           MOV LS[L30] TO WR[0]                    ;
U 2361,  3640,95 ;10427           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 2362,  0869,3C ;10428           JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.LS
U 2363,  0A35,C4 ;10429           JMP [LOOP.TF.6.1B1]                     ; зацикливание при ошибке, если разрешено
U 2364,  8A70,1C ;10430           JSR [INC.OTHER]                         ;
                 ;10431   LOOP.TF.6.1B2:
U 2365,  8643,15 ;10432           MOV LS[#2] TO WR[2]                     ; WR2 = 2
U 2366,  B69E,15 ;10433           MOV LS[ONES] TO WR[0]                   ;
U 2367,  BEA6,15 ;10434           MOV WR[0] TO LS[L53]                    ; LS = -1
U 2368,  ECA7,15 ;10435           ADD WR[2] TO LS[L53]                    ;
U 2369,  36A6,15 ;10436           MOV LS[L53] TO WR[0]                    ;
U 236A,  3640,95 ;10437           MOV LS[#1] TO WR[1]                     ; Ожидаемые данные = 1
U 236B,  0869,3C ;10438           JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.LS
U 236C,  0A36,54 ;10439           JMP [LOOP.TF.6.1B2]                     ; зацикливание при ошибке, если разрешено
U 236D,  8A70,1C ;10440           JSR [INC.OTHER]                         ;
                 ;10441   LOOP.TF.6.1B3:
U 236E,  8643,15 ;10442           MOV LS[#2] TO WR[2]                     ; WR2 = 2
U 236F,  B69E,15 ;10443           MOV LS[ONES] TO WR[0]                   ;
U 2370,  3EE6,15 ;10444           MOV WR[0] TO LS[L73]                    ; LS = -1
U 2371,  6CE7,15 ;10445           ADD WR[2] TO LS[L73]                    ;
U 2372,  B6E6,15 ;10446           MOV LS[L73] TO WR[0]                    ;
U 2373,  3640,95 ;10447           MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 2374,  0869,3C ;10448           JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.LS
U 2375,  8A36,E4 ;10449           JMP [LOOP.TF.6.1B3]                     ; зацикливание при ошибке, если разрешено
U 2376,  8A70,1C ;10450           JSR [INC.OTHER]                         ;
                 ;10451   LOOP.TF.6.1B4:
U 2377,  3699,15 ;10452           MOV LS[ALTER.ODD] TO WR[2]              ;
U 2378,  2045,15 ;10453           INC WR[2]                               ; WR2 = AAAAAAAB
U 2379,  369A,15 ;10454           MOV LS[ALTER.EVEN] TO WR[0]             ;
U 237A,  3E10,15 ;10455           MOV WR[0] TO LS[T8]                     ; LS = 55555555
U 237B,  ED11,15 ;10456           BIS WR[2] TO LS[T8]                     ;
U 237C,  B610,15 ;10457           MOV LS[T8] TO WR[0]                     ;
U 237D,  369E,95 ;10458           MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 237E,  0869,3C ;10459           JSR [CHECK.RESULT]                      ; проверка операции WR.OR.LS
U 237F,  0A37,74 ;10460           JMP [LOOP.TF.6.1B4]                     ; зацикливание при ошибке, если разрешено
```

```
U 2380, 8A70,1C ;10461          JSR [INC.OTHER]                         ;
                ;10462  LOOP.TF.6.1B5:
U 2381, 3699,15 ;10463          MOV LS[ALTER.ODD] TO WR[2]              ;
U 2382, 2045,15 ;10464          INC WR[2]                               ; WR2 = AAAAAAAB
U 2383, 369A,15 ;10465          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 2384, 8E60,15 ;10466          MOV WR[0] TO LS[L30]                    ; LS = 55555555
U 2385, 6D61,15 ;10467          BIS WR[2] TO LS[L30]                    ;
U 2386, 3660,15 ;10468          MOV LS[L30] TO WR[0]                    ;
U 2387, 369E,95 ;10469          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 2388, 0869,3C ;10470          JSR [CHECK.RESULT]                      ; проверка операции WR.OR.LS
U 2389, 0A38,14 ;10471          JMP [LOOP.TF.6.1B5]                     ; зацикливание при ошибке, если разрешено
U 238A, 8A70,1C ;10472          JSR [INC.OTHER]                         ;
                ;10473  LOOP.TF.6.1B6:
U 238B, 3699,15 ;10474          MOV LS[ALTER.ODD] TO WR[2]
U 238C, 2045,15 ;10475          INC WR[2]                               ; WR2 = AAAAAAAB
U 238D, 369A,15 ;10476          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 238E, BEA6,15 ;10477          MOV WR[0] TO LS[L53]                    ; LS = 55555555
U 238F, 6DA7,15  10478          BIS WR[2] TO LS[L53]                    ;
U 2390, 36A6,15 ;10479          MOV LS[L53] TO WR[0]                    ;
U 2391, 369E,95 ;10480          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 2392, 0869,3C ;10481          JSR [CHECK.RESULT]                      ; проверка операции WR.OR.LS
U 2393, 0A38,B4 ;10482          JMP [LOOP.TF.6.1B6]                     ; зацикливание при ошибке, если разрешено
U 2394, 8A70,1C ;10483          JSR [INC.OTHER]                         ;
                ;10484  LOOP.TF.6.1B7:
U 2395, 3699,15 ;10485          MOV LS[ALTER.ODD] TO WR[2]
U 2396, 2045,15 ;10486          INC WR[2]                               ; WR2 = AAAAAAAB
U 2397, 369A,15 ;10487          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 2398, 3EE6,15 ;10488          MOV WR[0] TO LS[L73]                    ; LS = 55555555
U 2399, EDE7,15 ;10489          BIS WR[2] TO LS[L73]                    ;
U 239A, B6E6,15 ;10490          MOV LS[L73] TO WR[0]                    ;
U 239B, 369E,95 ;10491          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 239C, 0869,3C ;10492          JSR [CHECK.RESULT]                      ; проверка операци WR.OR.LS
U 239D, 0A39,54 ;10493          JMP [LOOP.TF.6.1B7]                     ; зацикливание при ошибке, если разрешено
U 239E, 8A70,1C ;10494          JSR [INC.OTHER]                         ;
                ;10495  LOOP.TF.6.1B8:
U 239F, 3699,15 ;10496          MOV LS[ALTER.ODD] TO WR[2]              ; WR2 = AAAAAAAA
U 23A0, 3628,15 ;10497          MOV LS[#FFFF] TO WR[0]                  ;
U 23A1, 3E10,15 ;10498          MOV WR[0] TO LS[T8]                     ; LS = 0000FFFF
U 23A2, EE11,15 ;10499          AND WR[2] TO LS[T8]                     ;
U 23A3, B610,15 ;10500          MOV LS[T8] TO WR[0]                     ;
U 23A4, 3698,95 ;10501          MOV LS[ALTER.ODD] TO WR[1]              ;
U 23A5, 4518,95 ;10502          BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 23A6, 0869,3C ;10503          JSR [CHECK.RESULT]                      ; проверка WR.AND.LS
U 23A7, 0A39,F4 ;10504          JMP [LOOP.TF.6.1B8]                     ; зацикливание при ошибке, если разрешено
U 23A8, 8A70,1C ;10505          JSR [INC.OTHER]                         ;
                ;10506  LOOP.TF.6.1B9:
U 23A9, 3699,15 ;10507          MOV LS[ALTER.ODD] TO WR[2]              ; WR2 = AAAAAAAA
U 23AA, 3628,15 ;10508          MOV LS[#FFFF] TO WR[0]                  ;
U 23AB, BE60,15 ;10509          MOV WR[0] TO LS[L30]                    ; LS = 0000FFFF
U 23AC, 6E61,15 ;10510          AND WR[2] TO LS[L30]                    ;
U 23AD, 3660,15 ;10511          MOV LS[L30] TO WR[0]                    ;
U 23AE, 3698,95 ;10512          MOV LS[ALTER.ODD] TO WR[1]              ;
U 23AF, 4518,95 ;10513          BIC LS[T12] TO WR[1]                    ; ожидаемые данные = 0000AAAA
U 23B0, 0869,3C ;10514          JSR [CHECK.RESULT]                      ; проверка операции WR.AND.LS
U 23B1, 0A3A,94 ;10515          JMP [LOOP.TF.6.1B9]                     ; зацикливание при ошибке, если разрешено
```

```
U 23B2, 8A7C,1C ;10516          JSR [INC.OTHER]
                ;10517  LOOP.TF.6.1BA:
U 23B3, 3699,15 ;10518          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
U 23B4, 3628,15 ;10519          MOV LS[#FFFF] TO WR[0]              ;
U 23B5, BEA6,15 ;10520          MOV WR[0] TO LS[L53]                ; LS = 0000FFFF
U 23B6, 6EA7,15 ;10521          AND WR[2] TO LS[L53]                ;
U 23B7, 36A6,15 ;10522          MOV LS[L53] TO WR[0]                ;
U 23B8, 3698,95 ;10523          MOV LS[ALTER.ODD] TO WR[1]          ;
U 23B9, 4518,95 ;10524          BIC LS[T12] TO WR[1]                ; ожидаемые данные = 0000AAAA
U 23BA, 0869,3C ;10525          JSR [CHECK.RESULT]                  ; проверка операции WR.AND.LS
U 23BB, 8A3B,34 ;10526          JMP [LOOP.TF.6.1BA]                 ; зацикливание при ошибке, если разрешено
U 23BC, 8A70,1C ;10527          JSR [INC.OTHER]                     ;
                ;10528  LOOP.TF.6.1BB:
U 23BD, 3699,15 ;10529          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
U 23BE, 3628,15 ;10530          MOV LS[#FFFF] TO WR[0]              ;
U 23BF, 3EE6,15 ;10531          MOV WR[0] TO LS[L73]                ; LS = 0000FFFF
U 23C0, EEE7,15 ;10532          AND WR[2] TO LS[L73]                ;
U 23C1, B6E6,15 ;10533          MOV LS[L73] TO WR[0]                ;
U 23C2, 3698,95 ;10534          MOV LS[ALTER.ODD] TO WR[1]          ;
U 23C3, 4518,95 ;10535          BIC LS[T12] TO WR[1]                ; ожидаемые данные = 0000AAAA
U 23C4, 0869,3C ;10536          JSR [CHECK.RESULT]                  ; проверка операции WR.AND.LS
U 23C5, 0A3B,D4 ;10537          JMP [LOOP.TF.6.1BB]                 ; зацикливание при ошибке, если разрешено
U 23C6, 8A70,1C ;10538          JSR [INC.OTHER]                     ;
                ;10539  LOOP.TF.6.1BC:
U 23C7, 3699,15 ;10540          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
U 23C8, 3628,15 ;10541          MOV LS[#FFFF] TO WR[0]              ;
U 23C9, 3E10,15 ;10542          MOV WR[0] TO LS[T8]                 ; LS = 0000FFFF
U 23CA, 6F11,15 ;10543          XOR WR[2] TO LS[T8]                 ;
U 23CB, B610,15 ;10544          MOV LS[T8] TO WR[0]                 ;
U 23CC, 3722,95 ;10545          MOV LS[ALTER.MIX] TO WR[1]          ; ожидаемые данные = AAAA5555
U 23CD, 0869,3C ;10546          JSR [CHECK.RESULT]                  ; проверка операции WR.XOR.LS
U 23CE, 8A3C,74 ;10547          JMP [LOOP.TF.6.1BC]                 ; зацикливание при ошибке, если разрешено
U 23CF, 8A70,1C ;10548          JSR [INC.OTHER]                     ;
                ;10549  LOOP.TF.6.1BD:
U 23D0, 3699,15 ;10550          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
U 23D1, 3628,15 ;10551          MOV LS[#FFFF] TO WR[0]              ;
U 23D2, BE60,15 ;10552          MOV WR[0] TO LS[L30]                ; LS = 0000FFFF
U 23D3, EF61,15 ;10553          XOR WR[2] TO LS[L30]                ;
U 23D4, 3660,15 ;10554          MOV LS[L30] TO WR[0]                ;
U 23D5, 3722,95 ;10555          MOV LS[ALTER.MIX] TO WR[1]          ; ожидаемые данные = AAAA5555
U 23D6, 0869,3C ;10556          JSR [CHECK.RESULT]                  ; проверка операции WR.XOR.LS
U 23D7, 8A3D,04 ;10557          JMP [LOOP.TF.6.1BD]                 ; зацикливание при ошибке, если разрешено
U 23D8, 8A70,1C ;10558          JSR [INC.OTHER]                     ;
                ;10559  LOOP.TF.6.1BE:
U 23D9, 3699,15 ;10560          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
U 23DA, 3628,15 ;10561          MOV LS[#FFFF] TO WR[0]              ;
U 23DB, BEA6,15 ;10562          MOV WR[0] TO LS[L53]                ; LS = 0000FFFF
U 23DC, EFA7,15 ;10563          XOR WR[2] TO LS[L53]                ;
U 23DD, 36A6,15 ;10564          MOV LS[L53] TO WR[0]                ;
U 23DE, 3722,95 ;10565          MOV LS[ALTER.MIX] TO WR[1]          ; ожидаемые данные = AAAA5555
U 23DF, 0869,3C ;10566          JSR [CHECK.RESULT]                  ; проверка операции WR.XOR.LS
U 23E0, 8A3D,94 ;10567          JMP [LOOP.TF.6.1BE]                 ; зацикливание при ошибке, если разрешено
U 23E1, 8A70,1C ;10568          JSR [INC.OTHER]                     ;
                ;10569  LOOP.TF.6.1BF:
U 23E2, 3699,15 ;10570          MOV LS[ALTER.ODD] TO WR[2]          ; WR2 = AAAAAAAA
```

```
U 23E3, 3628,15 ;10571          MOV LS[#FFFF] TO WR[0]                  ;
U 23E4, 3EE6,15 ;10572          MOV WR[0] TO LS[L73]                    ; LS = 0000FFFF
U 23E5, 6FE7,15 ;10573          XOR WR[2] TO LS[L73]                    ;
U 23E6, B6E6,15 ;10574          MOV LS[L73] TO WR[0]                    ;
U 23E7, 3722,95 ;10575          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 23E8, 0869,3C ;10576          JSR [CHECK.RESULT]                      ; проверка операции WR.XOR.LS
U 23E9, 0A3E,24 ;10577          JMP [LOOP.TF.6.18F]                     ; зацикливание при ошибке, если разрешено
U 23EA, 8A70,1C ;10578          JSR [INC.OTHER]                         ;
                ;10579  LOOP.TF.6.1C0:
U 23EB, D842,15 ;10580          MOV LS[#2] TO Q                         ; Q = 2
U 23EC, B69E,15 ;10581          MOV LS[ONES] TO WR[0]                   ;
U 23ED, 3E10,15 ;10582          MOV WR[0] TO LS[T8]                     ; LS = -1
U 23EE, 7010,15 ;10583          ADD Q TO LS[T8]                         ;
U 23EF, B610,15 ;10584          MOV LS[T8] TO WR[0]                     ;
U 23F0, 3640,95 ;10585          MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 23F1, 0869,3C ;10586          JSR [CHECK.RESULT]                      ; проверка операции Q.PLUS.LS
U 23F2, 0A3E,B4 ;10587          JMP [LOOP.TF.6.1C0]                     ; зацикливание при ошибке, если разрешено
U 23F3, 8A70,1C ;10588          JSR [INC.OTHER]                         ;
                ;10589  LOOP.TF.6.1C1:
U 23F4, D842,15 ;10590          MOV LS[#2] TO Q                         ; Q = 2
U 23F5, B69E,15 ;10591          MOV LS[ONES] TO WR[0]                   ;
U 23F6, BE60,15 ;10592          MOV WR[0] TO LS[L30]                    ; LS = -1
U 23F7, F060,15 ;10593          ADD Q TO LS[L30]                        ;
U 23F8, 3660,15 ;10594          MOV LS[L30] TO WR[0]                    ;
U 23F9, 3640,95 ;10595          MOV LS[#1] TO WR[1]                     ; ожидаемые данные = 1
U 23FA, 0869,3C ;10596          JSR [CHECK.RESULT]                      ; проверка операции Q.PLUS.LS
U 23FB, BA3F,44 ;10597          JMP [LOOP.TF.6.1C1]                     ; зацикливание при ошибке, если разрешено
U 23FC, 8A70,1C ;10598          JSR [INC.OTHER]                         ;
                ;10599  LOOP.TF.6.1C2:
U 23FD, D842,15 ;10600          MOV LS[#2] TO Q                         ; Q = 2
U 23FE, B69E,15 ;10601          MOV LS[ONES] TO WR[0]                   ;
U 23FF, BEA6,15 ;10602          MOV WR[0] TO LS[L53]                    ; LS = -1
U 2400, F0A6,15 ;10603          ADD Q TO LS[L53]                        ;
U 2401, 36A6,15 ;10604          MOV LS[L53] TO WR[0]                    ;
U 2402, 3640,95 ;10605          MOV LS[#1] TO WR[1]                     ; Ожидаемые данные = 1
U 2403, 0869,3C ;10606          JSR [CHECK.RESULT]                      ; проверка операции Q.PLUS.LS
U 2404, 8A3F,D4 ;10607          JMP [LOOP.TF.6.1C2]                     ; зацикливание при ошибке, если разрешено
U 2405, 8A70,1C ;10608          JSR [INC.OTHER]                         ;
                ;10609  LOOP.TF.6.1C3:
U 2406, D842,15 ;10610          MOV LS[#2] TO Q                         ; Q = 2
U 2407, B69E,15 ;10611          MOV LS[ONES] TO WR[0]                   ;
U 2408, 3EE6,15 ;10612          MOV WR[0] TO LS[L73]                    ; LS = -1
U 2409, 70E6,15 ;10613          ADD Q TO LS[L73]                        ;
U 240A, B6E6,15 ;10614          MOV LS[L73] TO WR[0]                    ;
U 240B, 3640,95 ;10615          MOV LS[#1] TO WR[1]                     ; Ожидаемые данные = 1
U 240C, 0869,3C ;10616          JSR [CHECK.RESULT]                      ; проверка операции Q.PLUS.LS
U 240D, 8A40,64 ;10617          JMP [LOOP.TF.6.1C3]                     ; зацикливание при ошибке, если разрешено
U 240E, 8A70,1C ;10618          JSR [INC.OTHER]                         ;
U 240F, 0A41,A4 ;10619          JMP [LOOP.TF.6.1C4]                     ;
                ;10620  241A:
                ;10621  LOOP.TF.6.1C4:
U 241A, 3642,15 ;10622          MOV LS[#2] TO WR[0]                     ;
U 241B, 3E10,15 ;10623          MOV WR[0] TO LS[T8]                     ; LS = 2
U 241C, 5840,15 ;10624          MOV LS[#1] TO Q                         ; Q = 1
U 241D, F110,15 ;10625          SUB LS[T8] FROM Q TO LS[T8]             ;
```

```
U 241E, B610,15 ;10626          MOV LS[T8] TO WR[0]                     ;
U 241F, 369E,95 ;10627          MOV LS[ONES] TO WR[1]                   ; Ожидаемые данные = -1
U 2420, 0869,3C ;10628          JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.Q.LS
U 2421, 0A41,A4 ;10629          JMP [LOOP.TF.6.1C4]                     ; зацикливание при ошибке, если разрешено
U 2422, BA70,1C ;10630          JSR [INC.OTHER]                         ;
                ;10631  LOOP.TF.6.1C5:
U 2423, 3642,15 ;10632          MOV LS[#2] TO WR[0]                     ; LS = 2
U 2424, BE60,15 ;10633          MOV WR[0] TO LS[L30]                    ; Q = 1
U 2425, 5840,15 ;10634          MOV LS[#1] TO Q                         ;
U 2426, 7160,15 ;10635          SUB LS[L30] FROM Q TO LS[L30]           ;
U 2427, 3660,15 ;10636          MOV LS[L30] TO WR[0]                    ;
U 2428, 369E,95 ;10637          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 2429, 0869,3C ;10638          JSR [CHECK.RESULT]                      ; проверка LS.FROM.Q.LS
U 242A, 0A42,34 ;10639          JMP [LOOP.TF.6.1C5]                     ; зацикливание при ошибке, если разрешено
U 242B, BA70,1C ;10640          JSR [INC.OTHER]                         ;
                ;10641  LOOP.TF.6.1C6:
U 242C, 3642,15 ;10642          MOV LS[#2] TO WR[0]                     ;
U 242D, BEA6,15 ;10643          MOV WR[0] TO LS[L53]                    ; LS = 2
U 242E, 5840,15 ;10644          MOV LS[#1] TO Q                         ; Q = 1
U 242F, 71A6,15 ;10645          SUB LS[L53] FROM Q TO LS[L53]           ;
U 2430, 36A6,15 ;10646          MOV LS[L53] TO WR[0]                    ;
U 2431, 369E,95 ;10647          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 2432, 0869,3C ;10648          JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.Q.LS
U 2433, 0A42,C4 ;10649          JMP [LOOP.TF.6.1C6]                     ; зацикливание при ошибке, если разрешено
U 2434, BA70,1C ;10650          JSR [INC.OTHER]                         ;
                ;10651  LOOP.TF.6.1C7:
U 2435, 3642,15 ;10652          MOV LS[#2] TO WR[0]                     ;
U 2436, 3EE6,15 ;10653          MOV WR[0] TO LS[L73]                    ; LS = 2
U 2437, 5840,15 ;10654          MOV LS[#1] TO Q                         ; Q = 1
U 2438, F1E6,15 ;10655          SUB LS[L73] FROM Q TO LS[L73]           ;
U 2439, B6E6,15 ;10656          MOV LS[L73] TO WR[0]                    ;
U 243A, 369E,95 ;10657          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 243B, 0869,3C ;10658          JSR [CHECK.RESULT]                      ; проверка операции LS.FROM.Q.LS
U 243C, BA43,54 ;10659          JMP [LOOP.TF.6.1C7]                     ; зацикливание при ошибке, если разрешено
U 243D, BA70,1C ;10660          JSR [INC.OTHER]                         ;
                ;10661  LOOP.TF.6.1C8:
U 243E, D842,15 ;10662          MOV LS[#2] TO Q                         ; Q = 2
U 243F, B640,15 ;10663          MOV LS[#1] TO WR[0]                     ;
U 2440, 3E10,15 ;10664          MOV WR[0] TO LS[T8]                     ; LS = 1
U 2441, F210,15 ;10665          SUB Q FROM LS[T8]                       ;
U 2442, B610,15 ;10666          MOV LS[T8] TO WR[0]                     ;
U 2443, 369E,95 ;10667          MOV LS[ONES] TO WR[1]                   ; Ожидаемые данные = -1
U 2444, 0869,3C ;10668          JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.LS
U 2445, 0A43,E4 ;10669          JMP [LOOP.TF.6.1C8]                     ; зацикливание при ошибке, если разрешено
U 2446, BA70,1C ;10670          JSR [INC.OTHER]                         ;
                ;10671  LOOP.TF.6.1C9:
U 2447, D842,15 ;10672          MOV LS[#2] TO Q                         ; Q = 2
U 2448, B640,15 ;10673          MOV LS[#1] TO WR[0]                     ;
U 2449, BE60,15 ;10674          MOV WR[0] TO LS[L30]                    ; LS = 1
U 244A, 7260,15 ;10675          SUB Q FROM LS[L30]                      ;
U 244B, 3660,15 ;10676          MOV LS[L30] TO WR[0]                    ;
U 244C, 369E,95 ;10677          MOV LS[ONES] TO WR[1]                   ; Ожидаемые данные = -1
U 244D, 0869,3C ;10678          JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.LS
U 244E, BA44,74 ;10679          JMP [LOOP.TF.6.1C9]                     ; зацикливание при ошибке, если разрешено
U 244F, BA70,1C ;10680          JSR [INC.OTHER]                         ;
```

```
U 2450, 0A45,14 ;10681          JMP [LOOP.TF.6.1CA]                     ;
                ;10682  LOOP.TF.6.1CA:
U 2451, D842,15 ;10683          MOV LS[#2] TO Q                         ; Q = 2
U 2452, B640,15 ;10684          MOV LS[#1] TO WR[0]                     ;
U 2453, BEA6,15 ;10685          MOV WR[0] TO LS[L53]                    ; LS = 1
U 2454, 72A6,15 ;10686          SUB Q FROM LS[L53]                      ;
U 2455, 36A6,15 ;10687          MOV LS[L53] TO WR[0]                    ;
U 2456, 369E,95 ;10688          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 2457, 0869,3C ;10689          JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.LS
U 2458, 0A45,14 ;10690          JMP [LOOP.TF.6.1CA]                     ; цикл при ошибке, если разрешено
U 2459, 8A70,1C ;10691          JSR [INC.OTHER]                         ;
                ;10692  LOOP.TF.6.1CB:
U 245A, D842,15 ;10693          MOV LS[#2] TO Q                         ; Q = 2
U 245B, B640,15 ;10694          MOV LS[#1] TO WR[0]                     ;
U 245C, 3EE6,15 ;10695          MOV WR[0] TO LS[L73]                    ; LS = 1
U 245D, F2E6,15 ;10696          SUB Q FROM LS[L73]                      ;
U 245E, B6E6,15 ;10697          MOV LS[L73] TO WR[0]                    ;
U 245F, 369E,95 ;10698          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = -1
U 2460, 0869,3C ;10699          JSR [CHECK.RESULT]                      ; проверка Q.FROM.LS
U 2461, 8A45,A4 ;10700          JMP [LOOP.TF.6.1CB]                     ; цикл при ошибке, если разрешено
U 2462, 8A70,1C ;10701          JSR [INC.OTHER]                         ;
                ;10702  LOOP.TF.6.1CC:
U 2463, 5898,15 ;10703          MOV LS[ALTER.ODD] TO Q                  ;
U 2464, 2540,15 ;10704          INC Q                                   ; Q = AAAAAAAB
U 2465, 369A,15 ;10705          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 2466, 3E10,15 ;10706          MOV WR[0] TO LS[T8]                     ; LS = 55555555
U 2467, 7310,15 ;10707          BIS Q TO LS[T8]                         ;
U 2468, B610,15 ;10708          MOV LS[T8] TO WR[0]                     ;
U 2469, 369E,95 ;10709          MOV LS[ONES] TO WR[1]                   ; Ожидаемые данные = FFFFFFFF
U 246A, 0869,3C ;10710          JSR [CHECK.RESULT]                      ; проверка операции Q.OR.LS
U 246B, 8A46,34 ;10711          JMP [LOOP.TF.6.1CC]                     ; зацикливание при ошибке, если разрешено
U 246C, 8A70,1C ;10712          JSR [INC.OTHER]                         ;
                ;10713  LOOP.TF.6.1CD:
U 246D, 5898,15 ;10714          MOV LS[ALTER.ODD] TO Q
U 246E, 2540,15 ;10715          INC Q                                   ; Q = AAAAAAAB
U 246F, 369A,15 ;10716          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 2470, BE60,15 ;10717          MOV WR[0] TO LS[L30]                    ; LS = 55555555
U 2471, F360,15 ;10718          BIS Q TO LS[L30]                        ;
U 2472, 3660,15 ;10719          MOV LS[L30] TO WR[0]                    ;
U 2473, 369E,95 ;10720          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные = FFFFFFFF
U 2474, 0869,3C ;10721          JSR [CHECK.RESULT]                      ; проверка операции Q.OR.LS
U 2475, 0A46,D4 ;10722          JMP [LOOP.TF.6.1CD]                     ; зацикливание при ошибке, если разрешено
U 2476, 8A70,1C ;10723          JSR [INC.OTHER]                         ;
                ;10724  LOOP.TF.6.1CE:
U 2477, 5898,15 ;10725          MOV LS[ALTER.ODD] TO Q
U 2478, 2540,15 ;10726          INC Q                                   ; Q = AAAAAAAB
U 2479, 369A,15 ;10727          MOV LS[ALTER.EVEN] TO WR[0]             ;
U 247A, BEA6,15 ;10728          MOV WR[0] TO LS[L53]                    ; LS = 55555555
U 247B, F3A6,15 ;10729          BIS Q TO LS[L53]                        ;
U 247C, 36A6,15 ;10730          MOV LS[L53] TO WR[0]                    ;
U 247D, 369E,95 ;10731          MOV LS[ONES] TO WR[1]                   ; Ожидаемые данные = FFFFFFFF
U 247E, 0869,3C ;10732          JSR [CHECK.RESULT]                      ; проверка операции Q.OR.LS
U 247F, 8A47,74 ;10733          JMP [LOOP.TF.6.1CE]                     ; зацикливание при ошибке, если разрешено
U 2480, 8A70,1C ;10734          JSR [INC.OTHER]                         ;
                ;10735  LOOP.TF.6.1CF:
```

```
U 2481, 5898,15 ;10736           MOV LS[ALTER.ODD] TO Q
U 2482, 2540,15 ;10737           INC Q                                  ; Q = AAAAAAA8
U 2483, 369A,15 ;10738           MOV LS[ALTER.EVEN] TO WR[0]            ;
U 2484, 3EE6,15 ;10739           MOV WR[0] TO LS[L73]                   ; LS = 55555555
U 2485, 73E6,15 ;10740           BIS Q TO LS[L73]                       ;
U 2486, B6E6,15 ;10741           MOV LS[L73] TO WR[0]                   ;
U 2487, 369E,95 ;10742           MOV LS[ONES] TO WR[1]                  ; ожидаемые данные = FFFFFFFF
U 2488, 0869,3C ;10743           JSR [CHECK.RESULT]                     ; проверка операции Q.OR.LS
U 2489, 8A48,14 ;10744           JMP [LOOP.TF.6.1CF]                    ; зацикливание при ошибке, если разрешено
U 248A, 8A70,1C ;10745           JSR [INC.OTHER]                        ;
                ;10746   LOOP.TF.6.1D0:
U 248B, 5898,15 ;10747           MOV LS[ALTER.ODD] TO Q                 ; Q = AAAAAAA8
U 248C, 3628,15 ;10748           MOV LS[#FFFF] TO WR[0]                 ;
U 248D, 3E10,15 ;10749           MOV WR[0] TO LS[T8]                    ; LS = 0000FFFF
U 248E, F410,15 ;10750           AND Q TO LS[T8]                        ;
U 248F, B610,15 ;10751           MOV LS[T8] TO WR[0]                    ;
U 2490, 3698,95 ;10752           MOV LS[ALTER.ODD] TO WR[1]             ;
U 2491, 4518,95 ;10753           BIC LS[T12] TO WR[1]                   ; одидаемые данные = 0000AAAA
U 2492, 0869,3C ;10754           JSR [CHECK.RESULT]                     ; проверка операции Q.AND.LS
U 2493, 8A48 B4 ;10755           JMP [LOOP.TF.6.1D0]                    ; зацикливание при ошибке, если разрешено
U 2494, 8A70,1C ;10756           JSR [INC.OTHER]                        ;
                ;10757   LOOP.TF.6.1D1:
U 2495, 5898,15 ;10758           MOV LS[ALTER.ODD] TO Q                 ; Q = AAAAAAAA
U 2496, 3628,15 ;10759           MOV LS[#FFFF] TO WR[0]                 ;
U 2497, BE60,15 ;10760           MOV WR[0] TO LS[L30]                   ; LS = 0000FFFF
U 2498, 7460,15 ;10761           AND Q TO LS[L30]                       ;
U 2499, 3660,15 ;10762           MOV LS[L30] TO WR[0]                   ;
U 249A, 3698,95 ;10763           MOV LS[ALTER.ODD] TO WR[1]             ;
U 249B, 4518,95 ;10764           BIC LS[T12] TO WR[1]                   ; Ожидаемые данные = 0000AAAA
U 249C, 0869,3C ;10765           JSR [CHECK.RESULT]                     ; проверка операции Q.AND.LS
U 249D, 8A49,54 ;10766           JMP [LOOP.TF.6.1D1]                    ; зацикливание при ошибке, если разрешено
U 249E, 8A70,1C ;10767           JSR [INC.OTHER]                        ;
                ;10768   LOOP.TF.6.1D2:
U 249F, 5898,15 ;10769           MOV LS[ALTER.ODD] TO Q                 ; Q = AAAAAAAA
U 24A0, 3628,15 ;10770           MOV LS[#FFFF] TO WR[0]                 ;
U 24A1, BEA6,15 ;10771           MOV WR[0] TO LS[L53]                   ; LS = 0000FFFF
U 24A2, 74A6,15 ;10772           AND Q TO LS[L53]                       ;
U 24A3, 36A6,15 ;10773           MOV LS[L53] TO WR[0]                   ;
U 24A4, 3698,95 ;10774           MOV LS[ALTER.ODD] TO WR[1]             ;
U 24A5, 4518,95 ;10775           BIC LS[T12] TO WR[1]                   ; ожидаемые данные = 0000AAAA
U 24A6, 0869,3C ;10776           JSR [CHECK.RESULT]                     ; проверка операции Q.AND.LS
U 24A7, 8A49,F4 ;10777           JMP [LOOP.TF.6.1D2]                    ; зацикливание при ошибке, если разрешено
U 24A8, 8A70,1C ;10778           JSR [INC.OTHER]                        ;
                ;10779   LOOP.TF.6.1D3:
U 24A9, 5898,15 ;10780           MOV LS[ALTER.ODD] TO Q                 ; Q = AAAAAAAA
U 24AA, 3628,15 ;10781           MOV LS[#FFFF] TO WR[0]                 ;
U 24AB, 3EE6,15 ;10782           MOV WR[0] TO LS[L73]                   ; LS = 0000FFFF
U 24AC, F4E6,15 ;10783           AND Q TO LS[L73]                       ;
U 24AD, B6E6,15 ;10784           MOV LS[L73] TO WR[0]                   ;
U 24AE, 3698,95 ;10785           MOV LS[ALTER.ODD] TO WR[1]             ;
U 24AF, 4518,95 ;10786           BIC LS[T12] TO WR[1]                   ; Ожидаемые данные = 0000AAAA
U 2480, 0869,3C ;10787           JSR [CHECK.RESULT]                     ; проверка операции Q.AND.LS
U 24B1, 8A4A,94 ;10788           JMP [LOOP.TF.6.1D3]                    ; зацикливание при ошибке, если разрешено
U 24B2, 8A70,1C ;10789           JSR [INC.OTHER]                        ;
                ;10790   LOOP.TF.6.1D4:
```

```
U 24B3, 5898,15 ;10791          MOV LS[ALTER.ODD] TO Q                  ; Q = AAAAAAAA
U 24B4, 3628,15 ;10792          MOV LS[#FFFF] TO WR[0]                  ;
U 24B5, 3E10,15 ;10793          MOV WR[0] TO LS[T8]                     ; LS = 0000FFFF
U 24B6, 7510,15 ;10794          XOR Q TO LS[T8]                         ;
U 24B7, B610,15 ;10795          MOV LS[T8] TO WR[0]                     ;
U 24B8, 3722,95 ;10796          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 24B9, 0869,3C ;10797          JSR [CHECK.RESULT]                      ; проверка операции Q.XOR.LS
U 24BA, 0A48,34 ;10798          JMP [LOOP.TF.6.1D4]                     ; зацикливание при ошибке, если разрешено
U 24BB, 8A70,1C ;10799          JSR [INC.OTHER]                         ;
                ;10800  LOOP.TF.6.1D5:
U 24BC, 5898,15 ;10801          MOV LS[ALTER.ODD] TO Q                  ; Q = AAAAAAAA
U 24BD, 3628,15 ;10802          MOV LS[#FFFF] TO WR[0]                  ;
U 24BE, BE60,15 ;10803          MOV WR[0] TO LS[L30]                    ; LS = 0000FFFF
U 24BF, F560,15 ;10804          XOR Q TO LS[L30]                        ;
U 24C0, 3660,15 ;10805          MOV LS[L30] TO WR[0]                    ;
U 24C1, 3722,95 ;10806          MOV LS[ALTER.MIX] TO WR[1]              ; Ожидаемые данные = AAAA5555
U 24C2, 0869,3C ;10807          JSR [CHECK.RESULT]                      ; проверка операции Q.XOR.LS
U 24C3, 0A4B,C4 ;10808          JMP [LOOP.TF.6.1D5]                     ; зацикливание при ошибке, если разрешено
U 24C4, 8A70,1C ;10809          JSR [INC.OTHER]                         ;
                ;10810  LOOP.TF.6.1D6:
U 24C5, 5898,15 ;10811          MOV LS[ALTER.ODD] TO Q                  ; Q = AAAAAAAA
U 24C6, 3628,15 ;10812          MOV LS[#FFFF] TO WR[0]                  ;
U 24C7, BEA6,15 ;10813          MOV WR[0] TO LS[L53]                    ; LS = 0000FFFF
U 24C8, F5A6,15 ;10814          XOR Q TO LS[L53]                        ;
U 24C9, 36A6,15 ;10815          MOV LS[L53] TO WR[0]
U 24CA, 3722,95 ;10816          MOV LS[ALTER.MIX] TO WR[1]              ; Ожидаемые данные = AAAA5555
U 24CB, 0869,3C ;10817          JSR [CHECK.RESULT]                      ; проверка операции Q.XOR.LS
U 24CC, 8A4C,54 ;10818          JMP [LOOP.TF.6.1D6]                     ; зацикливание при ошибке, если разрешено
U 24CD, 8A70,1C ;10819          JSR [INC.OTHER]                         ;
                ;10820  LOOP.TF.6.1D7:
U 24CE, 5898,15 ;10821          MOV LS[ALTER.ODD] TO Q                  ; Q = AAAAAAAA
U 24CF, 3628,15 ;10822          MOV LS[#FFFF] TO WR[0]                  ;
U 24D0, 3EE6,15 ;10823          MOV WR[0] TO LS[L73]                    ; LS = 0000FFFF
U 24D1, 75E6,15 ;10824          XOR Q TO LS[L73]                        ;
U 24D2, B6E6,15 ;10825          MOV LS[L73] TO WR[0]                    ;
U 24D3, 3722,95 ;10826          MOV LS[ALTER.MIX] TO WR[1]              ; ожидаемые данные = AAAA5555
U 24D4, 0869,3C ;10827          JSR [CHECK.RESULT]                      ; проверка операции Q.XOR.LS
U 24D5, 0A4C,E4 ;10828          JMP [LOOP.TF.6.1D7]                     ; зацикливание при ошибке, если разрешено
U 24D6, 8A70,1C ;10829          JSR [INC.OTHER]                         ;
                ;10830  LOOP.TF.6.1D8:
U 24D7, D89A,15 ;10831          MOV LS[ALTER.EVEN] TO Q                 ; Q = 55555555
U 24D8, 7610,15 ;10832          MOV Q TO LS[T8]                         ;
U 24D9, B610,15 ;10833          MOV LS[T8] TO WR[0]                     ;
U 24DA, B69A,95 ;10834          MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 24DB, 0869,3C ;10835          JSR [CHECK.RESULT]                      ; проверка операции MOV.Q.LS
U 24DC, 8A4D,74 ;10836          JMP [LOOP.TF.6.1D8]                     ; зацикливание при ошибке, если разрешено
U 24DD, 8A70,1C ;10837          JSR [INC.OTHER]                         ;
                ;10838  LOOP.TF.6.1D9:
U 24DE, D89A,15 ;10839          MOV LS[ALTER.EVEN] TO Q                 ; Q = 55555555
U 24DF, F660,15 ;10840          MOV Q TO LS[L30]                        ;
U 24E0, 3660,15 ;10841          MOV LS[L30] TO WR[0]                    ;
U 24E1, B69A,95 ;10842          MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные = 55555555
U 24E2, 0869,3C ;10843          JSR [CHECK.RESULT]                      ; проверка операции MOV.Q.LS
U 24E3, 8A4D,E4 ;10844          JMP [LOOP.TF.6.1D9]                     ; зацикливание при ошибке, если разрешено
U 24E4, 8A70,1C ;10845          JSR [INC.OTHER]                         ;
```

```
                    ;10846   LOOP.TF.6.1DA:
U 24E5, D89A,15 ;10847           MOV LS[ALTER.EVEN] TO Q                ; Q = 55555555
U 24E6, F6A6,15 ;10848           MOV Q TO LS[L53]                       ;
U 24E7, 36A6,15 ;10849           MOV LS[L53] TO WR[0]                   ;
U 24E8, 969A,95 ;10850           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные = 55555555
U 24E9, 0869,3C ;10851           JSR [CHECK.RESULT]                     ; проверка операции MOV.Q.LS
U 24EA, 0A4E,54 ;10852           JMP [LOOP.TF.6.1DA]                    ; зацикливание при ошибке, если разрешено
U 24EB, 8A70,1C ;10853           JSR [INC.OTHER]                        ;
                    ;10854   LOOP.TF.6.1DB:
U 24EC, D89A,15 ;10855           MOV LS[ALTER.EVEN] TO Q                ; Q=55555555
U 24ED, 76E6,15 ;10856           MOV Q TO LS[L73]                       ;
U 24EE, B6E6,15 ;10857           MOV LS[L73] TO WR[0]                   ;
U 24EF, B69A,95 ;10858           MOV LS[ALTER.EVEN] TO WR[1]            ; ожидаемые данные = 55555555
U 24F0, 0869,3C ;10859           JSR [CHECK.RESULT]                     ; проверка операции MOV.Q.LS
U 24F1, 0A4E,C4 ;10860           JMP [LOOP.TF.6.1DB]                    ; зацикливание при ошибке, если разрешено
U 24F2, 8A70,1C ;10861           JSR [INC.OTHER]                        ;
                    ;10862   LOOP.TF.6.1DC:
U 24F3, 5840,15 ;10863           MOV LS[#1] TO Q                        ; Q=1
U 24F4, F710,15 ;10864           MNEG Q TO LS[T8]                       ;
U 24F5, B610,15 ;10865           MOV LS[T8] TO WR[0]                    ;
U 24F6, 369E,95 ;10866           MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 24F7, 0869,3C ;10867           JSR [CHECK.RESULT]                     ; проверка операции Q.NEG.LS
U 24F8, 8A4F,34 ;10868           JMP [LOOP.TF.6.1DC]                    ; зацикливание при ошибке, если разрешено
U 24F9, 8A70,1C ;10869           JSR [INC.OTHER]                        ;
                    ;10870   LOOP.TF.6.1DD:
U 24FA, 5840,15 ;10871           MOV LS[#1] TO Q                        ; Q=1
U 24FB, 7760,15 ;10872           MNEG Q TO LS[L30]                      ;
U 24FC, 3660,15 ;10873           MOV LS[L30] TO WR[0]                   ;
U 24FD, 369E,95 ;10874           MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 24FE, 0869,3C ;10875           JSR [CHECK.RESULT]                     ; проверка операции Q.NEG.LS
U 24FF, 8A4F,A4 ;10876           JMP [LOOP.TF.6.1DD]                    ; зацикливание при ошибке, если разрешено
U 2500, 8A70,1C ;10877           JSR [INC.OTHER]                        ;
                    ;10878   LOOP.TF.6.1DE:
U 2501, 5840,15 ;10879           MOV LS[#1] TO Q                        ; Q=1
U 2502, 77A6,15 ;10880           MNEG Q TO LS[L53]                      ;
U 2503, 36A6,15 ;10881           MOV LS[L53] TO WR[0]                   ;
U 2504, 369E,95 ;10882           MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 2505, 0869,3C ;10883           JSR [CHECK.RESULT]                     ; проверка операции Q.NEG.LS
U 2506, 8A50,14 ;10884           JMP [LOOP.TF.6.1DE]                    ; зацикливание при ошибке, если разрешено
U 2507, 8A70,1C ;10885           JSR [INC.OTHER]                        ;
                    ;10886   LOOP.TF.6.1DF:
U 2508, 5840,15 ;10887           MOV LS[#1] TO Q                        ; Q=1
U 2509, F7E6,15 ;10888           MNEG Q TO LS[L73]                      ;
U 250A, B6E6,15 ;10889           MOV LS[L73] TO WR[0]                   ;
U 250B, 369E,95 ;10890           MOV LS[ONES] TO WR[1]                  ; ожидаемые данные=-1
U 250C, 0869,3C ;10891           JSR [CHECK.RESULT]                     ; проверка операцииQ.NEG.LS
U 250D, 8A50,84 ;10892           JMP [LOOP.TF.6.1DF]                    ; зацикливание прри ошибке, если разре:ено
U 250E, 8A70,1C ;10893           JSR [INC.OTHER]                        ;
                    ;10894   LOOP.TF.6.1E0:
U 250F, B698,15 ;10895           MOV LS[ALTER.EVEN] TO WR[2]            ; WR2=55555555
U 2510, 7811,15 ;10896           MCOM WR[2] TO LS[T8]                   ;
U 2511, B610,15 ;10897           MOV LS[T8] TO WR[0]                    ;
U 2512, 3698,95 ;10898           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные=AAAAAAAA
U 2513, 0869,3C ;10899           JSR [CHECK.RESULT]                     ; проверка операции WR.COM.LS
U 2514, 0A50,F4 ;10900           JMP [LOOP.TF.6.1E0]                    ; зацикливание при ошибке, если разрешено
```

```
U 2515, 8A70,1C ;10901          JSR [INC.OTHER]                         ;
                ;10902  LOOP.TF.6.1E1:
U 2516, B698,15 ;10903          MOV LS[ALTER.EVEN] TO WR[2]             ; WR2=55555555
U 2517, F861,15 ;10904          MCOM WR[2] TO LS[L30]                   ;
U 2518, 3660,15 ;10905          MOV LS[L30] TO WR[0]                    ;
U 2519, 3698,95 ;10906          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные=AAAAAAAA
U 251A, 0869,3C ;10907          JSR [CHECK.RESULT]                      ; проверка операции WR.COM.LS
U 251B, 8A51,64 ;10908          JMP [LOOP.TF.6.1E1]                     ; зацикливание при ошибке, если разрешео
U 251C, 8A70,1C ;10909          JSR [INC.OTHER]                         ;
                ;10910  LOOP.TF.6.1E2:
U 251D, B698,15 ;10911          MOV LS[ALTER.EVEN] TO WR[2]             ; WR2=55555555
U 251E, F8A7,15 ;10912          MCOM WR[2] TO LS[L53]                   ;
U 251F, 36A6,15 ;10913          MOV LS[L53] TO WR[0]                    ;
U 2520, 3698,95 ;10914          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные=AAAAAAAA
U 2521, 0869,3C ;10915          JSR [CHECK.RESULT]                      ; проверка операции WR.COM.LS
U 2522, 0A51,D4 ;10916          JMP [LOOP.TF.6.1E2]                     ; зацикливание при ошибке, если разрешено
U 2523, 8A70,1C ;10917          JSR [INC.OTHER]                         ;
                ;10918  LOOP.TF.6.1E3:
U 2524, B698,15 ;10919          MOV LS[ALTER.EVEN] TO WR[2]             ; WR2=55555555
U 2525, 78E7,15 ;10920          MCOM WR[2] TO LS[L73]                   ;
U 2526, B6E6,15 ;10921          MOV LS[L73] TO WR[0]                    ;
U 2527, 3698,95 ;10922          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные=AAAAAAAA
U 2528, 0869,3C ;10923          JSR [CHECK.RESULT]                      ; проверка операции WR.COM.LS
U 2529, 0A52,44 ;10924          JMP [LOOP.TF.6.1E3]                     ; зацикливание при ошибке, если разрешено
U 252A, 8A70,1C ;10925          JSR [INC.OTHER]                         ;
                ;10926  LOOP.TF.6.1E4:
U 252B, 3641,15 ;10927          MOV LS[#1] TO WR[2]                     ; WR2=1
U 252C, F911,15 ;10928          MNEG WR[2] TO LS[T8]                    ;
U 252D, B610,15 ;10929          MOV LS[T8] TO WR[0]                     ;
U 252E, 369E,95 ;10930          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 252F, 0869,3C ;10931          JSR [CHECK.RESULT]                      ; проверка операции WR.NEG.LS
U 2530, 0A52,B4 ;10932          JMP [LOOP.TF.6.1E4]                     ; зацикливание при ошибке, если разрешено
U 2531, 8A70,1C ;10933          JSR [INC.OTHER]                         ;
                ;10934  LOOP.TF.6.1E5:
U 2532, 3641,15 ;10935          MOV LS[#1] TO WR[2]                     ; WR2=1
U 2533, 7961,15 ;10936          MNEG WR[2] TO LS[L30]                   ;
U 2534, 3660,15 ;10937          MOV LS[L30] TO WR[0]                    ;
U 2535, 369E,95 ;10938          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 2536, 0869,3C ;10939          JSR [CHECK.RESULT]                      ; проверка операции WR.NEG.LS
U 2537, 8A53,24 ;10940          JMP [LOOP.TF.6.1E5]                     ; зацикливание при ошибке, если разрешено
U 2538, 8A70,1C ;10941          JSR [INC.OTHER]                         ;
                ;10942  LOOP.TF.6.1E6:
U 2539, 3641,15 ;10943          MOV LS[#1] TO WR[2]                     ; WR2=1
U 253A, 79A7,15 ;10944          MNEG WR[2] TO LS[L53]                   ;
U 253B, 36A6,15 ;10945          MOV LS[L53] TO WR[0]                    ;
U 253C, 369E,95 ;10946          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 253D, 0869,3C ;10947          JSR [CHECK.RESULT]                      ; проверка операции WR.NEG.LS
U 253E, 0A53,94 ;10948          JMP [LOOP.TF.6.1E6]                     ; зацикливание при ошибке, если разрешено
U 253F, 8A70,1C ;10949          JSR [INC.OTHER]                         ;
                ;10950  LOOP.TF.6.1E7:
U 2540, 3641,15 ;10951          MOV LS[#1] TO WR[2]                     ; WR2=1
U 2541, F9E7,15 ;10952          MNEG WR[2] TO LS[L73]                   ;
U 2542, B6E6,15 ;10953          MOV LS[L73] TO WR[0]                    ;
U 2543, 369E,95 ;10954          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 2544, 0869,3C ;10955          JSR [CHECK.RESULT]                      ; проверка операции WR.NEG.LS
```

```
U 2545, 8A54,04 ;10956          JMP [LOOP.TF.6.1E7]                    ; зацикливание при ошибке, если разрешено
U 2546, 8A70,1C ;10957          JSR [INC.OTHER]                        ;
                ;10958  LOOP.TF.6.1E8:
U 2547, D842,15 ;10959          MOV LS[#2] TO Q                        ; Q=2
U 2548, 369F,15 ;10960          MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 2549, FA11,15 ;10961          ADD Q PLUS WR[2] TO LS[T8]             ;
U 254A, B610,15 ;10962          MOV LS[T8] TO WR[0]                    ;
U 254B, 3640,95 ;10963          MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 254C, 0869,3C ;10964          JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.WR.TO.LS
U 254D, 0A54,74 ;10965          JMP [LOOP.TF.6.1E8]                    ; зацикливание при ошибке, если разрешено
U 254E, 8A70,1C ;10966          JSR [INC.OTHER]                        ;
                ;10967  LOOP.TF.6.1E9:
U 254F, D842,15 ;10968          MOV LS[#2] TO Q                        ; Q=2
U 2550, 369F,15 ;10969          MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 2551, 7A61,15 ;10970          ADD Q PLUS WR[2] TO LS[L30]            ;
U 2552, 3660,15 ;10971          MOV LS[L30] TO WR[0]                   ;
U 2553, 3640,95 ;10972          MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 2554, 0869,3C ;10973          JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.WR.TO.LS
U 2555, 8A54,F4 ;10974          JMP [LOOP.TF.6.1E9]                    ; зацикливание при ошибке, если разрешено
U 2556, 8A70,1C ;10975          JSR [INC.OTHER]                        ;
                ;10976  LOOP.TF.6.1EA:
U 2557, D842,15 ;10977          MOV LS[#2] TO Q                        ; Q=2
U 2558, 369F,15 ;10978          MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 2559, 7AA7,15 ;10979          ADD Q PLUS WR[2] TO LS[L53]            ;
U 255A, 36A6,15 ;10980          MOV LS[L53] TO WR[0]                   ;
U 255B, 3640,95 ;10981          MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 255C, 0869,3C ;10982          JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.WR.TO.LS
U 255D, 8A55,74 ;10983          JMP [LOOP.TF.6.1EA]                    ; зацикливание при ошибке, если разрешено
U 255E, 8A70,1C ;10984          JSR [INC.OTHER]                        ;
                ;10985  LOOP.TF.6.1EB:
U 255F, D842,15 ;10986          MOV LS[#2] TO Q                        ; Q=2
U 2560, 369F,15 ;10987          MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 2561, FAE7,15 ;10988          ADD Q PLUS WR[2] TO LS[L73]            ;
U 2562, B6E6,15 ;10989          MOV LS[L73] TO WR[0]                   ;
U 2563, 3640,95 ;10990          MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 2564, 0869,3C ;10991          JSR [CHECK.RESULT]                     ; проверка операции Q.PLUS.WR.TO.LS
U 2565, 0A55,F4 ;10992          JMP [LOOP.TF.6.1EB]                    ; зацикливание при ошибке, если разрешено
U 2566, 8A70,1C ;10993          JSR [INC.OTHER]                        ;
                ;10994  LOOP.TF.6.1EC:
U 2567, D842,15 ;10995          MOV LS[#2] TO Q                        ; Q=2
U 2568, 3641,15 ;10996          MOV LS[#1] TO WR[2]                    ; WR2=1
U 2569, 7B11,15 ;10997          SUB Q FROM WR[2] TO LS[T8]             ;
U 256A, B610,15 ;10998          MOV LS[T8] TO WR[0]                    ;
U 256B, 369E,95 ;10999          MOV LS[ONES] TO WR[1]                  ; ожидаемые данные =-1
U 256C, 0869,3C ;11000          JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.WR.TO.LS
U 256D, 8A56,74 ;11001          JMP [LOOP.TF.6.1EC]                    ; зацикливание при ошибке, если разрешено
U 256E, 8A70,1C ;11002          JSR [INC.OTHER]                        ;
                ;11003  LOOP.TF.6.1ED:
U 256F, D842,15 ;11004          MOV LS[#2] TO Q                        ; Q=2
U 2570, 3641,15 ;11005          MOV LS[#1] TO WR[2]                    ; WR2=1
U 2571, FB61,15 ;11006          SUB Q FROM WR[2] TO LS[L30]            ;
U 2572, 3660,15 ;11007          MOV LS[L30] TO WR[0]                   ;
U 2573, 369E,95 ;11008          MOV LS[ONES] TO WR[1]                  ; ожидаемые данные =-1
U 2574, 0869,3C ;11009          JSR [CHECK.RESULT]                     ; проверка операции Q.FROM.WR.TO.LS
U 2575, 0A56,F4 ;11010          JMP [LOOP.TF.6.1ED]                    ; зацикливание при ошибке, если разрешено
```

```
U 2576, 8A70,1C ;11011          JSR [INC.OTHER]                         ;
                ;11012  LOOP.TF.6.1EE:
U 2577, D842,15 ;11013          MOV LS[#2] TO Q                         ; Q=2
U 2578, 3641,15 ;11014          MOV LS[#1] TO WR[2]                     ; WR2=1
U 2579, FBA7,15 ;11015          SUB Q FROM WR[2] TO LS[L53]             ;
U 257A, 36A6,15 ;11016          MOV LS[L53] TO WR[0]                    ;
U 257B, 369E,95 ;11017          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 257C, 0869,3C ;11018          JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.WR.TO.LS
U 257D, 0A57,74 ;11019          JMP [LOOP.TF.6.1EE]                     ; зацикливание при ошибке, если разрешено
U 257E, 8A70,1C ;11020          JSR [INC.OTHER]                         ;
                ;11021  LOOP.TF.6.1EF:
U 257F, D842,15 ;11022          MOV LS[#2] TO Q                         ; Q=2
U 2580, 3641,15 ;11023          MOV LS[#1] TO WR[2]                     ; WR2=1
U 2581, 78E7,15 ;11024          SUB Q FROM WR[2] TO LS[L73]             ;
U 2582, B6E6,15 ;11025          MOV LS[L73] TO WR[0]                    ;
U 2583, 369E,95 ;11026          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 2584, 0869,3C ;11027          JSR [CHECK.RESULT]                      ; проверка операции Q.FROM.WR.TO.LS
U 2585, 8A57,F4 ;11028          JMP [LOOP.TF.6.1EF]                     ; зацикливание при ошибке, если разрешено
U 2586, 8A70,1C ,11029          JSR [INC.OTHER]                         ;
                ;11030  LOOP.TF 6.1F0:
U 2587, 369C,15 ;1 031          MOV LS[ZERO] TO WR[0]                   ;
U 2588, 3E10,15 ;11032          MOV WR[0] TO LS[T8]                     ; LS=0
U 2589, 7C10,15 ;11033          DEC LS[T8]                              ;
U 258A, B610,15 ;11034          MOV LS[T8] TO WR[0]                     ;
U 258B, 369E,95 ;11035          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 258C, 0869,3C ;11036          JSR [CHECK.RESULT]                      ; проверка операции DEC.LS
U 258D, 0A58,74 ;11037          JMP [LOOP.TF.6.1F0]                     ; зацикливание при ошибке, если разрешено
U 258E, 8A70,1C ;11038          JSR [INC.OTHER]                         ;
                ;11039  LOOP.TF.6.1F1:
U 258F, 369C,15 ;11040          MOV LS[ZERO] TO WR[0]                   ;
U 2590, BE60,15 ;11041          MOV WR[0] TO LS[L30]                    ; LS=0
U 2591, FC60,15 ;11042          DEC LS[L30]                             ;
U 2592, 3660,15 ;11043          MOV LS[L30] TO WR[0]                    ;
U 2593, 369E,95 ;11044          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 2594, 0869,3C ;11045          JSR [CHECK.RESULT]                      ; проверка операции DEC.LS
U 2595, 8A58,F4 ;11046          JMP [LOOP.TF.6.1F1]                     ; зацикливание при ошибке, если разрешено
U 2596, 8A70,1C ;11047          JSR [INC.OTHER]                         ;
                ;11048  LOOP.TF.6.1F2.
U 2597, 369C,15 ;11049          MOV LS[ZERO] TO WR[0]                   ;
U 2598, BEA6,15 ;11050          MOV WR[0] TO LS[L53]                    ; LS=0
U 2599, FCA6,15 ;11051          DEC LS[L53]                             ;
U 259A, 36A6,15 ;11052          MOV LS[L53] TO WR[0]                    ;
U 259B, 369E,95 ;11053          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 259C, 0869,3C ;11054          JSR [CHECK.RESULT]                      ; проверка операции DEC.LS
U 259D, 8A59,74 ;11055          JMP [LOOP.TF.6.1F2]                     ; зацикливание при ошибке, если разрешено
U 259E, 8A70,1C ;11056          JSR [INC.OTHER]                         ;
                ;11057  LOOP.TF.6.1F3:
U 259F, 369C,15 ;11058          MOV LS[ZERO] TO WR[0]                   ;
U 25A0, 3EE6,15 ;11059          MOV WR[0] TO LS[L73]                    ; LS=0
U 25A1, 7CE6,15 ;11060          DEC LS[L73]                             ;
U 25A2, B6E6,15 ;11061          MOV LS[L73] TO WR[0]                    ;
U 25A3, 369E,95 ;11062          MOV LS[ONES] TO WR[1]                   ; ожидаемые данные =-1
U 25A4, 0869,3C ;11063          JSR [CHECK.RESULT]                      ; проверка операции DEC.LS
U 25A5, 0A59,F4 ;11064          JMP [LOOP.TF.6.1F3]                     ; зацикливание при ошибке, если разрешено
U 25A6, 8A70,1C ;11065          JSR [INC.OTHER]                         ;
```

```
                    ;11066    LOOP.TF.6.1F4:
U 25A7,  369A,15    ;11067              MOV LS[ALTER.EVEN] TO WR[0]              ;
U 25A8,  3E10,15    ;11068              MOV WR[0] TO LS[T8]                      ; LS=55555555
U 25A9,  FD10,15    ;11069              COM LS[T8]                               ;
U 25AA,  B610,15    ;11070              MOV LS[T8] TO WR[0]                      ;
U 25AB,  3698,95    ;11071              MOV LS[ALTER.ODD] TO WR[1]               ; ожидаемые данные =AAAAAAAA
U 25AC,  0869,3C    ;11072              JSR [CHECK.RESULT]                       ; проверка операции COM.LS
U 25AD,  8A5A,74    ;11073              JMP [LOOP.TF.6.1F4]                      ; зацикливание при ошибке, если разрешено
U 25AE,  8A70,1C    ;11074              JSR [INC.OTHER]                          ;
                    ;11075    LOOP.TF.6.1F5:
U 25AF,  369A,15    ;11076              MOV LS[ALTER.EVEN] TO WR[0]              ;
U 25B0,  BE60,15    ;11077              MOV WR[0] TO LS[L30]                     ; LS=55555555
U 25B1,  7D60,15    ;11078              COM LS[L30]                              ;
U 25B2,  3660,15    ;11079              MOV LS[L30] TO WR[0]                     ;
U 25B3,  3698,95    ;11080              MOV LS[ALTER.ODD] TO WR[1]               ; ожидаемые данные =AAAAAAAA
U 25B4,  0869,3C    ;11081              JSR [CHECK.RESULT]                       ; проверка операции COM.LS
U 25B5,  0A5A,F4    ;11082              JMP [LOOP.TF.6.1F5]                      ; зацикливание при ошибке, если разрешено
U 25B6,  8A70,1C    ;11083              JSR [INC.OTHER]                          ;
                    ;11084    LOOP.TF.6.1F6:
U 25B7,  369A,15    ;11085              MOV LS[ALTER.EVEN] TO WR[0]              ;
U 25B8,  BEA6,15    ;11086              MOV WR[0] TO LS[L53]                     ; LS=55555555
U 25B9,  7DA6,15    ;11087              COM LS[L53]                              ;
U 25BA,  36A6,15    ;11088              MOV LS[L53] TO WR[0]                     ;
U 25BB,  3698,95    ;11089              MOV LS[ALTER.ODD] TO WR[1]               ; ожидаемые данные =AAAAAAAA
U 25BC,  0869,3C    ;11090              JSR [CHECK.RESULT]                       ; проверка операции COM.LS
U 25BD,  0A5B,74    ;11091              JMP [LOOP.TF.6.1F6]                      ; зацикливание при ошибке, если разрешено
U 25BE,  8A70,1C    ;11092              JSR [INC.OTHER]                          ;
                    ;11093    LOOP.TF.6.1F7:
U 25BF,  369A,15    ;11094              MOV LS[ALTER.EVEN] TO WR[0]              ;
U 25C0,  3EE6,15    ;11095              MOV WR[0] TO LS[L73]                     ; LS=55555555
U 25C1,  FDE6,15    ;11096              COM LS[L73]                              ;
U 25C2,  B6E6,15    ;11097              MOV LS[L73] TO WR[0]                     ;
U 25C3,  3698,95    ;11098              MOV LS[ALTER.ODD] TO WR[1]               ; ожидаемые данные =AAAAAAAA
U 25C4,  0869,3C    ;11099              JSR [CHECK.RESULT]                       ; проверка операции COM.LS
U 25C5,  8A5B,F4    ;11100              JMP [LOOP.TF.6.1F7]                      ; зацикливание при ошибке, если разрешено
U 25C6,  8A70,1C    ;11101              JSR [INC.OTHER]                          ;
                    ;11102    LOOP.TF.6.1F8:
U 25C7,  B69E,15    ;11103              MOV LS[ONES] TO WR[0]                    ;
U 25C8,  3E10,15    ;11104              MOV WR[0] TO LS[T8]                      ; LS=-1
U 25C9,  FE10,15    ;11105              NEG LS[T8]                               ;
U 25CA,  B610,15    ;11106              MOV LS[T8] TO WR[0]                      ;
U 25CB,  3640,95    ;11107              MOV LS[#1] TO WR[1]                      ; ожидаемые данные =1
U 25CC,  0869,3C    ;11108              JSR [CHECK.RESULT]                       ; проверка операции NEG.LS
U 25CD,  8A5C,74    ;11109              JMP [LOOP.TF.6.1F8]                      ; зацикливание при ошибке, если разрешено
U 25CE,  8A70,1C    ;11110              JSR [INC.OTHER]                          ;
                    ;11111    LOOP.TF.6.1F9:
U 25CF,  B69E,15    ;11112              MOV LS[ONES] TO WR[0]                    ;
U 25D0,  BE60,15    ;11113              MOV WR[0] TO LS[L30]                     ; LS=-1
U 25D1,  7E60,15    ;11114              NEG LS[L30]                              ;
U 25D2,  3660,15    ;11115              MOV LS[L30] TO WR[0]                     ;
U 25D3,  3640,95    ;11116              MOV LS[#1] TO WR[1]                      ; ожидаемые данные =1
U 25D4,  0869,3C    ;11117              JSR [CHECK.RESULT]                       ; проверка операции NEG.LS
U 25D5,  0A5C,F4    ;11118              JMP [LOOP.TF.6.1F9]                      ; зацикливание при ошибке, если разрешено
U 25D6,  8A70,1C    ;11119              JSR [INC.OTHER]                          ;
                    ;11120    LOOP.TF.6.1FA:
```

```
U 25D7, B69E,15 ;11121           MOV LS[ONES] TO WR[0]                  ;
U 25D8, BEA6,15 ;11122           MOV WR[0] TO LS[L53]                   ; LS=-1
U 25D9, 7EA6,15 ;11123           NEG LS[L53]                            ;
U 25DA, 36A6,15 ;11124           MOV LS[L53] TO WR[0]                   ;
U 25DB, 3640,95 ;11125           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 25DC, 0869,3C ;11126           JSR [CHECK.RESULT]                     ; проверка операции NEG.LS
U 25DD, 0A5D,74 ;11127           JMP [LOOP.TF.6.1FA]                    ; зацикливание при ошибке, если разрешено
U 25DE, 8A70,1C ;11128           JSR [INC.OTHER]                        ;
                ;11129   LOOP.TF.6.1F8:
U 25DF, B69E,15 ;11130           MOV LS[ONES] TO WR[0]                  ;
U 25E0, 3EE6,15 ;11131           MOV WR[0] TO LS[L73]                   ; LS=-1
U 25E1, FEE6,15 ;11132           NEG LS[L73]                            ;
U 25E2, B6E6,15 ;11133           MOV LS[L73] TO WR[0]                   ;
U 25E3, 3640,95 ;11134           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 25E4, 0869,3C ;11135           JSR [CHECK.RESULT]                     ; проверка операции NEG.LS
U 25E5, 8A5D,F4 ;11136           JMP [LOOP.TF.6.1FB]                    ; зацикливание при ошибке, если разрешено
U 25E6, 8A70,1C ;11137           JSR [INC.OTHER]                        ;
                ;11138   LOOP.TF.6.1FC:
U 25E7, 2F80,15 ;11139           CLR WR[0]                              ;
U 25E8, 3E10,15 ;11140           MOV WR[0] TO LS[T8]                    ; LS=0
U 25E9, 7F10,15  11141           INC LS[T8]                             ;
U 25EA, B610,15 ;11142           MOV LS[T8] TO WR[0]                    ;
U 25EB, 3640,95 ;11143           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 25EC, 0869,3C ;11144           JSR [CHECK.RESULT]                     ; проверка операции INC.LS
U 25ED, 0A5E,74 ;11145           JMP [LOOP.TF.6.1FC]                    ; зацикливание при ошибке, если разрешено
U 25EE, 8A70,1C ;11146           JSR [INC.OTHER]                        ;
                ;11147   LOOP.TF.6.1FD:
U 25EF, 2F80,15 ;11148           CLR WR[0]                              ;
U 25F0, BE60,15 ;11149           MOV WR[0] TO LS[L30]                   ; LS=0
U 25F1, FF60,15 ;11150           INC LS[L30]                            ;
U 25F2, 3660,15 ;11151           MOV LS[L30] TO WR[0]                   ;
U 25F3, 3640,95 ;11152           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 25F4, 0869,3C ;11153           JSR [CHECK.RESULT]                     ; проверка операции INC.LS
U 25F5, 8A5E,F4 ;11154           JMP [LOOP.TF.6.1FD]                    ; зацикливание при ошибке, если разрешено
U 25F6, 8A70,1C ;11155           JSR [INC.OTHER]                        ;
                ;11156   LOOP.TF.6.1FE:
U 25F7, 2F80,15 ;11157           CLR WR[0]                              ;
U 25F8, 8EA6,15 ;11158           MOV WR[0] TO LS[L53]                   ; LS=0
U 25F9, FFA6,15 ;11159           INC LS[L53]                            ;
U 25FA, 36A6,15 ;11160           MOV LS[L53] TO WR[0]                   ;
U 25FB, 3640,95 ;11161           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 25FC, 0869,3C ;11162           JSR [CHECK.RESULT]                     ; проверка операции INC.LS
U 25FD, 8A5F,74 ;11163           JMP [LOOP.TF.6.1FE]                    ; зацикливание при ошибке, если разрешено
U 25FE, 8A70,1C ;11164           JSR [INC.OTHER]                        ;
                ;11165   LOOP.TF.6.1FF:
U 25FF, 2F80,15 ;11166           CLR WR[0]                              ;
U 2600, 3EE6,15 ;11167           MOV WR[0] TO LS[L73]                   ; LS=0
U 2601, 7FE6,15 ;11168           INC LS[L73]                            ;
U 2602, B6E6,15 ;11169           MOV LS[L73] TO WR[0]                   ;
U 2603, 3640,95 ;11170           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 2604, 0869,3C ;11171           JSR [CHECK.RESULT]                     ; проверка операции INC.LS
U 2605, 0A5F,F4 ;11172           JMP [LOOP.TF.6.1FF]                    ; зацикливание при ошибке, если разрешено
                ;11173    ;
                ;11174    ; ТЕСТЫ ИНСТРУКЦИЙ MOVE
                ;11175    ;
```

```
                ;11176    ;   Инструкции MOVE проверяются в восьми группах, т. е. каждая инструкция прове-
                ;11177    ;   ряется в восьми разных областях местной памяти: 0-1F, 20-3F, 40-5F, 60-7F,
                ;11178    ;   80-9F, A0-BF, C0-DF и E0-FF. Используются восемь текущих ячеек:
                ;11179    ;       LS  8 -  T8
                ;11180    ;       LS 30 - #26
                ;11181    ;       LS 53 - #F
                ;11182    ;       LS 73 - #3F
                ;11183    ;       LS 90 - L90
                ;11184    ;       LS B0 - LB0
                ;11185    ;       LS D0 - LD0
                ;11186    ;   и   LS F0 - LF0
                ;11187    ;
                ;11188    ;   ПРИМЕЧАНИЕ: не все инструкции пересылки проверяются в этом модуле. Все пере-
                ;11189    ;   сылки, которые требуют предварительно проверенной памяти, будут проверяться
                ;11190    ;   в следующем модуле.
                ;11191    TF.7:
U 2606, 8870,0C ;11192            JSR [INC.EN]                      ; увеличение номера ошибки до 7
U 2607, 3720,15 ;11193            MOV LS[L90] TO WR[0]              ; сохранение LS 90, B0, D0 и Fv
U 2608, 3760,95 ;11194            MOV LS[LB0] TO WR[1]              ;
U 2609, 37A1,15 ;11195            MOV LS[LD0] TO WR[2]              ;
U 260A, 37E1,95 ;11196            MOV LS[LF0] TO WR[3]              ;
U 2608, BE1B,15 ;11197            MOV WR[0] TO LS[T12]              ;
U 260C, BE1A,95 ;11198            MOV WR[1] TO LS[T13]              ;
U 260D, BE1D,15 ;11199            MOV WR[2] TO LS[T14]              ;
U 260E, BE1F,95 ;11200            MOV WR[3] TO LS[T15]              ;
U 260F, 364E,15 ;11201            MOV LS[BIT7] TO WR[0]             ;
U 2610, C74C,15 ;11202            BIS LS[BIT6] TO WR[0]             ;
U 2611, 4748,15 ;11203            BIS LS[BIT4] TO WR[0]             ;
U 2612, C746,15 ;11204            BIS LS[BIT3] TO WR[0]             ;
U 2613, BE88,15 ;11205            MOV WR[0] TO LS[ADDRESS.DATA]     ; установка начального значения D8 в поле "ДРУГИЕ ДАННЫЕ"
                ;11206    LOOP.TF.7.D8:
U 2614, B698,15 ;11207            MOV LS[ALTER.ODD] TO WR[0]        ;
U 2615, 3E10,15 ;11208            MOV WR[0] TO LS[T8]               ; пересылка данных в LS
U 2616, 2F80,15 ;11209            CLR WR[0]                         ;
U 2617, B610,15 ;11210            MOV LS[T8] TO WR[0]               ;
U 2618, 3698,95 ;11211            MOV LS[ALTER.ODD] TO WR[1]        ; ожидаемые данные = те же
U 2619, 0869,3C ;11212            JSR [CHECK.RESULT]                ; проверка операции MOV.LS.WR
U 261A, 0A61,44 ;11213            JMP [LOOP.TF.7.D8]                ; зацикливание при ошибке, если разрешено
U 261B, 8A70,1C ;11214            JSR [INC.OTHER]                   ;
                ;11215    LOOP.TF.7.D9:
U 261C, B698,15 ;11216            MOV LS[ALTER.ODD] TO WR[0]        ;
U 261D, BE60,15 ;11217            MOV WR[0] TO LS[L30]              ; пересылка данных в LS
U 261E, 2F80,15 ;11218            CLR WR[0]                         ;
U 261F, 3660,15 ;11219            MOV LS[L30] TO WR[0]              ;
U 2620, 3698,95 ;11220            MOV LS[ALTER.ODD] TO WR[1]        ; ожидаемые данные = те же
U 2621, 0869,3C ;11221            JSR [CHECK.RESULT]                ; проверка операции MOV.LS.WR
U 2622, 8A61,C4 ;11222            JMP [LOOP.TF.7.D9]                ; зацикливание при ошибке, если разрешено
U 2623, 8A70,1C ;11223            JSR [INC.OTHER]                   ;
                ;11224    LOOP.TF.7.DA:
U 2624, B698,15 ;11225            MOV LS[ALTER.ODD] TO WR[0]        ;
U 2625, BEA6,15 ;11226            MOV WR[0] TO LS[L53]              ; пересылка даных в LS
U 2626, 2F80,15 ;11227            CLR WR[0]                         ;
U 2627, 36A6,15 ;11228            MOV LS[L53] TO WR[0]              ;
U 2628, 3698,95 ;11229            MOV LS[ALTER.ODD] TO WR[1]        ; ожидаемые данные = те же
U 2629, 0869,3C ;11230            JSR [CHECK.RESULT]                ; проверка операции MOV.LS.WR
```

```
U 262A, 0A62,44 ;11231          JMP [LOOP.TF.7.DA]                      ; зацикливание при ошибке, если разрешено
U 262B, 8A70,1C ;11232          JSR [INC.OTHER]                         ;
                ;11233  LOOP.TF.7.DB:
U 262C, B698,15 ;11234          MOV LS[ALTER.ODD] TO WR[0]              ;
U 262D, 3EE6,15 ;11235          MOV WR[0] TO LS[L73]                    ; пересылка данных в LS
U 262E, 2F80,15 ;11236          CLR WR[0]                               ;
U 262F, B6E6,15 ;11237          MOV LS[L73] TO WR[0]                    ;
U 2630, 3698,95 ;11238          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 2631, 0869,3C ;11239          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.WR
U 2632, 8A62,C4 ;11240          JMP [LOOP.TF.7.DB]                      ; зацикливание при ошибке, если разрешено
U 2633, 8A70,1C ;11241          JSR [INC.OTHER]                         ;
                ;11242  LOOP.TF.7.DC:
U 2634, B698,15 ;11243          MOV LS[ALTER.ODD] TO WR[0]              ;
U 2635, BF20,15 ;11244          MOV WR[0] TO LS[L90]                    ; пересылка данных в LS
U 2636, 2F80,15 ;11245          CLR WR[0]                               ;
U 2637, 3720,15 ;11246          MOV LS[L90] TO WR[0]                    ;
U 2638, 3698,95 ;11247          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 2639, 0869,3C ;11248          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.WR
U 263A, 8A63,44 ;11249          JMP [LOOP.TF.7.DC]                      ; зацикливание при ошибке, если разрешено
U 263B, 8A70,1C ;11250          JSR [INC.OTHER]                         ;
                ;11251  LOOP.TF.7.DD:
U 263C, B698,15 ;11252          MOV LS[ALTER.ODD] TO WR[0]              ;
U 263D, 3F60,15 ;11253          MOV WR[0] TO LS[LB0]                    ; пересылка данных в LS
U 263E, 2F80,15 ;11254          CLR WR[0]                               ;
U 263F, B760,15 ;11255          MOV LS[LB0] TO WR[0]                    ;
U 2640, 3698,95 ;11256          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 2641, 0869,3C ;11257          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.WR
U 2642, 0A63,C4 ;11258          JMP [LOOP.TF.7.DD]                      ; зацикливание при ошибке, если разрешено
U 2643, 8A70,1C ;11259          JSR [INC.OTHER]                         ;
                ;11260  LOOP.TF.7.DE:
U 2644, B698,15 ;11261          MOV LS[ALTER.ODD] TO WR[0]              ;
U 2645, 3FA0,15 ;11262          MOV WR[0] TO LS[LD0]                    ; пересылка данных в LS
U 2646, 2F80,15 ;11263          CLR WR[0]                               ;
U 2647, B7A0,15 ;11264          MOV LS[LD0] TO WR[0]                    ;
U 2648, 3698,95 ;11265          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 2649, 0869,3C ;11266          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.WR
U 264A, 0A64,44 ;11267          JMP [LOOP.TF.7.DE]                      ; зацикливание при ошибке, если разрешено
U 264B, 8A70,1C ;11268          JSR [INC.OTHER]
                ;11269  LOOP.TF.7.DF:
U 264C, B698,15 ;11270          MOV LS[ALTER.ODD] TO WR[0]              ;
U 264D, BFE0,15 ;11271          MOV WR[0] TO LS[LF0]                    ; пересылка данных в LS
U 264E, 2F80,15 ;11272          CLR WR[0]                               ;
U 264F, 37E0,15 ;11273          MOV LS[LF0] TO WR[0]                    ;
U 2650, 3698,95 ;11274          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 2651, 0869,3C ;11275          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.WR
U 2652, 8A64,C4 ;11276          JMP [LOOP.TF.7.DF]                      ; зацикливание при  шибке, если разрешено
U 2653, 8A70,1C ;11277          JSR [INC.OTHER]                         ;
U 2654, 364E,15 ;11278          MOV LS[BIT7] TO WR[0]                   ;
U 2655, C74C,15 ;11279          BIS LS[BIT6] TO WR[0]                   ;
U 2656, C74A,15 ;11280          BIS LS[BIT5] TO WR[0]                   ;
U 2657, 4748,15 ;11281          BIS LS[BIT4] TO WR[0]                   ;
U 2658, BE88,15 ;11282          MOV WR[0] TO LS[ADDRESS.DATA]           ; установка начального значения F0 в поле "ДРУГИЕ ДАННЫЕ"
                ;11283  LOOP.TF.7.F0:
U 2659, 3642,15 ;11284          MOV LS[#2] TO WR[0]                     ;
U 265A, 3EF8,15 ;11285          MOV WR[0] TO LS[OS]                     ; OS=2
```

```
U 2658, 369F,15 ;11286          MOV LS[ONES] TO WR[2]                   ; WR2=1
U 265C, 3C11,15 ;11287          ADD OS PLUS WR[2] TO LS[T8]             ;
U 265D, B610,15 ;11288          MOV LS[T8] TO WR[0]                     ;
U 265E, 3640,95 ;11289          MOV LS[#1] TO WR[1]                     ; ожидаемые данные =1
U 265F, 0869,3C ;11290          JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.OS
U 2660, 0A65,94 ;11291          JMP [LOOP.TF.7.F0]                      ; зацикливание при ошибке, если разрешено
U 2661, 8A70,1C ;11292          JSR [INC.OTHER]                         ;
                ;11293  LOOP.TF.7.F1:
U 2662, 3642,15 ;11294          MOV LS[#2] TO WR[0]                     ;
U 2663, 3EF8,15 ;11295          MOV WR[0] TO LS[OS]                     ; OS=2
U 2664, 369F,15 ;11296          MOV LS[ONES] TO WR[2]                   ; WR2=1
U 2665, BC61,15 ;11297          ADD OS PLUS WR[2] TO LS[L30]            ;
U 2666, 3660,15 ;11298          MOV LS[L30] TO WR[0]                    ;
U 2667, 3640,95 ;11299          MOV LS[#1] TO WR[1]                     ; ожидаемые данные =1
U 2668, 0869,3C ;11300          JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.OS
U 2669, 8A66,24 ;11301          JMP [LOOP.TF.7.F1]                      ; зацикливание при ошибке, если разрешено
U 266A, 8A70,1C ;11302          JSR [INC.OTHER]                         ;
                ;11303  LOOP.TF.7.F2:
U 266B, 3642,15 ;11304          MOV LS[#2] TO WR[0]                     ;
U 266C, 3EF8,15 ;11305          MOV WR[0] TO LS[OS]                     ; OS=2
U 266D, 369F,15 ;11306          MOV LS[ONES] TO WR[2]                   ; WR2=-1
U 266E, BCA7,15 ;11307          ADD OS PLUS WR[2] TO LS[L53]            ;
U 266F, 36A6,15 ;11308          MOV LS[L53] TO WR[0]                    ;
U 2670, 3640,95 ;11309          MOV LS[#1] TO WR[1]                     ; ожидаемые данные =1
U 2671, 0869,3C ;11310          JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.OS
U 2672, 8A66,84 ;11311          JMP [LOOP.TF.7.F2]                      ; зацикливание при ошибке, если разрешено
U 2673, 8A70,1C ;11312          JSR [INC.OTHER]                         ;
                ;11313  LOOP.TF.7.F3:
U 2674, 3642,15 ;11314          MOV LS[#2] TO WR[0]                     ;
U 2675, 3EF8,15 ;11315          MOV WR[0] TO LS[OS]                     ; OS=2
U 2676, 369F,15 ;11316          MOV LS[ONES] TO WR[2]                   ; WR2=-1
U 2677, 3CE7,15 ;11317          ADD OS PLUS WR[2] TO LS[L73]            ;
U 2678, B6E6,15 ;11318          MOV LS[L73] TO WR[0]                    ;
U 2679, 3640,95 ;11319          MOV LS[#1] TO WR[1]                     ; ожидаемые данные =1
U 267A, 0869,3C ;11320          JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.OS
U 267B, 0A67,44 ;11321          JMP [LOOP.TF.7.F3]                      ; зацикливание при ошибке, если разрешено
U 267C, 8A70,1C ;11322          JSR [INC.OTHER]                         ;
U 267D, B62C,15 ;11323          MOV LS[#FFFFFF00] TO WR[0]              ;
U 267E, 3E8A,15 ;11324          MOV WR[0] TO LS[ERROR.MASK]             ; игнорирование расширения знака для регистра OS
                ;11325  LOOP.TF.7.F4:
U 267F, 3642,15 ;11326          MOV LS[#2] TO WR[0]                     ;
U 2680, 3EF8,15 ;11327          MOV WR[0] TO LS[OS]                     ; OS=2
U 2681, 369F,15 ;11328          MOV LS[ONES] TO WR[2]                   ; WR2=-1
U 2682, BD21,15 ;11329          ADD OS PLUS WR[2] TO LS[L90]            ;
U 2683, 3720,15 ;11330          MOV LS[L90] TO WR[0]                    ;
U 2684, 3640,95 ;11331          MOV LS[#1] TO WR[1]                     ; ожидаемые данные =1
U 2685, 0869,3C ;11332          JSR [CHECK.RESULT]                      ; проверка операции WR.PLUS.OS
U 2686, 8A67,F4 ;11333          JMP [LOOP.TF.7.F4]                      ; зацикливание при ошибке, если разрешено
U 2687, 8A70,1C ;11334          JSR [INC.OTHER]                         ;
                ;11335  LOOP.TF.7.F5:
U 2688, 3642,15 ;11336          MOV LS[#2] TO WR[0]                     ;
U 2689, 3EF8,15 ;11337          MOV WR[0] TO LS[OS]                     ; OS=2
U 268A, 369F,15 ;11338          MOV LS[ONES] TO WR[2]                   ; WR2=-1
U 268B, 3D61,15 ;11339          ADD OS PLUS WR[2] TO LS[LB0]            ;
U 268C, B760,15 ;11340          MOV LS[LB0] TO WR[0]                    ;
```

```
U 268D,  3640,95 ;11341           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 268E,  0869,3C ;11342           JSR [CHECK.RESULT]                     ; проверка операции WR.PLUS.OS
U 268F,  0A68,84 ;11343           JMP [LOOP.TF.7.F5]                     ; зацикливание при ошибке, если разрешено
U 2690,  8A70,1C ;11344           JSR [INC.OTHER]                        ;
                 ;11345   LOOP.TF.7.F6:
U 2691,  3642,15 ;11346           MOV LS[#2] TO WR[0]                    ;
U 2692,  3EF8,15 ;11347           MOV WR[0] TO LS[OS]                    ; OS=2
U 2693,  369F,15 ;11348           MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 2694,  3DA1,15 ;11349           ADD OS PLUS WR[2] TO LS[LD0]           ;
U 2695,  B7A0,15 ;11350           MOV LS[LD0] TO WR[0]                   ;
U 2696,  3640,95 ;11351           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 2697,  0869,3C ;11352           JSR [CHECK.RESULT]                     ; проверка операции WR.PLUS.OS
U 2698,  8A69,14 ;11353           JMP [LOOP.TF.7.F6]                     ; зацикливание при ошибке, если разрешено
U 2699,  8A70,1C ;11354           JSR [INC.OTHER]                        ;
                 ;11355   LOOP.TF.7.F7:
U 269A,  3642,15 ;11356           MOV LS[#2] TO WR[0]                    ;
U 269B,  3EF8,15 ;11357           MOV WR[0] TO LS[OS]                    ; OS=2
U 269C,  369F,15 ;11358           MOV LS[ONES] TO WR[2]                  ; WR2=-1
U 269D,  BDE1,15 ;11359           ADD OS PLUS WR[2] TO LS[LF0]           ;
U 269E,  37E0,15 ;11360           MOV LS[LF0] TO WR[0]                   ;
U 269F,  3640,95  11361           MOV LS[#1] TO WR[1]                    ; ожидаемые данные =1
U 26A0,  0869,3C ;11362           JSR [CHECK.RESULT]                     ; проверка операции WR PLUS.OS
U 26A1,  0A69,A4 ;11363           JMP [LOOP.TF.7.F7]                     ; зацикливание при ошибке, если разрешено
U 26A2,  8A70,1C ;11364           JSR [INC.OTHER]                        ;
U 26A3,  E58A,15 ;11365           CLR LS[ERROR.MASK]                     ; восстановление маски
                 ;11366   LOOP.TF.7.F8:
U 26A4,  E510,15 ;11367           CLR LS[T8]                             ;
U 26A5,  3699,15 ;11368           MOV LS[ALTER.ODD] TO WR[2]             ; пересылка данных в WR
U 26A6,  8E11,15 ;11369           MOV WR[2] TO LS[T8]                    ;
U 26A7,  8610,15 ;11370           MOV LS[T8] TO WR[0]                    ;
U 26A8,  3698,95 ;11371           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные = те же
U 26A9,  0869,3C ;11372           JSR [CHECK.RESULT]                     ; проверка операции MOV.WR.LS
U 26AA,  8A6A,44 ;11373           JMP [LOOP.TF.7.F8]                     ; зацикливание при ошибке, если разрешено
U 26AB,  8A70,1C ;11374           JSR [INC.OTHER]                        ;
                 ;11375   LOOP.TF.7.F9:
U 26AC,  6560,15 ;11376           CLR LS[L30]                            ;
U 26AD,  3699,15 ;11377           MOV LS[ALTER.ODD] TO WR[2]             ; пересылка данных в WR
U 26AE,  3E61,15 ;11378           MOV WR[2] TO LS[L30]                   ;
U 26AF,  3660,15 ;11379           MOV LS[L30] TO WR[0]                   ;
U 26B0,  3698,95 ;11380           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные = те же
U 26B1,  0869,3C ;11381           JSR [CHECK.RESULT]                     ; проверка операции MOV.WR.LS
U 26B2,  0A6A,C4 ;11382           JMP [LOOP.TF.7.F9]                     ; зацикливание при ошибке, если разрешено
U 26B3,  8A70,1C ;11383           JSR [INC.OTHER]                        ;
                 ;11384   LOOP.TF.7.FA:
U 26B4,  65A6,15 ;11385           CLR LS[L53]                            ;
U 26B5,  3699,15 ;11386           MOV LS[ALTER.ODD] TO WR[2]             ; пересылка данных в WR
U 26B6,  3EA7,15 ;11387           MOV WR[2] TO LS[L53]                   ;
U 26B7,  36A6,15 ;11388           MOV LS[L53] TO WR[0]                   ;
U 26B8,  3698,95 ;11389           MOV LS[ALTER.ODD] TO WR[1]             ; ожидаемые данные = те же
U 26B9,  0869,3C ;11390           JSR [CHECK.RESULT]                     ; проверка операции MOV.WR.LS
U 26BA,  0A68,44 ;11391           JMP [LOOP.TF.7.FA]                     ; зацикливание при ошибке, если разрешено
U 26BB,  8A70,1C ;11392           JSR [INC.OTHER]                        ;
                 ;11393   LOOP.TF.7.FB:
U 26BC,  E5E6,15 ;11394           CLR LS[L73]                            ;
U 26BD,  3699 15 ;11395           MOV LS[ALTER.ODD] TO WR[2]             ; пересылка данных в WR
```

```
U 26BE,  BEE7,15 ;11396           MOV WR[2] TO LS[L73]                    ;
U 26BF,  B6E6,15 ;11397           MOV LS[L73] TO WR[0]                    ;
U 26C0,  3698,95 ;11398           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 26C1,  0869,3C ;11399           JSR [CHECK.RESULT]                      ; проверка операции MOV.WR.LS
U 26C2,  8A68,C4 ;11400           JMP [LOOP.TF.7.FB]                      ; зацикливание при ошибке, если разрешено
U 26C3,  8A70,1C ;11401           JSR [INC.OTHER]                         ;
                 ;11402   LOOP.TF.7.FC:
U 26C4,  2F85,15 ;11403           CLR WR[2]                               ;
U 26C5,  3F21,15 ;11404           MOV WR[2] TO LS[L90]                    ;
U 26C6,  3699,15 ;11405           MOV LS[ALTER.ODD] TO WR[2]              ; пересылка данных в WR
U 26C7,  3F21,15 ;11406           MOV WR[2] TO LS[L90]                    ;
U 26C8,  3720,15 ;11407           MOV LS[L90] TO WR[0]                    ;
U 26C9,  3698,95 ;11408           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 26CA,  0869,3C ;11409           JSR [CHECK.RESULT]                      ; проверка операции MOV.WR.LS
U 26CB,  8A6C,44 ;11410           JMP [LOOP.TF.7.FC]                      ; зацикливание при ошибке, если разрешено
U 26CC,  8A70,1C ;11411           JSR [INC.OTHER]                         ;
                 ;11412   LOOP.TF.7.FD:
U 26CD,  2F85,15 ;11413           CLR WR[2]                               ;
U 26CE,  8F61,15 ;11414           MOV WR[2] TO LS[LB0]                    ;
U 26CF,  3699,15 ;11415           MOV LS[ALTER.ODD] TO WR[2]              ; пересылка данных в WR
U 26D0,  8F61,15 ;11416           MOV WR[2] TO LS[LB0]                    ;
U 26D1,  B760,15 ;11417           MOV LS[LB0] TO WR[0]                    ;
U 26D2,  3698,95 ;11418           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 26D3,  0869,3C ;11419           JSR [CHECK.RESULT]                      ; проверка операции MOV.WR.LS
U 26D4,  8A6C,D4 ;11420           JMP [LOOP.TF.7.FD]                      ; зацикливание при ошибке, если разрешено
U 26D5,  8A70,1C ;11421           JSR [INC.OTHER]                         ;
                 ;11422   LOOP.TF.7.FE:
U 26D6,  2F85,15 ;11423           CLR WR[2]                               ;
U 26D7,  BFA1,15 ;11424           MOV WR[2] TO LS[LD0]                    ;
U 26D8,  3699,15 ;11425           MOV LS[ALTER.ODD] TO WR[2]              ; пересылка данных в WR
U 26D9,  BFA1,15 ;11426           MOV WR[2] TO LS[LD0]                    ;
U 26DA,  B7A0,15 ;11427           MOV LS[LD0] TO WR[0]                    ;
U 26DB,  3698,95 ;11428           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 26DC,  0869,3C ;11429           JSR [CHECK.RESULT]                      ; проверка операции MOV.WR.LS
U 26DD,  8A6D,64 ;11430           JMP [LOOP.TF.7.FE]                      ; зацикливание при ошибке, если разрешено
U 26DE,  8A70,1C ;11431           JSR [INC.OTHER]                         ;
                 ;11432   LOOP.TF.7.FF:
U 26DF,  2F85,15 ;11433           CLR WR[2]                               ;
U 26E0,  3FE1,15 ;11434           MOV WR[2] TO LS[LF0]                    ;
U 26E1,  3699,15 ;11435           MOV LS[ALTER.ODD] TO WR[2]              ; пересылка данных в WR
U 26E2,  3FE1,15 ;11436           MOV WR[2] TO LS[LF0]                    ;
U 26E3,  37E0,15 ;11437           MOV LS[LF0] TO WR[0]                    ;
U 26E4,  3698,95 ;11438           MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = те же
U 26E5,  0869,3C ;11439           JSR [CHECK.RESULT]                      ; проверка операции MOV.WR.LS
U 26E6,  8A6D,F4 ;11440           JMP [LOOP.TF.7.FF]                      ; зацикливание при ошибке, если разрешено
                 ;11441   TF.RESTORE.LS:
U 26E7,  3612,15 ;11442           MOV LS[T9] TO WR[0]                     ; восстановление ячеек LS, используемых в частях 6 и 7
U 26E8,  B614,95 ;11443           MOV LS[T10] TO WR[1]                    ;
U 26E9,  3617,15 ;11444           MOV LS[T11] TO WR[2]                    ;
U 26EA,  BE60,15 ;11445           MOV WR[0] TO LS[L30]                    ;
U 26EB,  3EA6,95 ;11446           MOV WR[1] TO LS[L53]                    ;
U 26EC,  BEE7,15 ;11447           MOV WR[2] TO LS[L73]
U 26ED,  3618,15 ;11448           MOV LS[T12] TO WR[0]
U 26EE,  361A,95 ;11449           MOV LS[T13] TO WR[1]
U 26EF,  361D,15 ;11450           MOV LS[T14] TO WR[2]                    ;
```

```
U 26F0,  361F,95 ;11451           MOV LS[T15] TO WR[3]                    ;
U 26F1,  BF20,15 ;11452           MOV WR[0] TO LS[L90]                    ;
U 26F2,  BF60,95 ;11453           MOV WR[1] TO LS[LBQ]                    ;
U 26F3,  BFA1,15 ;11454           MOV WR[2] TO LS[LD0]                    ;
U 26F4,  BFE1,95 ;11455           MOV WR[3] TO LS[LF0]                    ;
U 26F5,  8A70,54 ;11456           JMP [END.TF]                            ;
                 ;11457   CHECK.WR3:
U 26F6,  A006,15 ;11458           MOV WR[3] TO WR[0]                      ;
U 26F7,  B699,95 ;11459           MOV LS[ALTER.ODD] TO WR[3]              ; восстановление WR3 для случая, если была запись
U 26F8,  2006,95 ;11460           MOV WR[3] TO WR[1]                      ; ожидаемые данные
U 26F9,  0869,3C ;11461           JSR [CHECK.RESULT]                      ; проверка
U 26FA,  5B00,14 ;11462           RETURN                                  ; зацикливание при ошибке, если разрешено
U 26FB,  DB00,16 ;11463           RETURN+1                                ; нормальный возврат
                 ;11464   CHECK.WR0:
U 26FC,  B69A,95 ;11465           MOV LS[ALTER.EVEN] TO WR[1]             ; ожидаемые данные
U 26FD,  0869,3C ;11466           JSR [CHECK.RESULT]                      ; проверка
U 26FE,  5B00,14 ;11467           RETURN                                  ; зацикливание при ошибке, если разрешено
U 26FF,  369A,15 ;11468           MOV LS[ALTER.EVEN] TO WR[0]             ; восстановление WR0 для случая, если была запись
U 2700,  DB00,16 ;11469           RETURN+1                                ;
                 ;11470   INC.OTHER:
U 2701,  B688,95 ;11471           MOV LS[ADDRESS.DATA] TO WR[1]           ;
U 2702,  2042,95 ;11472           INC WR[1]                               ;
U 2703,  3E88,95 ;11473           MOV WR[1] TO LS[ADDRESS.DATA]           ; увеличение "ДРУГИХ" данных
U 2704,  5B00,14 ;11474           RETURN                                  ;
                 ;11475   END.TF:
                 ;11476   END.SECTION:
U 2705,  365C,15 ;11477           MOV LS[BIT14] TO WR[0]                  ; установка WR0 для конца сегмента
U 2706,  3E80,15 ;11478           MOV WR[0] TO LS[CONTROL.STATUS]         ; установка слова управления и состояния для сообщения
                 ;11479                                                   ; конца сегмента
U 2707,  10E0,15 ;11480           MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для консольного процессора
                 ;11481   WAIT.EOS:
U 2708,  0A70,84 ;11482           JMP [WAIT.EOS]                          ; зацикливание для ожидания ответа консольного
                 ;11483                                                   ; процессора
                 ;11484
                 ;11485
```

Лист регистрации изменений

Изм	Номера листов(стр.) изме-ненных	новых	аннулированных	Всего листов в док.	N док.	Входящий N сопр. док. и дата	Подпись	Дата