КОМПЛЕКС ВЫЧИСЛИТЕЛЬНЫЙ СМ I700

Заводской № 0312 Год выпуска 1989

СМДО СМ I700

Подсистема процессора СМ 2700.2400

Микропрограммные тесты процессора СМ 2700.2400

Часть 3

Текст программы

00078-01 I2 03-3

Книга II

Утвержден

00076-01 12 03-1-ЛУ

СИСТЕМА МИКРОДИАГНОСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА СМ1700

СМДО СМ 1700

ПОДСИСТЕМА ПРОЦЕССОРА СМ2700.2400
Микропрограммные тесты процессора СМ 2700.2400

Текст программы
Часть 3

00076-01 12 03-3

Листов 211

1987

Перв.примен.
00076-01

Литера

АННОТАЦИЯ

Настоящий документ содержит тексты программной секции ENKCD, входящей в систему микродиагностического обеспечения вычислительного комплекса СМ 1700 - СМДО СМ1700 и реализующей часть 3 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 ENKCD.EXE, листинг трансляции - ENKCD.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            (разны)       +--------+-+---+-------+-----+-+-+---+-----------+
.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
```

```
.PAGE "ОПРЕДЕЛЕНИЯ ПОЛЕЙ ОСНОВНОГО МИКРОСЛОВА"
;
;    Следующие выражения используются при контроле истинности комбинаций полей
;
.SET/A.VAL=<.OR[<.EQL[<OPC2/>,4]>,<.EQL[<OPC2/>,5]>]>
.SET/B.VAL=<.AND[<.NEQ[<OPC2/>,0]>,<.NEQ[<OPC2/>,1]>,<.NEQ[<OPC2/>,2]>,
<.NEQ[<OPC2/>,3]>]>
.SET/D.VAL=<.AND[<.NEQ[<OPC2/>,0]>,<.NEQ[<OPC2/>,1]>,<.NEQ[<OPC2/>,2]>,
<.NEQ[<OPC2/>,4]>,<.NEQ[<OPC2/>,5]>,<.NEQ[<OPC2/>,6]>,
<.NEQ[<OPC2/>,7]>]>
.SET/XD.VAL=<.EQL[<OPC1/>,<OPC1/MOVE>]>
.SET/CC.VAL=<.AND[<.NEQ[<OPC2/>,0]>,<.NEQ[<OPC2/>,2]>,<.NEQ[<OPC2/>,3]>]>
.SET/DT.VAL=<.EQL[<OPC2/>,<OPC2/MEM.REQ>]>
.SET/SCTL.VAL=<.AND[<.NEQ[<OPC2/>,0]>,<.NEQ[<OPC2/>,1]>]>
.SET/JCTL.VAL=<.OR[<.EQL[<OPC2/>,0]>,<.EQL[<OPC2/>,1]>]>
.SET/JUMP.VAL=<.EQL[<OPC2/>,<OPC2/JUMP>]>
.SET/DECODE.VAL=<.EQL[<OPC2/>,<OPC2/DECODE>]>
.SET/MISC.VAL=<.EQL[<OPC2/>,<OPC2/MISC>]>
.SET/DP.VAL=<.EQL[<OPC/>,<OPC/BASIC>]>
.SET/PAGE.PROB=<.EQL[<.AND[<7FF>,<.>]>,<7FE>]>
.SET/SKIP.LEGAL2=<.OR[<.EQL[<SCTL/>,<SCTL/RET.NOERR>]>,<.EQL[<SCTL/>
<SCTL/POP.USTACK>]>]>
.SET/SKIP.LEGAL=<.OR[<.EQL[<SCTL/>,<SCTL/NO.SKIP>]>,<.EQL[<SCTL/>,
<SCTL/RETURN>]>,<.EQL[<SCTL/>,<SCTL/RETURN+1>]>,<SKIP.LEGAL2>]>
.SET/SKIP.PAGE.VAL=<.SELECT[<.EQL[<PAGE.PROB>,0]>,<1>,<.EQL[<PAGE.PROB>,1]>,
<SKIP.LEGAL>]>
.SET/JUMP.CHECK=<.OR[<.EQL[<JCTL/>,<JCTL/JSR>]>,
<.EQL[<JCTL/>,
<JCTL/JSR.VALID>]>]>
.SET/JUMP.PAGE.VAL=<.SELECT[<.EQL[<PAGE.PROB>,0]>,<1>,
<.EQL[<PAGE.PROB>,1]>,<.XOR[1,<JUMP.CHECK>]>]>

    Определения кодов операций микроинструкций

OPC/=<22>,.DEFAULT=<OPC/BASIC>
        BASIC=1             ; микроинструкция BASIC

OPC1/=<22:20>
        EXTENDED=2          ; микроинструкция EXTENDED
        MOVE=3              ; микроинструкция MOVE

OPC2/=<22:19>
        MEM.REQ=3           ; микроинструкция MEM.REQ
        MISC=2              ; микроинструкция MISC
        JUMP=1              ; микроинструкция JUMP
        DECODE=0            ; микроинструкция DECODE

          Определения адресных полей в микроинструкциях:

    A.ADRS - обращение к внутренней озу микропроцессора K1804BC1 через порт A
    B.ADRS - обращение к внутренней озу микропроцессора K1804BC1 через порт B
   XB.ADRS      "          "              "                "           "
;
A.ADRS/=<10:9>,.VALIDITY=<A.VAL>
        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])
;118
;119     ;
;120     ;        Поля управления путями данных
;121     ;
;122     ;  Управляющее поле путей данных микроинструкции BASIC
;123     ;
;124     DP/=<21:16>,.VALIDITY=<DP.VAL>,.DEFAULT=<.SHIFTL.ANDL<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>1>,.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
;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=<.ANDI[<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=<.CASEI[<.EQLI[<OPC2/>,<OPC2/DECODE>]>]OFI[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/=<8>,.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/=<18>
;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 Т-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 ]
;430    ;
;431    ;  WORD.SWAP         - выполняет 15-битовый циклический сдвиг, поэтому ответ должен
;432    ;                      корректироваться действием ROL WRD ]
;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=1B         ; 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.RB0=27            ; печать,имеется или нет RB0 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 К
;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 нарашивания 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.GNT.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            #0002(H)=65             ; константа для указателя переноса данных в LS
;843            BR7.IDENT=66            ; идентификатор BR7 (1010 (двоичн.) в битах 5-2)
;844            BC7=67                  ; адрес BC7 (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         TB=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=1B                ; 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 K
;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.BSSY=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=48         ; 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)=7B       ; 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            XQPR.OS=0FB            ; 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)=0FB    ; 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
;1295    ;
```

```
;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 KI804BC1. 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.AI
;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   ;                    обьединения 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. ФУНКЦИЕЙ 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 внутренее 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          0.Q=2                     ; R=0    S=Q
;1958          0.B=3                     ; R=0    S=B
;1959          A.Q=0                     ; R=A    S=Q
;1960          A.B=01                    ; R=A    S=B
;1961          D.Q=06                    ; R=D    S=Q
;1962          D.0=7                     ; R=D    S=0
;1963          0.A=4                     ; R=0    S=A
;1964          D.A=5                     ; 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
C 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 инстр ций
                 ;2067  ;  MISC не будут разрушены никакие данные внутри К1804ВС1
                 ;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
```

```
C 004E, 313      ;2085          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 004F, 313      ;2086          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0050, 313      ;2087          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0051, 313      ;2088          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0052, 313      ;2089          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0053, 313      ;2090          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0054, 313      ;2091          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0055, 313      ;2092          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0056, 313      ;2093          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0057, 313      ;2094          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0058, 313      ;2095          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 0059, 313      ;2096          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 005A, 313      ;2097          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 005B, 313      ;2098          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 005C, 313      ;2099          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 005D, 313      ;2100          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 005E, 313      ;2101          SRC/0.A,ALU/R.OR.S,DST/WRITE.B.A,CIN/CIN
C 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:
C 0060, 3DB      ;2112          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0061, 3DB      ;2113          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0062, 3DB      ;2114          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0063, 3DB      ;2115          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0064, 3DB      ;2116          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0065, 3DB      ;2117          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0066, 3DB      ;2118          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0067, 3DB      ;2119          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0068, 3DB      ;2120          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0069, 3DB      ;2121          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006A, 3DB      ;2122          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006B, 3DB      ;2123          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006C, 3DB      ;2124          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006D, 3DB      ;2125          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006E, 3DB      ;2126          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 006F, 3DB      ;2127          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0070, 3DB      ;2128          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0071, 3DB      ;2129          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0072, 3DB      ;2130          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0073, 3DB      ;2131          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0074, 3DB      ;2132          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0075, 3DB      ;2133          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0076, 3DB      ;2134          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0077, 3DB      ;2135          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0078, 3DB      ;2136          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 0079, 3DB      ;2137          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 007A, 3DB      ;2138          SRC/D.0,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
C 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, 10B      ;2159          SRC/0.A
                 ;2160  MOV.Q.WR:
C 0086, 29B      ;2161          SRC/0.Q,ALU/R.OR.S,DST/WRITE.B.F,CIN/CIN
                 ;2162  FILL87:
C 0087, 0BB      ;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, 04B      ;2169          SRC/A.B
                 ;2170  SHL2:
C 008B, 07B      ;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, 08B      ;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, 25B         ;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   LS.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, 3B4     ;2470          SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 0159, 3B4     ;2471          SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 015A, 3B4     ;2472          SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
C 015B, 3B4     ;2473          SRC/D.Q,ALU/R.AND.S,DST/LOAD.Q,CIN/CIN
                ;2474  LS.BIT.Q:
C 015C, 3BC     ;2475          SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015D, 3BC     ;2476          SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015E, 3BC     ;2477          SRC/D.Q,ALU/R.AND.S,DST/NOP,CIN/CIN
C 015F, 3BC     ;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
                ;2509  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, 34B             ;2591           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B5, 34B             ;2592           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B6, 34B             ;2593           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
C 01B7, 34B             ;2594           SRC/D.A,ALU/R.OR.S,DST/NOP,CIN/CIN
                        ;2595   WR.AND.LS:
C 01B8, 34C             ;2596           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01B9, 34C             ;2597           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01BA, 34C             ;2598           SRC/D.A,ALU/R.AND.S,DST/NOP,CIN/CIN
C 01BB, 34C             ;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, 0882,14     ;2697            JMP [T.1]                             ; переход к тесту 1
U 0002, 8888,F4     ;2698            JMP [T.2]                             ; переход к тесту 2
U 0003, 088D,D4     ;2699            JMP [T.3]                             ; переход к тесту 3
U 0004, 8893,54     ;2700            JMP [T.4]                             ; переход к тесту 4
U 0005, 08AD,34     ;2701            JMP [T.5]                             ; переход к тесту 5
U 0006, 08B5,F4     ;2702            JMP [T.6]                             ; переход к тесту 6
U 0007, 08BB,74     ;2703            JMP [T.7]                             ; переход к тесту 7
U 0008, 88C7,74     ;2704            JMP [T.8]                             ; переход к тесту 8
U 0009, 08CB,04     ;2705            JMP [T.9]                             ; переход к тесту 9
U 000A, 88D4,54     ;2706            JMP [T.A]                             ; переход к тесту A
U 000B, 88D8,34     ;2707            JMP [T.B]                             ; переход к тесту B
U 000C, 88DC,B4     ;2708            JMP [T.C]                             ; переход к тесту C
U 000D, 88DF,84     ;2709            JMP [T.D]                             ; переход к тесту D
U 000E, 08E1,84     ;2710            JMP [T.E]                             ; переход к тесту E
U 000F, 88EC,44     ;2711            JMP [T.F]                             ; переход к тесту F
U 0010, 08F2,94     ;2712            JMP [T.10]                            ; переход к тесту 10
U 0011, 886B,E4     ;2713            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2714                                                  ; в данном сегменте
U 0012, 886B,E4     ;2715            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2716                                                  ; в данном сегменте
U 0013, 886B,E4     ;2717            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2718                                                  ; в данном сегменте
U 0014, 886B,E4     ;2719            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2720                                                  ; в данном сегменте
U 0015, 886B,E4     ;2721            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2722                                                  ; в данном сегменте
U 0016, 886B,E4     ;2723            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2724                                                  ; в данном сегменте
U 0017, 886B,E4     ;2725            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2726                                                  ; в данном сегменте
U 0018, 886B,E4     ;2727            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2728                                                  ; в данном сегменте
U 0019, 886B,E4     ;2729            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2730                                                  ; в данном сегменте
U 001A, 886B,E4     ;2731            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2732                                                  ; в данном сегменте
U 001B, 886B,E4     ;2733            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2734                                                  ; в данном сегменте
U 001C, 886B,E4     ;2735            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2736                                                  ; в данном сегменте
U 001D, 886B,E4     ;2737            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2738                                                  ; в данном сегменте
U 001E, 886B,E4     ;2739            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2740                                                  ; в данном сегменте
U 001F, 886B,E4     ;2741            JMP [ERR.NO.TEST]                     ; переход к программе обработки ошибок. Тест отсутствует
                    ;2742                                                  ; в данном сегменте
```

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

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

```
                    ;2840      .PAGE  "ДИАГНОСТИЧЕСКИЕ УКАЗАТЕЛИ"
                    ;2841      ;
                    ;2842      ;   Этот раздел содержит все указатели, необходимые для данного диагностического
                    ;2843      ; сегмента. Первый указатель (по адресу WCS 0) указывает на программу переноса
                    ;2844      ; данных. Это первая ячейка, работающая после новой загрузки WCS. Инструкция, раз-
                    ;2845      ; мещенная здесь, является переходом (JUMP) к TRANSFER.POINT, где могут содержать-
                    ;2846      ; ся инструкции для переноса данных. Если отсутствуют данные, подлежащие переносу,
                    ;2847      ; или по завершении переноса данных, центральный процессор выставляет CPU ATTN,
                    ;2848      ; причем все биты слова управления и состояния очищены.
                    ;2849      ;   Следующие 80 ячеек (от ячейки WCS 1 до ячейки WCS 4F) содержат указатели для
                    ;2850      ; каждого теста в этом сегменте. Указателями являются инструкции JUMP к номеру
                    ;2851      ; теста, соответствующему номеру ячейки, т.е, ячейка 5 будет содержать переход к
                    ;2852      ; тесту 5. Все неиспользованные номера тестов содержат переход к программе обрабо-
                    ;2853      ; тки ошибок.
                    ;2854      ;   Наконец, следующие 32 ячейки (от ячейки WCS 50 до 6F) содержат указатели
                    ;2855      ; (инструкции JUMP) к подпрограммам WCS, которые подлежат использованию диагности-
                    ;2856      ; ческим монитором консольного процессора.
                    ;2857      ;
                    ;2858      0:
U 0000, 086C,24     ;2859            JMP [TRANSFER.POINT]                ; переход к подпрограмме, которая загружает LS 7
                    ;2860                                                ; указателем секции данных и выдает CPU ATTN с
                    ;2861                                                ; установленным битом переноса данных
                    ;2862      50:                                       ; начало указателей
                    ;2863                                                ; подпрограмм в ячейке WCS 50
                    ;2864      ;
                    ;2865      ;           Указатели подпрограмм
                    ;2866      ;
U 0050, 0860,84     ;2867            JMP [DEPOSIT.CSR]                   ; переход к подпрограмме WCS, которая выполняет запись в
                    ;2868                                                ; регистры управления и состояния MCT по команде DEPOSIT
                    ;2869                                                ; CSR
U 0051, 0860,D4     ;2870            JMP [EXAMINE.CSR]                   ; переход к подпрограмме WCS, выполняющей чтение
                    ;2871                                                ; регистров управления и состояния MCT по команде
                    ;2872                                                ; EXAMINE CSR
U 0052, 0861,F4     ;2873            JMP [DEPOSIT.TB]                    ; переход к подпрограмме WCS, выполняющей запись в буфер
                    ;2874                                                ; трансляции MCT по команде DEPOSIT для буфера
                    ;2875                                                ; трансляции
U 0053, 8863,A4     ;2876            JMP [EXAMINE.TB]                    ; переход к подпрограмме WCS, выполняющей чтение буфера
                    ;2877                                                ; трансляции MCT по команде EXAMINE для буфера
                    ;2878                                                ; трансляции
U 0054, 8865,C4     ;2879            JMP [DEPOSIT.MM]                    ; переход к подпрограмме WCS, которая записывает в
                    ;2880                                                ; основную память по команде DEPOSIT для основной
                    ;2881                                                ; памяти. Используется также для пересылки данных из
                    ;2882                                                ; WCS в основную память
U 0055, 0866,24     ;2883            JMP [EXAMINE.MM]                    ; переход к подпрограмме WCS, которая читает из основной
                    ;2884                                                ; памяти по команде EXAMINE для основной памяти
                    ;2885
U 0056, 0868,94     ;2886            JMP [SAVE.WR]                       ; переход к подпрограмме WCS, которая запоминает
                    ;2887                                                ; содержимое рабочих регистров WR0-WR3 в местной памяти
                    ;2888                                                ; LS 0-3
U 0057, 8868,E4     ;2889            JMP [RESTORE.WR]                    ; переход к подпрограмме WCS, которая восстанавливает
                    ;2890                                                ; содержимое WR0-WR3 из LS 0-3
U 0058, 8864,24     ;2891            JMP [DEPOSIT.UBS]                   ; переход к подпрограмме WCS, которая записывает в буфер
                    ;2892                                                ; трансляции общей шины контроллера памяти
U 0059, 0865,44     ;2893            JMP [EXAMINE.UBS]                   ; переход к подпрограмме WCS, которая читает из буфера
                    ;2894                                                ; трансляции общей шины контроллера памяти
```

```
U 005A, 0866,84 ;2895            JMP [EXAMINE.ICSR]          ; переход к подпрограмме WCS, которая читает CSR IDC
U 005B, 0866,B4 ;2896            JMP [EXAMINE.IDAR]          ; переход к подпрограмме WCS, которая читает DAR IDC
U 005C, 0866,E4 ;2897            JMP [EXAMINE.DBUF]          ; переход к подпрограмме WCS, которая читает DBUF IDC
U 005D, 8867,14 ;2898            JMP [EXAMINE.PATT]          ; переход к подпрограмме WCS, которая читает код
                ;2899                                        ; коррекции ECC из IDC
U 005E, 8867,44 ;2900            JMP [EXAMINE.POSIT]         ; переход к подпрограмме WCS, которая читает адрес
                ;2901                                        ; ошибки ECC из IDC
U 005F, 0867,94 ;2902            JMP [DEPOSIT.ICSR]          ; переход к подпрограмме WCS, которая записывает CSR
                ;2903                                        ; IDC
U 0060, 0867,C4 ;2904            JMP [DEPOSIT.IDAR]          ; переход к подпрограмме WCS, которая записывает DAR
                ;2905                                        ; IDC
U 0061, 0867,F4 ;2906            JMP [DEPOSIT.DBUF]          ; переход к подпрограмме WCS, которая записывает DBUF
                ;2907                                        ; IDC
U 0062, 8868,24 ;2908            JMP [CLEAR.FIFO.ADDR]       ; переход к подпрограмме WCS, которая очищает адрес FIFO
                ;2909                                        ; IDC
U 0063, 8868,44 ;2910            JMP [SELECT.FIFO.A]         ; переход к подпрограмме WCS, которая выбирает FIFO A
U 0064, 0868,64 ;2911            JMP [SELECT.FIFO.B]         ; переход к подпрограмме WCS, которая выбирает FIFO B
U 0065, DB00,15 ;2912            NOP                         ; не используется
U 0066, DB00,15 ;2913            NOP                         ; не используется
U 0067, DB00,15 ;2914            NOP                         ; не используется
U 0068, DB00,15 ;2915            NOP                         ; не используется
U 0069, DB00,15 ;2916            NOP                         ; не используется
U 006A, DB00,15 ;2917            NOP                         ; не используется
U 006B, DB00,15 ;2918            NOP                         ; не используется
U 006C, DB00,15 ;2919            NOP                         ; не используется
U 006D, DB00,15 ;2920            NOP                         ; не используется
U 006E, DB00,15 ;2921            NOP                         ; не используется
U 006F, DB00,15 ;2922            NOP                         ; не используется
                ;2923
                ;2924
                ;2925      .REGION/70,5FF
                ;2926      ;
```

```
                ;2927     .PAGE "СЕКЦИЯ ДАННЫХ МЕСТНОЙ ПАМЯТИ"
                ;2928     ;
                ;2929     ;       Этот раздел перечисляет данные, которые будут переданы в LS консольным
                ;2930     ;     процессором в качестве части процедуры инициации, выполняемой тестом 0.
                ;2931     ;     Программа переноса данных ссылается на эту область данных. LS содержит
                ;2932     ;     константы, слова управления и состояния и ячейки временного хранения, ис-
                ;2933     ;     пользуемые и подпрограммами микродиагностики.
                ;2934     ;       LS имеет ширину 32 бита. Каждое из слов здесь имеет ширину 16 битов,
                ;2935     ;     так, что два последовательных слова, перечисленные здесь, будут загружать-
                ;2936     ;     ся в каждую последовательную ячейку LS. Первое перечисленное слово будет
                ;2937     ;     занимать биты 0-15 ячейки LS. Второе перечисленное слово будет составлять
                ;2938     ;     биты 16-31 той же ячейки LS.
                ;2939     ;       Ячейки 78-7F первой половины LS и F8-FF второй половины LS не являются
                ;2940     ;     в действительности ячейками LS. Они задают доступ к одному из нескольких
                ;2941     ;     регистров центрального процессора или реализуют средство индексации других
                ;2942     ;     ячеек LS. По этой причине они не записываются посредством программы пере-
                ;2943     ;     носа данных.
                ;2944     ;       ПРИМЕЧАНИЕ: Эта таблица задана в шестнадцатеричном формате, т.е., каждый
                ;2945     ;     разряд представляется четырьмя битами, так, что четыре разряда представ-
                ;2946     ;     ляют полное 16-битовое слово. Микроассемблер требует, чтобы шестнадцате-
                ;2947     ;     ричному значению, которое начинается буквой (например, FFFF), предшество-
                ;2948     ;     вал 0 (0FFFF). Таким образом, некоторые значения в таблице содержат 5
                ;2949     ;     шестнадцатеричных разрядов. Пятый разряд в действительности не использу-
                 2950               ются
                ;2951
                ;2952     ;---
                ;2953     ;
                ;2954     TRANSFER.DATA.LS1:
                ;2955     ;
U 0070, 0000,78 ;2956           COUNT.LS[0078]                  ; счетчик длинных слов (число длинных слов, подлежащи)
                ;2957                                           ; пересылке в LS, равно 120, десятичн.). Приемником
                ;2958                                           ; является LS
U 0071, 0000,00 ;2959           START.ADR[0000]                 ; начальный адрес приемника (начинается при LS=0)
                ;2960     ;
U 0072, 0000,00 ;2961           WORD[0000]                      ; ячейка 0
U 0073, 0000,00 ;2962           WORD[0000]                      ;
U 0074, 0000,00 ;2963           WORD[0000]                      ; ячейка 1
U 0075, 0000,00 ;2964           WORD[0000]                      ;
U 0076, 0000,00 ;2965           WORD[0000]                      ; ячейка 2
U 0077, 0000,00 ;2966           WORD[0000]                      ;
U 0078, 0000,00 ;2967           WORD[0000]                      ; ячейка 3
U 0079, 0000,00 ;2968           WORD[0000]                      ;
U 007A, 0000,00 ;2969           WORD[0000]                      ; ячейка 4
U 007B, 0000,00 ;2970           WORD[0000]                      ;
U 007C, 0000,00 ;2971           WORD[0000]                      ; ячейка 5
U 007D, 0000,00 ;2972           WORD[0000]                      ;
U 007E, 0000,00 ;2973           WORD[0000]                      ; ячейка 6
U 007F, 0000,00 ;2974           WORD[0000]                      ;
U 0080, 0000,00 ;2975           WORD[0000]                      ; ячейка 7
U 0081, 0000,00 ;2976           WORD[0000]                      ;
U 0082, 0000,00 ;2977           WORD[0000]                      ; ячейка 8
U 0083, 0000,00 ;2978           WORD[0000]                      ;
U 0084, 0000,00 ;2979           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, 00B0,00 ;3015          WORD[B000]                          ; 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          WORD[0020]                          ; ячейка 25
U 00BD, 0000,00 ;3036          WORD[0000]                          ;
```

```
U 00BE, 0000,40 ;3037           WORD[0040]                          ; ячейка 26
U 00BF, 0000,00 ;3038           WORD[0000]                          ;
U 00C0, 0000,80 ;3039           WORD[0080]                          ; ячейка 27
U 00C1, 0000,00 ;3040           WORD[0000]                          ;
U 00C2, 0001,00 ;3041           WORD[0100]                          ; ячейка 28
U 00C3, 0000,00 ;3042           WORD[0000]                          ;
U 00C4, 0002,00 ;3043           WORD[0200]                          ; ячейка 29
U 00C5, 0000,00 ;3044           WORD[0000]                          ;
U 00C6, 0004,00 ;3045           WORD[0400]                          ; ячейка 2A
U 00C7, 0000,00 ;3046           WORD[0000]                          ;
U 00C8, 0008,00 ;3047           WORD[0800]                          ; ячейка 2B
U 00C9, 0000,00 ;3048           WORD[0000]                          ;
U 00CA, 0010,00 ;3049           WORD[1000]                          ; ячейка 2C
U 00CB, 0000,00 ;3050           WORD[0000]                          ;
U 00CC, 0020,00 ;3051           WORD[2000]                          ; ячейка 2D
U 00CD, 0000,00 ;3052           WORD[0000]                          ;
U 00CE, 0040,00 ;3053           WORD[4000]                          ; ячейка 2E
U 00CF, 0000,00 ;3054           WORD[0000]                          ;
U 00D0, 0080,00 ;3055           WORD[8000]                          ; ячейка 2F
U 00D1, 0000,00 ;3056           WORD[0000]                          ;
U 00D2, 0000,00 ;3057           WORD[0000]                          ; ячейка 30
U 00D3, 0000,01 ;3058           WORD[0001]                          ;
U 00D4, 0000,00 ;3059           WORD[0000]                          ; ячейка 31
U 00D5, 0000,02 ;3060           WORD[0002]                          ;
U 00D6, 0000,00 ;3061           WORD[0000]                          ; ячейка 32
U 00D7, 0000,04 ;3062           WORD[0004]                          ;
U 00D8, 0000,00 ;3063           WORD[0000]                          ; ячейка 33
U 00D9, 0000,08 ;3064           WORD[0008]                          ;
U 00DA, 0000,00 ;3065           WORD[0000]                          ; ячейка 34
U 00DB, 0000,10 ;3066           WORD[0010]                          ;
U 00DC, 0000,00 ;3067           WORD[0000]                          ; ячейка 35
U 00DD, 0000,20 ;3068           WORD[0020]                          ;
U 00DE, 0000,00 ;3069           WORD[0000]                          ; ячейка 36
U 00DF, 0000,40 ;3070           WORD[0040]                          ;
U 00E0, 0000,00 ;3071           WORD[0000]                          ; ячейка 37
U 00E1, 0000,80 ;3072           WORD[0080]                          ;
U 00E2, 0000,00 ;3073           WORD[0000]                          ; ячейка 38
U 00E3, 0001,00 ;3074           WORD[0100]                          ;
U 00E4, 0000,00 ;3075           WORD[0000]                          ; ячейка 39
U 00E5, 0002,00 ;3076           WORD[0200]                          ;
U 00E6, 0000,00 ;3077           WORD[0000]                          ; ячейка 3A
U 00E7, 0004,00 ;3078           WORD[0400]                          ;
U 00E8, 0000,00 ;3079           WORD[0000]                          ; ячейка 3B
U 00E9, 0008,00 ;3080           WORD[0800]                          ;
U 00EA, 0000,00 ;3081           WORD[0000]                          ; ячейка 3C
U 00EB, 0010,00 ;3082           WORD[1000]                          ;
U 00EC, 0000,00 ;3083           WORD[0000]                          ; ячейка 3D
U 00ED, 0020,00 ;3084           WORD[2000]                          ;
U 00EE, 0000,00 ;3085           WORD[0000]                          ; ячейка 3E
U 00EF, 0040,00 ;3086           WORD[4000]                          ;
U 00F0, 0000,00 ;3087           WORD[0000]                          ; ячейка 3F
U 00F1, 0080,00 ;3088           WORD[8000]                          ;
U 00F2, 0000,00 ;3089           WORD[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 микрокомструкции MOVE.
                    ;3206       ;        Эта секция загружается отдельной пересылкой.
                    ;3207       ;
                    ;3208      TRANSFER.DATA.LS2:
U 0162, 0000,78     ;3209               COUNT.LS[0078]                  ; счетчик длинных слов (число длинных слов, подлежащих
                    ;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 0178, 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, 0001,F0     ;3251               WORD[01F0]                      ;
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
U 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 01FB, 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
```

```
                ;3456      .PAGE "СЕКЦИЯ ДАННЫХ ОСНОВНОЙ ПАМЯТИ"
                ;3457      ;
                ;3458      ;    В этом разделе перечислены данные, которые будут переданы в основную
                ;3459      ;    память консольным процессором, как часть начальной установки, выпол-
                ;3460      ;    няемой в тесте 0. Программа переноса данных укажет эту область данных.
                ;3461      ;    Основная память содержит адреса микрослов, используемых для микродиаг-
                ;3462      ;    ностики, а также другие данные, используемые тестами.
                ;3463      ;       ПРИМЕЧАНИЕ: Данная таблица представлена в шестнадцатеричном коде,
                ;3464      ;    т.е., каждая цифра представляет четыре бита, так что четыре цифры пред-
                ;3465      ;    ставляют полное 16-битовое слово. Микроассемблер требует, чтобы шестнад-
                ;3466      ;    цатеричной величине, которая начинается с буквы (напр.FFFF), предшест-
                ;3467      ;    вовал 0 (0FFFF). Таким образом, некоторые величины в таблице будут со-
                ;3468      ;    держать 5 шестнадцатеричных цифр. Пятая цифра на самом деле не исполь-
                ;3469           зуется.
                ;3470      ;
                ;3471      ;
U 0254, 0101,90 ;3472           COUNT.MM[0190]                  ; счетчик длинных слов (количество длинных слов
                ;3473                                           ; пересылаемых в память). Приемником является 400(дес.)
                ;3474                                           ; длинных слов в основной памяти
U 0255, 0000,00 ;3475           START.ADR[0000]                 ; начальный адрес приемника (начинается с адреса 0
                ;3476                                           ; основной памяти)
U 0256, 00FF,00 ;3477           WORD[0FF00]                     ; ячейка 0
U 0257, 00FF,00 ;3478           WORD[0FF00]                     ;
                ;3479      ;
U 0258, 0000,FF ;3480           WORD[00FF]                      ; ячейка 4
U 0259, 0000,FF ;3481           WORD[00FF]                      ;
                ;3482
U 025A, 0000,01 ;3483           WORD[0001]                      ; ячейка 8
U 025B, 0000,00 ;3484           WORD[0000]                      ;
                ;3485      ;
U 025C, 0000,02 ;3486           WORD[0002]                      ; ячейка C
U 025D, 0000,00 ;3487           WORD[0000]                      ;
                ;3488      ;
U 025E, 0000,04 ;3489           WORD[0004]                      ; ячейка 10
U 025F, 0000,00 ;3490           WORD[0000]                      ;
                ;3491      ;
U 0260, 0000,08 ;3492           WORD[0008]                      ; ячейка 14
U 0261, 0000,00 ;3493           WORD[0000]                      ;
                ;3494      ;
U 0262, 0000,10 ;3495           WORD[0010]                      ; ячейка 18
U 0263, 0000,00 ;3496           WORD[0000]                      ;
                ;3497      ;
U 0264, 0000,20 ;3498           WORD[0020]                      ; ячейка 1C
U 0265, 0000,00 ;3499           WORD[0000]                      ;
                ;3500      ;
U 0266, 0000,40 ;3501           WORD[0040]                      ; ячейка 20
U 0267, 0000,00 ;3502           WORD[0000]                      ;
                ;3503      ;
U 0268, 0000,80 ;3504           WORD[0080]                      ; ячейка 24
U 0269, 0000,00 ;3505           WORD[0000]                      ;
                ;3506      ;
U 026A, 0001,00 ;3507           WORD[0100]                      ; ячейка 28
U 026B, 0000,00 ;3508           WORD[0000]                      ;
                ;3509      ;
U 026C, 0002,00 ;3510           WORD[0200]                      ; ячейка 2C
```

```
U 026D, 0000,00 ;3511           WORD[0000]                              ;
                ;3512     ;
U 026E, 0004,00 ;3513           WORD[0400]                              ; ячейка 30
U 026F, 0000,00 ;3514           WORD[0000]                              ;
                ;3515     ;
U 0270, 0008,00 ;3516           WORD[0800]                              ; ячейка 34
U 0271, 0000,00 ;3517           WORD[0000]                              ;
                ;3518     ;
U 0272, 0010,00 ;3519           WORD[1000]                              ; ячейка 38
U 0273, 0000,00 ;3520           WORD[0000]                              ;
                ;3521     ;
U 0274, 0020,00 ;3522           WORD[2000]                              ; ячейка 3C
U 0275, 0000,00 ;3523           WORD[0000]                              ;
                ;3524
U 0276, 0040,00 ;3525           WORD[4000]                              ; ячейка 40
U 0277, 0000,00 ;3526           WORD[0000]                              ;
                ;3527     ;
U 0278, 0080,00 ;3528           WORD[8000]                              ; ячейка 44
U 0279, 0000,00 ;3529           WORD[0000]                              ;
                ;3530     ;
U 027A, 0000,00 ;3531           WORD[0000]                              ; ячейка 48
U 027B, 0000,01 ;3532           WORD[0001]                              ;
                ;3533     ;
U 027C, 0000,00 ;3534           WORD[0000]                              ; ячейка 4C
U 027D, 0000,02 ;3535           WORD[0002]                              ;
                ;3536     ;
U 027E, 0000,00 ;3537           WORD[0000]                              ; ячейка 50
U 027F, 0000,04 ;3538           WORD[0004]                              ;
                ;3539     ;
U 0280, 0000,00 ;3540           WORD[0000]                              ; ячейка 54
U 0281, 0000,08 ;3541           WORD[0008]                              ;
                ;3542     ;
U 0282, 0000,00 ;3543           WORD[0000]                              ; ячейка 58
U 0283, 0000,10 ;3544           WORD[0010]                              ;
                ;3545     ;
U 0284, 0000,00 ;3546           WORD[0000]                              ; ячейка 5C
U 0285, 0000,20 ;3547           WORD[0020]                              ;
                ;3548     ;
U 0286, 0000,00 ;3549           WORD[0000]                              ; ячейка 60
U 0287, 0000,40 ;3550           WORD[0040]                              ;
                ;3551     ;
U 0288, 0000,00 ;3552           WORD[0000]                              ; ячейка 64
U 0289, 0000,80 ;3553           WORD[0080]                              ;
                ;3554     ;
U 028A, 0000,00 ;3555           WORD[0000]                              ; ячейка 68
U 028B, 0001,00 ;3556           WORD[0100]                              ;
                ;3557     ;
U 028C, 0000,00 ;3558           WORD[0000]                              ; ячейка 6C
U 028D, 0002,00 ;3559           WORD[0200]                              ;
                ;3560     ;
U 028E, 0000,00 ;3561           WORD[0000]                              ; ячейка 70
U 028F, 0004,00 ;3562           WORD[0400]                              ;
                ;3563     ;
U 0290, 0000,00 ;3564           WORD[0000]                              ; ячейка 74
U 0291, 0008,00 ;3565           WORD[0800]                              ;
```

```
                    ;3566     ;
U 0292,  0000,00    ;3567           WORD[0000]                              ; ячейка 78
U 0293,  0010,00    ;3568           WORD[1000]                              ;
                    ;3569     ;
U 0294,  0000,00    ;3570           WORD[0000]                              ; ячейка 7C
U 0295,  0020,00    ;3571           WORD[2000]                              ;
                    ;3572     ;
U 0296,  0000,00    ;3573           WORD[0000]                              ; ячейка 80
U 0297,  0040,00    ;3574           WORD[4000]                              ;
                    ;3575     ;
U 0298,  0000,00    ;3576           WORD[0000]                              ; ячейка 84
U 0299,  0080,00    ;3577           WORD[8000]                              ;
                    ;3578     ;
U 029A,  0000,00    ;3579           WORD[0000]                              ; ячейка 88
U 029B,  0000,00    ;3580           WORD[0000]                              ;
                    ;3581     ;
U 029C,  00FF,FF    ;3582           WORD[0FFFF]                             ; ячейка 8C
U 029D,  00FF,FF    ;3583           WORD[0FFFF]                             ;
                    ;3584     ;
U 029E,  0001,01    ;3585           WORD[0101]                              ; ячейка 90
U 029F,  0001,01    ;3586           WORD[0101]                              ;
                    ;3587     ;
U 02A0,  0000,00    ;3588           WORD[0000]                              ; ячейка 94
U 02A1,  0000,00    ;3589           WORD[0000]                              ;
                    ;3590     ;
                    ;3591     ;  Следующие данные представляют собой содержимое ПЗУ ДЕШИФРАТОР СПЕЦИФИКАТОРОВ
                    ;3592     ;  и используются тестом 4 части 3
                    ;3593     ;
U 02A2,  0004,04    ;3594           WORD[0404]                              ; ячейка 98
U 02A3,  0004,04    ;3595           WORD[0404]                              ;
                    ;3596     ;
U 02A4,  0004,04    ;3597           WORD[0404]                              ; ячейка 9C
U 02A5,  0004,04    ;3598           WORD[0404]                              ;
                    ;3599     ;
U 02A6,  0004,04    ;3600           WORD[0404]                              ; ячейка 0A0
U 02A7,  0004,04    ;3601           WORD[0404]                              ;
                    ;3602     ;
U 02A8,  0004,04    ;3603           WORD[0404]                              ; ячейка 0A4
U 02A9,  0004,04    ;3604           WORD[0404]                              ;
                    ;3605     ;
U 02AA,  0013,13    ;3606           WORD[1313]                              ; ячейка 0A8
U 02AB,  0013,00    ;3607           WORD[1300]                              ;
                    ;3608     ;
U 02AC,  0001,01    ;3609           WORD[0101]                              ; ячейка 0AC
U 02AD,  0001,01    ;3610           WORD[0101]                              ;
                    ;3611     ;
U 02AE,  000D,0D    ;3612           WORD[0D0D]                              ; ячейка 0B0
U 02AF,  000D,00    ;3613           WORD[0D00]                              ;
                    ;3614     ;
U 02B0,  0023,23    ;3615           WORD[2323]                              ; ячейка 0B4
U 02B1,  0023,00    ;3616           WORD[2300]                              ;
                    ;3617     ;
U 02B2,  002F,2F    ;3618           WORD[2F2F]                              ; ячейка 0B8
U 02B3,  002F,3A    ;3619           WORD[2F3A]                              ;
                    ;3620     ;
```

```
U 02B4, 0043,43 ;3621           WORD[4343]                          ; ячейка 0BC
U 02B5, 0043,51 ;3622           WORD[4351]                          ;
                ;3623   ;
U 02B6, 005B,5B ;3624           WORD[5B5B]                          ; ячейка 0C0
U 02B7, 005B,64 ;3625           WORD[5B64]                          ;
                ;3626   ;
U 02B8, 006D,6D ;3627           WORD[6D6D]                          ; ячейка 0C4
U 02B9, 006D,79 ;3628           WORD[6D79]                          ;
                ;3629   ;
U 02BA, 0085,85 ;3630           WORD[8585]                          ; ячейка 0C8
U 02BB, 0085,92 ;3631           WORD[8592]                          ;
                ;3632   ;
U 02BC, 009F,9F ;3633           WORD[9F9F]                          ; ячейка 0CC
U 02BD, 009F,AD ;3634           WORD[9FAD]                          ;
                ;3635   ;
U 02BE, 00BB,BB ;3636           WORD[0BBBB]                         ; ячейка 0D0
U 02BF, 00BB,C8 ;3637           WORD[0BBC8]                         ;
                ;3638   ;
U 02C0, 00D5,D5 ;3639           WORD[0D5D5]                         ; ячейка 0D4
U 02C1, 00D5,E3 ;3640           WORD[0D5E3]                         ;
                ;3641   ;
U 02C2, 0007,07 ;3642           WORD[0707]                          ; ячейка 0D8
U 02C3, 0007,07 ;3643           WORD[0707]                          ;
                ;3644   ;
U 02C4, 0007,07 ;3645           WORD[0707]                          ; ячейка 0DC
U 02C5, 0007,07 ;3646           WORD[0707]                          ;
                ;3647   ;
U 02C6, 0007,07 ;3648           WORD[0707]                          ; ячейка 0E0
U 02C7, 0007,07 ;3649           WORD[0707]                          ;
                ;3650   ;
U 02C8, 0007,07 ;3651           WORD[0707]                          ; ячейка 0E4
U 02C9, 0007,07 ;3652           WORD[0707]                          ;
                ;3653   ;
U 02CA, 0013,13 ;3654           WORD[1313]                          ; ячейка 0E8
U 02CB, 0013,00 ;3655           WORD[1300]                          ;
                ;3656   ;
U 02CC, 0001,01 ;3657           WORD[0101]                          ; ячейка 0EC
U 02CD, 0001,01 ;3658           WORD[0101]                          ;
                ;3659   ;
U 02CE, 000D,0D ;3660           WORD[0D0D]                          ; ячейка 0F0
U 02CF, 000D,00 ;3661           WORD[0D00]                          ;
                ;3662   ;
U 02D0, 0023,23 ;3663           WORD[2323]                          ; ячейка 0F4
U 02D1, 0023,00 ;3664           WORD[2300]                          ;
                ;3665   ;
U 02D2, 002F,2F ;3666           WORD[2F2F]                          ; ячейка 0F8
U 02D3, 002F,3A ;3667           WORD[2F3A]                          ;
                ;3668   ;
U 02D4, 0043,43 ;3669           WORD[4343]                          ; ячейка 0FC
U 02D5, 0043,51 ;3670           WORD[4351]                          ;
                ;3671   ;
U 02D6, 005B,5B ;3672           WORD[5B5B]                          ; ячейка 100
U 02D7, 005B,64 ;3673           WORD[5B64]                          ;
                ;3674   ;
U 02D8, 006D,6D ;3675           WORD[6D6D]                          ; ячейка 104
```

```
U 02D9, 006D,79 ;3676            WORD[6D79]                              ;
                ;3677     ;
U 02DA, 0085,85 ;3678            WORD[8585]                              ; ячейка 108
U 02DB, 0085,92 ;3679            WORD[8592]                              ;
                ;3680     ;
U 02DC, 009F,9F ;3681            WORD[9F9F]                              ; ячейка 10C
U 02DD, 009F,AD ;3682            WORD[9FAD]                              ;
                ;3683     ;
U 02DE, 00B8,BB ;3684            WORD[0BBBB]                             ; ячейка 110
U 02DF, 00B8,C8 ;3685            WORD[0BBC8]                             ;
                ;3686     ;
U 02E0, 00D5,D5 ;3687            WORD[0D5D5]                             ; ячейка 114
U 02E1, 00D5,E3 ;3688            WORD[0D5E3]                             ;
                ;3689     ;
U 02E2, 0000,00 ;3690            WORD[0000]                              ; ячейка 118
U 02E3, 0000,00 ;3691            WORD[0000]                              ;
                ;3692     ;
U 02E4, 0000,00 ;3693            WORD[0000]                              ; ячейка 11C
U 02E5, 0000,00 ;3694            WORD[0000]                              ;
                ;3695     ;
U 02E6, 0000,00 ;3696            WORD[0000]                              ; ячейка 120
U 02E7, 0000,00 ;3697            WORD[0000]                              ;
                ;3698     ;
U 02E8, 0000,00 ;3699            WORD[0000]                              ; ячейка 124
U 02E9, 0000,00 ;3700            WORD[0000]                              ;
                ;3701     ;
U 02EA, 0013,13 ;3702            WORD[1313]                              ; ячейка 128
U 02EB, 0013,00 ;3703            WORD[1300]                              ;
                ;3704     ;
U 02EC, 0000,00 ;3705            WORD[0000]                              ; ячейка 12C
U 02ED, 0000,00 ;3706            WORD[0000]                              ;
                ;3707     ;
U 02EE, 000D,0D ;3708            WORD[0D0D]                              ; ячейка 13
U 02EF, 000D,00 ;3709            WORD[0D00]                              ;
                ;3710     ;
U 02F0, 0023,23 ;3711            WORD[2323]                              ; ячейка 134
U 02F1, 0023,00 ;3712            WORD[2300]                              ;
                ;3713     ;
U 02F2, 002F,2F ;3714            WORD[2F2F]                              ; ячейка 138
U 02F3, 002F,3A ;3715            WORD[2F3A]                              ;
                ;3716     ;
U 02F4, 0043,43 ;3717            WORD[4343]                              ; ячейка 13C
U 02F5, 0043,51 ;3718            WORD[4351]                              ;
                ;3719     ;
U 02F6, 005B,5B ;3720            WORD[5B5B]                              ; ячейка 140
U 02F7, 005B,64 ;3721            WORD[5B64]                              ;
                ;3722     ;
U 02F8, 006D,6D ;3723            WORD[6D6D]                              ; ячейка 144
U 02F9, 006D,79 ;3724            WORD[6D79]                              ;
                ;3725     ;
U 02FA, 0085,85 ;3726            WORD[8585]                              ; ячейка 148
U 02FB, 0085,92 ;3727            WORD[8592]                              ;
                ;3728     ;
U 02FC, 009F,9F ;3729            WORD[9F9F]                              ; ячейка 14C
U 02FD, 009F,AD ;3730            WORD[9FAD]                              ;
```

```
                    ;3731     ;
U 02FE, 00BB,BB     ;3732               WORD[0BBBB]                         ; ячейка 150
U 02FF, 00BB,C8     ;3733               WORD[0BBC8]                         ;
                    ;3734     ;
U 0300, 00D5,D5     ;3735               WORD[0D5D5]                         ; ячейка 154
U 0301, 00D5,E3     ;3736               WORD[0D5E3]                         ;
                    ;3737     ;
U 0302, 00FF,FF     ;3738               WORD[0FFFF]                         ; ячейка 158
U 0303, 00FF,FF     ;3739               WORD[0FFFF]                         ;
                    ;3740     ;
U 0304, 00FF,FF     ;3741               WORD[0FFFF]                         ; ячейка 15C
U 0305, 00FF,FF     ;3742               WORD[0FFFF]                         ;
                    ;3743     ;
U 0306, 00FF,FF     ;3744               WORD[0FFFF]                         ; ячейка 160
U 0307, 00FF,FF     ;3745               WORD[0FFFF]                         ;
                    ;3746     ;
U 0308, 00FF,FF     ;3747               WORD[0FFFF]                         ; ячейка 164
U 0309, 00FF,FF     ;3748               WORD[0FFFF]                         ;
                    ;3749     ;
U 030A, 00FF,FF     ;3750               WORD[0FFFF]                         ; ячейка 168
U 030B, 00FF,FF     ;3751               WORD[0FFFF]                         ;
                    ;3752     ;
U 030C, 00FF,FF     ;3753               WORD[0FFFF]                         ; ячейка 16C
U 030D, 00FF,FF     ;3754               WORD[0FFFF]                         ;
                    ;3755     ;
U 030E, 00FF,FF     ;3756               WORD[0FFFF]                         ; ячейка 170
U 030F, 00FF,FF     ;3757               WORD[0FFFF]                         ;
                    ;3758     ;
U 0310, 00FF,FF     ;3759               WORD[0FFFF]                         ; ячейка 174
U 0311, 00FF,FF     ;3760               WORD[0FFFF]                         ;
                    ;3761     ;
U 0312, 00FF,FF     ;3762               WORD[0FFFF]                         ; ячейка 178
U 0313, 00FF,FF     ;3763               WORD[0FFFF]                         ;
                    ;3764     ;
U 0314, 00FF,FF     ;3765               WORD[0FFFF]                         ; ячейка 17C
U 0315, 00FF,FF     ;3766               WORD[0FFFF]                         ;
                    ;3767     ;
U 0316, 00FF,FF     ;3768               WORD[0FFFF]                         ; ячейка 180
U 0317, 00FF,FF     ;3769               WORD[0FFFF]                         ;
                    ;3770     ;
U 0318, 00FF,FF     ;3771               WORD[0FFFF]                         ; ячейка 184
U 0319, 00FF,FF     ;3772               WORD[0FFFF]                         ;
                    ;3773     ;
U 031A, 00FF,FF     ;3774               WORD[0FFFF]                         ; ячейка 188
U 031B, 00FF,FF     ;3775               WORD[0FFFF]                         ;
                    ;3776     ;
U 031C, 00FF,FF     ;3777               WORD[0FFFF]                         ; ячейка 18C
U 031D, 00FF,FF     ;3778               WORD[0FFFF]                         ;
                    ;3779     ;
U 031E, 00FF,FF     ;3780               WORD[0FFFF]                         ; ячейка 190
U 031F, 00FF,FF     ;3781               WORD[0FFFF]                         ;
                    ;3782     ;
U 0320, 00FF,FF     ;3783               WORD[0FFFF]                         ; ячейка 194
U 0321, 00FF,FF     ;3784               WORD[0FFFF]                         ;
                    ;3785     ;
```

```
U 0322, 0000,00 ;3786           WORD[0000]                          ; ячейка 198
U 0323, 0000,00 ;3787           WORD[0000]                          ;
                ;3788     ;
U 0324, 0000,00 ;3789           WORD[0000]                          ; ячейка 19C
U 0325, 0000,00 ;3790           WORD[0000]                          ;
                ;3791     ;
U 0326, 0000,00 ;3792           WORD[0000]                          ; ячейка 1A0
U 0327, 0000,00 ;3793           WORD[0000]                          ;
                ;3794     ;
U 0328, 0000,00 ;3795           WORD[0000]                          ; ячейка 1A4
U 0329, 0000,00 ;3796           WORD[0000]                          ;
                ;3797     ;
U 032A, 0013,13 ;3798           WORD[1313]                          ; ячейка 1A8
U 032B, 0013,00 ;3799           WORD[1300]                          ;
                ;3800     ;
U 032C, 0002,02 ;3801           WORD[0202]                          ; ячейка 1AC
U 032D, 0002,02 ;3802           WORD[0202]                          ;
                ;3803     ;
U 032E, 000D,0D ;3804           WORD[0D0D]                          ; ячейка 1B0
U 032F, 000D,00 ;3805           WORD[0D00]                          ;
                ;3806     ;
U 0330, 0023,23 ;3807           WORD[2323]                          ; ячейка 1B4
U 0331, 0023,00 ;3808           WORD[2300]                          ;
                ;3809     ;
U 0332, 002F,2F ;3810           WORD[2F2F]                          ; ячейка 1B8
U 0333, 002F,3A ;3811           WORD[2F3A]                          ;
                ;3812     ;
U 0334, 0043,43 ;3813           WORD[4343]                          ; ячейка 1BC
U 0335, 0043,51 ;3814           WORD[4351]                          ;
                ;3815     ;
U 0336, 0058,5B ;3816           WORD[5B5B]                          ; ячейка 1C0
U 0337, 005B,64 ;3817           WORD[5B64]                          ;
                ;3818     ;
U 0338, 006D,6D ;3819           WORD[6D6D]                          ; ячейка 1C4
U 0339, 006D,79 ;3820           WORD[6D79]                          ;
                ;3821     ;
U 033A, 0085,85 ;3822           WORD[8585]                          ; ячейка 1C8
U 033B, 0085,92 ;3823           WORD[8592]                          ;
                ;3824     ;
U 033C, 009F,9F ;3825           WORD[9F9F]                          ; ячейка 1CC
U 033D, 009F,AD ;3826           WORD[9FAD]                          ;
                ;3827     ;
U 033E, 00BB,BB ;3828           WORD[0BBBB]                         ; ячейка 1D0
U 033F, 00BB,C8 ;3829           WORD[0BBC8]                         ;
                ;3830     ;
U 0340, 00D5,D5 ;3831           WORD[0D5D5]                         ; ячейка 1D4
U 0341, 00D5,E3 ;3832           WORD[0D5E3]                         ;
                ;3833     ;
U 0342, 0000,00 ;3834           WORD[0000]                          ; ячейка 1D8
U 0343, 0000,00 ;3835           WORD[0000]                          ;
                ;3836     ;
U 0344, 0000,00 ;3837           WORD[0000]                          ; ячейка 1DC
U 0345, 0000,00 ;3838           WORD[0000]                          ;
                ;3839     ;
U 0346, 0000,00 ;3840           WORD[0000]                          ; ячейка 1E0
```

```
U 0347, 0000,00 ;3841           WORD[0000]                              ;
                ;3842     ;
U 0348, 0000,00 ;3843           WORD[0000]                              ; ячейка 1E4
U 0349, 0000,00 ;3844           WORD[0000]                              ;
                ;3845     ;
U 034A, 0022,22 ;3846           WORD[2222]                              ; ячейка 1E8
U 034B, 0022,22 ;3847           WORD[2222]                              ;
                ;3848     .;
U 034C, 0000,00 ;3849           WORD[0000]                              ; ячейка 1EC
U 034D, 0000,00 ;3850           WORD[0000]                              ;
                ;3851     ;
U 034E, 000D,0D ;3852           WORD[0D0D]                              ; ячейка 1F0
U 034F, 000D,00 ;3853           WORD[0D00]                              ;
                ;3854     ;
U 0350, 0023,23 ;3855           WORD[2323]                              ; ячейка 1F4
U 0351, 0023,00 ;3856           WORD[2300]                              ;
                ;3857     ;
U 0352, 002F,2F ;3858           WORD[2F2F]                              ; ячейка 1F8
U 0353, 002F,3A ;3859           WORD[2F3A]                              ;
                ;3860     ;
U 0354, 0043,43 ;3861           WORD[4343]                              ; ячейка 1FC
U 0355, 0043,51 ;3862           WORD[4351]                              ;
                ;3863     ;
U 0356, 005B,5B ;3864           WORD[5B5B]                              ; ячейка 200
U 0357, 005B,64 ;3865           WORD[5B64]                              ;
                ;3866     ;
U 0358, 006D,6D ;3867           WORD[6D6D]                              ; ячейка 204
U 0359, 006D,79 ;3868           WORD[6D79]                              ;
                ;3869     ;
U 035A, 0085,85 ;3870           WORD[8585]                              ; ячейка 208
U 035B, 0085,92 ;3871           WORD[8592]                              ;
                ;3872     ;
U 035C, 009F,9F ;3873           WORD[9F9F]                              ; ячейка 20C
U 035D, 009F,AD ;3874           WORD[9FAD]                              ;
                ;3875     ;
U 035E, 00BB,BB ;3876           WORD[0BBBB]                             ; ячейка 210
U 035F, 00BB,C8 ;3877           WORD[0BBC8]                             ;
                ;3878     ;
U 0360, 00D5,D5 ;3879           WORD[0D5D5]                             ; ячейка 214
U 0361, 00D5,E3 ;3880           WORD[0D5E3]                             ;
                ;3881     ;
U 0362, 002B,2B ;3882           WORD[2B2B]                              ; ячейка 218
U 0363, 002C,2C ;3883           WORD[2C2C]                              ;
                ;3884     ;
U 0364, 0031,31 ;3885           WORD[3131]                              ; ячейка 21C
U 0365, 0038,38 ;3886           WORD[3838]                              ;
                ;3887     ;
U 0366, 0041,41 ;3888           WORD[4141]                              ; ячейка 220
U 0367, 0049,49 ;3889           WORD[4949]                              ;
                ;3890     ;
U 0368, 0053,53 ;3891           WORD[5353]                              ; ячейка 224
U 0369, 005C,5C ;3892           WORD[5C5C]                              ;
                ;3893     ;
U 036A, 002B,2B ;3894           WORD[2B2B]                              ; ячейка 228
U 036B, 002C,2C ;3895           WORD[2C2C]                              :
```

```
                    ;3896     ;
U 036C, 0031,31     ;3897           WORD[3131]                          ; ячейка 22C
U 036D, 0038,38     ;3898           WORD[3838]                          ;
                    ;3899     ;
U 036E, 0041,41     ;3900           WORD[4141]                          ; ячейка 230
U 036F, 0049,49     ;3901           WORD[4949]                          ;
                    ;3902     ;
U 0370, 0053,53     ;3903           WORD[5353]                          ; ячейка 234
U 0371, 005C,5C     ;3904           WORD[5C5C]                          ;
                    ;3905     ;
U 0372, 002B,2B     ;3906           WORD[2B2B]                          ; ячейка 238
U 0373, 002C,2C     ;3907           WORD[2C2C]                          ;
                    ;3908     ;
U 0374, 0031,31     ;3909           WORD[3131]                          ; ячейка 23C
U 0375, 0038,38     ;3910           WORD[3838]                          ;
                    ;3911     ;
U 0376, 0041,41     ;3912           WORD[4141]                          ; ячейка 240
U 0377, 0049,49     ;3913           WORD[4949]                          ;
                    ;3914     ;
U 0378, 0053,53     ;3915           WORD[5353]                          ; ячейка 244
U 0379, 005C,5C     ;3916           WORD[5C5C]                          ;
                    ;3917     ;
U 037A, 002B,2B     ;3918           WORD[2B2B]                          ; ячейка 248
U 037B, 002C,2C     ;3919           WORD[2C2C]                          ;
                    ;3920     ;
U 037C, 0031,31     ;3921           WORD[3131]                          ; ячейка 24C
U 037D, 0038,38     ;3922           WORD[3838]                          ;
                    ;3923     ;
U 037E, 0041,41     ;3924           WORD[4141]                          ; ячейка 250
U 037F, 0049,49     ;3925           WORD[4949]                          ;
                    ;3926     ;
U 0380, 0053,53     ;3927           WORD[5353]                          ; ячейка 254
U 0381, 005C,5C     ;3928           WORD[5C5C]                          ;
                    ;3929     ;
U 0382, 0000,00     ;3930           WORD[0000]                          ; ячейка 258
U 0383, 0000,00     ;3931           WORD[0000]                          ;
                    ;3932     ;
U 0384, 0000,00     ;3933           WORD[0000]                          ; ячейка 25C
U 0385, 0000,00     ;3934           WORD[0000]                          ;
                    ;3935     ;
U 0386, 0000,00     ;3936           WORD[0000]                          ; ячейка 260
U 0387, 0000,00     ;3937           WORD[0000]                          ;
                    ;3938     ;
U 0388, 0000,00     ;3939           WORD[0000]                          ; ячейка 264
U 0389, 0000,00     ;3940           WORD[0000]                          ;
                    ;3941     ;
U 038A, 0001,01     ;3942           WORD[0101]                          ; ячейка 268
U 038B, 0001,01     ;3943           WORD[0101]                          ;
                    ;3944     ;
U 038C, 0001,01     ;3945           WORD[0101]                          ; ячейка 26C
U 038D, 0001,01     ;3946           WORD[0101]                          ;
                    ;3947     ;
U 038E, 0001,01     ;3948           WORD[0101]                          ; ячейка 270
U 038F, 0001,01     ;3949           WORD[0101]                          ;
                    ;3950     ;
```

```
U 0390, 0001,01 ;3951           WORD[0101]                              ; ячейка 274
U 0391, 0001,01 ;3952           WORD[0101]                              ;
                ;3953   ;
U 0392, 0002,02 ;3954           WORD[0202]                              ; ячейка 278
U 0393, 0002,02 ;3955           WORD[0202]                              ;
                ;3956   ;
U 0394, 0002,02 ;3957           WORD[0202]                              ; ячейка 27C
U 0395, 0002,02 ;3958           WORD[0202]                              ;
                ;3959   ;
U 0396, 0002,02 ;3960           WORD[0202]                              ; ячейка 280
U 0397, 0002,02 ;3961           WORD[0202]                              ;
                ;3962   ;
U 0398, 0002,02 ;3963           WORD[0202]                              ; ячейка 284
U 0399, 0002,02 ;3964           WORD[0202]                              ;
                ;3965   ;
U 039A, 0003,03 ;3966           WORD[0303]                              ; ячейка 288
U 039B, 0003,03 ;3967           WORD[0303]                              ;
                ;3968   ;
U 039C, 0003,03 ;3969           WORD[0303]                              ; ячейка 28C
U 039D, 0003,03 ;3970           WORD[0303]                              ;
                ;3971   ;
U 039E, 0003,03 ;3972           WORD[0303]                              ; ячейка 290
U 039F, 0003,03 ;3973           WORD[0303]                              ;
                ;3974   ;
U 03A0, 0003,03 ;3975           WORD[0303]                              ; ячейка 294
U 03A1, 0003,03 ;3976           WORD[0303]                              ;
                ;3977   ;
                ;3978   ;  Следующие данные представляют содержимое ПЗУ ДЕШИФРАТОР КОДОВ ОПЕРАЦИЙ и
                ;3979   ;  используются тестом 5 части 3
                ;3980   ;
U 03A2, 0004,02 ;3981           WORD[0402]                              ; ячейка 298
U 03A3, 0015,01 ;3982           WORD[1501]                              ;
                ;3983   ;
U 03A4, 0005,01 ;3984           WORD[0501]                              ; ячейка 29C
U 03A5, 0006,01 ;3985           WORD[0601]                              ;
                ;3986   ;
U 03A6, 0004,01 ;3987           WORD[0401]                              ; ячейка 2A0
U 03A7, 0015,01 ;3988           WORD[1501]                              ;
                ;3989   ;
U 03A8, 0004,39 ;3990           WORD[0439]                              ; ячейка 2A4
U 03A9, 000E,40 ;3991           WORD[0E40]                              ;
                ;3992   ;
U 03AA, 0027,08 ;3993           WORD[2708]                              ; ячейка 2A8
U 03AB, 0004,18 ;3994           WORD[0418]                              ;
                ;3995   ;
U 03AC, 0007,20 ;3996           WORD[0720]                              ; ячейка 2AC
U 03AD, 0030,40 ;3997           WORD[3040]                              ;
                ;3998   ;
U 03AE, 008D,01 ;3999           WORD[8D01]                              ; ячейка 2B0
U 03AF, 00B1,01 ;4000           WORD[0B101]                             ;
                ;4001   ;
U 03B0, 00B2,06 ;4002           WORD[0B206]                             ; ячейка 2B4
U 03B1, 0098,02 ;4003           WORD[9802]                              ;
                ;4004   ;
U 03B2, 007D,01 ;4005           WORD[7D01]                              ; ячейка 2B8
```

```
U 03B3, 0080,01 ;4006           WORD[8001]                              ;
                ;4007     ;
U 03B4, 0083,01 ;4008           WORD[8301]                              ; ячейка 2BC
U 03B5, 0086,01 ;4009           WORD[8601]                              ;
                ;4010     ;
U 03B6, 00BD,02 ;4011           WORD[0BD02]                             ; ячейка 2C0
U 03B7, 00BE,10 ;4012           WORD[0BE10]                             ;
                ;4013     ;
U 03B8, 00CC,10 ;4014           WORD[0CC10]                             ; ячейка 2C4
U 03B9, 00C9,10 ;4015           WORD[0C910]                             ;
                ;4016     ;
U 03BA, 00DB,10 ;4017           WORD[0DB10]                             ; ячейка 2C8
U 03BB, 00CF,10 ;4018           WORD[0CF10]                             ;
                ;4019     ;
U 03BC, 00D3,10 ;4020           WORD[0D310]                             ; ячейка 2CC
U 03BD, 008E,02 ;4021           WORD[8E02]                              ;
                ;4022     ;
U 03BE, 0090,02 ;4023           WORD[9002]                              ; ячейка 2D0
U 03BF, 0092,02 ;4024           WORD[9202]                              ;
                ;4025     ;
U 03C0, 0094,02 ;4026           WORD[9402]                              ; ячейка 2D4
U 03C1, 0096,02 ;4027           WORD[9602]                              ;
                ;4028     ;
U 03C2, 00BD,04 ;4029           WORD[0BD04]                             ; ячейка 2D8
U 03C3, 00A8,02 ;4030           WORD[0A802]                             ;
                ;4031     ;
U 03C4, 00B2,08 ;4032           WORD[0B208]                             ; ячейка 2DC
U 03C5, 00BB,01 ;4033           WORD[0BB01]                             ;
                ;4034     ;
U 03C6, 00BC,01 ;4035           WORD[0BC01]                             ; ячейка 2E0
U 03C7, 007D,01 ;4036           WORD[7D01]                              ;
                ;4037     ;
U 03C8, 0080,01 ;4038           WORD[8001]                              ; ячейка 2E4
U 03C9, 0083,01 ;4039           WORD[8301]                              ;
                ;4040     ;
U 03CA, 0089,01 ;4041           WORD[8901]                              ; ячейка 2E8
U 03CB, 00BD,02 ;4042           WORD[0BD02]                             ;
                ;4043     ;
U 03CC, 00C2,10 ;4044           WORD[0C210]                             ; ячейка 2EC
U 03CD, 00CC,10 ;4045           WORD[0CC10]                             ;
                ;4046     ;
U 03CE, 00C9,10 ;4047           WORD[0C910]                             ; ячейка 2F0
U 03CF, 00DB,10 ;4048           WORD[0DB10]                             ;
                ;4049     ;
U 03D0, 00CF,10 ;4050           WORD[0CF10]                             ; ячейка 2F4
U 03D1, 00D7,10 ;4051           WORD[0D710]                             ;
                ;4052     ;
U 03D2, 00BD,10 ;4053           WORD[0BD10]                             ; ячейка 2F8
U 03D3, 00EA,D8 ;4054           WORD[0EAD8]                             ;
                ;4055     ;
U 03D4, 00D9,00 ;4056           WORD[0D900]                             ; ячейка 2FC
U 03D5, 0000,00 ;4057           WORD[0000]                              ;
                ;4058     ;
U 03D6, 00DC,DB ;4059           WORD[0DCDB]                             ; ячейка 300
U 03D7, 0012,11 ;4060           WORD[1211]                              ;
```

```
                    ;4061     ;
U 03D8, 00E6,66     ;4062               WORD[0E666]                         ; ячейка 304
U 03D9, 00DA,D9     ;4063               WORD[0DAD9]                         ;
                    ;4064     ;
U 03DA, 003B,1B     ;4065               WORD[3B1B]                          ; ячейка 308
U 03DB, 0000,00     ;4066               WORD[0000]                          ;
                    ;4067     ;
U 03DC, 0000,00     ;4068               WORD[0000]                          ; ячейка 30C
U 03DD, 0000,00     ;4069               WORD[0000]                          ;
                    ;4070     ;
U 03DE, 0000,00     ;4071               WORD[0000]                          ; ячейка 310
U 03DF, 0000,00     ;4072               WORD[0000]                          ;
                    ;4073     ;
U 03E0, 0000,00     ;4074               WORD[0000]                          ; ячейка 314
U 03E1, 0000,00     ;4075               WORD[0000]                          ;
                    ;4076     ;
U 03E2, 0000,00     ;4077               WORD[0000]                          ; ячейка 318
U 03E3, 000B,07     ;4078               WORD[0B07]                          ;
                    ;4079     ;
U 03E4, 000A,09     ;4080               WORD[0A09]                          ; ячейка 31C
U 03E5, 000B,0F     ;4081               WORD[0B0F]                          ;
                    ;4082     ;
U 03E6, 000C,10     ;4083               WORD[0C10]                          ; ячейка 320
U 03E7, 0021,1D     ;4084               WORD[211D]                          ;
                    ;4085     ;
U 03E8, 0026,25     ;4086               WORD[2625]                          ; ячейка 324
U 03E9, 0022,1E     ;4087               WORD[221E]                          ;
                    ;4088     ;
U 03EA, 0020,1F     ;4089               WORD[201F]                          ; ячейка 328
U 03EB, 0000,00     ;4090               WORD[0000]                          ;
                    ;4091     ;
U 03EC, 00A0,96     ;4092               WORD[0A096]                         ; ячейка 32C
U 03ED, 0005,04     ;4093               WORD[0504]                          ;
                    ;4094     ;
U 03EE, 0006,0E     ;4095               WORD[060E]                          ; ячейка 330
U 03EF, 0027,E7     ;4096               WORD[27E7]                          ;
                    ;4097     ;
U 03F0, 0024,23     ;4098               WORD[2423]                          ; ячейка 334
U 03F1, 00E3,B2     ;4099               WORD[0E382]                         ;
                    ;4100     ;
U 03F2, 0004,02     ;4101               WORD[0402]                          ; ячейка 338
U 03F3, 000C,03     ;4102               WORD[0C03]                          ;
                    ;4103     ;
U 03F4, 0009,00     ;4104               WORD[0900]                          ; ячейка 33C
U 03F5, 004E,49     ;4105               WORD[4E49]                          ;
                    ;4106     ;
U 03F6, 002D,28     ;4107               WORD[2D28]                          ; ячейка 340
U 03F7, 00AD,A8     ;4108               WORD[0ADA8]                         ;
                    ;4109     ;
U 03F8, 00B6,B2     ;4110               WORD[0B6B2]                         ; ячейка 344
U 03F9, 00FC,FC     ;4111               WORD[0FCFC]                         ;
                    ;4112     ;
U 03FA, 003F,FC     ;4113               WORD[3FFC]                          ; ячейка 348
U 03FB, 0068,05     ;4114               WORD[6805]                          ;
                    ;4115     ;
```

```
U 03FC, 0029,04 ;4116            WORD[2904]                          ; ячейка 34C
U 03FD, 00DD,F1 ;4117            WORD[0DDF1]                         ;
                ;4118    ;
U 03FE, 00DF,CB ;4119            WORD[0DFCB]                         ; ячейка 350
U 03FF, 00DF,DA ;4120            WORD[0DFDA]                         ;
                ;4121    ';
U 0400, 00DF,DF ;4122            WORD[0DFDF]                         ; ячейка 354
U 0401, 00A0,64 ;4123            WORD[0A064]                         ;
                ;4124    ';
U 0402, 00E9,E8 ;4125            WORD[0E9E8]                         ; ячейка 358
U 0403, 0021,13 ;4126            WORD[2113]                          ;
                ;4127    ;
U 0404, 001E,10 ;4128            WORD[1E10]                          ; ячейка 35C
U 0405, 005F,52 ;4129            WORD[5F52]                          ;
                ;4130    ;
U 0406, 003F,31 ;4131            WORD[3F31]                          ; ячейка 360
U 0407, 0090,8B ;4132            WORD[908B]                          ;
                ;4133    ;
U 0408, 0099,95 ;4134            WORD[9995]                          ; ячейка 364
U 0409, 00F3,F3 ;4135            WORD[0F3F3]                         ;
                ;4136    ;
U 040A, 0077,F3 ;4137            WORD[77F3]                          ; ячейка 368
U 040B, 0076,11 ;4138            WORD[7611]                          ;
                ;4139    ;
U 040C, 001D,10 ;4140            WORD[1D10]                          ; ячейка 36C
U 040D, 00DE,F2 ;4141            WORD[0DEF2]                         ;
                ;4142    ;
U 040E, 00DF,EB ;4143            WORD[0DFEB]                         ; ячейка 370
U 040F, 00E0,01 ;4144            WORD[0E001]                         ;
                ;4145    ;
U 0410, 00FD,E6 ;4146            WORD[0FDE6]                         ; ячейка 374
U 0411, 0000,00 ;4147            WORD[0000]                          ;
                ;4148    ;
U 0412, 00E1,E0 ;4149            WORD[0E1E0]                         ; ячейка 378
U 0413, 0000,2A ;4150            WORD[002A]                          ;
                ;4151    ;
U 0414, 0000,3D ;4152            WORD[003D]                          ; ячейка 37C
U 0415, 001A,00 ;4153            WORD[1A00]                          ;
                ;4154    ;
U 0416, 0076,2C ;4155            WORD[762C]                          ; ячейка 380
U 0417, 004B,43 ;4156            WORD[4B43]                          ;
                ;4157    ;
U 0418, 0054,4C ;4158            WORD[544C]                          ; ячейка 384
U 0419, 005E,56 ;4159            WORD[5E56]                          ;
                ;4160    ;
U 041A, 004C,6A ;4161            WORD[4C6A]                          ; ячейка 388
U 041B, 0000,00 ;4162            WORD[0000]                          ;
                ;4163    ;
U 041C, 00CD,7A ;4164            WORD[0CD7A]                         ; ячейка 38C
U 041D, 00C6,00 ;4165            WORD[0C600]                         ;
                ;4166    ;
U 041E, 0000,00 ;4167            WORD[0000]                          ; ячейка 390
U 041F, 008D,86 ;4168            WORD[8D86]                          ;
                ;4169    ;
U 0420, 007D,7E ;4170            WORD[7D7E]                          ; ячейка 394
```

```
U 0421, 00E3,D1 ;4171           WORD[0E3D1]                         ;
                ;4172     ;
U 0422, 0004,01 ;4173           WORD[0401]                          ; ячейка 398
U 0423, 0000,2A ;4174           WORD[002A]                          ;
                ;4175     ;
U 0424, 0000,3D ;4176           WORD[003D]                          ; ячейка 39C
U 0425, 0020,0C ;4177           WORD[200C]                          ;
                ;4178     ;
U 0426, 007D,37 ;4179           WORD[7D37]                          ; ячейка 3A0
U 0427, 0048,43 ;4180           WORD[4843]                          ;
                ;4181     ;
U 0428, 0054,4C ;4182           WORD[544C]                          ; ячейка 3A4
U 0429, 005E,56 ;4183           WORD[5E56]                          ;
                ;4184     ;
U 042A, 004C,6A ;4185           WORD[4C6A]                          ; ячейка 3A8
U 042B, 0000,00 ;4186           WORD[0000]                          ;
                ;4187     ;
U 042C, 00CD,7A ;4188           WORD[0CD7A]                         ; ячейка 3AC
U 042D, 00C6,00 ;4189           WORD[0C600]                         ;
                ;4190     ;
U 042E, 0000,00 ;4191           WORD[0000]                          ; ячейка 3B0
U 042F, 003C,3D ;4192           WORD[3C3D]                          ;
                ;4193     ;
U 0430, 00A0,81 ;4194           WORD[0A081]                         ; ячейка 3B4
U 0431, 00DE,DB ;4195           WORD[0DEDB]                         ;
                ;4196     ;
U 0432, 00DD,DC ;4197           WORD[0DDDC]                         ; ячейка 3B8
U 0433, 0000,2A ;4198           WORD[002A]                          ;
                ;4199     ;
U 0434, 0000,3D ;4200           WORD[003D]                          ; ячейка 3BC
U 0435, 0028,15 ;4201           WORD[2815]                          ;
                ;4202     ;
U 0436, 0088,47 ;4203           WORD[8847]                          ; ячейка 3C0
U 0437, 0048,43 ;4204           WORD[4843]                          ;
                ;4205     ;
U 0438, 0054,4C ;4206           WORD[544C]                          ; ячейка 3C4
U 0439, 005E,56 ;4207           WORD[5E56]                          ;
                ;4208     ;
U 043A, 004C,6A ;4209           WORD[4C6A]                          ; ячейка 3C8
U 043B, 0000,00 ;4210           WORD[0000]                          ;
                ;4211     ;
U 043C, 00CD,7A ;4212           WORD[0CD7A]                         ; ячейка 3CC
U 043D, 00C6,00 ;4213           WORD[0C600]                         ;
                ;4214     ;
U 043E, 0000,00 ;4215           WORD[0000]                          ; ячейка 3D0
U 043F, 00C8,C1 ;4216           WORD[0C8C1]                         ;
                ;4217     ;
U 0440, 00DF,E4 ;4218           WORD[0DFE4]                         ; ячейка 3D4
U 0441, 005F,00 ;4219           WORD[5F00]                          ;
                ;4220     ;
U 0442, 0004,00 ;4221           WORD[0400]                          ; ячейка 3D8
U 0443, 0069,64 ;4222           WORD[6964]                          ;
                ;4223     ;
U 0444, 007E,6E ;4224           WORD[7E6E]                          ; ячейка 3DC
U 0445, 0086,76 ;4225           WORD[8676]                          ;
```

```
                    ;4226     ;
U 0446,  0090,8E    ;4227             WORD[908E]                          ; ячейка 3E0
U 0447,  0005,02    ;4228             WORD[0502]                          ;
                    ;4229     ;
U 0448,  0022,21    ;4230             WORD[2221]                          ; ячейка 3E4
U 0449,  0013,14    ;4231             WORD[1314]                          ;
                    ;4232     ;
U 044A,  0008,09    ;4233             WORD[0809]                          ; ячейка 3E8
U 044B,  00E3,E5    ;4234             WORD[0E3E5]                         ;
                    ;4235     ;
U 044C,  0000,00    ;4236             WORD[0000]                          ; ячейка 3EC
U 044D,  0000,00    ;4237             WORD[0000]                          ;
                    ;4238     ;
U 044E,  00B1,A3    ;4239             WORD[0B1A3]                         ; ячейка 3F0
U 044F,  000D,13    ;4240             WORD[0D13]                          ;
                    ;4241
U 0450,  0000,00    ;4242             WORD[0000]                          ; ячейка 3F4
U 0451,  00E3,E2    ;4243             WORD[0E3E2]                         ;
                    ;4244     ;
U 0452,  00E5,E4    ;4245             WORD[0E5E4]                         ; ячейка 3F8
U 0453,  00BC,BC    ;4246             WORD[0BCBC]                         ;
                    ;4247     ;
U 0454,  00BC,BD    ;4248             WORD[0BCBD]                         ; ячейка 3FC
U 0455,  00D8,BE    ;4249             WORD[0D8BE]                         ;
                    ;4250     ;
U 0456,  00BC,BC    ;4251             WORD[0BCBC]                         ; ячейка 400
U 0457,  0079,79    ;4252             WORD[7979]                          ;
                    ;4253     ;
U 0458,  00BC,66    ;4254             WORD[0BC66]                         ; ячейка 404
U 0459,  0089,89    ;4255             WORD[8989]                          ;
                    ;4256     ;
U 045A,  0086,86    ;4257             WORD[8686]                          ; ячейка 408
U 045B,  00B0,C1    ;4258             WORD[0B0C1]                         ;
                    ;4259     ;
U 045C,  00AF,AF    ;4260             WORD[0AFAF]                         ; ячейка 40C
U 045D,  00AF,AF    ;4261             WORD[0AFAF]                         ;
                    ;4262     ;
U 045E,  00E6,ED    ;4263             WORD[0E6ED]                         ; ячейка 410
U 045F,  00AF,AF    ;4264             WORD[0AFAF]                         ;
                    ;4265     ;
U 0460,  00AF,AF    ;4266             WORD[0AFAF]                         ; ячейка 414
U 0461,  00AF,AF    ;4267             WORD[0AFAF]                         ;
                    ;4268     ;
U 0462,  00AF,AF    ;4269             WORD[0AFAF]                         ; ячейка 418
U 0463,  0079,79    ;4270             WORD[7979]                          ;
                    ;4271     ;
U 0464,  0079,79    ;4272             WORD[7979]                          ; ячейка 41C
U 0465,  0079,79    ;4273             WORD[7979]                          ;
                    ;4274     ;
U 0466,  0079,79    ;4275             WORD[7979]                          ; ячейка 420
U 0467,  00BC,BC    ;4276             WORD[BCBC]                          ;
                    ;4277     ;
U 0468,  00BC,BC    ;4278             WORD[BCBC]                          ; ячейка 424
U 0469,  00BC,BC    ;4279             WORD[BCBC]                          ;
                    ;4280     ;
```

```
U 046A, 00BC,BC ;4281            WORD[BC8C]                          ; ячейка 428
U 046B, 00B5,C2 ;4282            WORD[0B5C2]                         ;
                ;4283     ;
U 046C, 0073,73 ;4284            WORD[7373]                          ; ячейка 42C
U 046D, 0079,79 ;4285            WORD[7979]                          ;
                ;4286     ;
U 046E, 0079,79 ;4287            WORD[7979]                          ; ячейка 430
U 046F, 008C,BC ;4288            WORD[8CBC]                          ;
                ;4289     ;
U 0470, 008C,8C ;4290            WORD[8C8C]                          ; ячейка 434
U 0471, 0043,5C ;4291            WORD[435C]                          ;
                ;4292     ;
U 0472, 0086,86 ;4293            WORD[8686]                          ; ячейка 438
U 0473, 001D,0B ;4294            WORD[1D0B]                          ;
                ;4295     ;
U 0474, 001D,0B ;4296            WORD[1D0B]                          ; ячейка 43C
U 0475, 001D,0B ;4297            WORD[1D0B]                          ;
                ;4298
U 0476, 001D,0B ;4299            WORD[1D0B]                          ; ячейка 440
U 0477, 004C,4C ;4300            WORD[4C4C]                          ;
                ;4301     ;
U 0478, 004C,4C ;4302            WORD[4C4C]                          ; ячейка 444
U 0479, 008D,8D ;4303            WORD[8D8D]                          ;
                ;4304     ;
U 047A, 0059,8D ;4305            WORD[598D]                          ; ячейка 44C
U 047B, 004C,59 ;4306            WORD[4C59]                          ;
                ;4307     ;
U 047C, 0059,59 ;4308            WORD[5959]                          ; ячейка 450
U 047D, 00BC,4C ;4309            WORD[0BC4C]                         ;
                ;4310     ;
U 047E, 00BC,4C ;4311            WORD[0BC4C]                         ; ячейка 454
U 047F, 00BC,BC ;4312            WORD[0BCBC]                         ;
                ;4313     ;
U 0480, 00BC,BC ;4314            WORD[0BCBC]                         ; ячейка 450
U 0481, 0086,86 ;4315            WORD[8686]                          ;
                ;4316     ;
U 0482, 00BC,BC ;4317            WORD[0BCBC]                         ; ячейка 458
U 0483, 003C,2B ;4318            WORD[3C2B]                          ;
                ;4319     ;
U 0484, 003C,2B ;4320            WORD[3C2B]                          ; ячейка 45C
U 0485, 003C,2B ;4321            WORD[3C2B]                          ;
                ;4322     ;
U 0486, 003C,2B ;4323            WORD[3C2B]                          ; ячейка 460
U 0487, 0062,62 ;4324            WORD[6262]                          ;
                ;4325     ;
U 0488, 0062,62 ;4326            WORD[6262]                          ; ячейка 464
U 0489, 008D,8D ;4327            WORD[8D8D]                          ;
                ;4328     ;
U 048A, 0071,8D ;4329            WORD[718D]                          ; ячейка 468
U 048B, 0062,71 ;4330            WORD[6271]                          ;
                ;4331     ;
U 048C, 0071,71 ;4332            WORD[7171]                          ; ячейка 46C
U 048D, 00BC,62 ;4333            WORD[0BC62]                         ;
                ;4334     ;
U 048E, 00BC,62 ;4335            WORD[0BC62]                         ; ячейка 470
```

```
U 048F, 0089,66 ;4336           WORD[8966]                              ;
                ;4337     ;
U 0490, 0089,89 ;4338           WORD[8989]                              ; ячейка 474
U 0491, 0097,00 ;4339           WORD[9700]                              ;
                ;4340     ;
U 0492, 00BC,BC ;4341           WORD[0BCBC]                             ; ячейка 478
U 0493, 00C4,04 ;4342           WORD[0C404]                             ;
                ;4343     ;
U 0494, 00CE,04 ;4344           WORD[0CE04]                             ; ячейка 47C
U 0495, 0043,24 ;4345           WORD[4324]                              ;
                ;4346     ;
U 0496, 0043,24 ;4347           WORD[4324]                              ; ячейка 480
U 0497, 0033,04 ;4348           WORD[3304]                              ;
                ;4349     ;
U 0498, 0033,04 ;4350           WORD[3304]                              ; ячейка 484
U 0499, 0033,04 ;4351           WORD[3304]                              ;
                ;4352     ;
U 049A, 0066,5C ;4353           WORD[665C]                              ; ячейка 488
U 049B, 0098,91 ;4354           WORD[9891]                              ;
                ;4355     ;
U 049C, 0033,5C ;4356           WORD[335C]                              ; ячейка 48C
U 049D, 005C,AA ;4357           WORD[5CAA]                              ;
                ;4358     ;
U 049E, 00A5,A0 ;4359           WORD[0A5A0]                             ; ячейка 490
U 049F, 006B,6B ;4360           WORD[6B6B]                              ;
                ;4361     ;
U 04A0, 005C,5C ;4362           WORD[5C5C]                              ; ячейка 494
U 04A1, 0043,5C ;4363           WORD[435C]                              ;
                ;4364     ;
U 04A2, 0086,86 ;4365           WORD[8686]                              ; ячейка 498
U 04A3, 00C4,04 ;4366           WORD[0C404]                             ;
                ;4367     ;
U 04A4, 00CE,04 ;4368           WORD[0CE04]                             ; ячейка 49C
U 04A5, 0043,24 ;4369           WORD[4324]                              ;
                ;4370     ;
U 04A6, 0043,24 ;4371           WORD[4324]                              ; ячейка 4A0
U 04A7, 0033,04 ;4372           WORD[3304]                              ;
                ;4373     ;
U 04A8, 0033,04 ;4374           WORD[3304]                              ; ячейка 4A4
U 04A9, 0033,04 ;4375           WORD[3304]                              ;
                ;4376     ;
U 04AA, 0066,5C ;4377           WORD[665C]                              ; ячейка 4A8
U 04AB, 0098,91 ;4378           WORD[9891]                              ;
                ;4379     ;
U 04AC, 0033,5C ;4380           WORD[335C]                              ; ячейка 4AC
U 04AD, 005C,AA ;4381           WORD[5CAA]                              ;
                ;4382     ;
U 04AE, 00A5,A0 ;4383           WORD[0A5A0]                             ; ячейка 4B0
U 04AF, 0066,66 ;4384           WORD[6666]                              ;
                ;4385     ;
U 04B0, 0066,66 ;4386           WORD[6666]                              ; ячейка 4B4
U 04B1, 0089,89 ;4387           WORD[8989]                              ;
                ;4388     ;
U 04B2, 0089,89 ;4389           WORD[8989]                              ; ячейка 4B8
U 04B3, 00C4,04 ;4390           WORD[0C404]
```

```
                    ;4391     ;
U 04B4, 00CE,04 ;4392               WORD[0CE04]                         ; ячейка 4BC
U 04B5, 0050,14 ;4393               WORD[5014]                          ;
                    ;4394     ;
U 04B6, 0050,14 ;4395               WORD[5014]                          ; ячейка 4C0
U 04B7, 0033,04 ;4396               WORD[3304]                          ;
                    ;4397     ;
U 04B8, 0033,04 ;4398               WORD[3304]                          ; ячейка 4C4
U 04B9, 0033,04 ;4399               WORD[3304]                          ;
                    ;4400     ;
U 04BA, 0066,5C ;4401               WORD[665C]                          ; ячейка 4C8
U 04BB, 0098,91 ;4402               WORD[9891]                          ;
                    ;4403
U 04BC, 0033,5C ;4404               WORD[335C]                          ; ячейка 4CC
U 04BD, 005C,AA ;4405               WORD[5CAA]                          ;
                    ;4406
U 04BE, 00A5,A0 ;4407               WORD[0A5A0]                         ; ячейка 4D0
U 04BF, 0004,04 ;4408               WORD[0404]                          ;
                    ;4409
U 04C0, 0089,43 ;4410               WORD[8943]                          ; ячейка 4D4
U 04C1, 005C,C3 ;4411               WORD[5CC3]                          ;
                    ;4412     ;
U 04C2, 0086,DF ;4413               WORD[86DF]                          ; ячейка 4D8
U 04C3, 007A,7A ;4414               WORD[7A7A]                          ;
                    ;4415
U 04C4, 007A,7A ;4416               WORD[7A7A]                          ; ячейка 4DC
U 04C5, 007A,7A ;4417               WORD[7A7A]                          ;
                    ;4418
U 04C6, 007A,7A ;4419               WORD[7A7A]                          ; ячейка 4E0
U 04C7, 0066,66 ;4420               WORD[6666]
                    ;4421     ;
U 04C8, 0066,66 ;4422               WORD[6666]                          ; ячейка 4E4
U 04C9, 0066,66 ;4423               WORD[6666]                          ;
                    ;4424
U 04CA, 0066,66 ;4425               WORD[6666]                          ; ячейка 4E8
U 04CB, 0043,43 ;4426               WORD[4343]                          ;
                    ;4427     ;
U 04CC, 00F9,FA ;4428               WORD[0F9FA]                         ; ячейка 4EC
U 04CD, 00C0,BF ;4429               WORD[0C0BF]                         ;
                    ;4430
U 04CE, 00B0,B0 ;4431               WORD[B0B0]                          ; ячейка 4F0
U 04CF, 0079,79 ;4432               WORD[7979]                          ;
                    ;4433
U 04D0, 0066,86 ;4434               WORD[6686]                          ; ячейка 4F4
U 04D1, 00BC,BC ;4435               WORD[0BCBC]                         ;
                    ;4436     ;
U 04D2, 00BC,BC ;4437               WORD[0BCBC]                         ; ячейка 4F8
U 04D3, 0000,00 ;4438               WORD[0000]                          ;
                    ;4439     ;
U 04D4, 0000,00 ;4440               WORD[0000]                          ; ячейка 4FC
U 04D5, 0000,00 ;4441               WORD[0000]                          ;
                    ;4442
U 04D6, 0003,08 ;4443               WORD[0308]                          ; ячейка 500
U 04D7, 0007,02 ;4444               WORD[0702]                          ;
                    ;4445
```

```
U 04D8, 000C,01 ;4446           WORD[0C01]                          ; ячейка 504
U 04D9, 0003,03 ;4447           WORD[0303]                          ;
                ;4448     ;
U 04DA, 0002,02 ;4449           WORD[0202]                          ; ячейка 508
U 04DB, 0000,04 ;4450           WORD[0004]                          ;
                ;4451     ;
U 04DC, 0004,02 ;4452           WORD[0402]                          ; ячейка 50C
U 04DD, 0000,02 ;4453           WORD[0002]                          ;
                ;4454     ;
U 04DE, 0006,02 ;4455           WORD[0602]                          ; ячейка 510
U 04DF, 0001,02 ;4456           WORD[0102]                          ;
                ;4457     ;
U 04E0, 0002,02 ;4458           WORD[0202]                          ; ячейка 514
U 04E1, 0003,02 ;4459           WORD[0302]                          ;
                ;4460
U 04E2, 0004,04 ;4461           WORD[0404]                          ; ячейка 518
U 04E3, 0007,01 ;4462           WORD[0701]                          ;
                ;4463     ;
U 04E4, 0004,01 ;4464           WORD[0401]                          ; ячейка 51C
U 04E5, 0007,01 ;4465           WORD[0701]                          ;
                ;4466     ;
U 04E6, 0004,01 ;4467           WORD[0401]                          ; ячейка 520
U 04E7, 0007,01 ;4468           WORD[0701]                          ;
                ;4469     ;
U 04E8, 0006,01 ;4470           WORD[0601]                          ; ячейка 524
U 04E9, 0007,02 ;4471           WORD[0702]                          ;
                ;4472     ;
U 04EA, 0006,04 ;4473           WORD[0604]                          ; ячейка 528
U 04EB, 0000,02 ;4474           WORD[0002]                          ;
                ;4475     ;
U 04EC, 0004,02 ;4476           WORD[0402]                          ; ячейка 52C
U 04ED, 0007,01 ;4477           WORD[0701]                          ;
                ;4478     ;
U 04EE, 0004,01 ;4479           WORD[0401]                          ; ячейка 530
U 04EF, 0007,01 ;4480           WORD[0701]                          ;
                ;4481
U 04F0, 0004,02 ;4482           WORD[0402]                          ; ячейка 534
U 04F1, 0007,01 ;4483           WORD[0701]                          ;
                ;4484     ;
U 04F2, 0003,02 ;4485           WORD[0302]                          ; ячейка 538
U 04F3, 0007,04 ;4486           WORD[0704]                          ;
                ;4487
U 04F4, 000C,0C ;4488           WORD[0C0C]                          ; ячейка 53C
U 04F5, 0000,01 ;4489           WORD[0001]                          ;
                ;4490     ;
U 04F6, 0004,01 ;4491           WORD[0401]                          ; ячейка 540
U 04F7, 000C,01 ;4492           WORD[0C01]                          ;
                ;4493     ;
U 04F8, 000F,02 ;4494           WORD[0F02]                          ; ячейка 544
U 04F9, 000C,06 ;4495           WORD[0C06]                          ;
                ;4496     ;
U 04FA, 000F,01 ;4497           WORD[0F01]                          ; ячейка 548
U 04FB, 0004,01 ;4498           WORD[0401]                          ;
                ;4499     ;
U 04FC, 000F,03 ;4500           WORD[0F03]                          ; ячейка 54C
```

```
U 04FD, 0000,04 ;4501           WORD[0004]                              ;
                ;4502     ;
U 04FE, 000C,0C ;4503           WORD[0C0C]                              ; ячейка 550
U 04FF, 0000,01 ;4504           WORD[0001]                              ;
                ;4505     ;
U 0500, 0004,01 ;4506           WORD[0401]                              ; ячейка 554
U 0501, 000C,01 ;4507           WORD[0C01]                              ;
                ;4508     ;
U 0502, 000F,02 ;4509           WORD[0F02]                              ; ячейка 558
U 0503, 000C,06 ;4510           WORD[0C06]                              ;
                ;4511     ;
U 0504, 000F,01 ;4512           WORD[0F01]                              ; ячейка 55C
U 0505, 0000,02 ;4513           WORD[0002]                              ;
                ;4514     ;
U 0506, 000C,02 ;4515           WORD[0C02]                              ; ячейка 560
U 0507, 000F,04 ;4516           WORD[0F04]                              ;
                ;4517     ;
U 0508, 0000,02 ;4518           WORD[0002]                              ; ячейка 564
U 0509, 0001,02 ;4519           WORD[0102]                              ;
                ;4520     ;
U 050A, 0000,04 ;4521           WORD[0004]                              ; ячейка 568
U 050B, 0003,06 ;4522           WORD[0306]                              ;
                ;4523     ;
U 050C, 0000,01 ;4524           WORD[0001]                              ; ячейка 56C
U 050D, 0002,01 ;4525           WORD[0201]                              ;
                ;4526     ;
U 050E, 0003,01 ;4527           WORD[0301]                              ; ячейка 570
U 050F, 0002,01 ;4528           WORD[0201]                              ;
                ;4529     ;
U 0510, 0003,03 ;4530           WORD[0303]                              ; ячейка 574
U 0511, 0000,02 ;4531           WORD[0002]                              ;
                ;4532     ;
U 0512, 0001,01 ;4533           WORD[0101]                              ; ячейка 578
U 0513, 0000,02 ;4534           WORD[0002]                              ;
                ;4535     ;
U 0514, 0003,06 ;4536           WORD[0306]                              ; ячейка 57C
U 0515, 0004,02 ;4537           WORD[0402]                              ;
                ;4538
U 0516, 0005,02 ;4539           WORD[0502]                              ; ячейка 580
U 0517, 0004,04 ;4540           WORD[0404]                              ;
                ;4541     ;
U 0518, 0007,06 ;4542           WORD[0706]                              ; ячейка 584
U 0519, 0004,01 ;4543           WORD[0401]                              ;
                ;4544     ;
U 051A, 0006,01 ;4545           WORD[0601]                              ; ячейка 588
U 051B, 0007,01 ;4546           WORD[0701]                              ;
                ;4547     ;
U 051C, 0006,01 ;4548           WORD[0601]                              ; ячейка 58C
U 051D, 0007,03 ;4549           WORD[0703]                              ;
                ;4550     ;
U 051E, 0004,02 ;4551           WORD[0402]                              ; ячейка 590
U 051F, 0005,01 ;4552           WORD[0501]                              ;
                ;4553     ;
U 0520, 0004,02 ;4554           WORD[0402]                              ; ячейка 594
U 0521, 0007,06 ;4555           WORD[0706]                              ;
```

```
                    ;4556     ;
U 0522, 000C,02     ;4557           WORD[0C02]                      ; ячейка 598
U 0523, 000D,02     ;4558           WORD[0D02]                      ;
                    ;4559     ;
U 0524, 000C,04     ;4560           WORD[0C04]                      ; ячейка 59C
U 0525, 000F,06     ;4561           WORD[0F06]                      ;
                    ;4562     ;
U 0526, 000C,01     ;4563           WORD[0C01]                      ; ячейка 5A0
U 0527, 000E,01     ;4564           WORD[0E01]                      ;
                    ;4565     ;
U 0528, 000F,01     ;4566           WORD[0F01]                      ; ячейка 5A4
U 0529, 000E,01     ;4567           WORD[0E01]                      ;
                    ;4568     ;
U 052A, 000F,03     ;4569           WORD[0F03]                      ; ячейка 5A8
U 052B, 000C,02     ;4570           WORD[0C02]                      ;
                    ;4571     ;
U 052C, 000D,01     ;4572           WORD[0D01]                      ; ячейка 5AC
U 052D, 000C,01     ;4573           WORD[0C01]                      ;
                    ;4574     ;
U 052E, 000D,01     ;4575           WORD[0D01]                      ; ячейка 5B0
U 052F, 000F,10     ;4576           WORD[0F10]                      ;
                    ;4577     ;
U 0530, 000C,02     ;4578           WORD[0C02]                      ; ячейка 5B4
U 0531, 000E,02     ;4579           WORD[0E02]                      ;
                    ;4580     ;
U 0532, 000F,02     ;4581           WORD[0F02]                      ; ячейка 5B8
U 0533, 000C,01     ;4582           WORD[0C01]                      ;
                    ;4583     ;
U 0534, 000F,05     ;4584           WORD[0F05]                      ; ячейка 5BC
U 0535, 000C,02     ;4585           WORD[0C02]                      ;
                    ;4586     ;
U 0536, 0000,01     ;4587           WORD[0001]                      ; ячейка 5C0
U 0537, 000F,01     ;4588           WORD[0F01]                      ;
                    ;4589     ;
U 0538, 0003,01     ;4590           WORD[0301]                      ; ячейка 5C4
U 0539, 000F,05     ;4591           WORD[0F05]                      ;
                    ;4592     ;
U 053A, 0007,01     ;4593           WORD[0701]                      ; ячейка 5C8
U 053B, 0000,03     ;4594           WORD[0003]                      ;
                    ;4595     ;
U 053C, 0007,02     ;4596           WORD[0702]                      ; ячейка 5CC
U 053D, 0005,02     ;4597           WORD[0502]                      ;
                    ;4598     ;
U 053E, 0007,03     ;4599           WORD[0703]                      ; ячейка 5D0
U 053F, 0005,01     ;4600           WORD[0501]                      ;
                    ;4601     ;
U 0540, 0004,01     ;4602           WORD[0401]                      ; ячейка 5D4
U 0541, 0007,13     ;4603           WORD[0713]                      ;
                    ;4604     ;
U 0542, 0005,10     ;4605           WORD[0510]                      ; ячейка 5D8
U 0543, 0007,30     ;4606           WORD[0730]                      ;
                    ;4607     ;
U 0544, 0004,10     ;4608           WORD[0410]                      ; ячейка 5DC
U 0545, 0007,10     ;4609           WORD[0710]                      ;
                    ;4610     ;
```

```
U 0546, 0006,02 ;4611           WORD[0602]                      ; ячейка 5E0
U 0547, 0001,02 ;4612           WORD[0102]                      ;
                ;4613     ;
U 0548, 0002,02 ;4614           WORD[0202]                      ; ячейка 5E4
U 0549, 0003,02 ;4615           WORD[0302]                      ;
                ;4616     ;
U 054A, 0007,02 ;4617           WORD[0702]                      ; ячейка 5E8
U 054B, 0003,01 ;4618           WORD[0301]                      ;
                ;4619     ;
U 054C, 0001,01 ;4620           WORD[0101]                      ; ячейка 5EC
U 054D, 0000,01 ;4621           WORD[0001]                      ;
                ;4622     ;
U 054E, 0007,03 ;4623           WORD[0703]                      ; ячейка 5F0
U 054F, 0003,10 ;4624           WORD[0310]                      ;
                ;4625     ;
U 0550, 0001,10 ;4626           WORD[0110]                      ; ячейка 5F4
U 0551, 0003,30 ;4627           WORD[0330]                      ;
                ;4628     ;
U 0552, 0005,10 ;4629           WORD[0510]                      ; ячейка 5F8
U 0553, 0007,10 ;4630           WORD[0710]                      ;
                ;4631     ;
U 0554, 0000,00 ;4632           WORD[0000]                      ; ячейка 5FC
U 0555, 0000,00 ;4633           WORD[0000]                      ;
                ;4634     ;
U 0556, 0000,00 ;4635           WORD[0000]                      ; ячейка 600
U 0557, 0000,00 ;4636           WORD[0000]                      ;
                ;4637     ;
U 0558, 0000,00 ;4638           WORD[0000]                      ; ячейка 604
U 0559, 0000,00 ;4639           WORD[0000]                      ;
                ;4640     ;
U 055A, 0000,00 ;4641           WORD[0000]                      ; ячейка 608
U 055B, 0000,00 ;4642           WORD[0000]                      ;
                ;4643     ;
U 055C, 0000,00 ;4644           WORD[0000]                      ; ячейка 60C
U 055D, 0000,00 ;4645           WORD[0000]                      ;
                ;4646     ;
U 055E, 0000,00 ;4647           WORD[0000]                      ; ячейка 610
U 055F, 0000,00 ;4648           WORD[0000]                      ;
                ;4649     ;
U 0560, 0000,00 ;4650           WORD[0000]                      ; ячейка 614
U 0561, 0000,00 ;4651           WORD[0000]                      ;
                ;4652     ;
U 0562, 0000,00 ;4653           WORD[0000]                      ; ячейка 618
U 0563, 0000,00 ;4654           WORD[0000]                      ;
                ;4655
U 0564, 0000,00 ;4656           WORD[0000]                      ; ячейка 61C
U 0565, 0000,00 ;4657           WORD[0000]                      ;
                ;4658     ;
U 0566, 0000,00 ;4659           WORD[0000]                      ; ячейка 620
U 0567, 0000,00 ;4660           WORD[0000]                      ;
                ;4661     ;
U 0568, 0000,00 ;4662           WORD[0000]                      ; ячейка 624
U 0569, 0000,00 ;4663           WORD[0000]                      ;
                ;4664     ;
U 056A, 0000,00 ;4665           WORD[0000]                      ; ячейка 628
```

```
U 056B, 0000,00 ;4666           WORD[0000]                              ;
                ;4667     ;
U 056C, 0000,00 ;4668           WORD[0000]                              ; ячейка 62C
U 056D, 0000,00 ;4669           WORD[0000]                              ;
                ;4670     ;
U 056E, 0000,00 ;4671           WORD[0000]                              ; ячейка 630
U 056F, 0000,00 ;4672           WORD[0000]                              ;
                ;4673     ;
U 0570, 0000,00 ;4674           WORD[0000]                              ; ячейка 634
U 0571, 0000,00 ;4675           WORD[0000]                              ;
                ;4676     ;
U 0572, 0000,00 ;4677           WORD[0000]                              ; ячейка 638
U 0573, 0000,00 ;4678           WORD[0000]                              ;
                ;4679     ;
U 0574, 0000,00 ;4680           WORD[0000]                              ; ячейка 63C
U 0575, 0000,00 ;4681           WORD[0000]                              ;
                ;4682     ;
U 0576, 0000,00 ;4683           WORD[0000]                              ; ячейка 640
U 0577, 0000,00 ;4684           WORD[0000]                              ;
                ;4685     ;
U 0578, 0000,00 ;4686           WORD[0000]                              ; ячейка 644
U 0579, 0000,00 ;4687           WORD[0000]                              ;
                ;4688     ;
U 057A, 0000,00 ;4689           WORD[0000]                              ; ячейка 648
U 057B, 0000,00 ;4690           WORD[0000]                              ;
                ;4691     ;
U 057C, 0000,00 ;4692           WORD[0000]                              ; ячейка 64C
U 057D, 0000,00 ;4693           WORD[0000]                              ;
                ;4694     ;
U 057E, 0000,00 ;4695           WORD[0000]                              ; ячейка 650
U 057F, 0000,00 ;4696           WORD[0000]                              ;
                ;4697     ;
U 0580, 0000,00 ;4698           WORD[0000]                              ; ячейка 654
U 0581, 0000,00 ;4699           WORD[0000]                              ;
                ;4700     ;
U 0582, 0000,00 ;4701           WORD[0000]                              ; ячейка 658
U 0583, 0000,00 ;4702           WORD[0000]                              ;
                ;4703     ;
U 0584, 0000,00 ;4704           WORD[0000]                              ; ячейка 65C
U 0585, 0000,00 ;4705           WORD[0000]                              ;
                ;4706     ;
U 0586, 0000,00 ;4707           WORD[0000]                              ; ячейка 660
U 0587, 0000,00 ;4708           WORD[0000]                              ;
                ;4709     ;
U 0588, 0000,00 ;4710           WORD[0000]                              ; ячейка 664
U 0589, 0000,00 ;4711           WORD[0000]                              ;
                ;4712     ;
U 058A, 0000,00 ;4713           WORD[0000]                              ; ячейка 668
U 058B, 0000,00 ;4714           WORD[0000]                              ;
                ;4715     ;
U 058C, 0000,00 ;4716           WORD[0000]                              ; ячейка 66C
U 058D, 0000,00 ;4717           WORD[0000]                              ;
                ;4718     ;
U 058E, 0000,00 ;4719           WORD[0000]                              ; ячейка 670
U 058F, 0000,00 ;4720           WORD[0000]                              ;
```

```
                    ;4721     ;
U 0590,  0000,00    ;4722               WORD[0000]                          ; ячейка 674
U 0591,  0000,00    ;4723               WORD[0000]                          ;
                    ;4724     ;
U 0592,  0000,00    ;4725               WORD[0000]                          ; ячейка 678
U 0593,  0000,00    ;4726               WORD[0000]                          ;
                    ;4727     ;
U 0594,  0000,00    ;4728               WORD[0000]                          ; ячейка 67C
U 0595,  0000,00    ;4729               WORD[0000]                          ;
                    ;4730     ;
U 0596,  0000,00    ;4731               WORD[0000]                          ; ячейка 680
U 0597,  0000,00    ;4732               WORD[0000]                          ;
                    ;4733     ;
U 0598,  0000,00    ;4734               WORD[0000]                          ; ячейка 684
U 0599,  0000,00    ;4735               WORD[0000]                          ;
                    ;4736     ;
U 059A,  0000,00    ;4737               WORD[0000]                          ; ячейка 688
U 059B,  0000,00    ;4738               WORD[0000]                          ;
                    ;4739     ;
U 059C,  0000,00    ;4740               WORD[0000]                          ; ячейка 68C
U 059D,  0000,00    ;4741               WORD[0000]                          ;
                    ;4742     ;
U 059E,  0000,00    ;4743               WORD[0000]                          ; ячейка 690
U 059F,  0000,00    ;4744               WORD[0000]                          ;
                    ;4745     ;
U 05A0,  0000,00    ;4746               WORD[0000]                          ; ячейка 694
U 05A1,  0000,00    ;4747               WORD[0000]                          ;
                    ;4748     ;
U 05A2,  0000,00    ;4749               WORD[0000]                          ; ячейка 698
U 05A3,  0000,00    ;4750               WORD[0000]                          ;
                    ;4751     ;
U 05A4,  0000,00    ;4752               WORD[0000]                          ; ячейка 69C
U 05A5,  0000,00    ;4753               WORD[0000]                          ;
                    ;4754     ;
U 05A6,  0000,00    ;4755               WORD[0000]                          ; ячейка 6A0
U 05A7,  0000,00    ;4756               WORD[0000]                          ;
                    ;4757     ;
U 05A8,  0000,00    ;4758               WORD[0000]                          ; ячейка 6A4
U 05A9,  0000,00    ;4759               WORD[0000]                          ;
                    ;4760     ;
U 05AA,  0000,00    ;4761               WORD[0000]                          ; ячейка 6A8
U 05AB,  0000,00    ;4762               WORD[0000]                          ;
                    ;4763     ;
U 05AC,  0000,00    ;4764               WORD[0000]                          ; ячейка 6AC
U 05AD,  0000,00    ;4765               WORD[0000]                          ;
                    ;4766     ;
U 05AE,  0000,00    ;4767               WORD[0000]                          ; ячейка 6B0
U 05AF,  0000,00    ;4768               WORD[0000]                          ;
                    ;4769     ;
U 05B0,  0000,00    ;4770               WORD[0000]                          ; ячейка 6B4
U 05B1,  0000,00    ;4771               WORD[0000]                          ;
                    ;4772     ;
U 05B2,  0000,00    ;4773               WORD[0000]                          ; ячейка 6B8
U 05B3,  0000,00    ;4774               WORD[0000]                          ;
                    ;4775     ;
```

```
U 05B4, 0000,00 ;4776            WORD[0000]                          ; ячейка 6BC
U 05B5, 0000,00 ;4777            WORD[0000]                          ;
                ;4778    ;
U 05B6, 0000,00 ;4779            WORD[0000]                          ; ячейка 6C0
U 05B7, 0000,00 ;4780            WORD[0000]                          ;
                ;4781    ;
U 05B8, 0000,00 ;4782            WORD[0000]                          ; ячейка 6C4
U 05B9, 0000,00 ;4783            WORD[0000]                          ;
                ;4784    ;
U 05BA, 0000,00 ;4785            WORD[0000]                          ; ячейка 6CC
U 05BB, 0000,00 ;4786            WORD[0000]                          ;
                ;4787    ;
U 05BC, 0000,00 ;4788            WORD[0000]                          ; ячейка 6D0
U 05BD, 0000,00 ;4789            WORD[0000]                          ;
                ;4790    ;
U 05BE, 0000,00 ;4791            WORD[0000]                          ; ячейка 6D4
U 05BF, 0000,00 ;4792            WORD[0000]                          ;
                ;4793    ;
U 05C0, 0000,00 ;4794            WORD[0000]                          ; ячейка 6D8
U 05C1, 0000,00 ;4795            WORD[0000]                          ;
                ;4796
U 05C2, 0000,00 ;4797            WORD[0000]                          ; ячейка 6DC
U 05C3, 0000,00 ;4798            WORD[0000]                          ;
                ;4799    ;
U 05C4, 0000,00 ;4800            WORD[0000]                          ; ячейка 6E0
U 05C5, 0000,00 ;4801            WORD[0000]                          ;
                ;4802    ;
U 05C6, 0000,00 ;4803            WORD[0000]                          ; ячейка 6E4
U 05C7, 0000,00 ;4804            WORD[0000]                          ;
                ;4805    ;
U 05C8, 0000,00 ;4806            WORD[0000]                          ; ячейка 6E8
U 05C9, 0000,00 ;4807            WORD[0000]                          ;
                ;4808    ;
U 05CA, 0000,00 ;4809            WORD[0000]                          ; ячейка 6EC
U 05CB, 0000,00 ;4810            WORD[0000]                          ;
                ;4811    ;
U 05CC, 0000,00 ;4812            WORD[0000]                          ; ячейка 6F0
U 05CD, 0000,00 ;4813            WORD[0000]                          ;
                ;4814    ;
U 05CE, 0000,00 ;4815            WORD[0000]                          ; ячейка 6F4
U 05CF, 0000,00 ;4816            WORD[0000]                          ;
                ;4817    ;
U 05D0, 0000,00 ;4818            WORD[0000]                          ; ячейка 6F8
U 05D1, 0000,00 ;4819            WORD[0000]                          ;
                ;4820
U 05D2, 0000,00 ;4821            WORD[0000]
U 05D3, 0000,00 ;4822            WORD[0000]
                ;4823
U 05D4, 0000,00 ;4824            WORD[0000]
U 05D5, 0000,00 ;4825            WORD[0000]
                ;4826    ;
                ;4827      .REGION/600,6FF
```

```
                ;4828     .PAGE "ПОДПРОГРАММЫ В УПРАВЛЯЮЩЕЙ ПАМЯТИ (WCS), ИСПОЛЬЗУЕМЫЕ МИКРОМОНИТОРОМ"
                ;4829     ;
                ;4830     ;                        Программа генерации данных
                ;4831     ;
                ;4832     ;           Эта программа используется диагностическим монитором для загрузки рабо-
                ;4833     ;         чего регистра WR0 кодом данных из консольного процесора. Монитор устанав-
                ;4834     ;         ливает CONSOLE ATTN, если должна генерироваться единица, или очищает
                ;4835     ;         CONSOLE ATTN, если должен генерироваться 0. Затем он проходит эту программу
                ;4836     ;         один раз с целью вдвигания бита данных в WR0. Эта программа загружает 32-
                ;4837     ;         битовые значения намного быстрее, чем было бы при вдвигании каждой инструк-
                ;4838     ;         ции в CSR из монитора, и используется, главным образом, для формирования
                ;4839     ;         данных, подлежащих загрузке в LS в качестве констант.
                ;4840     ;
                ;4841     ;                             *** ПРЕДУПРЕЖДЕНИЕ ***
                ;4842     ;
                ;4843     ;         Эта программа должна начинаться адресом 600 и заканчиваться 605.
                ;4844     ;
                ;4845     ;
                ;4846     GEN.XFER.DATA:
U 0600,  2F80,15 ;4847             CLR WR[0]                                ; очистка рабочего регистра
U 0601,  A0C0,15 ;4848             COM WR[0]                                ; рабочий регистр WR0 содержит все единицы
                ;4849     LOOP.XFER:
U 0602,  5B00,18 ;4850             SKIP.IF[CONSOLE.ATTN]                    ; пропуск следующей инструкции, если установлен CONSOLE
                ;4851                                                      ; ATTN (генерация единицы)
                ;4852               ASHL WR[0],                            ; вдвигание 0 в бит 0 WR0
U 0603,  A3D0,1E ;4853             SKIP                                     ; пропуск инструкции ROL
U 0604,  A3C0,15 ;4854             ROL WR[0]                                ; вдвигание единицы в бит 0 WR0
U 0605,  0B60,24 ;4855             JMP [LOOP.XFER]                          ; повторение
                ;4856     ;
                ;4857     ;         Здесь хранится номер версии диагностики
                ;4858     ;
                ;4859     ;                             *** ПРЕДУПРЕЖДЕНИЕ ***
                ;4860     ;
                ;4861     ;         Эти слова должны быть в ячейках 606 и 607.
                ;4862     ;
                ;4863     ;
U 0606,  0003,00 ;4864             WORD[0300]                               ; версия 03.00
U 0607,  0043,44 ;4865             ASCII [CD]                               ; последние две буквы имени файла [ENKCD]
                ;4866
                ;4867     ;              Программа записи в регистр управления и состояния МСТ
                ;4868     ;***
                ;4869     ;
                ;4870     ; Описание функционирования:
                ;4871     ;
                ;4872     ;           Программа записи CSR используется для записи в регистры управления и
                ;4873     ;         состояния контроллера памяти. Контроллер памяти имеет три регистра уп-
                ;4874     ;         равления и состояния, идентифицируемые как CSR0, CSR1 и CSR2.
                ;4875     ;           CSR0 содержит конрольные биты или биты синдрома, полученные из
                ;4876     ;         логических схем корректирующего кода (ECC) после определенных программ
                ;4877     ;         диагностирования. Его нельзя непосредственно записывать при помощи этой
                ;4878     ;         программы.
                ;4879     ;           CSR1 содержит биты ошибок, установленные, если произошла ошибка во
                ;4880     ;         время обращения к памяти. Он содержит 5 битов управления (29-25),
                ;4881     ;         допускающих запись. имеется также 7 проверочных битов (биты 6-0),
                ;4882     ;         которые допускают запись, но не считываются. Эти биты используются для
```

```
                ;4883       ;       записи контрольных битов в микросхемы кодов коррекции (ECC).
                ;4884       ;         CSR2 содержит биты ошибок, установленные, если произошла ошибка во
                ;4885       ;       время обращения к общей шине. Эти биты только считываются и не могут
                ;4886       ;       записываться данной программой.
                ;4887       ;         Поэтому CSR1 является единственным регистром управления и состояния,
                ;4888       ;       допускающим запись, и только биты 25-29 могут как записываться, так и
                ;4889       ;       считываться. Таким образом, эта программа при записи CSR вынуждает
                ;4890       ;       запись в CSR1.
                ;4891       ;         Эта программа заносит в LS5 данные для обращения к CSR1, затем запи-
                ;4892       ;       сывает в CSR данные, хранящиеся в LS6. Затем она выдает CPU ATTN с целью
                ;4893       ;       информирования монитора, что она завершила свою функцию.
                ;4894       ;         ПРИМЕЧАНИЕ: биты ошибок в CSR1 очищаются при любой записи в CSR1. Эти-
                ;4895       ;       ми битами являются 31,30, и 23-14. Биты 13-0 и бит 24 не используются.
                ;4896       ;
                ;4897       ;  Процедура вызова:
                ;4898       ;       Консольный процессор загружает адрес указателя этой программы (инструк-
                ;4899       ;       ции JMP в ячейке WCS 50) в счетчик микроинструкций (UPC) и запускает
                ;4900       ;       центральный процессор.
                ;4901       ;
                ;4902       ;  Входные параметры:
                ;4903       ;       В ячейку LS6 до выполнения этой программы должны быть записаны из кон-
                ;4904       ;       соли требуемые данные, подлежащие занесению в CSR1.
                ;4905       ;
                ;4906       ;  Неявные входные данные:
                ;4907       ;       Отсутствуют.
                ;4908       ;
                ;4909       ;  Параметры выхода:
                ;4910       ;       В биты CSR 29-25 и 6-0 записаны данные из LS6.
                ;4911       ;
                ;4912       ;  Неявные выходы:
                ;4913       ;       отсутствуют.
                ;4914       ;
                ;4915       ;  Побочный эффект:
                ;4916       ;       Этой программой модифицируется WR0.
                ;4917       ;       Этой программой модифицируется LS5.
                ;4918       ;       Биты CSR1 31,30 и 23-14 очищаются.
                ;4919       ;
                ;4920       ;---
                ;4921       DEPOSIT.CSR:
U 060B, 3644,15 ;4922               MOV LS[CSR1] TO WR[0]                   ; установка бита 2 в WR0 в качестве номера CSR,в который
                ;4923                                                       ; будет производиться запись (биты 2 и 3 являются
                ;4924                                                       ; номерами CSR)
U 0609, BE0A,15 ;4925               MOV WR[0] TO LS[T5.SUB]                  ; занесение  номера CSR в ячейку LS5 (CSR1)
U 060A, 1D0B,F5 ;4926               MEM.REQ[WRITE.CSR] ADRS[T5.SUB] DT[LONG]    ; выдача запроса памяти для доступа к CSR1
U 060B, B20C,15 ;4927               WRITE.MEM LS[T6.SUB]                    ; пересылка данных из ячейки 6 в CSR1
U 060C, 8868,74 ;4928               JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;4929       ;
                ;4930       ;         Программа индикации регистра управления и состояния MCT
                ;4931       ;***
                ;4932       ;
                ;4933       ;  Описание функционирования:
                ;4934       ;
                ;4935       ;           Программа индикации CSR читает регистр управления и состояния контрол-
                ;4936       ;       лера памяти. Существуют 3 регистра управления и состояния, идентифициру-
                ;4937       ;       мые как CSR0, CSR1 и CSR2.
```

```
                ;4938     ;         CSR0 содержит контрольные биты или биты синдрома из логических схем
                ;4939     ;       корректирующего кода (ECC). Они содержатся в CSR в битах 6-0. Другие биты
                ;4940     ;       непредсказуемые. Биты 6-0 допускают только запись специальными диагнос-
                ;4941     ;       тическими программами (они не могут записываться командой DEPOSIT CSR).
                ;4942     ;         CSR1 содержит биты ошибок и биты управления. Битами ошибок являются
                ;4943     ;       31,30 и 23-14. Битами управления являются биты 29-25. другие биты не-
                ;4944     ;       предсказуемые. Биты ошибок записываются во время выполнения функций па-
                ;4945     ;       мяти и не могут записываться посредством команды DEPOSIT CSR. Биты управ-
                ;4946     ;       ления допускают запись. Заметим, что команда DEPOSIT CSR записывает биты
                ;4947     ;       управления и очищает все биты ошибок.
                ;4948     ;         CSR2 содержит 3 бита (биты 16-14), которые могут считываться. Они уста-
                ;4949     ;       навливаются, если произошла ошибка общей шины при обращении к общей шине.
                ;4950     ;       Они не могут записываться непосредственно командой DEPOSIT CSR.
                ;4951     ;         Консольный процессор загружает LS5 номером CSR (0,1 или 2), который
                ;4952     ;       подлежит считывать. Для правильного его расположения эта программа сдви-
                ;4953     ;       гает LS5 на два места влево. Затем она выполняет чтение CSR и помещает
                ;4954     ;       данные в LS6 для считывания с консоли.
                ;4955     ;
                ;4956     ; Процедура вызова:
                ;4957     ;
                ;4958     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                ;4959     ;       указателя этой программы (инструкции JMP в ячейке WCS 51) и запускает
                ;4960     ;       центральный процессор.
                ;4961     ;
                ;4962     ; Входные параметры:
                ;4963     ;
                ;4964     ;       LS5 содержит номер CSR, подлежащего чтению.
                ;4965     ;
                ;4966     ; Неявные входные данные:
                ;4967     ;
                ;4968     ;       Отсутствуют.
                ;4969     ;
                ;4970     ; Выходные параметры:
                ;4971     ;
                ;4972     ;       Данные, считанные с выбранного CSR, в LS6.
                ;4973     ;
                ;4974     ; Неявные выходные данные:
                ;4975     ;
                ;4976     ;       Отсутствуют.
                ;4977     ;
                ;4978     ; Побочный эффект:
                ;4979     ;
                ;4980     ;        LS5 смещена на 2 места влево.
                ;4981     ;
                ;4982     ;---
                ;4983     EXAMINE.CSR:
U 060D, 360A,15 ;4984             MOV LS[T5.SUB] TO WR[0]                 ; выборка из LS5 номера CSR
U 060E, A2D0,15 ;4985             SHL2 WR[0]                              ; сдвиг числа на 2 места влево для получения номера CSR
                ;4986                                                     ; в битах 2 и 3
U 060F, BE0A,15 ;4987             MOV WR[0] TO LS[T5.SUB]                 ; занесение сдвинутого числа в LS5
U 0610, 9D0B,75 ;4988             MEM.REQ[READ.CSR] ADRS[T5.SUB] DT[LONG]  ; выдача запроса памяти для доступа к заданному CSR
U 0611, 3022,15 ;4989             MOV MEM.DATA TO WR[0]                   ; чтение выбранного CSR
                ;4990     CHECK.CSR2:
U 0612, B60A,95 ;4991             MOV LS[T5.SUB] TO WR[1]                 ; выборка сдвинутого номера CSR
                ;4992               CMP LS[#8] WITH WR[1],                ; проверка, был ли выбран CSR2
```

```
U 0613,  CE46,B5 ;4993            DT(LONG)&SET.ALU.CC                    ; установка кодов условий
U 0614,  0B61,61 ;4994            JMP.IF[NEQ] TO [CHECK.CSR1]            ; переход, если выбран не CSR2
U 0615,  C530,15 ;4995            BIC LS[CSR2.MASK] TO WR[0]             ; очистка неиспользуемых (непредсказуемых) битов в CSR2
                 ;4996     CHECK.CSR1:
                 ;4997               CMP LS[#4] WITH WR[1],              ; проверка был ли выбран CSR1
U 0616,  4E44,B5 ;4998            DT(LONG)&SET.ALU.CC                    ; установка кодов условий
U 0617,  0B61,91 ;4999            JMP.IF[NEQ] TO [CHECK.CSR0]            ; переход, если выбран не CSR1
U 0618,  C52E,15 ;5000            BIC LS[CSR1.MASK] TO WR[0]             ; очистка неиспользуемых (непредсказуемых) битов в CSR1
                 ;5001     CHECK.CSR0:
                 ;5002               CMP LS[#0] WITH WR[1],              ; проверка был ли выбран CSR0
U 0619,  4E9C,B5 ;5003            DT(LONG)&SET.ALU.CC                    ; установка кодов условий
U 061A,  8B61,D1 ;5004            JMP.IF[NEQ] TO [LOAD.LS6]              ; переход, если выбран не CSR0
U 061B,  452C,15 ;5005            BIC LS[CSR0.MASK] TO WR[0]             ; очистка неиспользуемых (непредсказуемых) битов в CSR0
U 061C,  C54E,15 ;5006            BIC LS[BIT7] TO WR[0]                  ; бит 7 также не используется
                 ;5007     LOAD.LS6:
U 061D,  BE0C,15 ;5008            MOV.WR[0] TO LS[T6.SUB]                ; загрузка данных CSR в LS6 для печати
U 061E,  8868,74 ;5009            JMP [WAIT.SUB]                         ; выдача CPU ATTN и ожидание ответа
                 ;5010      ;
                 ;5011      ;        Программа записи в буфер трансляции MCT
                 ;5012      ;***
                 ;5013      ;
                 ;5014      ; Описание функционирования:
                 ;5015      ;
                 ;5016      ;          Эта программа дает возможность записывать в ту часть контроллера памя-
                 ;5017      ;        ти, которая составляет буфер трансляции. Область буфера содержит 128 дес.
                 ;5018      ;        ячеек, адресуемых от 0 до 7F (шестнадцатер.). Остаток памяти буфера транс-
                 ;5019      ;        ляции (TB) либо не используется, либо содержит данные отображения адресов
                 ;5020      ;        общей шины. Для записи в ту часть ОЗУ TB, которая содержит отображение
                 ;5021      ;        общей шины, используется команда DEPOSIT UB. Адрес, заданный командой
                 ;5022      ;        DEPOSIT, является действительным адресом обращения к ОЗУ буфера. Эта прог-
                 ;5023      ;        рамма выполняет все необходимые сдвиги заданного адреса для правильного
                 ;5024      ;        его расположения. Заданный адрес загружен в ячейку LS5 до выполнения
                 ;5025      ;        этой программы. Адрес должен быть в шестнадцатеричном формате.
                 ;5026      ;          Требуемые данные должны размещаться консольным процессором в LS6 до вы-
                 ;5027      ;        полнения этой программы. Биты 07-01 являются битами VALID (бит действи-
                 ;5028      ;        тельности), PROT (биты зашиты), MODIFY (бит модификации) и BYTE OFFSET
                 ;5029      ;        (бит смещения байта)(битов PROT имеется четыре). Биты 22-08 содержат би-
                 ;5030      ;        ты физического адреса (PA) от PA 23 до PA 09 соответственно. Биты от 31
                 ;5031      ;        до 23 и бит 0 не используются. Эта программа делает все сдвиги, необходи-
                 ;5032      ;        мые, чтобы записать биты в буфер трансляции так, как описано. Нужные
                 ;5033      ;        данные должны представляться в шестнадцатеричном формате.
                 ;5034      ;
                 ;5035      ; Процедура вызова:
                 ;5036      ;
                 ;5037      ;          Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                 ;5038      ;        указателя для данной программы (инструкции JMP в ячейке WCS 52) и запус-
                 ;5039      ;        кает центральный процессор.
                 ;5040      ;
                 ;5041      ; Входные параметры:
                 ;5042      ;
                 ;5043      ;        LS5 - адрес буфера трансляци, по которому предстоит записывать (диапазон
                 ;5044      ;              от 0 до 7F).
                 ;5045      ;        LS6 - данные, подлежащие записи (22 бита, от 1 до 22). Биты 0 и 23-31
                 ;5046      ;              игнорируются.
                 ;5047      ;
```

```
                ;5048     ; Неявные входные данные:
                ;5049     ;
                ;5050     ;       Отсутствуют.
                ;5051     ;
                ;5052     ; Выходные параметры:
                ;5053     ;
                ;5054     ;       В ТВ по заданному адресу будет записана заданная информация.
                ;5055     ;
                ;5056     ; Неявные выходные данные:
                ;5057     ;
                ;5058     ;       Отсутствуют.
                ;5059     ;
                ;5060     ; Побочный эффект:
                ;5061     ;
                ;5062     ;       WR0 - будет модифицирован
                ;5063     ;       WR1 - будет модифицирован
                ;5064     ;
                ;5065     ;---
                ;5066     DEPOSIT.TB:
U 061F, 8862,FC ;5067             JSR [SHIFT.TB.ADR]                    ; переход к подпрограмме правильного расположения адреса
                ;5068                                                   ; ТВ
U 0620, 360C,15 ;5069             MOV LS[T6.SUB] TO WR[0]               ; выборка данных из LS6
U 0621, 4526,15 ;5070             BIC LS[#FF] TO WR[0]                  ; очистка младшего байта заданной информации. Младший
                ;5071                                                   ; байт будет позднее размещен в битах 24-31
U 0622, 8862,AC ;5072             JSR [SHIFT.LOOP]                      ; переход к подпрограмме для сдвига WR0 на 8 позиций
                ;5073                                                   ; вправо
U 0623, E40C,15 ;5074             SWAP LS[T6.SUB] WITH WR[0]            ; замена исходных данных модифицированными
                ;5075                                                   ; данными. Теперь LS6 содержит биты 8-22 в битах 0-14,
                ;5076                                                   ; как и требовалось. WR0 содержит заданные исходные
                ;5077                                                   ; данные
U 0624, 452C,15 ;5078             BIC LS[#FFFFFF00] TO WR[0]            ; очистка всех заданных исходных данных, кроме младшего
                ;5079                                                   ; байта
U 0625, 8862,AC ;5080             JSR [SHIFT.LOOP]                      ; переход к подпрограмме сдвига WR0 на 8 позиций
                ;5081                                                   ; вправо. После возврата WR0 будет содержать биты 0-7 на
                ;5082                                                   ; месте битов 24-31. Другие биты очищены
U 0626, ED0C,15 ;5083             BIS WR[0] TO LS[T6.SUB]               ; теперь LS6 содержит данные, подлежащие записи, в
                ;5084                                                   ; правильном формате, т.е., биты PA в битах 0-14, BYTE
                ;5085                                                   ; OFFSET, MODIFY, биты PROT от A до D и VALID в битах
                ;5086                                                   ; 25-31
U 0627, 980A,F5 ;5087             MEM.REQ[WRITE.TB] ADRS[T5.SUB] DT[LONG]   ; выдача запроса памяти для записи в ТВ
U 0628, B20C,15 ;5088             WRITE.MEM LS[T6.SUB]                  ; запись данных из LS6 в ТВ
U 0629, 8868,74 ;5089             JMP [WAIT.SUB]                        ; переход к выдаче CPU ATTN и ожиданию
                ;5090     ;
                ;5091     ;       Подпрограмма сдвига ТВ на 8 мест вправо
                ;5092     ;
                ;5093     SHIFT.LOOP:
U 062A, 3646,95 ;5094             MOV LS[#8] TO WR[1]                   ; в WR1 находится счетчик сдвига на 8 позиций
                ;5095     SHIFT.LOOP.1:
U 062B, 2340,15 ;5096             ROR WR[0]                             ; сдвиг WR0 на одну позицию вправо
                ;5097               DEC WR[1],                          ; уменьшение счетчика и установка
U 062C, A102,B5 ;5098             DT(LONG)&SET.ALU.CC                   ; кодов условий
U 062D, 8862,B1 ;5099             JMP.IF[NEQ] TO [SHIFT.LOOP.1]         ; повторение до тех пор, пока будет выполнено 8
                ;5100                                                   ; сдвигов, затем
U 062E, 5B00,14 ;5101             RETURN                                ; возврат
                ;5102     ;
```

```
                    ;5103           Подпрограмма размещения адреса буфера TB в битах 9-14 и бите 31
                    ;5104           (бит 0 адреса TB должен находиться на месте бита 31)
                    ;5105
                    ;5106   SHIFT_TB_ADR:
U 062F, 360A,15     ;5107           MOV LS[T5.SUB] TO WR[0]                 ; выборка адреса TB из LS5
U 0630, 452C,15     ;5108           BIC LS[#FFFFFF00] TO WR[0]              ; очистка всех битов, кроме младшего байта (8-битовый
                    ;5109                                                   ; адрес)
U 0631, A2D0,15     ;5110           SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0632, A2D0,15     ;5111           SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0633, A2D0,15     ;5112           SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0634, A2D0,15     ;5113           SHL2 WR[0]                              ; сдвиг на 2 бита влево для расположения адреса
U 0635, 23D0,15     ;5114           ASHL WR[0]                                сдвиг еще на одну позицию. Теперь биты 0-6 находятся в
                    ;5115                                                     битах 9-15
                    ;5116             BIT LS[BIT15] WITH WR[0],             ; проверка, установлен или очищен бит 15 (старший бит
U 0636, 595E,35     ;5117           DT(LONG)&SET.ALU.CC                     ; для правильной адресации TB старший бит должен перейти
                    ;5118                                                   ; в 31
U 0637, 5B00,09     ;5119           SKIP.IF[EQL]                            ; пропуск следующей инструкции, если старший бит равен
                    ;5120                                                   ; нулю
U 0638, 477E,15     ;5121           BIS LS[BIT31] TO WR[0]                  ; в противном случае поместить бит 31 в старший бит.
                    ;5122             MOV WR[0] TO LS[T5.SUB],              ; теперь адрес TB в правильной форме находится в ячейке
                    ;5123                                                     LS5
U 0639, 3E0A,14     ;5124           RETURN                                    возврат к основному коду
                    ;5125
                    ;5126           Программа индикации буфера трансляции MC
                    ;5127   ;***
                    ;5128
                    ;5129     Описание функционирования.
                    ;5130
                     5131           Эта программа выполняет чтение из буфера трансляции (TB) контроллера
                    ;5132         памяти. Содержимое TB будет получено с битами BYTE OFFSET, MODIFY, PROT
                    ;5133         от A до D и VALID в битах 01-07 соответственно. Биты PA 09-23 поступают
                    ;5134         в битах 08-22 соответственно. Биты 23-31 и бит 0 не используются, поэтому
                    ;5135         будут очищены до печати результата. Результат для печати на консоли будет
                     5136         помещен в LS6
                    ;5137            Программа выдает запрос к памяти с функцией памяти READ.TB и типом дан-
                    ;5138         ных LONGWORD (длинное слово). Консольный процессор загружает LS5 заданным
                    ;5139         адресом прежде, чем выполняется эта программа. Эта программа сдвигает ад-
                    ;5140         рес так, как требуется для адресации заданной ячейки TB. Заданный адрес
                    ;5141         должен быть в 16-ричном формате.
                    ;5142            TB состоит из 128 дес.ячеек (адреса от 0 до 7F). Оставшиеся ячейки CS
                    ;5143         буфера трансляции либо не используются, либо используются для отображения
                    ;5144         адресов общей шины. Адреса из области отображения общей шины считываются
                    ;5145         посредством программы EXAMINE CS.
                    ;5146
                    ;5147     Процедура вызова
                    ;5148
                    ;5149           Консольный процессор загружает в счетчик микроинструкций (UPC) адрес
                    ;5150         указателя этой программы (инструкции JMP в ячейке WCS 53).
                    ;5151
                    ;5152     Входные параметры
                    ;5153
                    ;5154           LS5 - адрес буфера трансляции (0-7F) в шестнадцатеричном формате.
                    ;5155
                    ;5156     Неявные входные данные
                    ;5157
```

```
                ;5158       ;       Отсутствуют.
                ;5159       ;
                ;5160       ; Выходные параметры:
                ;5161       ;
                ;5162       ;       LS6 - данные из TB по заданному адресу
                ;5163       ;
                ;5164       ; Неявные выходные данные.
                ;5165       ;
                ;5166       ;       Отсутствуют.
                ;5167       ;
                ;5168       ; Побочный эффект:
                ;5169       ;
                ;5170       ;       LS5 - модифицируется
                ;5171       ;       WR0 - модифицируется
                ;5172       ;
                ;5173       ;---
                ;5174       EXAMINE.TB:
U 063A, 8862,FC ;5175       ;       JSR [SHIFT.TB.ADR]                          ; заданный в LS5 адрес сдвигается в правильное положение
                ;5176       ;                                                   ; и замещает исходный адрес в LS5
U 063B, 9C0B,F5 ;5177       ;       MEM.REQ[READ.TB] ADRS[T5.SUB] DT[LONG]      ; выдача запроса к памяти для чтения TB по адресу,
                ;5178       ;                                                   ; содержашемуся в LS5
U 063C, 3022,15 ;5179       ;       MOV MEM.DATA TO WR[0]                       ; занесение результата в WR0
U 063D, 452A,15 ;5180       ;       BIC LS[#FF000000] TO WR[0]                  ; очистка старшего байта результата (старший байт не
                ;5181       ;                                                   ; является частью TB и является непредсказуемым
U 063E, 456E,15 ;5182       ;       BIC LS[BIT23] TO WR[0]                      ; то же действие для бита 23
U 063F, 4540,15 ;5183       ;       BIC LS[BIT0] TO WR[0]                       ; то же действие для бита 0. WR0 теперь содержит
                ;5184       ;                                                   ; результат, готовый к печати
U 0640, BE0C,15 ;5185       ;       MOV WR[0] TO LS[T6.SUB]                     ; результат занесен в LS6
U 0641, 8868,74 ;5186       ;       JMP [WAIT.SUB]                              ; переход к установке ATTN и ожиданию
                ;5187       ;
                ;5188       ;       Программа записи в буфер трансляции адресов общей шины
                ;5189       ;***
                ;5190       ;
                ;5191       ; Описание функционирования:
                ;5192       ;
                ;5193       ;          Эта программа позволяет записывать в ту часть буфера трансляции (TB)
                ;5194       ;       контроллера памяти, которая содержит данные отображения адресов общей ши-
                ;5195       ;       ны. Область отображения содержит 512 дес. ячеек, адресуемых от 200 до 3FF.
                ;5196       ;       Оставшаяся часть ОЗУ TB либо не используется, либо содержит информацию бу-
                ;5197       ;       фера трансляции. Для записи в ОЗУ буфера трансляции используется команда
                ;5198       ;       DEPOSIT TB. Адрес, заданный командой DEPOSIT, является действительным ад-
                ;5199       ;       ресом обращения к ОЗУ. Эта программа выполняет все сдвиги заданного адреса,
                ;5200       ;       необходимые для правильного его расположения. Заданный адрес загружается
                ;5201       ;       консольным процессором в ячейку LS5 прежде, чем выполняется эта программа.
                ;5202       ;       Адрес должен иметь 16-ричный формат.
                ;5203       ;          Требуемые данные размещаются консольным процессором в LS6 до выполне-
                ;5204       ;       ния этой программы. Биты 07-01 являются битами VALID, PROT, MODIFY и BYTE
                ;5205       ;       OFFSET (имеются четыре бита PROT). Биты 22-08 содержат биты PA от PA23 до
                ;5206       ;       PA9 соответственно. Биты от 31 до 23 и бит 0 не используются. Эта програм-
                ;5207       ;       ма выполняет все сдвиги, необходимые для записи этих битов в TB, как опи-
                ;5208       ;       сано. Заданная информация должна быть в 16-ричном формате.
                ;5209       ;
                ;5210       ;     Процедура вызова:
                ;5211       ;
                ;5212       ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес уке-
```

```
                ;5213     ;       зателя этой программы (инструкции JMP в ячейке 58) и запускает централь-
                ;5214     ;       ный процессор.
                ;5215     ;
                ;5216     ; Входные параметры:
                ;5217     ;
                ;5218     ;       LS5 - адрес области отображения общей шины в ТВ (должен находиться в диа-
                ;5219     ;             пазоне 200-3FF шестнадцатер.).
                ;5220     ;       LS6 - в битах 1-22 содержит данные, подлежащие записи в область отобра-
                ;5221     ;             жения общей шины.
                ;5222     ;
                ;5223     ; Неявные входные данные:
                ;5224     ;
                ;5225     ;       Отсутствуют
                ;5226     ;
                ;5227     ; Выходные параметры:
                ;5228     ;
                ;5229     ;       Часть ОЗУ ТВ, составляющая область отображения общей шины, записывается
                ;5230     ;       данными из LS6 по адресу, заданному в LS5.
                ;5231     ;
                ;5232     ; Неявные выходные данные:
                ;5233     ;
                ;5234     ;       Отсутствуют.
                ;5235     ;
                ;5236     ; Побочный эффект:
                ;5237     ;
                ;5238     ;       Будет модифицирован WR0.
                ;5239     ;       Будет модифицирован WR1.
                ;5240     ;
                ;5241     ;---
                ;5242     DEPOSIT.UBS:
U 0642, 0864,DC ;5243             JSR [SHIFT.UB.ADR]                      ; переход к подпрограмме для получения правильного
                ;5244                                                     ; адреса области отображения общей шины
U 0643, 360C,15 ;5245             MOV LS[T6.SUB] TO WR[0]                 ; выборка данных из LS6
U 0644, 4526,15 ;5246             BIC LS[#FF] TO WR[0]                    ; очистка младшего байта заданной информации. Младший
                ;5247                                                     ; байт будет позднее размещен в битах 24-31
U 0645, 8862,AC ;5248             JSR [SHIFT.LOOP]                        ; переход к подпрограмме сдвига WR0 на 8 позиций вправо
U 0646, E40C,15 ;5249             SWAP LS[T6.SUB] WITH WR[0]              ; взаимная замена заданных исходных данных с
                ;5250                                                     ; модифицированными данными. LS6 теперь содержит биты
                ;5251                                                     ; 8-22 в битах 0-14, как и требовалось. WR0 содержит
                ;5252                                                     ; заданные исходные данные
U 0647, 452C,15 ;5253             BIC LS[#FFFFFF00] TO WR[0]               ; очистка всех битов за исключением младшего байта
                ;5254                                                     ; заданных исходных данных
U 0648, 8862,AC ;5255             JSR [SHIFT.LOOP]                        ; переход к подпрограмме сдвига WR0 на 8 позиций
                ;5256                                                     ; вправо. После возврата WR0 будет содержать биты 0-7 на
                ;5257                                                     ; месте битов 24-31. Другие биты очищены
U 0649, ED0C,15 ;5258             BIS WR[0] TO LS[T6.SUB]                  ; теперь LS6 содержит данные, подлежащие записи в ТВ, в
                ;5259                                                     ; правильном формате, т.е. биты PA в битах 0-14 и BYTE
                ;5260                                                     ; OFFSET, MODIFY, PROT от A до D и VALID в битах 25-31
U 064A, 1B0B,F5 ;5261             MEM.REQ[WRITE.UBS.MAP] ADRS[T5.SUB] DT[LONG]   ; выдача запроса к памяти для записи в область
                ;5262                                                     ; отображения общей шины
U 064B, B20C,15 ;5263             WRITE.MEM LS[T6.SUB]                    ; запись данных из LS6 в область отображения общей шины
U 064C, 8868,74 ;5264             JMP [WAIT.SUB]                          ; переход к установке ATTN и ожиданию
                ;5265     ;
                ;5266     ;       Подпрограмма правильного позиционирования заданного адреса для
                ;5267     ;       адресации области отображения адресов общей шины.
```

```
                ;5268     ;
                ;5269     SHIFT.UB.ADR:
U 064D, 360A,15 ;5270             MOV LS[T5.SUB] TO WR[0]                 ; выборка заданного адреса в WR0
U 064E, A2D0,15 ;5271             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
U 064F, A2D0,15 ;5272             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
U 0650, A2D0,15 ;5273             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
U 0651, A2D0,15 ;5274             SHL2 WR[0]                              ; сдвиг адреса на 2 позиции влево
U 0652, 23D0,15 ;5275             ASHL WR[0]                              ; сдвиг еще на 1 позицию. Теперь адрес находится в битах
                ;5276                                                     ; 9-17 WR0
                ;5277               MOV WR[0] TO LS[T5.SUB],              ; теперь LS5 содержит правильный адрес
U 0653, 3E0A,14 ;5278             RETURN                                  ; возврат к основной программе
                ;5279     ;
                ;5280     ;       Программа индикации буфера трансляции адресов общей шины
                ;5281     ;***
                ;5282     ;          Эта программа читает из той части контроллера памяти, которая составляет
                ;5283     ;       область отображения общей шины. Содержимое области отображения общей шины
                ;5284     ;       будет получено с битами BYTE OFFSET, MODIFY, PROT от A до D и VALID в битах
                ;5285     ;       01-07 соответственно. Биты PA 09-23 поступают в битах 08-22 соответственно.
                ;5286     ;       Биты 23-31 и бит 0 не используются, поэтому будут очищены до печати резуль-
                ;5287     ;       тата. результат для печати на консоли будет помещен в LS6.
                ;5288     ;          Эта программа выдает запрос к памяти с функцией памяти READ.UBS.MAP и
                ;5289     ;       типом данных=LONGWORD (длинное слово). Консольный процессор загружает LS5
                ;5290     ;       заданным адресом прежде, чем выполняется данная программа. Эта программа
                ;5291     ;       сдвигает адрес так, как требуется для адресации заданной ячейки области
                ;5292     ;       отображения. Заданный адрес должен быть в 16-ричном формате.
                ;5293     ;          Область отображения ОШ состоит из 512 дес. ячеек (адреса от 200 до 3FF).
                ;5294     ;       Оставшиеся ячейки в ОЗУ ТВ либо не используются, либо используются буфером
                ;5295     ;       трансляции. Содержимое буфера трансляции считывается посредством программы
                ;5296             EXAMINE TB.
                ;5297     ;
                ;5298     ; Процедура вызова:
                ;5299     ;
                ;5300     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;5301     ;       зателя этой программы (инструкции JUMP в ячейке WCS 59) и запускает цент-
                ;5302     ;       ральный процессор.
                ;5303     ;
                ;5304     ; Входные параметры:
                ;5305     ;
                ;5306     ;       LS5 - адрес ТВ в шестнадцатеричном формате.
                ;5307     ;
                ;5308     ; Неявные входные данные:
                ;5309     ;
                ;5310     ;       Отсутствуют.
                ;5311     ;
                ;5312     ; Выходные параметры:
                ;5313     ;
                ;5314     ;       LS6 - данные из области отображения общей шины по заданному адресу.
                ;5315     ;
                ;5316     ; Неявные выходные данные:
                ;5317     ;
                ;5318     ;       Отсутствуют.
                ;5319     ;
                ;5320     ; Побочный эффект:
                ;5321     ;
                ;5322     ;       LS5 - будет модифицироваться.
```

```
                ;5323      ;       WR0 - будет модифицироваться.
                ;5324      ;
                ;5325      ;---
                ;5326      EXAMINE.UBS:
U 0654, 0864,DC ;5327              JSR [SHIFT.UB.ADR]                          ; сдвиг адреса, заданного в LS5, в правильное положение и
                ;5328                                                          ; замена LS5
U 0655, 1C0B,75 ;5329              MEM.REQ[READ.UBS.MAP] ADRS[T5.SUB] DT[LONG]  ; выдача запроса к памяти для чтения области
                ;5330                                                          ; отображения ОШ по адресу, содержащемуся в LS5
U 0656, 3022,15 ;5331              MOV MEM.DATA TO WR[0]                       ; занесение результата в WR0
U 0657, 452A,15 ;5332              BIC LS[#FF000000] TO WR[0]                  ; очистка старшего байта результата (старший байт не
                ;5333                                                          ; является частью ТВ и является непредсказуемым)
U 0658, 456E,15 ;5334              BIC LS[BIT23] TO WR[0]                      ; то же для бита 23
U 0659, 4540,15 ;5335              BIC LS[BIT0] TO WR[0]                       ; то же для бита 0. WR0 теперь содержит результат, готовый
                ;5336                                                          ; для печати
U 065A, BE0C,15 ;5337              MOV WR[0] TO LS[T6.SUB]                     ; результат находится в LS6
U 065B, 8868,74 ;5338              JMP [WAIT.SUB]                              ; переход к установке ATTN и ожиданию
                ;5339      ;
                ;5340      ;       Программа записи в основную память
                ;5341      ;***
                ;5342      ;
                ;5343      ; Описание функционирования:
                ;5344      ;
                ;5345      ;          Эта программа записывает в основную память по заданному адресу длинное
                ;5346      ;       слово данных. Адрес не обязательно должен быть на границе длинного слова.
                ;5347      ;       Консольный процессор использует эту программу для пересылки данных в ос-
                ;5348      ;       новную память, а также и для команды DEPOSIT.MM. Консольный процессор дол-
                ;5349      ;       жен загружать адрес в LS5 и данные в LS6 до выполнения этой программы.
                ;5350      ;          Эта программа выдает запрос для памяти с функцией памяти WRITE.P.и ти-
                ;5351      ;       пом данных=длинное слово. Затем она выдает инструкцию WRITE.MEM LS[6] для
                ;5352      ;       записи данных в память. Для того, чтобы убедиться, что запись произошла
                ;5353      ;       без ошибок, контролируется ERR.SUM. Если установлен сигнал ERR.SUM, то
                ;5354      ;       центральный процессор, прежде, чем выставить CPU ATTN, устанавливает CPU
                ;5355      ;       ACK для информирования консольного процессора, что произошла ошибка.
                ;5356      ;
                ;5357      ; Процедура вызова:
                ;5358      ;
                ;5359      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;5360      ;       зателя этой программы (инструкции JMP в ячейке WCS 54) и запускает цент-
                ;5361      ;       ральный процессор.
                ;5362      ;
                ;5363      ; Входные параметры:
                ;5364      ;
                ;5365      ;       LS5 - физический адрес основной памяти
                ;5366      ;       LS6 - длинное слово данных для основной памяти
                ;5367      ;
                ;5368      ; Неявные входные данные:
                ;5369      ;
                ;5370      ;       Отсутствуют
                ;5371      ;
                ;5372      ; Выходные параметры:
                ;5373      ;
                ;5374      ;       В основной памяти по заданному адресу записаны данные из LS6
                ;5375      ;
                ;5376      ; Неявные выходные данные:
                ;5377      ;
```

```
                ;5378      ;       Ошибка памяти вызывает установку CPU ACK для информирования консольного
                ;5379      ;       процессора об ошибке записи
                ;5380      ;
                ;5381      ; Побочный эффект:
                ;5382      ;
                ;5383      ;       Отсутствует
                ;5384      ;
                ;5385      ;---
                ;5386      DEPOSIT.MM:
U 065C, 990A,75 ;5387              MEM.REQ[WRITE.P] ADRS[T5.SUB] DT[LONG]   ; инициирование записи в основную память с
                ;5388                                                       ; использованием физического адреса, хранящегося в LS5
                ;5389                WRITE.MEM LS[T6.SUB],                  ; запись данных из LS6 в основную память и
U 065D, 320C,1D ;5390              SKIP.IF[MEM.REF.OK]                      ; пропуск следующей инструкции, если ошибки памяти
                ;5391                                                       ; отсутствуют (нет ERR SUM)
U 065E, 10D0,15 ;5392              MISC [SET.CP.ACK]                        ; установка CPU ACK, если выставлен сигнал ERR.SUM
U 065F, 3644,15 ;5393              MOV LS[#4] TO WR[0]                      ; занесение числа 4 в WR0
U 0660, 6C0A,15 ;5394              ADD WR[0] TO LS[T5.SUB]                  ; увеличение LS5 для записи следующего длинного слова
U 0661, 8868,74 ;5395              JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5396      ;
                ;5397      ;       Программа индикации основной памяти
                ;5398      ;-**
                ;5399      ;
                ;5400      ; Описание функционирования:
                ;5401      ;
                ;5402      ;          Эта программа используется для чтения из основной памяти длинного сло-
                ;5403      ;       ва по заданному физическому адресу. Адрес не обязательно должен быть на
                ;5404      ;       границе длинного слова. Эта программа используется консольным процессором
                ;5405      ;       при команде EXAMINE.MM. Прежде, чем выполняется эта программа, консольный
                ;5406      ;       процессор должен загрузить в LS5 заданный физический адрес.
                ;5407      ;          Эта программа производит вначале запись в CSR1 для установки бита INH
                ;5408      ;       REP CRD (запрет сообщения об исправимых ошибках чтения), чтобы препятст-
                ;5409      ;       вовать появлению ошибки системы памяти (ERR SUM) при одиночной ошибке, ко-
                ;5410      ;       торая была исправлена. Затем программа выдает запрос к памяти с функцией
                ;5411      ;       READ.P и типом данных=длинное слово. Затем она читает ячейку памяти и по-
                ;5412      ;       мещает результат в LS6. Она также проверяет ERR SUM и, если произошла
                ;5413      ;       ошибка, выдает CPU ACK (исправленные ошибки в одиночных битах не вызывают
                ;5414      ;       ERR SUM). Если произошла ошибка, то бит CPU ACK будет установлен до выдачи
                ;5415      ;       CPU ATTN для информирования консольного процессора, что имеется ошибка.
                ;5416      ;
                ;5417      ; Процедура вызова:
                ;5418      ;
                ;5419      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;5420      ;       зателя этой программы (инструкция JMP в ячейке WCS 55) и запускает цент-
                ;5421      ;       ральный процессор.
                ;5422      ;
                ;5423      ; Входные параметры:
                ;5424      ;
                ;5425      ;       LS5 - физический адрес основной памяти, по которому будет происходить
                ;5426      ;             чтение
                ;5427      ;
                ;5428      ; Неявные входные данные:
                ;5429      ;
                ;5430      ;       LS[INH.CRD] - данные для записи в CSR1 с целью запрета ERR SUM для
                ;5431      ;                     исправимых ошибок чтения (CRD)
                ;5432      ;       LS[CSR1]    - адрес для записи CSR1
```

```
                ;5433     ;
                ;5434     ; Выходные параметры:
                ;5435     ;
                ;5436     ;       LS6 - длинное слово данных, считанное по заданному физическому адресу
                ;5437     ;             памяти
                ;5438     ;
                ;5439     ; Неявные выходные данные:
                ;5440     ;
                ;5441     ;       CPU ACK, если зафиксирована ошибка
                ;5442     ;
                ;5443     ; Побочный эффект.
                ;5444     ;
                ;5445     ;       Отсутствует
                ;5446     ;
                ;5447     ;---
                ;5448     EXAMINE.MM:
U 0662,  1D45,F5 ;5449             MEM.REQ[WRITE.CSR] ADRS[CSR1] DT[LONG]   ; выдача запроса к памяти для записи CSR1
U 0663,  B278,15 ;5450             WRITE.MEM LS[INH.CRD]                    ; установка в CSR1 бита INH REP CRD и очистка запрета
                ;5451                                                      ; коррекции ошибок (ECC)
U 0664,  180B,F5 ;5452             MEM.REQ[READ.P] ADRS[T5.SUB] DT[LONG]    ; выдача запроса к памяти для чтения длинного слова
                ;5453                                                      ; данных из основной памяти. Физический адрес находится
                ;5454                                                      ; в LS5
                ;5455               MOV MEM.DATA TO LS[T6.SUB],            ; чтение данных и занесение в LS6
U 0665,  380C,1D ;5456             SKIP.IF[MEM.REF.OK]                      ; пропуск следующей инструкции, если ошибки памяти
                ;5457                                                      ; отсутствуют (нет ERR SUM)
U 0666,  10D0,15 ;5458             MISC [SET.CP.ACK]                        ; установка CPU ACK, если произошла ошибка
U 0667,  8868,74 ;5459             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5460     ;
                ;5461     ;       Программа индикации регистров IDC
                ;5462     ;***
                ;5463     ;
                ;5464     ; Описание функционирования:
                ;5465     ;
                ;5466     ;       Следующие программы используются для чтения данных из регистров необяза-
                ;5467     ;       тельного модуля IDC. Результат для печати на консоли заносится в LS6.
                ;5468     ;
                ;5469     ; Процедура вызова:
                ;5470     ;
                ;5471     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;5472     ;       зателя этой программы (инструкции JMP в ячейках WCS от 5A до 5E) и запус-
                ;5473     ;       кает центральный процессор
                ;5474     ;
                ;5475     ; Входные параметры:
                ;5476     ;
                ;5477     ;       Отсутствуют
                ;5478     ;
                ;5479     ; Неявные входные данные:
                ;5480     ;
                ;5481     ;       Отсутствуют
                ;5482     ;
                ;5483     ; Выходные параметры:
                ;5484     ;
                ;5485     ;       LS6 - данные из заданного регистра IDC
                ;5486     ;
                ;5487     ; Неявные выходные данные:
```

```
                ;5488     ;
                ;5489     ;       Отсутствуют
                ;5490     ;
                ;5491     ; Побочный эффект:
                ;5492     ;
                ;5493     ;       Отсутствует
                ;5494     ;
                ;5495     ;---
                ;5496     EXAMINE.ICSR:
U 066B, 9090,15 ;5497             MISC [SET.PORT.I.O]                   ; выборка порта на шину
U 0669, 9780,15 ;5498             PORT.READ PORT.ADRS[CSR]              ; пересылка команды чтения
U 066A, 0867,64 ;5499             JMP [READ.IDC]                        ; переход к чтению данных
                ;5500     EXAMINE.IDAR:
U 066B, 9090,15 ;5501             MISC [SET.PORT.I.O]                   ; выборка порта на шину
U 066C, 1784,15 ;5502             PORT.READ PORT.ADRS[IDAR]             ; пересылка команды чтения
U 066D, 0867,64 ;5503             JMP [READ.IDC]                        ; переход к чтению данных
                ;5504     EXAMINE.DBUF:
U 066E, 9090,15 ;5505             MISC [SET.PORT.I.O]                   ; выборка порта на шину
U 066F, 978C,15 ;5506             PORT.READ PORT.ADRS[DATA.WORD]        ; пересылка команды чтения
U 0670, 0867,64 ;5507             JMP [READ.IDC]                        ; переход к чтению данных
                ;5508     EXAMINE.PATT:
U 0671, 9090,15 ;5509             MISC [SET.PORT.I.O]                   ; выборка порта на шину
U 0672, 1790,15 ;5510             PORT.READ PORT.ADRS[PATTERN]          ; пересылка команды чтения
U 0673, 0867,64 ;5511             JMP [READ.IDC]                        ; переход к чтению данных
                ;5512     EXAMINE.POSIT:
U 0674, 9090,15 ;5513             MISC [SET.PORT.I.O]                   ; выборка порта на шину
U 0675, 9794,15 ;5514             PORT.READ PORT.ADRS[POSITION]         ; пересылка команды чтения
                ;5515     READ.IDC:
U 0676, 3A0C,15 ;5516             MOV PORT TO LS[T6.SUB]                ; чтение данных и занесение в LS6
U 0677, 1080,15 ;5517             MISC [SET.ACC.I.O]                    ; возврат к выборке ускорителя
U 0678, 8868,74 ;5518             JMP [WAIT.SUB]                        ; переход к установке ATTN и ожиданию
                ;5519     ;
                ;5520     ;       Программы записи в регистры IDC
                ;5521     ;***
                ;5522     ;
                ;5523     ; Описание функционирования:
                ;5524     ;
                ;5525     ;       Следующие программы используются для записи данных в регистры необяза-
                ;5526     ;       тельного модуля IDC. Данные берутся из LS6, которая предварительно заг-
                ;5527     ;       ружается монитором, если выполняется команда DEPOSIT.
                ;5528     ;
                ;5529     ; Процедура вызова:
                ;5530     ;
                ;5531     ;       Консольный процессор загружает в счетчик микроинструкций (UPC) указатель
                ;5532     ;       этой программы (инструкции JMP в ячейках WCS от 5F до 61) и запускает
                ;5533     ;       центральный процессор.
                ;5534     ;
                ;5535     ; Входные параметры:
                ;5536     ;
                ;5537     ;       LS6 - код данных, подлежащих записи
                ;5538     ;
                ;5539     ; Неявные входные данные:
                ;5540     ;
                ;5541     ;       Отсутствуют
                ;5542     ;
```

```
                ;5543     ; Выходные параметры:
                ;5544     ;
                ;5545     ;       Отсутствуют
                ;5546     ;
                ;5547     ; Неявные выходные параметры:
                ;5548     ;
                ;5549     ;       Отсутствуют
                ;5550     ;
                ;5551     ; Побочный эффект:
                ;5552     ;
                ;5553     ;       Отсутствует
                ;5554     ;
                ;5555     ;---
                ;5556     DEPOSIT.ICSR:
U 0679, 360C,15 ;5557             MOV LS[T6.SUB] TO WR[0]                  ; выборка данных, подлежащих записи
U 067A, 17A0,15 ;5558             PORT.WRITE PORT.ADRS[CSR] WITH WR[0]     ; запись в IDC
U 067B, 8868,74 ;5559             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5560     DEPOSIT.IDAR:
U 067C, 360C,15 ;5561             MOV LS[T6.SUB] TO WR[0]                  ; выборка данных, подлежащих записи
U 067D, 97A4,15 ;5562             PORT.WRITE PORT.ADRS[DAR] WITH WR[0]     ; запись в IDC
U 067E, 8868,74 ;5563             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5564     DEPOSIT.DBUF:
U 067F, 360C,15 ;5565             MOV LS[T6.SUB] TO WR[0]                  ; выборка данных, подлежащих записи
U 0680, 17AC,15 ;5566             PORT.WRITE PORT.ADRS[DATA.WORD] WITH WR[0]  ; запись в IDC
U 0681, 8868,74 ;5567             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5568     CLEAR.FIFO.ADDR:
U 0682, 17C0,15 ;5569             PORT.CONTROL [CLEAR.FIFO.CNTR]           ; очистка адреса FIFO A или FIFO B
U 0683, 8868,74 ;5570             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5571     SELECT.FIFO.A:
U 0684, 17D8,15 ;5572             PORT.CONTROL [SEL.FIFO.A]                ; выборка FIFO A
U 0685, 8868,74 ;5573             JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5574     SELECT.FIFO.B:
U 0686, 97DC,15 ;5575             PORT.CONTROL [SEL.FIFO.B]                ; выборка FIFO B
                ;5576     WAIT.SUB:
U 0687, 10E0,15 ;5577             MISC [SET.CP.ATTN]                       ; выдача консольному процессору сигнала CPU ATTN
                ;5578     WAIT.DE.EX:
U 0688, 8868,84 ;5579             JMP [WAIT.DE.EX]                         ; зацикливание на себя до тех пор, пока консольный
                ;5580                                                      ; процессор возьмет управление
                ;5581     ;
                ;5582     ;       Программа запоминания рабочих регистров
                ;5583     ;***
                ;5584     ;
                ;5585     ; Описание функционирования:
                ;5586     ;
                ;5587     ;          Эта программа запоминает рабочие регистры 0-3 в ячейках местной памяти
                ;5588     ;       LS 1-4. Если консольный процессор забирает управление от центрального про-
                ;5589     ;       цессора, он вызывает эту программу для того, чтобы рабочие регистры могли
                ;5590     ;       использоваться консольным процессором. Рабочие регистры восстанавливаются
                ;5591     ;       другой программой, вызываемой консольным процессором прежде, чем централь-
                ;5592     ;       ный процессор возобновляет выполнение (например, при команде CONTINUE).
                ;5593     ;          Эта программа просто пересылает данные из WR 0-3 в ячейки LS 1-4 и за-
                ;5594     ;       тем выдает для консольного процессора CPU ATTN. Затем она зацикливается
                ;5595     ;       инструкцией JMP на себя до тех пор, пока консольный процессор возьмет
                ;5596     ;       управление.
                ;5597     ;
```

```
                ;5598      ; Процедура вызова:
                ;5599      ;
                ;5600      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) указатель
                ;5601      ;       этой программы (инструкцию JMP в ячейке WCS 46) и запускает центральный
                ;5602      ;       процессор.
                ;5603      ;
                ;5604      ; Входные параметры:
                ;5605      ;
                ;5606      ;       Отсутствуют
                ;5607      ;
                ;5608      ; Неявные входные данные:
                ;5609      ;
                ;5610      ;       Отсутствуют
                ;5611      ;
                ;5612      ; Выходные параметры:
                ;5613      ;
                ;5614      ;       LS1 - содержимое WR0
                ;5615      ;       LS2 - содержимое WR1
                ;5616      ;       LS3 - содержимое WR2
                ;5617      ;       LS4 - содержимое WR3
                ;5618      ;
                ;5619      ; Неявные выходные данные:
                ;5620      ;
                ;5621      ;       Отсутствуют
                ;5622      ;
                ;5623      ; Побочный эффект:
                ;5624      ;
                ;5625      ;       Отсутствует
                ;5626      ;
                ;5627      ;---
                ;5628      SAVE.WR:
U 0689,  3E02,15 ;5629              MOV WR[0] TO LS[SAV.WR0]             ; запоминание WR0 в ячейке LS1
U 068A,  8E04,95 ;5630              MOV WR[1] TO LS[SAV.WR1]             ; запоминание WR1 в ячейке LS2
U 068B,  3E07,15 ;5631              MOV WR[2] TO LS[SAV.WR2]             ; запоминание WR2 в ячейке LS3
U 068C,  3E09,95 ;5632              MOV WR[3] TO LS[SAV.WR3]             ; запоминание WR3 в ячейке LS4
U 068D,  8868,74 ;5633              JMP [WAIT.SUB]                       ; переход к установке ATTN и ожиданию
                ;5634      ;
                ;5635      ;       Программа восстановления рабочих регистров
                ;5636      ;***
                ;5637      ;
                ;5638      ; Описание функционирования:
                ;5639      ;
                ;5640      ;          Эта программа восстанавливает рабочие регистры К1804ВС1, хранящиеся в
                ;5641      ;       LS 1-4. Консольный процессор вызывает эту программу прежде, чем запускает-
                ;5642      ;       ся центральный процессор после останова по инициативе консольного процес-
                ;5643      ;       сора.
                ;5644      ;          Эта программа просто пересылает данные из ячеек LS 1-4 в рабочие регис-
                ;5645      ;       тры (WR) 0-3 и затем выдает для консольного процессора CPU ATTN. После
                ;5646      ;       этого выполняется инструкция JMP на себя, пока консольный процессор не
                ;5647      ;       возобновит управление.
                ;5648      ;
                ;5649      ; Процедура вызова:
                ;5650      ;
                ;5651      ;       Консольный процессор загружает в счетчик микроинструкций (UPC) адрес ука-
                ;5652      ;       зателя этой программы (инструкций JMP в ячейке WCS 47) и запускает цент-
```

```
                ;5653     ;       ральный процессор.
                ;5654     ;
                ;5655     ; Входные параметры:
                ;5656     ;
                ;5657     ;       LS1 содержит данные для восстановления WR0
                ;5658     ;       LS2 содержит данные для восстановления WR1
                ;5659     ;       LS3 содержит данные для восстановления WR2
                ;5660     ;       LS4 содержит данные для восстановления WR3
                ;5661     ;
                ;5662     ; Неявные входные данные:
                ;5663     ;
                ;5664     ;       Отсутствуют
                ;5665     ;
                ;5666     ; Выходные параметры:
                ;5667     ;
                ;5668     ;       WR0 получает данные из LS1
                ;5669     ;       WR1 получает данные из LS2
                ;5670     ;       WR2 получает данные из LS3
                ;5671     ;       WR3 получает данные из LS4
                ;5672     ;
                ;5673     ; Неявные выходные данные:
                ;5674     ;
                ;5675     ;       Отсутствуют
                ;5676     ;
                ;5677     ; Побочный эффект:
                ;5678     ;
                ;5679     ;       Отсутствует
                ;5680     ;
                ;5681     ;---
                ;5682     RESTORE.WR:
U 068E,  B602,15 ;5683             MOV LS[SAV.WR0] TO WR[0]              ; восстановление WR0
U 068F,  3604,95 ;5684             MOV LS[SAV.WR1] TO WR[1]              ; восстановление WR1
U 0690,  B607,15 ;5685             MOV LS[SAV.WR2] TO WR[2]              ; восстановление WR2
U 0691,  B609,95 ;5686             MOV LS[SAV.WR3] TO WR[3]              ; восстановление WR3
U 0692,  8868,74 ;5687             JMP [WAIT.SUB]                       ; переход к установке ATTN и ожиданию
                ;5688     ;
```

```
;5689      .PAGE "ПОДПРОГРАММЫ WCS ДЛЯ НУЖД ЦЕНТРАЛЬНОГО ПРОЦЕССОРА"
;5690      ;
;5691      ;            Программа обработки ошибок
;5692      ;
;5693      ; Описание функционирования:
;5694      ;
;5695      ;     Эта программа используется для обнаружения ошибок и выдачи диагностических
;5696      ;   сообщений об ошибках, появляющихся во время выполнения микродиагностики, ба-
;5697      ;   зируемой на WCS. Микрокод WCS устанавливает параметры, перечисленные ниже, и
;5698      ;   вызывает эту программу. Эта программа сравнивает WR1 (ожидаемые данные) и
;5699      ;   WR0 (полученные данные) в соответствии с маской ошибок. Установленные биты
;5700      ;   маски указывают те биты, которые не проверяются на наличие ошибки.
;5701      ;     Если WR0 и WR1 равны (ошибок нет), программа выполняет возврат+1 к точке
;5702      ;   вызова. Единственным исключением является условие, когда установлено зацикли-
;5703      ;   вание на ошибке. В этом случае проверяется номер ошибки, чтобы убедиться,
;5704      ;   происходила ли эта ошибка ранее. Если да, то вынуждается цикл по ошибке, пу-
;5705      ;   тем выполнения возврата. Таким образом фиксируется цикл для неустойчивых
;5706      ;   ошибок.
;5707      ;     Если WR0 и WR1 не равны (ошибка), тогда в слове управления и состояния уста-
;5708      ;   навливается бит 8, ожидаемые и полученные данные запоминаются в LS для печа-
;5709          ти и проверяются признаки в слове управления ошибками. Оно управляет запре-
;5710          том печати ошибок, разрешением зацикливания на ошибке и т.д. После печати
;5711          ошибки выполняется возврат+1 к точке вызова. Зацикливание на ошибке вместо
;5712      ;   возврата+1 вызывает возврат, чтобы вынудить зацикливание по ошибке. Также, в
;5713      ;   конце программы переключается CPU ACK для образования точки синхронизации
;5714      ;   осциллографа.
;5715      ;
;5716      ; Процедура вызова:
;5717      ;
;5718      ;   JSR [CHECK.RESULT]
;5719      ;
;5720      ; Входные параметры:
;5721      ;
;5722      ;   WR[0] = полученные данные, т.е. действительный результат теста
;5723      ;   WR[1] = ожидаемые данные, т.е. такой результат, каким он должен быть
;5724      ;
;5725      ; Неявные входные данные:
;5726      ;
;5727      ;   LS[ERROR.MASK]      = используется для маскирования битов, которые не подле-
;5728      ;                         жат проверке
;5729      ;   LS[ERROR.NUMBER]    = номер текущей ошибки
;5730      ;   LS[PREVIOUS.ERROR]  = номер ошибки предыдущих данных
;5731      ;   LS[CONTROL.STATUS]  = слово управления и состояния. Содержит информацию для
;5732      ;                         монитора о необходимых действиях, записанную централь-
;5733      ;                         ным процессором
;5734      ;   LS[ERROR.CONTROL]   = информация такого типа, как зацикливание по ошибке,
;5735      ;                         звуковой сигнал при ошибке, запрет сообщений об ошиб-
;5736      ;                         ках и останов при ошибке.
;5737      ;
;5738      ; Выходные параметры:
;5739      ;
;5740      ;   Отсутствуют
;5741      ;
;5742      ; Неявные выходные данные:
;5743      ;
```

```
                    ;5744      ;    LS[CONTROL.STATUS] = слово управления и состояния с новой информацией
                    ;5745      ;    LS[PREVIOUS.ERROR] = номер последней ошибки (модифицируется только при ошибке,
                    ;5746      ;                         если разрешено зацикливание при ошибке)
                    ;5747      ;    LS[EXPECTED.DATA]  = ожидаемый результат, если была обнаружена ошибка
                    ;5748      ;    LS[RECEIVED.DATA]  = действительный результат, если была обнаружена ошибка
                    ;5749      ;
                    ;5750      ; Побочный эффект:
                    ;5751      ;
                    ;5752      ;    WR0, WR1 и регистр Q модифицированы и не восстановлены перед возвратом.
                    ;5753      ;    Если разрешено зацикливание при ошибке и должен выполняться цикл по ошибке,
                    ;5754      ;    будет переключен CPU ACK для синхронизации осциллографа.
                    ;5755      ;
                    ;5756      CHECK.RESULT:
U 0693, 458A,15     ;5757              BIC LS[ERROR.MASK] TO WR[0]                 ; маскирование непроверяемых битов (очистка) для
                    ;5758                                                          ; полученных данных
U 0694, C58A,95     ;5759              BIC LS[ERROR.MASK] TO WR[1]                 ; то же для ожидаемых данных
                    ;5760                XOR WR[0] WITH WR[1] TO Q,                ; сравнение полученных данных в WR[0] с ожидаемыми
                    ;5761                                                          ; данными в WR[1] для обнаружения ошибки
U 0695, AB80,B5     ;5762              DT(LONG)&SET.ALU.CC                         ;
U 0696, 8869,F1     ;5763              JMP.IF[NEQ] TO [ERROR]                      ; переход, если ошибка
U 0697, 3690,15     ;5764              MOV LS[ERROR.CONTROL] TO WR[0]              ; выборка из LS слова управления ошибками
                    ;5765                BIT WR[0] WITH LS[LOE],                   ; проверка, разрешено ли зацикливание по ошибке и
U 0698, 5940,35     ;5766              DT(LONG)&SET.ALU.CC                         ; установка кодов условий
U 0699, DB00,01     ;5767              SKIP.IF[BIT.SET]                            ; пропуск следующей инструкции, если да
U 069A, DB00,16     ;5768              RETURN+1                                    ; иначе возврат+1 (отсутствует ошибка и не разрешено
                    ;5769                                                          ; зацикливание)
U 069B, 36A0,15     ;5770              MOV LS[PREVIOUS.ERROR] TO WR[0]             ; если отсутствует ошибка, но зацикливание по ошибке
                    ;5771                                                          ; разрешено, проверка, была ли предыдущая ошибка
                    ;5772                XOR WR[0] WITH LS[ERROR.NUMBER] TO Q,     ;
U 069C, CDB2,35     ;5773              DT(LONG)&SET.ALU.CC                         ; является ли это место тем же, в котором раньше была
                    ;5774                                                          ; ошибка (неустойчивая ошибка)?
U 069D, 0868,91     ;5775              JMP.IF[NEQ] TO [RET+1]                      ; если нет, переход к возврату без зацикливания по
                    ;5776                                                          ; ошибке (возврат+1)
U 069E, 0868,C4     ;5777              JMP [RET]                                   ; иначе цикл при ошибке (зацикливание будет происходить
                    ;5778                                                          ; даже если ошибка изчезла)
                    ;5779      ERROR:
U 069F, 3E86,15     ;5780              MOV WR[0] TO LS[RECEIVED.DATA]               ; занесение результата теста в LS для печати
U 06A0, 3690,15     ;5781              MOV LS[ERROR.CONTROL] TO WR[0]               ; выборка битов управления ошибками для проверки
                    ;5782                                                          ; разрешения зацикливания
                    ;5783                BIT WR[0] WITH LS[LOOP.COMMAND],          ; проверка, установлен ли бит 6
U 06A1, 594C,35     ;5784              DT(LONG)&SET.ALU.CC                         ; и установка кодов условий
U 06A2, 0868,C1     ;5785              JMP.IF[BIT.SET] TO [RET]                    ; переход на зацикливание, если установлен (быстрый
                    ;5786                                                          ; цикл)
U 06A3, 3E84,95     ;5787              MOV WR[1] TO LS[EXPECTED.DATA]               ; занесение в LS ожидаемых данных для печати
U 06A4, 3680,95     ;5788              MOV LS[CONTROL.STATUS] TO WR[1]              ; выборка из LS слова управления и состояния
U 06A5, C532,95     ;5789              BIC LS[#FE7FFFFF] TO WR[1]                   ; очистка всех битов слова управления и состояния, кроме
                    ;5790                                                          ; 23 и 24
U 06A6, C750,95     ;5791              BIS LS[ERROR] TO WR[1]                       ; установка в слове управления и состояния бита 8
                    ;5792                                                          ; (указывает, что обнаружена ошибка)
U 06A7, BE80,95     ;5793              MOV WR[1] TO LS[CONTROL.STATUS]              ; запись откорректированного слова управления и
                    ;5794                                                          ; состояния обратно в LS 40
                    ;5795                BIT WR[0] WITH LS[BELL],                  ; проверка, установлен ли бит звукового сигнала при
                    ;5796                                                          ; ошибке (WR0 все еще содержит слово управления
                    ;5797                                                          ; ошибками)
U 06A8, D944,35     ;5798              DT(LONG)&SET.ALU.CC                         ; установка кодов условий
```

```
U 06A9, 886A,D9 ;5799            JMP.IF[BITS.CLR] TO [CHECK.NER]          ; если бит "звуковой сигнал при ошибке" не установлен,
                ;5800                                                     ; переход для проверки, запрещаются или нет сообщения об
                ;5801                                                     ; ошибках
U 06AA, 10E0,15 ;5802            MISC [SET.CP.ATTN]                       ; иначе установка CPU ATTN (для звукового сигнала)
                ;5803    WAIT.1:
U 06AB, 086A,B4 ;5804            JMP [WAIT.1]                             ; ожидание ответа консольного процессора
U 06AC, 086B,64 ;5805            JMP [LOOP]                               ; переход к проверке цикла при ошибке после того, как
                ;5806                                                     ; консольный процессор закончил
                ;5807    CHECK.NER:
                ;5808            BIT WR[0] WITH LS[NER],                  ; если не звуковой сигнал, проверка, запрещены ли
                ;5809                                                     ; сообщения об ошибках
U 06AD, D942,35 ;5810            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06AE, 886B,21 ;5811            JMP.IF[BIT.SET] TO [CHECK.HALT]          ; если сообщения об ошибках запрещены, переход к
                ;5812                                                     ; проверке останова по ошибке
U 06AF, 10E0,15 ;5813            MISC [SET.CP.ATTN]                       ; установка CPU ATTN (сообщение об ошибке)
                ;5814    WAIT.2:
U 06B0, 086B,04 ;5815            JMP [WAIT.2]                             ; ожидание ответа консольного процессора
U 06B1, 086B,64 ;5816            JMP [LOOP]                               ; переход к проверке зацикливания по ошибке после того,
                ;5817                                                     ; как консольный процессор закончил
                ;5818    CHECK.HALT:
                ;5819                BIT WR[0] WITH LS[HOE],              ; проверка, разрешен ли останов при ошибке и
U 06B2, 5946,35 ;5820            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06B3, 886B,69 ;5821            JMP.IF[BITS.CLR] TO [LOOP]               ; если нет, переход к проверке зацикливания при ошибке
U 06B4, 10E0,15 ;5822            MISC [SET.CP.ATTN]                       ; иначе установка CPU ATTN (останов при ошибке)
                ;5823    WAIT.3:
U 06B5, 086B,54 ;5824            JMP [WAIT.3]                             ; ожидание ответа консольного процессора
                ;5825    LOOP:
U 06B6, 3690,15 ;5826            MOV LS[ERROR.CONTROL] TO WR[0]           ; выборка битов управления ошибками (NER, HOE и т.д.)
                ;5827              BIT WR[0] WITH LS[LOE],                ; проверка, разрешено ли зацикливание при ошибке и
U 06B7, 5940,35 ;5828            DT(LONG)&SET.ALU.CC                      ; установка кодов условий
U 06B8, DB00,01 ;5829            SKIP.IF[BIT.SET]                         ; пропуск инструкции, если разрешено зацикливание при
                ;5830                                                     ; ошибке
                ;5831    RET+1:
U 06B9, DB00,16 ;5832            RETURN+1                                 ; иначе возврат+1 в цикл по отсутствию ошибок
U 06BA, 3682,15 ;5833            MOV LS[ERROR.NUMBER] TO WR[0]            ; если зацикливание, выборка номера ошибки
U 06BB, BEA0,15 ;5834            MOV WR[0] TO LS[PREVIOUS.ERROR]          ; запоминание его по месту хранения предыдущей ошибки
                ;5835    RET:
U 06BC, 10D0,15 ;5836            MISC [SET.CP.ACK]                        ; установка CPU ACK для синхронизации осциллографа
                ;5837              MISC [CLR.CP.ATTN.AND.ACK],            ; и его очистка
U 06BD, 10C0,14 ;5838            RETURN                                   ; возврат к тесту и цикл при ошибке
                ;5839     ;
                ;5840     ;
                ;5841    ERR.NO.TEST:
U 06BE, DB00,15 ;5842            NOP                                      ; этот NOP вызывает ошибку в следовании тестов
U 06BF, B65E,15 ;5843            MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;5844                                                     ; состояния
U 06C0, 3E80,15 ;5845            MOV WR[0] TO LS[CONTROL.STATUS]          ; установка бита 15 в слове управления и состояния. Бит
                ;5846                                                     ; 15 указывает конс.процессору начало теста
U 06C1, 8B68,74 ;5847            JMP [WAIT.SUB]                           ; переход к установке ATTN и ожиданию
                ;5848     ;
                ;5849     ;
                ;5850     ;                    Программа инициации переноса данных
                ;5851     ;
                ;5852     ;    Эта программа загружает LS данными, необходимыми для выполнения тестов. За-
                ;5853     ;    тем она выдает CPU ATTN при очищенном слове управления и состояния, что ука-
```

```
                ;5854      ;    зывает, что все пересылки завершены.
                ;5855      ;
                ;5856     TRANSFER.POINT:
U 06C2, 2F80,15 ;5857             CLR WR[0]                          ; начало. Очистка WR0
U 06C3, BFFE,15 ;5858             MOV WR[0] TO LS[PSL.HW]             ; обеспечение условия, что бит режим совместимости
                ;5859                                                ; очищен
U 06C4, 2F80,15 ;5860             CLR WR[0]                          ; WR0 теперь содержит 0
U 06C5, 2040,15 ;5861             INC WR[0]                          ; WR0 теперь содержит 1
U 06C6, 23D0,15 ;5862             ASHL WR[0]                         ; 0 в бите 0, 1 в бите 1
U 06C7, 2040,15 ;5863             INC WR[0]                          ; 11(B)
U 06C8, 23D0,15 ;5864             ASHL WR[0]                         , 110(B)
U 06C9, 2040,15 ;5865             INC WR[0]                          ; 111(B)
U 06CA, 23D0,15 ;5866             ASHL WR[0]                         ; 1110(B)
U 06CB, 23D0,15 ;5867             ASHL WR[0]                         ; 1 1100(B)
U 06CC, 23D0,15 ;5868             ASHL WR[0]                         ; 11 1000(B)
U 06CD, 23D0,15 ;5869             ASHL WR[0]                         ; 111 0000(B). WR0 теперь содержит 70(H). Это правильный
                ;5870                                                ; начальный адрес секции данных (включая счетчик
                ;5871                                                ; длинных слов, приемник и адрес приемника)
U 06CE, 3E0E,15 ;5872             MOV WR[0] TO LS[TRANSFER.POINTER]   ; загрузка LS 7 указателем для пересылки секции данных
U 06CF, 2F80,15 ;5873             CLR WR[0]                          ; 0 в WR0
U 06D0, 2040,15 ;5874             INC WR[0]                          ; 1 в WR0
U 06D1, 23D0,15 ;5875             ASHL WR[0]                         ; 10(B)
U 06D2, 23D0,15 ;5876             ASHL WR[0]                         ; 100(B)
U 06D3, 23D0,15 ;5877             ASHL WR[0]                         ; 1000(B)
U 06D4, 23D0,15 ;5878             ASHL WR[0]                         ; 1 0000(B)
U 06D5, 23D0,15 ;5879             ASHL WR[0]                         ; 10 0000(B)
U 06D6, 23D0,15 ;5880             ASHL WR[0]                         ; 100 0000(B)
U 06D7, 23D0,15 ;5881             ASHL WR[0]                         ; 1000 0000(B)
U 06D8, 23D0,15 ;5882             ASHL WR[0]                         ; 1 0000 0000(B)
U 06D9, 23D0,15 ;5883             ASHL WR[0]                         ; 10 0000 0000(B)
U 06DA, 23D0,15 ;5884             ASHL WR[0]                         ; 100 0000 0000(B)
U 06DB, 23D0,15 ;5885             ASHL WR[0]                         ; 1000 0000 0000(B)
U 06DC, 23D0,15 ;5886             ASHL WR[0]                         ; 1 0000 0000 0000(B)
U 06DD, 23D0,15 ;5887             ASHL WR[0]                         ; 10 0000 0000 0000(B)
U 06DE, 3EB0,15 ;5888             MOV WR[0] TO LS[CONTROL.STATUS]     ; установка бита 13 в слове управления и состояния в LS
                ;5889                                                ; (бит 13 указывает, что консольный процессор должен
                ;5890                                                ; выполнять перенос данных)
U 06DF, 8881,EC ;5891             JSR [SET.ATTN]                     ; выполнение вначале пересылок в ячейки от 0 до 77(H)
U 06E0, 36CA,15 ;5892             MOV LS[#162(H)] TO WR[0]           ; выборка начального адреса второй половины данных в LS
U 06E1, 3E0E,15 ;5893             MOV WR[0] TO LS[TRANSFER.POINTER]   ; загрузка в LS начального адреса для консольного
                ;5894                                                ; процессора
U 06E2, 365A,15 ;5895             MOV LS[XFER.DATA] TO WR[0]          ; установка в WR бита 13
U 06E3, 3EB0,15 ;5896             MOV WR[0] TO LS[CONTROL.STATUS]     ; установка бита для выполнения переноса данных
U 06E4, 8881,EC ;5897             JSR [SET.ATTN]                     ; выполнение второго переноса данных в ячейки местной
                ;5898                                                ; памяти от 80 до F7(H)
U 06E5, 3022,15 ;5899             MOV MEM.DATA TO WR[0]              ; эта инструция используется для освобождения
                ;5900                                                ; памяти, если произошло "ЗАВИСАНИЕ" при передаче
                ;5901                                                ; восьмикратных слов
U 06E6, 3022,15 ;5902             MOV MEM.DATA TO WR[0]              ; таких инструкций требуются 4
U 06E7, 3022,15 ;5903             MOV MEM.DATA TO WR[0]              ;
U 06E8, 3022,15 ;5904             MOV MEM.DATA TO WR[0]              ; инициализация памяти завершена
U 06E9, 3740,15 ;5905             MOV LS[#254] TO WR[0]              ; выборка начального адреса данных, пересылаемых в
                ;5906                                                ; основную память
U 06EA, 3E0E,15 ;5907             MOV WR[0] TO LS[TRANSFER.POINTER]   ; загрузка начального адреса в LS для консоли
U 06EB, 8881,EC ;5908             JSR [SET.ATTN]                     ; выполнение пересылок в основную память в соответствии
```

```
                ;5909                                                   ; с ячейками, заданными в файле данных
U 06EC, 1D45,F5 ;5910           MEM.REQ[WRITE.CSR] ADRS[CSR1] DTI[LONG] ;
U 06ED, B29C,15 ;5911           WRITE.MEM LS[ZERO]                      ; запрет диспетчера памяти
U 06EE, B724,15 ;5912           MOV LS[HI.IPL] TO WR[0]                 ;
U 06EF, BFFE,15 ;5913           MOV WR[0] TO LS[PSL.HW]                 ; установка IPL=1F
U 06F0, 17CC,15 ;5914           PORT.CONTROL [CLEAR.IDC]                ; сброс IDC в холостой цикл
U 06F1, 17CC,15 ;5915           PORT.CONTROL [CLEAR.IDC]                ;
U 06F2, 17CC,15 ;5916           PORT.CONTROL [CLEAR.IDC]                ;
U 06F3, 17CC,15 ;5917           PORT.CONTROL [CLEAR.IDC]                ;
U 06F4, B64E,95 ;5918           MOV LS[#80] TO WR[1]                    ; установка в WR1 бита 7
U 06F5, 97A0,95 ;5919           PORT.WRITE PORT.ADRS[CSR] WITH WR[1]    ; очистка в IDC бита CRDY
U 06F6, 17D4,15 ;5920           PORT.CONTROL [CLEAR.AUTO]               ; очистка в IDC режима автоматической пересылки
U 06F7, 97C4,15 ;5921           PORT.CONTROL [RESET.BR]                 ; очистка в IDC BR7
U 06F8, 3660,15 ;5922           MOV LS[BIT16] TO WR[0]                  ;
U 06F9, C77A,15 ;5923           BIS LS[UBS.DCLO] TO WR[0]               ;
U 06FA, 3E80,15 ;5924           MOV WR[0] TO LS[CONTROL.STATUS]         ; очистка всех прерываний путем выдачи DC LO
U 06FB, 8881,EC ;5925           JSR [SET.ATTN]                          ; выдача CPU ATTN и ожидание повторного запуска
U 06FC, E580,15 ;5926           CLR LS[CONTROL.STATUS]                  ;
U 06FD, 0881,E4 ;5927           JMP [SET.ATTN]                          ; выдача CPU ATTN с очищенным словом управления и
                ;5928                                                   ; состояния
                ;5929    .LIST
                ;5930    ;
                ;5931    .REGION/700,3FFF
                ;5932    ;
                ;5933    700:
U 0700, 2047,95 ;5934           INC WR[3]                               ; блок из 256 приращений
U 0701, 2047,95 ;5935           INC WR[3]                               ;  !
U 0702, 2047,95 ;5936           INC WR[3]                               ;  !
U 0703, 2047,95 ;5937           INC WR[3]                               ;  !
U 0704, 2047,95 ;5938           INC WR[3]                               ;  !
U 0705, 2047,95 ;5939           INC WR[3]                               ;  !
U 0706, 2047,95 ;5940           INC WR[3]                               ;  !
U 0707, 2047,95 ;5941           INC WR[3]                               ;  !
U 0708, 2047,95 ;5942           INC WR[3]                               ;  !
U 0709, 2047,95 ;5943           INC WR[3]                               ;  !
U 070A, 2047,95 ;5944           INC WR[3]                               ;  !
U 070B, 2047,95 ;5945           INC WR[3]                               ;  !
U 070C, 2047,95 ;5946           INC WR[3]                               ;  !
U 070D, 2047,95 ;5947           INC WR[3]                               ;  !
U 070E, 2047,95 ;5948           INC WR[3]                               ;  !
U 070F, 2047,95 ;5949           INC WR[3]                               ;  !
U 0710, 2047,95 ;5950           INC WR[3]                               ;  !
U 0711, 2047,95 ;5951           INC WR[3]                               ;  !
U 0712, 2047,95 ;5952           INC WR[3]                               ;  !
U 0713, 2047,95 ;5953           INC WR[3]                               ;  !
U 0714, 2047,95 ;5954           INC WR[3]                               ;  !
U 0715, 2047,95 ;5955           INC WR[3]                               ;  !
U 0716, 2047,95 ;5956           INC WR[3]                               ;  !
U 0717, 2047,95 ;5957           INC WR[3]                               ;  !
U 0718, 2047,95 ;5958           INC WR[3]                               ;  !
U 0719, 2047,95 ;5959           INC WR[3]                               ;  !
U 071A, 2047,95 ;5960           INC WR[3]                               ;  !
U 071B, 2047,95 ;5961           INC WR[3]                               ;  !
U 071C, 2047,95 ;5962           INC WR[3]                               ;  !
U 071D, 2047,95 ;5963           INC WR[3]                               ;  !
```

```
U 071E, 2047,95 ;5964          INC WR[3]                              ;  !
U 071F, 2047,95 ;5965          INC WR[3]                              ;  !
U 0720, 2047,95 ;5966          INC WR[3]                              ;  !
U 0721, 2047,95 ;5967          INC WR[3]                              ;  !
U 0722, 2047,95 ;5968          INC WR[3]                              ;  !
U 0723, 2047,95 ;5969          INC WR[3]                              ;  !
U 0724, 2047,95 ;5970          INC WR[3]                              ;  !
U 0725, 2047,95 ;5971          INC WR[3]                              ;  !
U 0726, 2047,95 ;5972          INC WR[3]                              ;  !
U 0727, 2047,95 ;5973          INC WR[3]                              ;  !
U 0728, 2047,95 ;5974          INC WR[3]                              ;  !
U 0729, 2047,95 ;5975          INC WR[3]                              ;  !
U 072A, 2047,95 ;5976          INC WR[3]                              ;  !
U 072B, 2047,95 ;5977          INC WR[3]                              ;  !
U 072C, 2047,95 ;5978          INC WR[3]                              ;  !
U 072D, 2047,95 ;5979          INC WR[3]                              ;  !
U 072E, 2047,95 ;5980          INC WR[3]                              ;  !
U 072F, 2047,95 ;5981          INC WR[3]                              ;  !
U 0730, 2047,95 ;5982          INC WR[3]                              ;  !
U 0731, 2047,95 ;5983          INC WR[3]                              ;  !
U 0732, 2047,95 ;5984          INC WR[3]                              ;  !
U 0733, 2047,95 ;5985          INC WR[3]                              ;  !
U 0734, 2047,95 ;5986          INC WR[3]                              ;  !
U 0735, 2047,95 ;5987          INC WR[3]                              ;  !
U 0736, 2047,95 ;5988          INC WR[3]                              ;  !
U 0737, 2047,95 ;5989          INC WR[3]                              ;  !
U 0738, 2047,95 ;5990          INC WR[3]                              ;  !
U 0739, 2047,95 ;5991          INC WR[3]                              ;  !
U 073A, 2047,95 ;5992          INC WR[3]                              ;  !
U 073B, 2047,95 ;5993          INC WR[3]                              ;  !
U 073C, 2047,95 ;5994          INC WR[3]                              ;  !
U 073D, 2047,95 ;5995          INC WR[3]                              ;  !
U 073E, 2047,95 ;5996          INC WR[3]                              ;  !
U 073F, 2047,95 ;5997          INC WR[3]                              ;  !
U 0740, 2047,95 ;5998          INC WR[3]                              ;  !
U 0741, 2047,95 ;5999          INC WR[3]                              ;  !
U 0742, 2047,95 ;6000          INC WR[3]                              ;  !
U 0743, 2047,95 ;6001          INC WR[3]                              ;  !
U 0744, 2047,95 ;6002          INC WR[3]                              ;  !
U 0745, 2047,95 ;6003          INC WR[3]                              ;  !
U 0746, 2047,95 ;6004          INC WR[3]                              ;  !
U 0747, 2047,95 ;6005          INC WR[3]                              ;  !
U 0748, 2047,95 ;6006          INC WR[3]                              ;  !
U 0749, 2047,95 ;6007          INC WR[3]                              ;  !
U 074A, 2047,95 ;6008          INC WR[3]                              ;  !
U 074B, 2047,95 ;6009          INC WR[3]                              ;  !
U 074C, 2047,95 ;6010          INC WR[3]                              ;  !
U 074D, 2047,95 ;6011          INC WR[3]                              ;  !
U 074E, 2047,95 ;6012          INC WR[3]                              ;  !
U 074F, 2047,95 ;6013          INC WR[3]                              ;  !
U 0750, 2047,95 ;6014          INC WR[3]                              ;  !
U 0751, 2047,95 ;6015          INC WR[3]                              ;  !
U 0752, 2047,95 ;6016          INC WR[3]                              ;  !
U 0753, 2047,95 ;6017          INC WR[3]                              ;  !
U 0754, 2047,95 ;6018          INC WR[3]                              ;  !
```

```
U 0755, 2047,95 ;6019           INC WR[3]                       ;   !
U 0756, 2047,95 ;6020           INC WR[3]                       ;   !
U 0757, 2047,95 ;6021           INC WR[3]                       ;   !
U 0758, 2047,95 ;6022           INC WR[3]                       ;   !
U 0759, 2047,95 ;6023           INC WR[3]                       ;   !
U 075A, 2047,95 ;6024           INC WR[3]                       ;   !
U 075B, 2047,95 ;6025           INC WR[3]                       ;   !
U 075C, 2047,95 ;6026           INC WR[3]                       ;   !
U 075D, 2047,95 ;6027           INC WR[3]                       ;   !
U 075E, 2047,95 ;6028           INC WR[3]                       ;   !
U 075F, 2047,95 ;6029           INC WR[3]                       ;   !
U 0760, 2047,95 ;6030           INC WR[3]                       ;   !
U 0761, 2047,95 ;6031           INC WR[3]                       ;   !
U 0762, 2047,95 ;6032           INC WR[3]                       ;   !
U 0763, 2047,95 ;6033           INC WR[3]                       ;   !
U 0764, 2047,95 ;6034           INC WR[3]                       ;   !
U 0765, 2047,95 ;6035           INC WR[3]                       ;   !
U 0766, 2047,95 ;6036           INC WR[3]                       ;   !
U 0767, 2047,95 ;6037           INC WR[3]                       ;   !
U 0768, 2047,95 ;6038           INC WR[3]                       ;   !
U 0769, 2047,95 ;6039           INC WR[3]                       ;   !
U 076A, 2047,95 ;6040           INC WR[3]                       ;   !
U 076B, 2047,95 ;6041           INC WR[3]                       ;   !
U 076C, 2047,95 ;6042           INC WR[3]                       ;   !
U 076D, 2047,95 ;6043           INC WR[3]                       ;   !
U 076E, 2047,95 ;6044           INC WR[3]                       ;   !
U 076F, 2047,95 ;6045           INC WR[3]                       ;   !
U 0770, 2047,95 ;6046           INC WR[3]                       ;   !
U 0771, 2047,95 ;6047           INC WR[3]                       ;   !
U 0772, 2047,95 ;6048           INC WR[3]                       ;   !
U 0773, 2047,95 ;6049           INC WR[3]                       ;   !
U 0774, 2047,95 ;6050           INC WR[3]                       ;   !
U 0775, 2047,95 ;6051           INC WR[3]                       ;   !
U 0776, 2047,95 ;6052           INC WR[3]                       ;   !
U 0777, 2047,95 ;6053           INC WR[3]                       ;   !
U 0778, 2047,95 ;6054           INC WR[3]                       ;   !
U 0779, 2047,95 ;6055           INC WR[3]                       ;   !
U 077A, 2047,95 ;6056           INC WR[3]                       ;   !
U 077B, 2047,95 ;6057           INC WR[3]                       ;   !
U 077C, 2047,95 ;6058           INC WR[3]                       ;   !
U 077D, 2047,95 ;6059           INC WR[3]                       ;   !
U 077E, 2047,95 ;6060           INC WR[3]                       ;   !
U 077F, 2047,95 ;6061           INC WR[3]                       ;   !
U 0780, 2047,95 ;6062           INC WR[3]                       ;   !
U 0781, 2047,95 ;6063           INC WR[3]                       ;   !
U 0782, 2047,95 ;6064           INC WR[3]                       ;   !
U 0783, 2047,95 ;6065           INC WR[3]                       ;   !
U 0784, 2047,95 ;6066           INC WR[3]                       ;   !
U 0785, 2047,95 ;6067           INC WR[3]                       ;   !
U 0786, 2047,95 ;6068           INC WR[3]                       ;   !
U 0787, 2047,95 ;6069           INC WR[3]                       ;   !
U 0788, 2047,95 ;6070           INC WR[3]                       ;   !
U 0789, 2047,95 ;6071           INC WR[3]                       ;   !
U 078A, 2047,95 ;6072           INC WR[3]                       ;   !
U 078B, 2047,95 ;6073           INC WR[3]                       ;   !
```

```
U 078C, 2047,95 ;6074            INC WR[3]                              ;  !
U 078D, 2047,95 ;6075            INC WR[3]                              ;  !
U 078E, 2047,95 ;6076            INC WR[3]                              ;  !
U 078F, 2047,95 ;6077            INC WR[3]                              ;  !
U 0790, 2047,95 ;6078            INC WR[3]                              ;  !
U 0791, 2047,95 ;6079            INC WR[3]                              ;  !
U 0792, 2047,95 ;6080            INC WR[3]                              ;  !
U 0793, 2047,95 ;6081            INC WR[3]                              ;  !
U 0794, 2047,95 ;6082            INC WR[3]                              ;  !
U 0795, 2047,95 ;6083            INC WR[3]                              ;  !
U 0796, 2047,95 ;6084            INC WR[3]                              ;  !
U 0797, 2047,95 ;6085            INC WR[3]                              ;  !
U 0798, 2047,95 ;6086            INC WR[3]                              ;  !
U 0799, 2047,95 ;6087            INC WR[3]                              ;  !
U 079A, 2047,95 ;6088            INC WR[3]                              ;  !
U 079B, 2047,95 ;6089            INC WR[3]                              ;  !
U 079C, 2047,95 ;6090            INC WR[3]                              ;  !
U 079D, 2047,95 ;6091            INC WR[3]                              ;  !
U 079E, 2047,95 ;6092            INC WR[3]                              ;  !
U 079F, 2047,95 ;6093            INC WR[3]                              ;  !
U 07A0, 2047,95 ;6094            INC WR[3]                              ;  !
U 07A1, 2047,95 ;6095            INC WR[3]                              ;  !
U 07A2, 2047,95 ;6096            INC WR[3]                              ;  !
U 07A3, 2047,95 ;6097            INC WR[3]                              ;  !
U 07A4, 2047,95 ;6098            INC WR[3]                              ;  !
U 07A5, 2047,95 ;6099            INC WR[3]                              ;  !
U 07A6, 2047,95 ;6100            INC WR[3]                              ;  !
U 07A7, 2047,95 ;6101            INC WR[3]                              ;  !
U 07A8, 2047,95 ;6102            INC WR[3]                              ;  !
U 07A9, 2047,95 ;6103            INC WR[3]                              ;  !
U 07AA, 2047,95 ;6104            INC WR[3]                              ;  !
U 07AB, 2047,95 ;6105            INC WR[3]                              ;  !
U 07AC, 2047,95 ;6106            INC WR[3]                              ;  !
U 07AD, 2047,95 ;6107            INC WR[3]                              ;  !
U 07AE, 2047,95 ;6108            INC WR[3]                              ;  !
U 07AF, 2047,95 ;6109            INC WR[3]                              ;  !
U 07B0, 2047,95 ;6110            INC WR[3]                              ;  !
U 07B1, 2047,95 ;6111            INC WR[3]                              ;  !
U 07B2, 2047,95 ;6112            INC WR[3]                              ;  !
U 07B3, 2047,95 ;6113            INC WR[3]                              ;  !
U 07B4, 2047,95 ;6114            INC WR[3]                              ;  !
U 07B5, 2047,95 ;6115            INC WR[3]                              ;  !
U 07B6, 2047,95 ;6116            INC WR[3]                              ;  !
U 07B7, 2047,95 ;6117            INC WR[3]                              ;  !
U 07B8, 2047,95 ;6118            INC WR[3]                              ;  !
U 07B9, 2047,95 ;6119            INC WR[3]                              ;  !
U 07BA, 2047,95 ;6120            INC WR[3]                              ;  !
U 07BB, 2047,95 ;6121            INC WR[3]                              ;  !
U 07BC, 2047,95 ;6122            INC WR[3]                              ;  !
U 07BD, 2047,95 ;6123            INC WR[3]                              ;  !
U 07BE, 2047,95 ;6124            INC WR[3]                              ;  !
U 07BF, 2047,95 ;6125            INC WR[3]                              ;  !
U 07C0, 2047,95 ;6126            INC WR[3]                              ;  !
U 07C1, 2047,95 ;6127            INC WR[3]                              ;  !
U 07C2, 2047,95 ;6128            INC WR[3]                              ;  !
```

```
U 07C3, 2047,95 ;6129            INC WR[3]                              ;  !
U 07C4, 2047,95 ;6130            INC WR[3]                              ;  !
U 07C5, 2047,95 ;6131            INC WR[3]                              ;  !
U 07C6, 2047,95 ;6132            INC WR[3]                              ;  !
U 07C7, 2047,95 ;6133            INC WR[3]                              ;  !
U 07C8, 2047,95 ;6134            INC WR[3]                              ;  !
U 07C9, 2047,95 ;6135            INC WR[3]                              ;  !
U 07CA, 2047,95 ;6136            INC WR[3]                              ;  !
U 07CB, 2047,95 ;6137            INC WR[3]                              ;  !
U 07CC, 2047,95 ;6138            INC WR[3]                              ;  !
U 07CD, 2047,95 ;6139            INC WR[3]                              ;  !
U 07CE, 2047,95 ;6140            INC WR[3]                              ;  !
U 07CF, 2047,95 ;6141            INC WR[3]                              ;  !
U 07D0, 2047,95 ;6142            INC WR[3]                              ;  !
U 07D1, 2047,95 ;6143            INC WR[3]                              ;  !
U 07D2, 2047,95 ;6144            INC WR[3]                              ;  !
U 07D3, 2047,95 ;6145            INC WR[3]                              ;  !
U 07D4, 2047,95 ;6146            INC WR[3]                              ;  !
U 07D5, 2047,95 ;6147            INC WR[3]                              ;  !
U 07D6, 2047,95 ;6148            INC WR[3]                              ;  !
U 07D7, 2047,95 ;6149            INC WR[3]                              ;  !
U 07D8, 2047,95 ;6150            INC WR[3]                              ;  !
U 07D9, 2047,95 ;6151            INC WR[3]                              ;  !
U 07DA, 2047,95 ;6152            INC WR[3]                              ;  !
U 07DB, 2047,95 ;6153            INC WR[3]                              ;  !
U 07DC, 2047,95 ;6154            INC WR[3]                              ;  !
U 07DD, 2047,95 ;6155            INC WR[3]                              ;  !
U 07DE, 2047,95 ;6156            INC WR[3]                              ;  !
U 07DF, 2047,95 ;6157            INC WR[3]                              ;  !
U 07E0, 2047,95 ;6158            INC WR[3]                              ;  !
U 07E1, 2047,95 ;6159            INC WR[3]                              ;  !
U 07E2, 2047,95 ;6160            INC WR[3]                              ;  !
U 07E3, 2047,95 ;6161            INC WR[3]                              ;  !
U 07E4, 2047,95 ;6162            INC WR[3]                              ;  !
U 07E5, 2047,95 ;6163            INC WR[3]                              ;  !
U 07E6, 2047,95 ;6164            INC WR[3]                              ;  !
U 07E7, 2047,95 ;6165            INC WR[3]                              ;  !
U 07E8, 2047,95 ;6166            INC WR[3]                              ;  !
U 07E9, 2047,95 ;6167            INC WR[3]                              ;  !
U 07EA, 2047,95 ;6168            INC WR[3]                              ;  !
U 07EB, 2047,95 ;6169            INC WR[3]                              ;  !
U 07EC, 2047,95 ;6170            INC WR[3]                              ;  !
U 07ED, 2047,95 ;6171            INC WR[3]                              ;  !
U 07EE, 2047,95 ;6172            INC WR[3]                              ;  !
U 07EF, 2047,95 ;6173            INC WR[3]                              ;  !
U 07F0, 2047,95 ;6174            INC WR[3]                              ;  !
U 07F1, 2047,95 ;6175            INC WR[3]                              ;  !
U 07F2, 2047,95 ;6176            INC WR[3]                              ;  !
U 07F3, 2047,95 ;6177            INC WR[3]                              ;  !
U 07F4, 2047,95 ;6178            INC WR[3]                              ;  !
U 07F5, 2047,95 ;6179            INC WR[3]                              ;  !
U 07F6, 2047,95 ;6180            INC WR[3]                              ;  !
U 07F7, 2047,95 ;6181            INC WR[3]                              ;  !
U 07F8, 2047,95 ;6182            INC WR[3]                              ;  !
U 07F9, 2047,95 ;6183            INC WR[3]                              ;  !
```

```
U 07FA, 2047,95 ;6184            INC WR[3]                              ;  !
U 07FB, 2047,95 ;6185            INC WR[3]                              ;  !
U 07FC, 2047,95 ;6186            INC WR[3]                              ;  !
U 07FD, 2047,95 ;6187            INC WR[3]                              ;  !
U 07FE, 2047,95 ;6188            INC WR[3]                              ;  !
U 07FF, 2047,95 ;6189            INC WR[3]                              ;<------
U 0800, 5800,14 ;6190            RETURN
                ;6191
```

```
                ;6192      .PAGE "ПОДПРОГРАММЫ ОБЩЕГО ИСПОЛЬЗОВАНИЯ"
                ;6193      ;
                ;6194      ; Эта подпрограмма устанавливает признак N/A (не применяются) для полей ожида-
                ;6195      ; емых и полученных данных
                ;6196     B10:
                ;6197     ASSERT.NA:
U 0810, B66E,15 ;6198             MOV LS[BIT23] TO WR[0]                    ;
U 0811, 6DB0,15 ;6199             BIS WR[0] TO LS[CONTROL.STATUS]           ; CSR=бит 23
U 0812, 5B00,14 ;6200             RETURN                                    ;
                ;6201      ;
                ;6202      ; Эта подпрограмма снимает признак N/A (не применяются) для полей ожидаемых и по-
                ;6203      ; лученных данных
                ;6204     DEASSERT.NA:
U 0813, B6B0,15 ;6205             MOV LS[CONTROL.STATUS] TO WR[0]           ;
U 0814, 456E,15 ;6206             BIC LS[BIT23] TO WR[0]                    ;
U 0815, 3EB0,15 ;6207             MOV WR[0] TO LS[CONTROL.STATUS]           ;
U 0816, 5B00,14 ;6208             RETURN                                    ;
                ;6209      ;
                ;6210      ; Эта подпрограмма устанавливает признак для поля "OTHER" (другие данные)
                ;6211     ASSERT.OTHER:
U 0817, B670,15 ;6212             MOV LS[BIT24] TO WR[0]                    ;
U 0818, 6DB0,15 ;6213             BIS WR[0] TO LS[CONTROL.STATUS]           ;
U 0819, 5B00,14 ;6214             RETURN                                    ;
                ;6215      ;
                ;6216      ; Эта подпрограмма снимает признак для поля "OTHER" (другие данные)
                ;6217     DEASSERT.OTHER:
U 081A, B6B0,15 ;6218             MOV LS[CONTROL.STATUS] TO WR[0]           ;
U 081B, 4570,15 ;6219             BIC LS[BIT24] TO WR[0]                    ;
U 081C, 3EB0,15 ;6220             MOV WR[0] TO LS[CONTROL.STATUS]           ;
U 081D, 5B00,14 ;6221             RETURN                                    ;
                ;6222      ;
                ;6223      ; Эта подпрограмма выдает сигнал CPU ATTENTION и ожидает, пока консоль снова не
                ;6224      ; запустит центральный процессор
                ;6225     SET.ATTN:
U 081E, 10E0,15 ;6226             MISC [SET.CP.ATTN]                        ; выдача для консольного процессора CPU ATTN
                ;6227     WAIT.XFER:
U 081F, 8881,F4 ;6228             JMP [WAIT.XFER]                           ; здесь цикл ожидания реакции консоли
U 0820, 5B00,14 ;6229             RETURN                                    ; возврат к основной программе
                ;6230
```

```
;6231      .PAGE "ТЕСТ 1 – тест шины IB и буферов (модуль DAP)"
;6232      ;
;6233      ; ОПИСАНИЕ ТЕСТА:
;6234      ;
;6235      ;    Этот тест проверяет IB BUS и регистры модуля DAP. В начале выдается за-
;6236      ;    прос памяти с заполнением регистра предвыборки инструкций (IB.FILL),
;6237      ;    а после этого проверяется каждый из четырех байтов, чтобы убедиться, что
;6238      ;    в четыре буфера IB записан новый 32-битовый набор данных, для случая, если при
;6239      ;    выдаче запроса памяти не выдан IB.FILL (заполнение регистра предвыборки инструкий).
;6240      ;    Данные каждого байта проверяются отдельно. Пути данных работают следующим обра-
;6241      ;    зом: шина от MC D00 до D31 посредством четырех буферов формирует BUS IB D00 до
;6242      ;    D07. В определенный момент времени выход только одного из этих буферов пос-
;6243      ;    тупит на BUS IB D00-D07. Буфер определяется двумя младшими битами шины Y в
;6244      ;    начале инструкции DECODE. Инструкция DECODE загружает биты OS 0 H по OS 7 H из
;6245      ;    BUS IB 0 H по BUS IB 7 H. Биты OS через три мультиплексора поступают на шину
;6246      ;    D, где проверяются схемами АЛУ. Одновременно проверяются 8 битов. Другие би-
;6247      ;    ты маскируются. Ожидаемые данные создаются сдвигом и дополнением 32-битово-
;6248      ;    го набора. Для проверки каждого байта используется 8 младших битов, которые
;6249      ;    сдвигаются 8 раз для получения ожидаемых данных следующего байта. Этот тест
;6250      ;    также проверяет, что MOV.IB.DATA TO OS выполняет пропуск при IB.VALID и
;6251      ;    сообщает об ошибке, если пропуск не выполняется. LS [10] содержит адрес
;6252      ;    длинного слова памяти, который пересылается в регистр предвыборки инструк-
;6253      ;    ций. Тест также проверяет, что LS[10] (макро PC) увеличивается инструкцией
;6254      ;    DECODE. Используемыми тестовыми данными являются единицы, нули, сдвигаемые
;6255      ;    единицы и сдвигаемые нули. Во второй части этого теста проверяется инструкция
;6256      ;    DECODE, которая использует пути данных таким же образом, как и в первой части
;6257      ;    этого теста. Так как пути данных и регистр предвыборки инструкций работают, нет
;6258      ;    необходимости проверять DECODE со всеми наборами данных. В качестве тестовых
;6259      ;    данных используются все единицы и все нули. Инструкцией MEM.REQ[IB.FILL] регистр
;6260      ;    предвыборки инструкций загружается нулями, затем с использованием инструк-
;6261      ;    ции DECODE регистр OS перегружается байтами из регистра предвыборки инс-
;6262      ;    трукций, пока достигается PC=3 и выставляется запрос памяти. Для проверки
;6263      ;    инструкции DECODE сбасывается IB VALID, устанавливается признак STATE 0
;6264      ;    и выполняется пропуск по IB VALID или STATE ZERO CLR. Пропуска не должно
;6265      ;    быть. Затем сбрасывается признак STATE 0 и выполняется пропуск по IB VALID.
;6266      ;    Пропуска не должно быть.
;6267      ;
;6268      ; ПРЕДПОЛОЖЕНИЯ:
;6269      ;
;6270      ;    Предполагается, что микротесты контроллера ОЗУ и модуля ОЗУ выполнены ус-
;6271      ;    пешно и данные памяти достоверны. Однако, сигнал LOAD IB, генерируемый контро-
;6272      ;    ллером памяти по сигналу запроса памяти с IB FILL, предварительно не прове-
;6273      ;    рен и может быть причиной ошибки. Также предполагается, что входы мульти-
;6274      ;    плексоров OS, которые поступают на шину D, и сами мультиплексоры исправны.
;6275      ;
;6276      ; ШАГИ ТЕСТА:
;6277      ;
;6278      ;    1. Начальная установка регистров и ячеек местной памяти.
;6279      ;    2. Выполнение MEM.REQ[IB.FILL] для загрузки регистра предвыборки инструкций PFR.
;6280      ;    3. Загрузка в OS байта из регистра предвыборки PFR и пропуск, если IB VALID.
;6281      ;       Если признак IB VALID не появляется, тогда выдается сообщение об ошиб-
;6282      ;       ке 1.
;6283      ;    4. Загрузка ожидаемых данных из памяти в WR1.
;6284      ;    5. Проверка данных в OS. Если неправильные, выдается сообщение об ошибке 2.
;6285      ;    6. Увеличение суммарного счетчика и проверка увеличения LS[10] (макро PC).
```

```
;6286      ;       Если нет, сообщение об ошибке 3.
;6287      ;    7. Повторение шагов с 2 по 6 пока не будут проверены все наборы данных.
;6288      ;    8. Заполнение PFR нулями при помощи MEM.REQ[IB.FILL].
;6289      ;    9. Проверка четырех байтов PFR с использованием микроинструкции DECODE в инс-
;6290      ;       трукции MOV IB.DATA TO OS. При ошибке, сообщение об ошибке 4.
;6291      ;   10. Чтение нового байта данных PFR для проверки, что инструкция DECODE выстав-
;6292      ;       ляет запрос памяти, когда PC=3 на шаге 9. При ошибке, сообщение об ошиб-
;6293      ;       КЕ 5.
;6294      ;   11. Сброс сигнала IB VALID и проверка, что инструкция DECODE не вызывает
;6295      ;       пропуска. При ошибке, сообщение об ошибке 6.
;6296      ;   12. Проверка при очищенном IB VALID и функции невыполнения пропуска при
;6297      ;       IB VALID=0, что инструкция DECODE не вызывает пропуска. При ошибке,
;6298      ;       сообщение об ошибке 7
;6299      ;
;6300      ; ОШИБКИ:
;6301      ;
;6302      ;   ошибка 1 - тест не выполняет пропуска при IB VALID.
;6303      ;   ошибка 2 - тест неправильно передает данные из памяти в OS инструкцией
;6304      ;              MEM.REQ[IB.FILL].
;6305      ;   ошибка 3 - неправильно увеличивается LS[10] (макро PC) инструкцией DECODE.
;6306      ;   ошибка 4 - тест неправильно передает данные из памяти в PFR инструкцией
;6307      ;              MEM.REQ[IB.FILL].
;6308      ;   ошибка 5 - инструкция DECODE не выставляет запроса памяти, когда PC=3.
;6309      ;   ошибка 6 - тест выполняет пропуск, когда IB VALID не установлен.
;6310      ;   ошибка 7 - тест выполняет пропуск, когда IB VALID сброшен.
;6311      ;
;6312      ; НАЛАДКА:
;6313      ;
;6314      ;   ОШИБКА 1: Возможны три причины невыполнения пропуска при IB VALID. Первой
;6315      ;             причиной может быть неисправная работа схем пропуска при сигнале
;6316      ;             IB VALID. Это может случиться в ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ или на
;6317      ;             входе IB VALID этой ПМЛ. Второй причиной может быть отсутствие
;6318      ;             сигнала IB VALID. В этом случае подозревается ПМЛ УПР.IB, РЕГ.
;6319      ;             ПРЕДВЫБ. ИНСТРУКЦИЙ или ее входы. В обоих случаях данные регистра
;6320      ;             предвыборки инструкций должны быть правильные. Третьей причиной
;6321      ;             может быть то, что память не выставляет сигнала LOAD IB H после за-
;6322      ;             проса памяти с заполнением буфера. В этом случае память неправильно
;6323      ;             выполняет функцию IB FILL или сигнал LOAD IB H не возвращается на
;6324      ;             модуль DAP. Также возможно, что поле управления переходами инструк-
;6325      ;             ции DECODE неправильно интерпретируется на ПМЛ УПР.МИАСС ЕКВЕНСЕ-
;6326      ;             РОМ. В этом случае, если переход разрешен, возможен переход на не-
;6327      ;             определенное место программы.
;6328      ;   ОШИБКА 2: Эта ошибка скорее всего появляется при неисправностях путей данных.
;6329      ;             Можно просмотреть регистр предвыборки при останове по ошибке. Оши-
;6330      ;             бка здесь указывает, что сигнал LOAD PREFETCH H не стробирует бу-
;6331      ;             фера, или данные  не поступают на входы буфера, или буфер неиспра-
;6332      ;             вен. Сигнал LOAD PREFETCH H формируется от сигнала LOAD IB H, по-
;6333      ;             ступающего из контроллера памяти. Могут быть оборваны линии данных
;6334      ;             к буферу или память не выполняет функции IB.FILL. Сам буфер может
;6335      ;             быть неисправен. В этом случае, вероятно, будет неправильным один
;6336      ;             байт из четырех. Необходимо проверить LS[10]. Два младших бита оп-
;6337      ;             ределяют, который байт не работает. Если регистр предвыборки испра-
;6338      ;             вен, неисправность может быть в регистре OS или в его управлении.
;6339      ;             Имеются две микросхемы ПМЛ OS[7:1] НЕЧЕТН. и OS[6:0] ЧЕТН. Одна из
;6340      ;             них может быть неисправна. Управление осуществляет ПМЛ РЕЖ.СОВМЕС-
```

```
                ;6341     ;              ТИМОСТИ, РЕГ. АДРЕСАЦИЯ. Оба сигнала OS CTL 0 и OS CTL 1
                ;6342     ;              должны быть низкими при возбужденном синхросигнале регистров. Дру-
                ;6343     ;              гая неисправность может быть в формировании сигналов ENABLE IB с 0
                ;6344     ;              по 3. Формированием этих сигналов управляет ПМЛ УПР.IB, РЕГ.ПРЕД-.
                ;6345     ;              ВЫБ.ИНСТРУКЦИЙ и дешифратор. Разрешение дешифрации появляется при
                ;6346     ;              инструкции DECODE, когда установлен IB.REQ. Пути данных после ПМЛ
                ;6347     ;              OS были проверены раньше.
                ;6348     ;   ОШИБКА 3: Возможны два источника ошибки, если LS[10] не увеличивается. Одним
                ;6349     ;              источником ошибки может быть выборка ячейки, другим — неправильно
                ;6350     ;              сформированные источник-приемник или функция для управления АЛУ из
                ;6351     ;              ПЗУ УПР.ФУНКЦИЕЙ АЛУ или из ПМЛ УПР.ИСТ.ПРИЕМН.АЛУ. В пошаго-
                ;6352     ;              вом режиме возможно просмотреть правильность управляющих сигналов.
                ;6353     ;              Источник должен иметь значение 7, приемник — 1, функция — 0 и бит
                ;6354     ;              переноса должен быть установлен. Если эти сигналы не соответствуют
                ;6355     ;              перечисленным, может быть неисправна ячейка ПЗУ или неисправна ПМЛ.
                ;6356     ;              Если управляющие сигналы правильные, тогда, по-видимому, неправиль-
                ;6357     ;              ный сигнал ENABLE BYTE. Сигнал ENABLE BYTE формируется ПМЛ УПР.
                ;6358     ;              МЕСТНОЙ ПАМЯТИ.
                ;6359     ;   ОШИБКА 4: Эта ошибка скорее всего указывает, что обращение к памяти выполня-
                ;6360     ;              ется до PC=3 и изменяются данные в регистре предвыборки инструкций.
                ;6361     ;              Необходимо проверить сигнал PC=3 на выходе ПМЛ УПР.IB, РЕГ.ПРЕДВЫБ.
                ;6362     ;              ИНСТРУКЦИЙ, чтобы убедиться, что он не становится высоким до выпол-
                ;6363     ;              нения последней инструкции DECODE.
                ;6364     ;   ОШИБКА 5: Если инструкция DECODE с установленным IB REQ и PC=3 не выставляет
                ;6365     ;              запроса памяти, тогда возможным источником ошибки могут быть схемы
                ;6366     ;              управления, которые первоначально загружают ПМЛ УПР.IB, РЕГ.ПРЕДВЫБ.
                ;6367     ;              ИНСТРУКЦИЙ. Необходимо проверить высокий уровень на входах BUS Y
                ;6368     ;              D00 H, BUS Y D01 H, DECODE INSTR H и CSR 08 H. Если входы правиль-
                ;6369     ;              ные, тогда неисправна ПМЛ. В этом случае выход PC EQUALS 3 H будет
                ;6370     ;              неправильным. Если этот выход правильный, тогда может быть неиспр-
                ;6371     ;              равен 8-входовый мультиплексор который выдает запрос памяти. Выход
                ;6372     ;              мультиплексора для выдачи запроса памяти должен иметь низкий уро-
                ;6373     ;              вень. схемы от этой точки уже были проверены при проверке инструкции
                ;6374     ;              MEM.REQ.
                ;6375     ;   ОШИБКА 6: Эта ошибка скорее всего указывает на неисправность ПМЛ УПР.IB,
                ;6376     ;              РЕГИСТРОМ ПРЕДВЫБ. ИНСТРУКЦИЙ. Проверьте сигнал DECODE INSTR H
                ;6377     ;              (контакт 5). Если этот сигнал правильный (должен быть 1), то
                ;6378     ;              проверьте сигнал IB VALID L (контакт 18), который должен быть 0.
                ;6379     ;              Если DECODE INSTR H=0, проверьте на ПМЛ РЕГ.STATE, ДЕКОД.MISC
                ;6380     ;              сигнал STATE 0 H (должен быть 1).
                ;6381     ;   ОШИБКА 7: Эта ошибка указывает на, то что неправильно сформирован с нал
                ;6382     ;              DECODE INSTR H (должен быть 1) на ПМЛ УПР.МЕСТНОЙ ПАМЯТИ (кон-
                ;6383     ;              такт 19). Проверьте контакты 1-4 этой ПМЛ (комбинацию 0000(Б)
                ;6384     ;              на контактах 1-4 этой ПМЛ).
                ;6385     ;
                ;6386     ;
                ;6387     T.1:
U 0821, B65E,15 ;6388               MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                ;6389                                                    ; состояния
U 0822, 3E80,15 ;6390               MOV WR[0] TO LS[CONTROL.STATUS]      ; установка бита 15 в слове управления и состояния. Бит
                ;6391                                                    ; 15 указывает начало теста для конс.процессора
U 0823, 10E0,15 ;6392               MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для конс.процессора
                ;6393     WAIT.T1.0:
U 0824, 0882,44 ;6394               JMP [WAIT.T1.0]                      ; зацикливание для ожидания ответа конс.процессора
U 0825, B724,15 ;6395               MOV LS[HI.IPL] TO WR[0]              ;
```

```
U 0826, BFFE,15 ;6396            MOV WR[0] TO LS[PSL.HW]                ; сброс сигнала COMPAT MODE и установка значения 1F в
                ;6397                                                   ; IPL
U 0827, DF26,15 ;6398            MCOM LS[#FF] TO WR[0]                  ;
U 0828, 3E8A,15 ;6399            MOV WR[0] TO LS[ERROR.MASK]            ; маска ошибки=FFFFFF00
U 0829, 65A0,15 ;6400            CLR LS[PREVIOUS.ERROR]                 ; очистка предыдущего номера ошибки
U 082A, 3642,15 ;6401            MOV LS[CPU] TO WR[0]                   ; установка 2 в WR0
U 082B, 3E8C,15 ;6402            MOV WR[0] TO LS[MODULE.NUM]            ; установка кода модуля для модуля DAP
U 082C, E520,15 ;6403            CLR LS[PC]                             ; начальная установка PC
U 082D, B646,15 ;6404            MOV LS[#B] TO WR[0]                    ;
U 082E, 474E,15 ;6405            BIS LS[#B0] TO WR[0]                   ;
U 082F, BE12,15 ;6406            MOV WR[0] TO LS[T9]                    ; T9=88 (конец цикла)
                ;6407    LOOP.T1:
U 0830, 1B21,75 ;6408            MEM.REQ[IB.FILL] ADRS[PC] DT[LONG]     ; заполнение PFR данными из памяти
U 0831, B022,95 ;6409            MOV MEM.DATA TO WR[1]                  ; должно произойти "ЗАВИСАНИЕ" здесь, если
                ;6410                                                   ; микропрограмма застряла в цикле заполнения
                ;6411                                                   ; регистра предвыборки
                ;6412                                                   ; инструкций. Следует проверить сигнал DATA RCVD H.
                ;6413    LOOP.T1.1:
U 0832, 0881,0C ;6414            JSR [ASSERT.NA]                        ; ожидаемые и полученные данные неприменимы в тесте
                ;6415                                                   ; пропуска
U 0833, B640,15 ;6416            MOV LS[#1] TO WR[0]                    ;
U 0834, BE82,15 ;6417            MOV WR[0] TO LS[ERROR.NUMBER]          ; номер ошибки=1
U 0835, 3621,15 ;6418            MOV LS[PC] TO WR[2]                    ; запоминание PC
U 0836, B401,4E ;6419            MOV IB.DATA TO OS                      ; пересылка байта в OS, увеличение PC и пропуск, если
                ;6420                                                   ; IB.VALID
U 0837, 5B00,1E ;6421            SKIP                                   ; ошибка, если не было пропуска при IB.VALID
U 0838, 8883,D4 ;6422            JMP [T1.2]                             ; пропуск был. Переход к следующей части
U 0839, 369E,95 ;6423            MOV LS[ONES] TO WR[1]                  ; установка ошибочных данных для индикации ошибки
U 083A, 2F80,15 ;6424            CLR WR[0]                              ;
U 083B, 0869,3C ;6425            JSR [CHECK.RESULT]                     ; сообщение об ошибке 1
U 083C, 0888,44 ;6426            JMP [T1.LOE]                           ;
                ;6427    T1.2:
U 083D, FF82,15 ;6428            INC LS[ERROR.NUMBER]                   ; увеличение номера ошибки до 2
U 083E, 0881,3C ;6429            JSR [DEASSERT.NA]                      ;
U 083F, B6F8,15 ;6430            MOV LS[OS] TO WR[0]                    ; выборка записанных данных
U 0840, BE11,15 ;6431            MOV WR[2] TO LS[T8]                    ; T8 <- PC-1
U 0841, B022,95 ;6432            MOV MEM.DATA TO WR[1]                  ; должно произойти "зависание" здесь, если
                ;6433                                                   ; микропрограмма застряла в потоке инструкций
                ;6434                                                   ; процессора. Цикл ожидания в результате запроса
                ;6435                                                   ; пересылки IB.DATA TO  S. Следует проверить сигналы
                ;6436                                                   ; DATA RCVD H или MEM RQ H
U 0842, 9811,95 ;6437            MEM.REQ[READ.P] ADRS[T8] DT[BYTE]      ;
U 0843, B022,95 ;6438            MOV MEM.DATA TO WR[1]                  ; выборка ожидаемых данных
U 0844, 0869,3C ;6439            JSR [CHECK.RESULT]                     ; проверка данных
U 0845, 0888,44 ;6440            JMP [T1.LOE]                           ; зацикливание при ошибке, если разрешено
                ;6441    T1.3:
U 0846, FF82,15 ;6442            INC LS[ERROR.NUMBER]                   ; увеличение номера ошибки до 3 .
U 0847, B620,15 ;6443            MOV LS[PC] TO WR[0]                    ;
U 0848, A004,95 ;6444            MOV WR[2] TO WR[1]                     ; выборка первоначального PC
U 0849, 2042,95 ;6445            INC WR[1]                              ; ожидаемые данные=PC+1 (PC должен быть увеличен)
U 084A, 0869,3C ;6446            JSR [CHECK.RESULT]                     ; проверка PC
U 084B, 0888,44 ;6447            JMP [T1.LOE]                           ; зацикливание при ошибке, если разрешено
U 084C, 2045,15 ;6448            INC WR[2]                              ;
                ;6449              XOR LS[T9] WITH WR[2] TO Q,          ; счетчик=88(H)?
U 084D, 4D13,35 ;6450            DT(LONG)&SET.ALU.CC                    ;
```

```
U 084E, 0883,01 ;6451            JMP.IF[NEQ] TO [LOOP.T1]                ; установка других тестовых данных, если нет
                ;6452    T1.4.RES:
U 084F, 3612,15 ;6453            MOV LS[T9] TO WR[0]                     ; 88(H) в WR0
U 0850, C044,15 ;6454            ADD LS[#4] TO WR[0]                     ; 8C(H) в WR0
U 0851, 3E10,15 ;6455            MOV WR[0] TO LS[T8]                     ; T8=8C(H)
                ;6456    LOOP.T1.4:
U 0852, 3644,15 ;6457            MOV LS[#4] TO WR[0]                     ;
U 0853, BE82,15 ;6458            MOV WR[0] TO LS[ERROR.NUMBER]           ; увеличение номера ошибки до 4
U 0854, 3612,15 ;6459            MOV LS[T9] TO WR[0]                     ;
U 0855, 3E20,15 ;6460            MOV WR[0] TO LS[PC]                     ; PC=88(H)(адресуется длинное слово нулей в памяти)
U 0856, B022,95 ;6461            MOV MEM.DATA TO WR[1]                   ; эта инструкция освобождает MCT в случае "зависания"
                ;6462                                                    ; при запросе дешифрации инструкции
U 0857, 1B21,75 ;6463            MEM.REQ[IB.FILL] ADRS[PC] DT[LONG]      ; заполнение PFR нулями
                ;6464    T1.4:
U 0858, 8401,4E ;6465            MOV IB.DATA TO OS                       ; пересылка байта в регистр OS, увеличение PC и пропуск,
                ;6466                                                    ; если IB.VALID
U 0859, DB00,15 ;6467            NOP                                     ;
U 085A, B6F8,15 ;6468            MOV LS[OS] TO WR[0]                     ; полученные данные=содержание регистра OS
U 085B, 2F82,95 ;6469            CLR WR[1]                               ; ожидаемые данные =0
U 085C, 0869,3C ;6470            JSR [CHECK.RESULT]                      ; проверка наличия 0 в регистре OS
U 085D, 8888,A4 ;6471            JMP [T1.LOE.4]                          ;
U 085E, 3621,15 ;6472            MOV LS[PC] TO WR[2]                     ;
                ,6473              XOR WR[2] WITH LS[T8] TO Q,           ;
U 085F, CD11,35 ;6474            DT(LONG)&SET.ALU.CC                     ; проверены четыре байта?
U 0860, 8885,81 ;6475            JMP.IF[NEQ] TO [T1.4]                   ; переход в цикл, если нет
                ;6476    T1.5:
U 0861, FF82,15 ;6477            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 5
U 0862, 8401,4E ;6478            MOV IB.DATA TO OS                       ; выборка следующего байта (байт 0 следующего длинного
                ;6479                                                    ; слова)
U 0863, DB00,15 ;6480            NOP                                     ;
U 0864, B6F8,15 ;6481            MOV LS[OS] TO WR[0]                     ;
U 0865, 3626,95 ;6482            MOV LS[#FF] TO WR[1]                    ; ожидаемые данные=FF
U 0866, 0869,3C ;6483            JSR [CHECK.RESULT]                      ; проверка наличия FF в регистре OS
U 0867, 8888,A4 ;6484            JMP [T1.LOE.4]                          ;
                ;6485    T1.6:
U 0868, FF82,15 ;6486            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 6
U 0869, 3660,15 ;6487            MOV LS[INTERUPT.EN] TO WR[0]            ; подготовка маски для запрета обращений к памяти
U 086A, 476C,15 ;6488            BIS LS[DISABLE.MEM.REF] TO WR[0]        ;
U 086B, 3E80,15 ;6489            MOV WR[0] TO LS[CONTROL.STATUS]         ; запрет обращений к памяти
U 086C, 10E0,15 ;6490            MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для конс.процессора
                ;6491    WAIT.T1.6.1:
U 086D, 8886,D4 ;6492            JMP [WAIT.T1.6.1]                       ; ожидание ответа конс.процессора
U 086E, 1B21,75 ;6493            MEM.REQ[READ.V.RCHK.IFILL] ADRS[PC] DT[LONG] ; сброс IB VALID
U 086F, 3660,15 ;6494            MOV LS[INTERUPT.EN] TO WR[0]            ; подготовка разрешения обращений к памяти
U 0870, 3E80,15 ;6495            MOV WR[0] TO LS[CONTROL.STATUS]         ; разрешение обращений к памяти
U 0871, 10E0,15 ;6496            MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для конс.процессора
                ;6497    WAIT.T1.6.2:
U 0872, 0887,24 ;6498            JMP [WAIT.T1.6.2]                       ; ожидание ответа конс.процессора
                ;6499    LOOP.T1.6:
U 0873, 9050,15 ;6500            SET.STATE.ZERO                          ; установка признака для пропуска при IB VALID
U 0874, E520,15 ;6501            CLR LS[PC]                              ; очистка PC=3 так, что IB VALID остается сброшенным
U 0875, 8401,4E ;6502            MOV IB.DATA TO OS                       ; инструкция DECODE для пропуска при IB VALID
U 0876, 0887,B4 ;6503            JMP [T1.7]                              ; правильно - пропуска не было
U 0877, 369E,95 ;6504            MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0878, 2F80,15 ;6505            CLR WR[0]
```

```
U 0879, 0869,3C ;6506            JSR [CHECK.RESULT]                    ; сообщение об ошибке 6
U 087A, 8887,34 ;6507            JMP [LOOP.T1.6]                       ; зацикливание при ошибке
                ;6508    T1.7:
U 087B, FF82,15 ;6509            INC LS[ERROR.NUMBER]                  ; увеличение номера ошибки до 7
                ;6510    LOOP.T1.7:
U 087C, 9030,15 ;6511            CLR.STATE.ZERO                        ; сброс признака пропуска при IB VALID
U 087D, E520,15 ;6512            CLR LS[PC]                            ; очистка PC=3 так, что IB VALID остается сброшенным
U 087E, 8401,4E ;6513            MOV IB.DATA TO OS                     ; инструкция DECODE для пропуска при IB VALID
U 087F, 8888,F4 ;6514            JMP [END.T1]                          ; правильно - пропуска не было
U 0880, 369E,95 ;6515            MOV LS[ONES] TO WR[1]                 ; установка ошибочных данных для индикации ошибки
U 0881, 2F80,15 ;6516            CLR WR[0]                             ;
U 0882, 0869,3C ;6517            JSR [CHECK.RESULT]                    ; сообщение об ошибке
U 0883, 8887,C4 ;6518            JMP [LOOP.T1.7]                       ; зацикливание при ошибке
                ;6519    T1.LOE:
U 0884, 2045,15 ;6520            INC WR[2]                             ; увеличение копии старого PC
                ;6521              BIT WR[2] WITH LS[#3(H)],           ;
U 0885, 59C1,35 ;6522            DT(LONG)&SET.ALU.CC                   ; два младших бита увеличенного PC сброшены?
U 0886, 2105,15 ;6523            DEC WR[2]                             ; восстановление копии старого PC
U 0887, BE21,15 ;6524            MOV WR[2] TO LS[PC]                   ; если происходит зацикливание на ошибке, PC должен быть
                ;6525                                                  ; восстановлен так, чтобы проверялся тот же самый байт
U 0888, 8883,21 ;6526            JMP.IF[NEQ] TO [LOOP.T1.1]            ; зацикливание при ошибке без перезаписи PFR
U 0889, 0883,04 ;6527            JMP [LOOP.T1]                         ; зацикливание при ошибке с перезаписью PFR
                ;6528    T1.LOE.4:
U 088A, 36C1,95 ;6529            MOV LS[#3(H)] TO WR[3]                ; установка двух младших битов
                ;6530              BIT WR[3] WITH LS[PC],              ;
U 088B, 5921,85 ;6531            DT(LONG)&SET.ALU.CC                   ; два младших бита увеличенного PC сброшены?
U 088C, 7C20,15 ;6532            DEC LS[PC]                            ; восстановление PC
U 088D, 8885,81 ;6533            JMP.IF[NEQ] TO [T1.4]                 ; зацикливание при ошибке без перезаписи PFR
U 088E, 8885,24 ;6534            JMP [LOOP.T1.4]                       ; зацикливание при ошибке с перезаписью PFR
                ;6535    END.T1:
                ;6536
```

```
;6537      .PAGE "ТЕСТ 2 - запрет IB.REQ в реж. совместимости и расширения знака (модуль DAP)"
;6538      ;
;6539      ; ОПИСАНИЕ ТЕСТА:
;6540      ;
;6541      ;       Этот тест проверяет, что установленный режим совместимости
;6542      ;       приводит к запрету IB.REQ. Устанавливается высокий уровень
;6543      ;       сигнала CSR 08 для гарантии того, что именно режим совмести-
;6544      ;       мости блокирует запрос памяти. Высокий уровень сигнала режима
;6545      ;       совместимости поступает на ПМЛ УПР.КОД. УСЛОВ. и формирует низ-
;6546      ;       кий уровень сигнала DATA REQ H, который блокирует заполнение
;6547      ;       регистра предвыборки инструкций.
;6548      ;         Этот тест также проверяет схемы расширения знака, которые не
;6549      ;       были проверены тестом расширения знака ENKCB, так как еще не была
;6550      ;       проверена память.
;6551      ;
;6552      ; ПРЕДПОЛОЖЕНИЯ:
;6553      ;
;6554      ;         Предполагается, что тест шины IB и буферов выполнен успешно и
;6555      ;       вся диагностика памяти выполнена успешно.
;6556      ;
;6557      ; ШАГИ ТЕСТА:
;6558      ;
;6559      ;       1) Выполнение запроса памяти для загрузки данными регистра
;6560      ;          предвыборки инструкций.
;6561      ;       2) Установка режима совместимости.
;6562      ;       3) Выполнение пяти инструкций DECODE (MOV IB.DATA TO OS).
;6563      ;          Пятая инструкция DECODE предназначена для проверки, что данные
;6564      ;          такие же, как и при первой инструкции DECODE, а не из следующей
;6565      ;          ячейки памяти (PFR не был перезаписан).
;6566      ;       4) Сброс режима совместимости.
;6567      ;       5) Запись и чтение байта нулей в/из памяти и проверка, что знак
;6568      ;          расширяется правильно.
;6569      ;       6) Запись и чтение байта "80(H)" в/из памяти и проверка, что
;6570      ;          расширяется единица.
;6571      ;       7) Запись и чтение слова "80(H)" в/из памяти и проверка, что
;6572      ;          расширяется нуль.
;6573      ;       8) Запись и чтение слова "8000(H)" в/из памяти и проверка, что
;6574      ;          расширяется единица.
;6575      ;
;6576      ; ОШИБКИ:
;6577      ;
;6578      ;       ошибка 1 - запрос памяти не блокируется режимом совместимости.
;6579      ;       ошибка 2 - неправильно расширяется байт, состоящий из нулей.
;6580      ;       ошибка 3 - неправильно расширяется единица из байта.
;6581      ;       ошибка 4 - неправильно расширяется нуль из слова.
;6582      ;       ошибка 5 - неправильно расширяется единица из слова.
;6583      ;
;6584      ; НАЛАДКА:
;6585      ;
;6586      ;       ОШИБКА 1: Проверьте ПМЛ УПР. КОД. УСЛОВ. На контакт 5 должен пос-
;6587      ;                 тупать сигнал COMPAT MODE H, а на контакт 12 выдается сиг-
;6588      ;                 нал DATA REQ L. Если эти сигналы правильные, проверьте схему
;6589      ;                 "НЕ-И" и схему "И-ИЛИ-НЕ", на выходе которой (контакт 8) дол-
;6590      ;                 жен быть сигнал CLOCK STALL. Возможны обрывы проводника или
;6591      ;                 неисправность микросхем.
```

```
                ;6592     ;
                ;6593     ;       ОШИБКА 2-5:Неправильно работает или ПМЛ УПР. РАСШИРЕНИЕМ ЗНАКА, или 2
                ;6594     ;                  четырехвходовых мультиплексора, или три регистра, выходами ко-
                ;6595     ;                  торых являются сигналы BUS D D31 H - BUS D D00 H. Проверьте
                ;6596     ;                  сигналы EN SXT B1 L и EN SXT HI WD L (должны быть 1). Если
                ;6597     ;                  эти сигналы правильные, подозревается неисправность мик-
                ;6598     ;                  росхем.
                ;6599     ;
                ;6600     T.2:
U 088F, B65E,15 ;6601             MOV LS[BEGIN.TEST] TO WR[0]                ; установка бита 15 в WR0 для слова управления и
                ;6602                                                        ; состояния
U 0890, 3E80,15 ;6603             MOV WR[0] TO LS[CONTROL.STATUS]             ; установка бита 15 в слове управления и состояния. Бит
                ;6604                                                        ; 15 указывает начало теста для консольного процессора
U 0891, 10E0,15 ;6605             MISC [SET.CP.ATTN]                         ; выдача сигнала CPU ATTN для консольного процессора
                ;6606     WAIT.T2.0:
U 0892, 8889,24 ;6607             JMP [WAIT.T2.0]                            ; зацикливание для ожидания ответа консольного
                ;6608                                                        ; процессора
U 0893, B724,15 ;6609             MOV LS[HI.IPL] TO WR[0]                    ;
U 0894, BFFE,15 ;6610             MOV WR[0] TO LS[PSL.HW]                    ; сброс режима совместимости
U 0895, DF26,15 ;6611             MCOM LS[#FF] TO WR[0]                      ;
U 0896, 3E8A,15 ;6612             MOV WR[0] TO LS[ERROR.MASK]                ; маска ошибки=FFFFFF00
U 0897, 65A0,15 ;6613             CLR LS[PREVIOUS.ERROR]                     ; очистка предыдущего номера ошибки
U 0898, B640,15 ;6614             MOV LS[#1] TO WR[0]                        ; установка 1 в WR0
U 0899, BE82,15 ;6615             MOV WR[0] TO LS[ERROR.NUMBER]              ; установка номера ошибки для первой ошибки
U 089A, A3C0,15 ;6616             ROL WR[0]                                  ; установка 2 в WR0
U 089B, 3E8C,15 ;6617             MOV WR[0] TO LS[MODULE.NUM]                ; установка кода модуля для модуля DAP
U 089C, 364E,15 ;6618             MOV LS[#80] TO WR[0]                       ;
U 089D, C746,15 ;6619             BIS LS[#8] TO WR[0]                        ;
U 089E, 4744,15 ;6620             BIS LS[#4] TO WR[0]                        ;
U 089F, 3E10,15 ;6621             MOV WR[0] TO LS[T8]                        ; T8=8C(H)
U 08A0, B724,15 ;6622             MOV LS[HI.IPL] TO WR[0]                    ;
U 08A1, 477E,15 ;6623             BIS LS[BIT31] TO WR[0]                     ;
U 08A2, BFFE,15 ;6624             MOV WR[0] TO LS[PSL.HW]                    ; установка режима совместимости и установка значения 1F
                ;6625                                                        ; в IPL
                ;6626     LOOP.T2.1:
U 08A3, 364E,15 ;6627             MOV LS[#80] TO WR[0]                       ;
U 08A4, C746,15 ;6628             BIS LS[#8] TO WR[0]                        ;
U 08A5, 3E20,15 ;6629             MOV WR[0] TO LS[PC]                        ; PC=88(H)
U 08A6, 1B21,75 ;6630             MEM.REQ[IB.FILL] ADRS[PC] DT[LONG]         ; заполнение PFR нулями
                ;6631     T2.1:
U 08A7, B401,4E ;6632             MOV IB.DATA TO OS                          ; пересылка байта в регистр OS, увеличение PC и пропуск
                ;6633                                                        ; при IB VALID
U 08A8, DB00,15 ;6634             NOP                                        ;
U 08A9, 3621,15 ;6635             MOV LS[PC] TO WR[2]                        ;
                ;6636               XOR LS[T8] WITH WR[2] TO Q,              ;
U 08AA, CD11,35 ;6637             DT(LONG)&SET.ALU.CC                        ; четыре байта проверены?
U 08AB, 888A,71 ;6638             JMP.IF[NEQ] TO [T2.1]                      ; возврат, если нет
U 08AC, B401,4E ;6639             MOV IB.DATA TO OS                          ; пересылка байта в регистр OS, увеличение PC и пропуск
                ;6640                                                        ; при IB VALID
U 08AD, DB00,15 ;6641             NOP                                        ;
U 08AE, 2FB2,95 ;6642             CLR WR[1]                                  ; ожидаемые данные=0
U 08AF, B6F8,15 ;6643             MOV LS[OS] TO WR[0]                        ; выборка регистра OS
U 08B0, 0869,3C ;6644             JSR [CHECK.RESULT]                         ; проверка наличия 0 в регистре OS
U 08B1, 088A,34 ;6645             JMP [LOOP.T2.1]                            ; зацикливание при ошибке, если разрешено
U 08B2, B724,15 ;6646             MOV LS[HI.IPL] TO WR[0]                    ;
```

```
U 08B3, BFFE,15 ;6647            MOV WR[0] TO LS[PSL.HW]                  ; сброс режима совместимости
U 08B4, E58A,15 ;6648            CLR LS[ERROR.MASK]                       ; очистка маски ошибки
U 08B5, FF82,15 ;6649            INC LS[ERROR.NUMBER]                     ; увеличение номера ошибки до 2
                ;6650    T2.2:
U 08B6, 6512,15 ;6651            CLR LS[T9]                               ; используется как физический адрес основной памяти
U 08B7, 2F82,95 ;6652            CLR WR[1]                                ; ожидаемые данные равны 0
U 08B8, 9912,15 ;6653            MEM.REQ[WRITE.P] ADRS[T9] DT[BYTE]       ; подготовка записи в память
U 08B9, B29C,15 ;6654            WRITE.MEM LS[ZERO]                       ; запись 0 в память
U 08BA, 1813,95 ;6655            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ; подготовка чтения памяти
U 08BB, 3022,15 ;6656            MOV MEM.DATA TO WR[0]                    ; выборка данных из памяти
U 08BC, 0869,3C ;6657            JSR [CHECK.RESULT]                       ; проверка нулевого результата
U 08BD, 8888,64 ;6658            JMP [T2.2]                               ; зацикливание при ошибке, если разрешено
U 08BE, FF82,15 ;6659            INC LS[ERROR.NUMBER]                     ; увеличение номера ошибки до 3
                ;6660    T2.3:
U 08BF, 6512,15 ;6661            CLR LS[T9]                               ; используется как физический адрес основной памяти
U 08C0, 362C,95 ;6662            MOV LS[#FFFFFF00] TO WR[1]               ; ожидаемые данные=FFFFFF80
U 08C1, C74E,95 ;6663            BIS LS[#80] TO WR[1]                     ;
U 08C2, 9912,15 ;6664            MEM.REQ[WRITE.P] ADRS[T9] DT[BYTE]       ; подготовка записи в память
U 08C3, B24E,15 ;6665            WRITE.MEM LS[#80]                        ; запись значения 80(H) в память
U 08C4, 1813,95 ;6666            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ; подготовка чтения памяти
U 08C5, 3022,15 ;6667            MOV MEM.DATA TO WR[0]                    ; выборка данных из памяти
U 08C6, 0869,3C ;6668            JSR [CHECK.RESULT]                       ; проверка нулевого результата
U 08C7, 8888,F4 ;6669            JMP [T2.3]                               ; зацикливание при ошибке, если разрешено
U 08C8, FF82,15 ;6670            INC LS[ERROR.NUMBER]                     ; увеличение номера ошибки до 4
                ;6671    T2.4:
U 08C9, 6512,15 ;6672            CLR LS[T9]                               ; используется как физический адрес основной памяти
U 08CA, 2F82,95 ;6673            CLR WR[1]                                ; ожидаемые данные=00000080
U 08CB, C74E,95 ;6674            BIS LS[#80] TO WR[1]                     ;
U 08CC, 1912,35 ;6675            MEM.REQ[WRITE.P] ADRS[T9] DT[WORD]       ; подготовка записи в память
U 08CD, B24E,15 ;6676            WRITE.MEM LS[#80]                        ; запись значения 80(H) в память
U 08CE, 9813,B5 ;6677            MEM.REQ[READ.P] ADRS[T9] DT[WORD]        ; подготовка чтения памяти
U 08CF, 3022,15 ;6678            MOV MEM.DATA TO WR[0]                    ; выборка данных из памяти
U 08D0, 0869,3C ;6679            JSR [CHECK.RESULT]                       ; проверка нулевого результата
U 08D1, 088C,94 ;6680            JMP [T2.4]                               ; зацикливание при ошибке, если разрешено
U 08D2, FF82,15 ;6681            INC LS[ERROR.NUMBER]                     ; увеличение номера ошибки до 5
                ;6682    T2.5:
U 08D3, 6512,15 ;6683            CLR LS[T9]                               ; используется как физический адрес основной памяти
U 08D4, 369E,95 ;6684            MOV LS[#FFFFFFFF] TO WR[1]               ; ожидаемые данные=FFFF8000
U 08D5, 4528,95 ;6685            BIC LS[#FFFF] TO WR[1]                   ;
U 08D6, 475E,95 ;6686            BIS LS[#8000] TO WR[1]                   ;
U 08D7, 1912,35 ;6687            MEM.REQ[WRITE.P] ADRS[T9] DT[WORD]       ; подготовка записи в память
U 08D8, 325E,15 ;6688            WRITE.MEM LS[#8000]                      ; запись значения 8000 в память
U 08D9, 9813,B5 ;6689            MEM.REQ[READ.P] ADRS[T9] DT[WORD]        ; подготовка чтения памяти
U 08DA, 3022,15 ;6690            MOV MEM.DATA TO WR[0]                    ; выборка данных из памяти
U 08DB, 0869,3C ;6691            JSR [CHECK.RESULT]                       ; проверка нулевого результата
U 08DC, 888D,34 ;6692            JMP [T2.5]                               ; зацикливание при ошибке, если разрешено
                ;6693    END.T2:
```

```
;6694      .PAGE "ТЕСТ 3 - тест РЕГИСТРА КОДОВ ОПЕРАЦИЙ (модуль DAP)"
;6695      ;
;6696      ;
;6697      ; ОПИСАНИЕ ТЕСТА:
;6698      ;
;6699      ;  Этот тест проверяет, что работают правильно регистр кода операции,
;6700      ;  пути данных в/из регистра кода операции и схемы управления. Значи-
;6701      ;  тельная часть путей данных, используемых для проверки регистра ко-
;6702      ;  да операции, уже проверена в тесте шины IB и буферов. Путь данных
;6703      ;  начинается в памяти, где запрос памяти формирует сигнал загрузки
;6704      ;  данными регистра предвыборки инструкций. Отсюда данные пересылаются
;6705      ;  в регистр кода операции. Единственным способом проверки правильности
;6706      ;  данных регистра кода операции является пересылка этих данных в АЛУ,
;6707      ;  где они сравниваются с эталонными данными. Для этого данные пересыла-
;6708      ;  ются в регистр OS. Данные из регистра OS пересылаются в АЛУ и прове-
;6709      ;  ряются. Все восемь битов регистра кода операции должны быть проверены
;6710      ;  на "застревание" в состоянии "1" или в состоянии "0". В режиме совмес-
;6711      ;  тимости может разрешаться выход или регистра кода операции или регист-
;6712      ;  ра предвыборки инструкций. Для регистра предвыборки инструкций это было
;6713      ;  проверено. Для проверки регистра кода операции в WR должны быть установ-
;6714      ;  лены все нули. Тогда выдается инструкция  MOV WR[] TO LS[] для установ-
;6715      ;  ки низкого уровня сигнала COMPAT MODE. Затем инструкция DECODE со сбро-
;6716      ;  шенным CSR 08 должна загрузить регистр OS данными из регистра кода
;6717      ;  операции. Чтобы различить, что данные поступили из регистра предвыборки
;6718      ;  инструкций или из регистра кода операции, регистр предвыборки инструкций
;6719      ;  и регистр кода операции должны содержать разные данные.
;6720      ;
;6721      ; ПРЕДПОЛОЖЕНИЯ:
;6722      ;
;6723      ;  Предполагается, что тест инструкции DECODE выполнен успешно. 10 байтов
;6724      ;  эталона, используемых в тесте регистра кода операции, должны находиться
;6725      ;  в памяти.
;6726      ;
;6727      ; ШАГИ ТЕСТА:
;6728      ;
;6729      ;  1)Подготовка в LS[10] начального адреса эталонных данных в оперативной
;6730      ;    памяти. Начальная установка низкого уровня сигнала COMPAT MODE очисткой
;6731      ;    WR и выполнением инструкции MOV WR[] TO LS[].
;6732      ;  2)Загрузка регистра предвыборки инструкций из памяти для проверки регистра
;6733      ;    кода операции.
;6734      ;  3)Загрузка байта данных из регистра предвыборки инструкций в регистр кода
;6735      ;    операции инструкцией DECODE.OPC1. Поле управления переходами должно со-
;6736      ;    держать  NO JUMP,SKIP IF IB VALID.
;6737      ;  4)Для пересылки байта данных из регистра кода операции в регистр OS
;6738      ;     используется микроинструкция типа
;6739      ;        DECODE.IFUNC/SPEC,IB.REQ/NOP,CM.HI/NOP,LOAD.OS/LOAD.OS,
;6740      ;        RDEST/NOP,OPC.SPEC/SPEC
;6741      ;  5)Загрузка байта полученных данных в WR[0].
;6742      ;  6)Вызов подпрограммы генерации ожидаемых данных.
;6743      ;  7)Загрузка ожидаемых данных в WR[1].
;6744      ;  8)Если данные правильные и это левый байт данных, тогда переход на шаг 2),
;6745      ;    в противном случае, если ошибка, переход к программе обработки ошибок.
;6746      ;  9)Загрузка регистра предвыборки инструкций и регистра кода операции
;6747      ;    различными и известными данными. Если данные поступают из регистра
;6748      ;    предвыборки инструкций, тогда переход к концу теста, в противном случае
```

```
                ;6749       ;     переход на обработку ошибки.
                ;6750       ;
                ;6751       ;     ОШИБКИ:
                ;6752       ;
                ;6753       ;       ошибка 1 - неправильно работают логические схемы после сигнала COMPAT MODE
                ;6754       ;                  (данные из регистра предвыборки инструкций).
                ;6755       ;       ошибка 2 - данные из регистра кода операции неправильные или в результа-
                ;6756       ;                  те застревания линий регистра кода операции, или в результате
                ;6757       ;                  застревания битов регистра кода операции, или в результате
                ;6758       ;                  неисправной работы схем управления.
                ;6759       ;       ошибка 3 - неправильно формируется запрос памяти, когда сигнал
                ;6760       ;                  IB.REQ сброшен.
                ;6761       ;       ошибка 4 - PC увеличивается, когда сигнал IB.REQ сброшен.
                ;6762       ;
                ;6763       ;     НАЛАДКА:
                ;6764       ;
                ;6765       ;       ОШИБКА 1: Если данные в регистре OS не такие, какие должны быть, тогда
                ;6766       ;                 возможно, что они поступают из регистра предвыборки инструк-
                ;6767       ;                 ций, что означает, что сигнал COMPAT MODE был установлен в
                ;6768       ;                 низкий уровень, когда он должен быть высоким. Возможным источ-
                ;6769       ;                 ником неправильной работы может быть ПМЛ РЕЖИМ СОВМЕСТ.,РЕГ.
                ;6770       ;                 АДРЕСАЦИЯ или входы LOAD PSL и BUS Y D31 этой ПМЛ.
                ;6771       ;       ОШИБКА 2: Различие между ожидаемыми и полученными данными может быть
                ;6772       ;                 по двум причинам. Первой причиной может быть неправильная ра-
                ;6773       ;                 бота буфера или связей к нему. При неисправности такого типа
                ;6774       ;                 будет постоянно неправильным один бит полученных данных из-за
                ;6775       ;                 застревания бита в состоянии "1" или "0". Другой причиной мо-
                ;6776       ;                 жет быть неправильная работа схем управления. Так как нет воз-
                ;6777       ;                 можности определить, был ли загружен регистр кода операции, обе
                ;6778       ;                 инструкции DECODE в одинаковой степени могут выполняться
                ;6779       ;                 неправильно, если данные не поступили. Для стробирования данных
                ;6780       ;                 в регистр кода операции формируется сигнал IRD STATE L. Если
                ;6781       ;                 этот сигнал не возбужден, тогда необходимо проверить ПМЛ РЕЖИМ
                ;6782       ;                 СОВМЕСТ., РЕГ. АДРЕСАЦИЯ. Для разрешения выхода из регистра
                ;6783       ;                 кода операции в регистр OS сигнал CSR 08 H должен иметь
                ;6784       ;                 низкий уровень, также, как и сигнал COMPAT MODE H. Если
                ;6785       ;                 сигнал CSR 08 H неправильный, тогда неисправна связь. Если
                ;6786       ;                 сигнал COMPAT MODE H неправильный, тогда возможно, что неис-
                ;6787       ;                 правна ПМЛ РЕЖИМ СОВМЕСТ.,РЕГ. АДРЕСАЦИЯ.
                ;6788       ;       ОШИБКА 3: Если сигнал IB.REQ сброшен, не должен формироваться сиг-
                ;6789       ;                 нал запроса памяти. Если запрос формируется, тогда, по-видимому,
                ;6790       ;                 неправильный вход CSR 08 и ПМЛ УПР.IB, РЕГ.ПРЕДВЫБ.ИНСТРУКЦИЙ.
                ;6791       ;                 Вход должен иметь низкий уровень, чтобы выход PC EQUALS 3 H
                ;6792       ;                 был низким, что предотвращает формирование сигнала запроса
                ;6793       ;                 памяти.
                ;6794       ;       ОШИБКА 4: СМ.ОШИБКУ 3.
                ;6795       T.3:
U 08DD, B65E,15 ;6796               MOV LS[BEGIN.TEST] TO WR[0]             ; установка бита 15 в WR0 для слова управления и
                ;6797                                                       ; состояния
U 08DE, 3E80,15 ;6798               MOV WR[0] TO LS[CONTROL.STATUS]         ; установка бита 15 в слове управления и состояния
                ;6799                                                       ; Бит 15 указывает начало теста для консольного
                ;6800                                                       ; процессора
U 08DF, 10E0,15 ;6801               MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для консольного
                ;6802                                                       ; процессора
                ;6803       WAIT.T3.0:
```

```
U 08E0, 888E,04 ;6804            JMP [WAIT.T3.0]                          ; зацикливание для ожидания ответа консольного
                ;6805                                                     ; процессора
U 08E1, B724,15 ;6806            MOV LS[HI.IPL] TO WR[0]                  ;
U 08E2, BFFE,15 ;6807            MOV WR[0] TO LS[PSL.HW]                  ; сброс сигнала COMPAT MODE для гарантии, что данные
                ;6808                                                     ; поступают из регистра кода операции
U 08E3, DF26,15 ;6809            MCOM LS[#FF] TO WR[0]                    ;
U 08E4, 3E8A,15 ;6810            MOV WR[0] TO LS[ERROR.MASK]              ; маска ошибки = FFFFFF00
U 08E5, 65A0,15 ;6811            CLR LS[PREVIOUS.ERROR]                   ; очистка предыдущего номера ошибки
U 08E6, B640,15 ;6812            MOV LS[#1] TO WR[0]                      ; установка 1 в WR0
U 08E7, BE82,15 ;6813            MOV WR[0] TO LS[ERROR.NUMBER]            ; установка номера для первой ошибки
U 08E8, A3C0,15 ;6814            ROL WR[0]                                ; установка 2 в  WR0
U 08E9, 3E8C,15 ;6815            MOV WR[0] TO LS[MODULE.NUM]              ; установка кода модуля для модуля DAP
U 08EA, B646,15 ;6816            MOV LS[BIT3] TO WR[0]                    ;
U 08EB, 3E20,15 ;6817            MOV WR[0] TO LS[PC]                      ; начальная установка указателя памяти для первого
                ;6818                                                     ; эталона сдвигаемых единиц
U 08EC, 364E,15 ;6819            MOV LS[#80] TO WR[0]                     ;
U 08ED, C746,15 ;6820            BIS LS[#8] TO WR[0]                      ;
U 08EE, 3E10,15 ;6821            MOV WR[0] TO LS[T8]                      ; T8=88(H) - конец эталона сдвигаемых единиц
U 08EF, 2100,15 ;6822            DEC WR[0]                                ;
U 08F0, BE14,15 ;6823            MOV WR[0] TO LS[T10]                     ; T10=87(H)
U 08F1, 2040,15 ;6824            INC WR[0]                                ;
U 08F2, 4744,15 ;6825            BIS LS[BIT2] TO WR[0]                    ;
U 08F3, BE12,15 ;6826            MOV WR[0] TO LS[T9]                      ; T9=8C(H) - указатель слова, состоящего из единиц
U 08F4, 2F87,95 ;6827            CLR WR[3]                                ;
                ;6828    LOOP.T3.1:
U 08F5, 1B21,75 ;6829            MEM.REQ[IB.FILL] ADRS[PC] DT[LONG]       ; заполнение PFR эталоном сдвигаемых единиц
                ;6830    T3.1:
U 08F6, 3642,15 ;6831            MOV LS[#2] TO WR[0]                      ;
U 08F7, BE82,15 ;6832            MOV WR[0] TO LS[ERROR.NUMBER]            ; номер ошибки=2
U 08F8, B620,15 ;6833            MOV LS[PC] TO WR[0]                      ;
U 08F9, 3E16,15 ;6834            MOV WR[0] TO LS[T11]                     ; сохранение PC
                ;6835              DECODE.OPC1 ADRS[T3.JUST.IN.CASE],     ;
U 08FA, 0487,1E ;6836            SKIP.IF(IB VALID)                        ; пересылка IB в регистр кода операции и увеличение PC
                ;6837    T3.JUST.IN.CASE:
U 08FB, DB00,15 ;6838            NOP                                      ;
U 08FC, 9B13,75 ;6839            MEM.REQ[IB.FILL] ADRS[T9] DT[LONG]       ; установка новых данных в PFR
                ;6840              TST WR[3],                             ;
U 08FD, 2007,B5 ;6841            DT(LONG)&SET.ALU.CC                      ; первый проход ?
U 08FE, 0890,E1 ;6842            JMP.IF[NEQ] TO [T3.NOT1]                 ; проверка только при первом проходе для первой ошибки
U 08FF, B69F,95 ;6843            MOV LS[ONES] TO WR[3]                    ; установка признака
U 0900, FC82,15 ;6844            DEC LS[ERROR.NUMBER]                     ; номер ошибки=1
U 0901, B724,15 ;6845            MOV LS[HI.IPL] TO WR[0]                  ;
U 0902, 477E,15 ;6846            BIS LS[BIT31] TO WR[0]                   ;
U 0903, BFFE,15 ;6847            MOV WR[0] TO LS[PSL.HW]                  ; установка режима совместимости
U 0904, 8401,4E ;6848            MOV IB.DATA TO OS                        ; чтение регистра предвыборки инструкций
U 0905, DB00,15 ;6849            NOP                                      ;
U 0906, B6F8,15 ;6850            MOV LS[OS] TO WR[0]                      ; пересылка полученных данных
U 0907, 3626,95 ;6851            MOV LS[#FF] TO WR[1]                     ; ожидаемые данные=FF
U 0908, 0869,3C ;6852            JSR [CHECK.RESULT]                       ;
U 0909, 0892,F4 ;6853            JMP [T3.LOE]                             ;
U 090A, 7C20,15 ;6854            DEC LS[PC]                               ; восстановление PC
U 090B, B724,15 ;6855            MOV LS[HI.IPL] TO WR[0]                  ;
U 090C, BFFE,15 ;6856            MOV WR[0] TO LS[PSL.HW]                  ; сброс регистра совместимости
U 090D, FF82,15 ;6857            INC LS[ERROR.NUMBER]                     ; увеличение номера ошибки до 2
                ;6858    T3.NOT1:
```

```
U 090E, 0400,4E ;6859            MOV CM.IB.DATA TO OS                    ; пересылка регистра кода операции в регистр OS
                ;6860                                                    ; примечание: PC не увеличивается!
U 090F, DB00,15 ;6861            NOP                                     ;
U 0910, B6F8,15 ;6862            MOV LS[OS] TO WR[0]                     ; пересылка полученных данных
U 0911, 9817,95 ;6863            MEM.REQ[READ.P] ADRS[T11] DT[BYTE]      ;
U 0912, B022,95 ;6864            MOV MEM.DATA TO WR[1]                   ; чтение ожидаемых данных
U 0913, 0869,3C ;6865            JSR [CHECK.RESULT]                      ; проверка регистра кода операции на правильность
                ;6866                                                    ; данных
U 0914, 0892,F4 ;6867            JMP [T3.LOE]                            ;
U 0915, B620,15 ;6868            MOV LS[PC] TO WR[0]                     ; чтение обновленного PC
                ;6869              XOR.LS[T9] WITH WR[0] TO Q,           ;
U 0916, CD12,35 ;6870            DT(LONG)&SET.ALU.CC                     ; выполнено ?
U 0917, 088F,51 ;6871            JMP.IF[NEQ] TO [LOOP.T3.1]              ; возврат, если нет
                ;6872    T3.3:
U 0918, 36C0,15 ;6873            MOV LS[#3(H)] TO WR[0]                  ;
U 0919, BEB2,15 ;6874            MOV WR[0] TO LS[ERROR.NUMBER]           ; номер ошибки=3
U 091A, 3612,15 ;6875            MOV LS[T9] TO WR[0]                     ;
U 091B, 3E20,15 ;6876            MOV WR[0] TO LS[PC]                     ; PC=BC(H)
U 091C, 1B21,75 ;6877            MEM.REQ[IB.FILL] ADRS[PC] DT[LONG]      ; заполнение PFR единицами
                ;6878              DECODE.OPC1 ADRS[T3.JUST.IN.CASE2],   ;
U 091D, 8497,1E ;6879            SKIP.IF(IB VALID)                       ; пересылка IB в регистр кода операции и увеличение PC
                ;6880    T3.JUST.IN.CASE2:
U 091E, DB00,15 ;6881            NOP                                     ;
U 091F, 3614,15 ;6882            MOV LS[T10] TO WR[0]                    ;
U 0920, 3E20,15 ;6883            MOV WR[0] TO LS[PC]                     ; PC=B7(H)
U 0921, 0400,4E ;6884            MOV CM.IB.DATA TO OS                    ; пересылка данных в OS и пропуск, если IB.VALID, без
                ;6885                                                    ; запроса памяти
U 0922, DB00,15 ;6886            NOP                                     ;
U 0923, 0400,4E ;6887            MOV CM.IB.DATA TO OS                    ; чтение байта и проверка, был ли запрос памяти
U 0924, DB00,15 ;6888            NOP                                     ;
U 0925, B6F8,15 ;6889            MOV LS[OS] TO WR[0]                     ;
U 0926, 3626,95 ;6890            MOV LS[#FF] TO WR[1]                    ;
U 0927, 0869,3C ;6891            JSR [CHECK.RESULT]                      ; проверка, что запрос памяти не выдавался
U 0928, 8891,84 ;6892            JMP [T3.3]                              ; переход на обработку ошибки
U 0929, FF82,15 ;6893            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 4
U 092A, B620,15 ;6894            MOV LS[PC] TO WR[0]                     ; чтение PC
U 092B, B614,95 ;6895            MOV LS[T10] TO WR[1]                    ; ожидаемые данные=B7(H)
U 092C, 0869,3C ;6896            JSR [CHECK.RESULT]                      ; проверка, что PC не увеличился
U 092D, 8891,84 ;6897            JMP [T3.3]                              ; переход на обработку ошибки
U 092E, 8893,54 ;6898            JMP [END.T3]                            ;
                ;6899    T3.LOE:
U 092F, 7C20,15 ;6900            DEC LS[PC]                              ; при зацикливании на ошибке PC должен быть
                ;6901                                                    ; уменьшен, чтобы проверялся тот же самый байт
U 0930, B643,15 ;6902            MOV LS[BIT1] TO WR[2]                   ;
U 0931, 2045,15 ;6903            INC WR[2]                               ; установка 2 младших битов
                ;6904              BIT WR[2] WITH LS[PC],                ;
U 0932, D921,35 ;6905            DT(LONG)&SET.ALU.CC                     ; младшие биты PC установлены?
U 0933, 088F,61 ;6906            JMP.IF[NEQ] TO [T3.1]                   ; зацикливание при ошибке без перезаписи PFR
U 0934, 088F,54 ;6907            JMP [LOOP.T3.1]                         ; зацикливание при ошибке с перезаписью PFR
                ;6908    END.T3:
```

```
;6909      .PAGE "ТЕСТ 4 - тест ПЗУ ДЕШИФРАТОРА СПЕЦИФИКАТОРОВ (модуль DAP)"
;6910      ;
;6911      ; ОПИСАНИЕ ТЕСТА:
;6912      ;
;6913      ;   Этот тест проверяет содержимое и адресацию ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ. Для
;6914      ;   обращения ко всем ячейкам ПЗУ используется поле IFUNC микроинструкции
;6915      ;   DECODE совместно со значением шины IB. Изменение в поле IFUNC вызывает
;6916      ;   изменение типа выполняемой инструкции DECODE. Поле IFUNC содержит три бита
;6917      ;   и поэтому для обращения ко всей микросхеме ПЗУ может использоваться 8
;6918      ;   различных инструкций DECODE. Инструкция DECODE, которой разрешает работу
;6919      ;   ПЗУ ДЕШ  СПЕЦИФИКАТОРОВ, вызывает переход в перезаписываемой управляющей памяти.
;6920      ;   Адрес ячейки перехода определяется восемью битами из ПЗУ (младшие биты) и
;6921      ;   битами из инструкции DECODE (старшие). Так как старшие биты будут одни и те
;6922      ;   же, возможны только 256 последовательных ячеек, куда допускается переход, и
;6923      ;   каждая из этих ячеек содержит инструкцию INC WR[3]. Так как блок начинается
;6924      ;   от нуля, адрес перехода можно вычислить вычитанием содержимого WR[3] из
;6925      ;   256 (конец блока). Действительный адрес перехода загружается в WR[0]. Ожи-
;6926      ;   даемое значение генерируется в подпрограмме с использованием цикла. Ожи-
;6927      ;   даемое значение загружается в WR[1] и вызывается программа проверки резуль-
;6928      ;   тата. Если имеется ошибка и зацикливание при ошибке разрешено, счетчик, ко-
;6929      ;   торый указывает ячейку ПЗУ, не увеличивается, так что выбирается тот же ад-
;6930      ;   рес. Младший бит адреса ПЗУ формируется как функция "НЕ-И" из трех младших
;6931      ;   битов шины IB. Для проверки каждого бита IB должны использоваться коды от
;6932      ;   0 до 7. Нулевой код уже проверялся.
;6933      ;
;6934      ; ПРЕДПОЛОЖЕНИЯ:
;6935      ;
;6936      ;   Предполагается, что память содержит значения, необходимые для адресации
;6937      ;   всех 512 байтов ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ. Также предполагается что тест ре-
;6938      ;   гистра кода операции выполнен успешно.
;6939      ;
;6940      ; ШАГИ ТЕСТА:
;6941      ;
;6942      ;   1) Начальная установка ячеек местной памяти, ячеек оперативной памяти и
;6943      ;      рабочих регистров, которые будут использоваться.
;6944      ;   2) Заполнение IB данными из памяти, которые будут вызывать нарастающий
;6945      ;      код на входе ПЗУ ДЕШИФРАТОРА СПЕЦИФИКАТОРОВ.
;6946      ;   3) Выполнение инструкции DECODE с IFUNC = 0, которая будет вызывать пере-
;6947      ;      ход к адресу в блоке из 256 инструкций INC WR[3] и возврат.
;6948      ;   4) Вычитание WR3 из 256(десятичн.) и помещение результата в WR0.
;6949      ;   5) Пересылка ожидаемых данных в WR1 и вызов программы обработки ошибок.
;6950      ;   6) Переключение бита 2 IB.
;6951      ;   7) Повторение шагов 2-5.
;6952      ;   8) Добавление 8 к коду, пересылаемому на IB и переключение битов 0-2.
;6953      ;   9) Повторение шагов 2-5.
;6954      ;  10) Повторение шагов 2-9 для IFUNC = 1 до 7.
;6955      ;  11) Выполнение инструкции DECODE с IFUNC = 0 и перемещающимся "0" в трех
;6956      ;      младших битах IB.
;6957      ;
;6958      ; ОШИБКИ:
;6959      ;
;6960      ;   ДРУГИЕ ДАННЫЕ: содержимое ячейки 1B0.
;6961      ;
;6962      ;   Ошибка 1 - неправильно работает адресация ПЗУ или неправильное содержимое
;6963      ;              ПЗУ. IFUNC = 0
```

```
                ;6964      ;    ошибка 2 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6965      ;               ПЗУ. IFUNC = 1
                ;6966      ;    ошибка 3 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6967      ;               ПЗУ. IFUNC = 2
                ;6968      ;    ошибка 4 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6969      ;               ПЗУ. IFUNC = 3
                ;6970      ;    ошибка 5 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6971      ;               ПЗУ. IFUNC = 4
                ;6972      ;    ошибка 6 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6973      ;               ПЗУ. IFUNC = 5
                ;6974      ;    ошибка 7 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6975      ;               ПЗУ. IFUNC = 6
                ;6976      ;    ошибка 8   неправильно работает адресация ПЗУ или неправильное содержимое
                ;6977      ;               ПЗУ. IFUNC = 7
                ;6978      ;    ошибка 9 - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6979      ;               ПЗУ. Входы схемы "НЕ-И" = 011
                ;6980      ;    ошибка A - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6981      ;               ПЗУ. Входы схемы "НЕ-И" = 101
                ;6982      ;    ошибка B - неправильно работает адресация ПЗУ или неправильное содержимое
                ;6983      ;               ПЗУ. Входы схемы "НЕ-И" = 110
                ;6984      ;
                ;6985      ; НАЛАДКА:
                ;6986      ;
                ;6987      ;    ОШИБКИ 1 - 8: Появление этих ошибок может быть вызвано тремя похожими
                ;6988      ;                  неисправностями. Может быть неправильное содержимое ПЗУ
                ;6989      ;                  или неправильно работает адресация, что может вызвать
                ;6990      ;                  двойную адресацию, если имеется неисправность в схемах
                ;6991      ;                  управления формированием следующего адреса,и переход
                ;6992      ;                  выполняется в то же самое место в блоке. Схемы управления
                ;6993      ;                  будут проверяться в следующем тесте. Входы ПЗУ можно опреде-
                ;6994      ;                  лить проверкой WR[2] и вычитанием этого значения из 512.
                ;6995      ;
                ;6996      ;    ОШИБКИ 9 - B: Или схема "НЕ-И" на входе ПЗУ неисправна, или можно подозревать
                ;6997      ;                  связи к этой схеме.
                ;6998      T.4:
U 0935, B65E,15 ;6999              MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;7000                                                       ; состояния
U 0936, 3E80,15 ;7001              MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
                ;7002                                                       ; 15 указывает начало теста для консольного процессора
U 0937, 10E0,15 ;7003              MISC [SET.CP.ATTN]                       ; выдача сигнала CPU ATTN для консольного процессора
                ;7004      WAIT.T4.0:
U 0938, 0893,84 ;7005              JMP [WAIT.T4.0]                          ; зацикливание для ожидания ответа консольного
                ;7006                                                       ; процессора
U 0939, B724,15 ;7007              MOV LS[HI.IPL] TO WR[0]                  ;
U 093A, BFFE,15 ;7008              MOV WR[0] TO LS[PSL.HW]                  ; сброс режима совместимости
U 093B, DF26,15 ;7009              MCOM LS[#FF] TO WR[0]                    ;
U 093C, 3E8A,15 ;7010              MOV WR[0] TO LS[ERROR.MASK]              ; маска ошибки = FFFFFF00
U 093D, 65A0,15 ;7011              CLR LS[PREVIOUS.ERROR]                   ; очистка предыдущего номера ошибки
U 093E, B640,15 ;7012              MOV LS[#1] TO WR[0]                      ; установка 1 в WR0
U 093F, BE82,15 ;7013              MOV WR[0] TO LS[ERROR.NUMBER]            ; установка номера ошибки для первой ошибки
U 0940, A3C0,15 ;7014              ROL WR[0]                                ; установка 2 в WR0
U 0941, 3EBC,15 ;7015              MOV WR[0] TO LS[MODULE.NUM]              ; установка кода модуля для модуля DAP
U 0942, E520,15 ;7016              CLR.LS[PC]                               ; всегда проверяется младший байт регистра предвыборки инстр
U 0943, 8881,7C ;7017              JSR [ASSERT.OTHER]                       ;
                ;7018      ;    WR2 - данные, пересылаемые в IB для формирования адреса ПЗУ (используется ячейка памяти 94)
```

```
                    ;7019      ;    WR3 - блок, содержащий 256 микроинструкций INC WR[3]
                    ;7020      ;    T8  = 94 (указатель ячейки памяти 94)
                    ;7021      ;    T9  - указатель ответов (начальное значение 98)
                    ;7022      ;    T10 - ячейка временного хранения
                    ;7023      ;    T11 = 107 (конец цикла)
                    ;7024      ;    T12 = 0111(в) (биты, подлежащие переключению)
U 0944,  364E,15    ;7025             MOV LS[#80] TO WR[0]                          ;
U 0945,  C048,15    ;7026             ADD LS[#10] TO WR[0]                          ;
U 0946,  C044,15    ;7027             ADD LS[#4] TO WR[0]                           ;
U 0947,  3E10,15    ;7028             MOV WR[0] TO LS[T8]                           ; T8 = 94
U 0948,  C044,15    ;7029             ADD LS[#4] TO WR[0]                           ;
U 0949,  BE12,15    ;7030             MOV WR[0] TO LS[T9]                           ; T9 = 98
U 094A,  3650,15    ;7031             MOV LS[#100] TO WR[0]                         ;
U 094B,  4046,15    ;7032             ADD LS[#8] TO WR[0]                           ;
U 094C,  2100,15    ;7033             DEC WR[0]                                     ;
U 094D,  3E16,15    ;7034             MOV WR[0] TO LS[T11]                          ; T11 =107
U 094E,  3647,15    ;7035             MOV LS[#8] TO WR[2]                           ;
U 094F,  2105,15    ;7036             DEC WR[2]                                     ; WR2 = 7
U 0950,  3E15,15    ;7037             MOV WR[2] TO LS[T10]                          ;
U 0951,  3E19,15    ;7038             MOV WR[2] TO LS[T12]                          ; T12 = 0111(B)
U 0952,  1910,75    ;7039             MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0953,  B214,15    ;7040             WRITE.MEM LS[T10]                             ; ячейка памяти 94 = 7
U 0954,  B724,15    ;7041             MOV LS[HI.IPL] TO WR[0]                       ;
U 0955,  477E,15    ;7042             BIS LS[BIT31] TO WR[0]                        ;
U 0956,  BFFE,15    ;7043             MOV WR[0] TO LS[PSL.HW]                       ; установка режима совместимости для передачи данных из
                    ;7044                                                           ; регистра предвыборки инструкций, а не из регистра кода
                    ;7045                                                           ; операции
                    ;7046     LOOP.T4.1:
U 0957,  1B11,75    ;7047             MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций данными из
                    ;7048                                                           ; ячейки памяти 94
U 0958,  3615,95    ;7049             MOV LS[T10] TO WR[3]                          ;
U 0959,  BE89,95    ;7050             MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 095A,  2F87,95    ;7051             CLR WR[3]                                     ;
                    ;7052      ;    Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                    ;7053      ;    ванному следующим образом:
                    ;7054      ;
                    ;7055      ;        7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                    ;7056      ;
U 095B,  8470,0C    ;7057             OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 095C,  CA51,95    ;7058             SUB WR[3] FROM LS[#100] TO Q                   ;
U 095D,  A180,15    ;7059             MOV Q TO WR[0]                                ; полученные данные =100(H) (256 десятичное)-WR3
U 095E,  1813,95    ;7060             MEM.REQ[READ.P] ADRS[T9] DT[BYTE]             ;
U 095F,  B022,95    ;7061             MOV MEM.DATA TO WR[1]                         ; выборка ожидаемых данных
U 0960,  0869,3C    ;7062             JSR [CHECK.RESULT]                            ; проверка
U 0961,  0895,74    ;7063             JMP [LOOP.T4.1]                               ; зацикливание при ошибке, если разрешено
U 0962,  FF12,15    ;7064             INC LS[T9]                                    ; увеличение указателя ответов
U 0963,  4319,15    ;7065             XOR LS[T12] TO WR[2]                          ; переключение битов 0-2 в WR2
U 0964,  3E15,15    ;7066             MOV WR[2] TO LS[T10]                          ;
U 0965,  1910,75    ;7067             MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0966,  B214,15    ;7068             WRITE.MEM LS[T10]                             ; запись нового кода для пересылки на IB в ячейку памяти 94
                    ;7069     LOOP2.T4.1:
U 0967,  1B11,75    ;7070             MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций данными из ячей
U 0968,  3615,95    ;7071             MOV LS[T10] TO WR[3]                          ;
U 0969,  BE89,95    ;7072             MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 096A,  2F87,95    ;7073             CLR WR[3]
```

```
U 096B, B470,0C ;7074            OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 096C, CA51,95 ;7075            SUB WR[3] FROM LS[#100] TO Q                  ;
U 096D, A180,15 ;7076            MOV Q TO WR[0]                                ; выборка полученных данных
U 096E, 1813,95 ;7077            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]             ;
U 096F, B022,95 ;7078            MOV MEM.DATA TO WR[1]                         ; выборка ожидаемых данных
U 0970, 0869,3C ;7079            JSR [CHECK.RESULT]                            ; проверка
U 0971, 0896,74 ;7080            JMP [LOOP2.T4.1]                              ; зацикливание при ошибке, если разрешено
U 0972, FF12,15 ;7081            INC LS[T9]                                    ;
U 0973, C047,15 ;7082            ADD LS[#8] TO WR[2]                           ; добавление 8(H) к коду для пересылки на IB и переключение
U 0974, 4319,15 ;7083            XOR LS[T12] TO WR[2]                          ; битов 0-2
U 0975, 3E15,15 ;7084            MOV WR[2] TO LS[T10]                          ;
U 0976, 1910,75 ;7085            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0977, B214,15 ;7086            WRITE.MEM LS[T10]                             ; запись нового кода в ячейку памяти 94
U 0978, 5816,15 ;7087            MOV LS[T11] TO Q                              ;
                ;7088              CMP WR[2] WITH Q,                           ;
U 0979, A484,35 ;7089            DT(LONG)&SET.ALU.CC                           ; код = 107(H)?
U 097A, 0895,71 ;7090            JMP.IF[NEQ] TO [LOOP.T4.1]                    ; зацикливание, если нет
                ;7091    T4.2:
U 097B, FF82,15 ;7092            INC LS[ERROR.NUMBER]                          ; номер ошибки = 2
U 097C, 3647,15 ;7093            MOV LS[#8] TO WR[2]                           ;
U 097D, 2105,15 ;7094            DEC WR[2]                                     ; восстановление кода для пересылки на IB до 7
U 097E, 3E15,15 ;7095            MOV WR[2] TO LS[T10]                          ;
U 097F, 1910,75 ;7096            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0980, B214,15 ;7097            WRITE.MEM LS[T10]                             ; восстановление этого кода в памяти
                ;7098    LOOP.T4.2:
U 0981, 1B11,75 ;7099            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций данными из
                ;7100                                                          ; ячейки памяти 94
U 0982, 3615,95 ;7101            MOV LS[T10] TO WR[3]                          ;
U 0983, BE89,95 ;7102            MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0984, 2F87,95 ;7103            CLR WR[3]                                     ;
                ;7104     ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7105     ;  ванному следующим образом:
                ;7106     ;
                ;7107     ;     7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7108     ;
U 0985, 0472,0C ;7109            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0986, CA51,95 ;7110            SUB WR[3] FROM LS[#100] TO Q                  ;
U 0987, A180,15 ;7111            MOV Q TO WR[0]                                ; полученные данные = 100(H) (256 десятичн.)-WR3
U 0988, 1813,95 ;7112            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]             ;
U 0989, B022,95 ;7113            MOV MEM.DATA TO WR[1]                         ; выборка ожидаемых данных из памяти
U 098A, 0869,3C ;7114            JSR [CHECK.RESULT]                            ; проверка
U 098B, 8898,14 ;7115            JMP [LOOP.T4.2]                               ; зацикливание при ошибке, если разрешено
U 098C, FF12,15 ;7116            INC LS[T9]                                    ; увеличение указателя ответов
U 098D, 4319,15 ;7117            XOR LS[T12] TO WR[2]                          ; переключение битов 0-2 в WR2
U 098E, 3E15,15 ;7118            MOV WR[2] TO LS[T10]                          ;
U 098F, 1910,75 ;7119            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0990, B214,15 ;7120            WRITE.MEM LS[T10]                             ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7121    LOOP2.T4.2:
U 0991, 1B11,75 ;7122            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций из ячейки
                ;7123                                                          ; памяти 94
U 0992, 3615,95 ;7124            MOV LS[T10] TO WR[3]                          ;
U 0993, BE89,95 ;7125            MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0994, 2F87,95 ;7126            CLR WR[3]                                     ;
U 0995, 0472,0C ;7127            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0996, CA51,95 ;7128            SUB WR[3] FROM LS[#100] TO Q                  ;
```

```
U 0997, A180,15 ;7129            MOV Q TO WR[0]                          ; выборка полученных данных
U 0998, 1813,95 ;7130            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 0999, B022,95 ;7131            MOV MEM.DATA TO WR[1]                   ; выборка ожидаемых данных
U 099A, 0869,3C ;7132            JSR [CHECK.RESULT]                      ; проверка
U 099B, 0899,14 ;7133            JMP [LOOP2.T4.2]                        ; зацикливание при ошибке, если разрешено
U 099C, FF12,15 ;7134            INC LS[T9]                              ;
U 099D, C047,15 ;7135            ADD LS[#8] TO WR[2]                     ; добавление 8(H) к коду для пересылки на IB и переключение
U 099E, 4319,15 ;7136            XOR LS[T12] TO WR[2]                    ; битов 0-2
U 099F, 3E15,15 ;7137            MOV WR[2] TO LS[T10]                    ;
U 09A0, 1910,75 ;7138            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 09A1, B214,15 ;7139            WRITE.MEM LS[T10]                       ; запись нового кода в ячейку памяти 94
U 09A2, 5816,15 ;7140            MOV LS[T11] TO Q                        ;
                ;7141              CMP WR[2] WITH Q,                     ;
U 09A3, A484,35 ;7142            DT(LONG)&SET.ALU.CC                     ; код = 107(H)?
U 09A4, 8898,11 ;7143            JMP.IF[NEQ] TO [LOOP.T4.2]              ; зацикливание, если нет
                ;7144    T4.3:
U 09A5, FF82,15 ;7145            INC LS[ERROR.NUMBER]                    ; номер ошибки = 3
U 09A6, 3647,15 ;7146            MOV LS[#8] TO WR[2]                     ;
U 09A7, 2105,15 ;7147            DEC WR[2]                               ; восстановление кода для пересылки на IB до 7
U 09A8, 3E15,15 ;7148            MOV WR[2] TO LS[T10]                    ;
U 09A9, 1910,75 ;7149            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 09AA, B214,15 ;7150            WRITE.MEM LS[T10]                       ; восстановление втого кода в памяти
                ;7151    LOOP.T4.3:
U 09AB, 1B11,75 ;7152            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; заполнение регистра предвыборки инструкций из ячейки
                ;7153                                                    ; памяти 94
U 09AC, 3615,95 ;7154            MOV LS[T10] TO WR[3]                    ;
U 09AD, BE89,95 ;7155            MOV WR[3] TO LS[ADDRESS.DATA]           ; установка поля "ДРУГИЕ ДАННЫЕ"
U 09AE, 2F87,95 ;7156            CLR WR[3]                               ;
                ;7157     ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7158     ;  ванному следующим образом:
                ;7159     ;
                ;7160     ;    7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7161     ;
U 09AF, 0474,0C ;7162            OPC2/DECODE,DEC.ADRS/7,IFUNC/2,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 09B0, CA51,95 ;7163            SUB WR[3] FROM LS[#100] TO Q            ;
U 09B1, A180,15 ;7164            MOV Q TO WR[0]                          ; полученные данные = 100(H) (256 десятичн.)-WR3
U 09B2, 1813,95 ;7165            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 09B3, B022,95 ;7166            MOV MEM.DATA TO WR[1]                   ; выборка ожидаемых данных из памяти
U 09B4, 0869,3C ;7167            JSR [CHECK.RESULT]                      ; проверка
U 09B5, 089A,B4 ;7168            JMP [LOOP.T4.3]                         ; зацикливание при ошибке, если разрешено
U 09B6, FF12,15 ;7169            INC LS[T9]                              ; увеличение указателя ответов
U 09B7, 4319,15 ;7170            XOR LS[T12] TO WR[2]                    ; переключение битов 0-2 в WR2
U 09B8, 3E15,15 ;7171            MOV WR[2] TO LS[T10]                    ;
U 09B9, 1910,75 ;7172            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 09BA, B214,15 ;7173            WRITE.MEM LS[T10]                       ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7174    LOOP2.T4.3:
U 09BB, 1B11,75 ;7175            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; заполнение регистра предвыборки инструкций данными из
                ;7176                                                    ; памяти
U 09BC, 3615,95 ;7177            MOV LS[T10] TO WR[3]                    ;
U 09BD, BE89,95 ;7178            MOV WR[3] TO LS[ADDRESS.DATA]           ; установка поля "ДРУГИЕ ДАННЫЕ"
U 09BE, 2F87,95 ;7179            CLR WR[3]                               ;
U 09BF, 0474,0C ;7180            OPC2/DECODE,DEC.ADRS/7,IFUNC/2,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 09C0, CA51,95 ;7181            SUB WR[3] FROM LS[#100] TO Q            ;
U 09C1, A180,15 ;7182            MOV Q TO WR[0]                          ; выборка полученных данных
U 09C2, 1813,95 ;7183            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
```

```
U 09C3, B022,95 ;7184            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных
U 09C4, 0869,3C ;7185            JSR [CHECK.RESULT]                       ; проверка
U 09C5, 8898,B4 ;7186            JMP [LOOP2.T4.3]                         ; зацикливание при ошибке, если разрешено
U 09C6, FF12,15 ;7187            INC LS[T9]                               ;
U 09C7, C047,15 ;7188            ADD LS[#8] TO WR[2]                      ; добавление 8(H) к коду для пересылки на IB и переключение
U 09C8, 4319,15 ;7189            XOR LS[T12] TO WR[2]                     ; битов 0-2
U 09C9, 3E15,15 ;7190            MOV WR[2] TO LS[T10]                     ;
U 09CA, 1910,75 ;7191            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 09CB, B214,15 ;7192            WRITE.MEM LS[T10]                        ; запись нового кода в ячейку памяти 94
U 09CC, 5816,15 ;7193            MOV LS[T11] TO Q                         ;
                ;7194              CMP WR[2] WITH Q,                      ;
U 09CD, A484,35 ;7195            DT(LONG)&SET.ALU.CC                      ; код = 107(H)?
U 09CE, 089A,B1 ;7196            JMP.IF[NEQ] TO [LOOP.T4.3]               ; зацикливание, если нет
                ;7197    T4.4:
U 09CF, FF82,15 ;7198            INC LS[ERROR.NUMBER]                     ; номер ошибки = 4
U 09D0, 3647,15 ;7199            MOV LS[#8] TO WR[2]                      ;
U 09D1, 2105,15 ;7200            DEC WR[2]                                ; восстановление кода для записи на IB до 7
U 09D2, 3F15,15 ;7201            MOV WR[2] TO LS[T10]                     ;
U 09D3, 1910,75 ;7202            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 09D4, B214,15 ;7203            WRITE.MEM LS[T10]                        ; восстановление этого кода в памяти
                ;7204    LOOP.T4.4:
U 09D5, 1B11,75 ;7205            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; заполнение регистра предвыборки инструкций данными из
                ;7206                                                     ; ячейки памяти 94
U 09D6, 3615,95 ;7207            MOV LS[T10] TO WR[3]                     ;
U 09D7, BF89,95 ;7208            MOV WR[3] TO LS[ADDRESS.DATA]            ; установка поля "ДРУГИЕ ДАННЫЕ"
U 09D8, 2F87,95 ;7209            CLR WR[3]                                ;
                ;7210     ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7211     ;  ванному следующим образом:
                ;7212     ;
                ;7213     ;     7 в DEC.ADRS+ПЗУ.ДЕШ.СПЕЦИФИКАТОРОВ(IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7214     ;
U 09D9, 8476,0C ;7215            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 09DA, CA51,95 ;7216            SUB WR[3] FROM LS[#100] TO Q             ;
U 09DB, A180,15 ;7217            MOV Q TO WR[0]                           ; полученные данные = 100(H) (256 десятичн.)-WR3
U 09DC, 1B13,95 ;7218            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 09DD, B022,95 ;7219            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных из памяти
U 09DE, 0869,3C ;7220            JSR [CHECK.RESULT]                       ; проверка
U 09DF, 089D,54 ;7221            JMP [LOOP.T4.4]                          ; зацикливание при ошибке, если разрешено
U 09E0, FF12,15 ;7222            INC LS[T9]                               ; увеличение указателя ответов
U 09E1, 4319,15 ;7223            XOR LS[T12] TO WR[2]                     ; переключение битов 0-2 в WR2
U 09E2, 3E15,15 ;7224            MOV WR[2] TO LS[T10]                     ;
U 09E3, 1910,75 ;7225            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 09E4, B214,15 ;7226            WRITE.MEM LS[T10]                        ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7227    LOOP2.T4.4:
U 09E5, 1B11,75 ;7228            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; заполнение регистра предвыборки инструкций данными из
                ;7229                                                     ; ячейки памяти 94
U 09E6, 3615,95 ;7230            MOV LS[T10] TO WR[3]                     ;
U 09E7, BE89,95 ;7231            MOV WR[3] TO LS[ADDRESS.DATA]            ; установка других данных
U 09E8, 2F87,95 ;7232            CLR WR[3]                                ;
U 09E9, 8476,0C ;7233            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 09EA, CA51,95 ;7234            SUB WR[3] FROM LS[#100] TO Q             ;
U 09EB, A180,15 ;7235            MOV Q TO WR[0]                           ; выборка полученных данных
U 09EC, 1B13,95 ;7236            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 09ED, B022,95 ;7237            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных
U 09EE, 0869,3C ;7238            JSR [CHECK.RESULT]                       ; проверка
```

```
U 09EE, 089E,54 ;7259            JMP [LOOP2.T4.4]                        ; зацикливание при ошибке, если разрешено
U 09F0, FF12,15 ;7260            INC LS[T9]                              ;
U 09F1, C047,15 ;7261            ADD LS[#8] TO WR[2]                     ; добавление 8(H) к коду для пересылки на IB и переключение
U 09F2, 4319,15 ;7262            XOR LS[T12] TO WR[2]                    ; битов 0-2
U 09F3, 1910,75 ;7263            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 09F4, B214,15 ;7264            WRITE.MEM LS[T10]                       ; запись нового кода в ячейку памяти 94
U 09F5, 5816,15 ;7265            MOV LS[T11] TO Q                        ;
                ;7266              CMP WR[2] WITH Q,                     ;
U 09F6, A484,35 ;7267            DT(LONG)&SET.ALU.CC                     ; код = 107(H)?
U 09F7, 089D,51 ;7268            JMP.IF[NEQ] TO [LOOP.T4.4]              ; зацикливание, если нет
                ;7269    T4.5:
U 09F8, FF82,15 ;7270            INC LS[ERROR.NUMBER]                    ; номер ошибки = 5
U 09F9, 3647,15 ;7271            MOV LS[#8] TO WR[2]                     ;
U 09FA, 2105,15 ;7272            DEC WR[2]                               ; восстановление кода для пересылки на IB до 7
U 09FB, 3E15,15 ;7273            MOV WR[2] TO LS[T10]                    ;
U 09FC, 1910,75 ;7274            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 09FD, B214,15 ;7275            WRITE.MEM LS[T10]                       ; восстановление второго кода в памяти
                ;7276
                ;7277    LOOP.T4.5:
U 09FE, 1B11,75 ;7278            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; заполнение регистра предвыборки инструкций данными из
                ;7279                                                    ; ячейки памяти 94
U 09FF, 3615,95 ;7280            MOV LS[T10] TO WR[3]                    ;
U 0A00, BE89,95 ;7281            MOV WR[3] TO LS[ADDRESS.DATA]           ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A01, 2F87,95 ;7282            CLR WR[3]                               ;
                ;7283      ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7284      ;  ванному следующим образом:
                ;7285      ;
                ;7286      ;     7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7287      ;
U 0A02, 0478,0C ;7288            OPC2/DECODE,DEC.ADRS/7,IFUNC/4,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A03, CA51,95 ;7289            SUB WR[3] FROM LS[#100] TO Q            ;
U 0A04, A180,15 ;7290            MOV Q TO WR[0]                          ; полученные данные = 100(H) (256 десятичн.)-WR3
U 0A05, 1813,95 ;7291            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 0A06, B022,95 ;7292            MOV MEM.DATA TO WR[1]                   ; выборка ожидаемых данных из памяти
U 0A07, 0869,3C ;7293            JSR [CHECK.RESULT]                      ; проверка
U 0A08, 089F,E4 ;7294            JMP [LOOP.T4.5]                         ; зацикливание при ошибке, если разрешено
U 0A09, FF12,15 ;7295            INC LS[T9]                              ; увеличение указателя ответов
U 0A0A, 4319,15 ;7296            XOR LS[T12] TO WR[2]                    ; переключение битов 0-2 в WR2
U 0A0B, 3E15,15 ;7297            MOV WR[2] TO LS[T10]                    ;
U 0A0C, 1910,75 ;7298            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0A0D, B214,15 ;7299            WRITE.MEM LS[T10]                       ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7300    LOOP2.T4.5:
U 0A0E, 1B11,75 ;7301            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; заполнение регистра предвыборки инструкций данными из
                ;7302                                                    ; ячейки памяти 94
U 0A0F, 3615,95 ;7303            MOV LS[T10] TO WR[3]                    ;
U 0A10, BE89,95 ;7304            MOV WR[3] TO LS[ADDRESS.DATA]           ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A11, 2F87,95 ;7305            CLR WR[3]                               ;
U 0A12, 0478,0C ;7306            OPC2/DECODE,DEC.ADRS/7,IFUNC/4,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A13, CA51,95 ;7307            SUB WR[3] FROM LS[#100] TO Q            ;
U 0A14, A180,15 ;7308            MOV Q TO WR[0]                          ; выборка полученных данных
U 0A15, 1813,95 ;7309            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 0A16, B022,95 ;7310            MOV MEM.DATA TO WR[1]                   ; выборка ожидаемых данных
U 0A17, 0869,3C ;7311            JSR [CHECK.RESULT]                      ; проверка
U 0A18, 08A0,E4 ;7312            JMP [LOOP2.T4.5]                        ; зацикливание при ошибке, если разрешено
U 0A19, FF12,15 ;7313            INC LS[T9]
```

```
U 0A1A, C047,15 ;7294            ADD LS[#8] TO WR[2]                    ; добавление 8(H) к коду для пересылки на IB и переключение
U 0A1B, 4319,15 ;7295            XOR LS[T12] TO WR[2]                   ; битов 0-2
U 0A1C, 3E15,15 ;7296            MOV WR[2] TO LS[T10]                   ;
U 0A1D, 1910,75 ;7297            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0A1E, B214,15 ;7298            WRITE.MEM LS[T10]                      ; запись нового кода в ячейку памяти 94
U 0A1F, 5816,15 ;7299            MOV LS[T11] TO Q                       ;
                ;7300              CMP WR[2] WITH Q,                    ;
U 0A20, A484,35 ;7301            DT(LONG)&SET.ALU.CC                    ; код = 107(H)?
U 0A21, 089F,E1 ;7302            JMP.IF[NEQ] TO [LOOP.T4.5]             ; зацикливание, если нет
                ;7303    T4.6:
U 0A22, FF82,15 ;7304            INC LS[ERROR.NUMBER]                   ; номер ошибки = 6
U 0A23, 3647,15 ;7305            MOV LS[#8] TO WR[2]                    ;
U 0A24, 2105,15 ;7306            DEC WR[2]                              ; восстановление кода для пересылки на IB до 7
U 0A25, 3E15,15 ;7307            MOV WR[2] TO LS[T10]                   ;
U 0A26, 1910,75 ;7308            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0A27, B214,15 ;7309            WRITE.MEM LS[T10]                      ; восстановление втого кода в памяти
                ;7310    LOOP.T4.6:
U 0A28, 1811,75 ;7311            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение регистра предвыборки инструкций данными из
                ;7312                                                   ; ячейки памяти 94
U 0A29, 3615,95 ;7313            MOV LS[T10] TO WR[3]                   ;
U 0A2A, BE89,95 ;7314            MOV WR[3] TO LS[ADDRESS.DATA]          ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A2B, 2F87,95 ;7315            CLR WR[3]                              ;
                ;7316      ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7317      ;  ванному следующим образом:
                ;7318      ;
                ;7319      ;      7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7320      ;
U 0A2C, 847A,0C ;7321            OPC2/DECODE,DEC.ADRS/7,IFUNC/5,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A2D, CA51,95 ;7322            SUB WR[3] FROM LS[#100] TO Q           ;
U 0A2E, A180,15 ;7323            MOV Q TO WR[0]                         ; полученные данные = 100(H) (256 десятичн.)-WR3
U 0A2F, 1813,95 ;7324            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]      ;
U 0A30, B022,95 ;7325            MOV MEM.DATA TO WR[1]                  ; выборка ожидаемых данных из памяти
U 0A31, 0869,3C ;7326            JSR [CHECK.RESULT]                     ; проверка
U 0A32, 89A2,84 ;7327            JMP [LOOP.T4.6]                        ; зацикливание при ошибке, если разрешено
U 0A33, FF12,15 ;7328            INC LS[T9]                             ; увеличение указателя ответов
U 0A34, 4319,15 ;7329            XOR LS[T12] TO WR[2]                   ; переключение битов 0-2 в WR2
U 0A35, 3E15,15 ;7330            MOV WR[2] TO LS[T10]                   ;
U 0A36, 1910,75 ;7331            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0A37, B214,15 ;7332            WRITE.MEM LS[T10]                      ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7333    LOOP2.T4.6:
U 0A38, 1811,75 ;7334            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение регистра предвыборки инструкций данными из
                ;7335                                                   ; ячейки памяти 94
U 0A39, 3615,95 ;7336            MOV LS[T10] TO WR[3]                   ;
U 0A3A, BE89,95 ;7337            MOV WR[3] TO LS[ADDRESS.DATA]          ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A3B, 2F87,95 ;7338            CLR WR[3]                              ;
U 0A3C, 847A,0C ;7339            OPC2/DECODE,DEC.ADRS/7,IFUNC/5,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A3D, CA51,95 ;7340            SUB WR[3] FROM LS[#100] TO Q           ;
U 0A3E, A180,15 ;7341            MOV Q TO WR[0]                         ; выборка полученных данных
U 0A3F, 1813,95 ;7342            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]      ;
U 0A40, B022,95 ;7343            MOV MEM.DATA TO WR[1]                  ; выборка ожидаемых данных
U 0A41, 0869,3C ;7344            JSR [CHECK.RESULT]                     ; проверка
U 0A42, 08A3,84 ;7345            JMP [LOOP2.T4.6]                       ; зацикливание при ошибке, если разрешено
U 0A43, FF12,15 ;7346            INC LS[T9]                             ;
U 0A44, C047,15 ;7347            ADD LS[#8] TO WR[2]                    ; добавление 8(H) к коду для пересылки на IB и переключение
U 0A45, 4319,15 ;7348            XOR LS[T12] TO WR[2]                   ; битов 0-2
```

```
U 0A46, 3E15,15 ;7349            MOV WR[2] TO LS[T10]                          ;
U 0A47, 1910,75 ;7350            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0A48, B214,15 ;7351            WRITE.MEM LS[T10]                             ; запись нового кода в ячейку памяти 94
U 0A49, 5816,15 ;7352            MOV LS[T11] TO Q                              ;
                ;7353              CMP WR[2] WITH Q,                           ;
U 0A4A, A484,35 ;7354            DT(LONG)&SET.ALU.CC                           ; код = 107(H)?
U 0A4B, 88A2,81 ;7355            JMP.IF[NEQ] TO [LOOP.T4.6]                    ; зацикливание, если нет
                ;7356    T4.7:
U 0A4C, FF82,15 ;7357            INC LS[ERROR.NUMBER]                          ; номер ошибки = 7
U 0A4D, 3647,15 ;7358            MOV LS[#8] TO WR[2]                           ;
U 0A4E, 2105,15 ;7359            DEC WR[2]                                     ; восстановление кода для пересылки на IB до 7
U 0A4F, 3E15,15 ;7360            MOV WR[2] TO LS[T10]                          ;
U 0A50, 1910,75 ;7361            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0A51, B214,15 ;7362            WRITE.MEM LS[T10]                             ; восстановление этого кода в памяти
                ;7363    LOOP.T4.7:
U 0A52, 1B11,75 ;7364            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций данными из
                ;7365                                                          ; ячейки памяти 94
U 0A53, 3615,95 ;7366            MOV LS[T10] TO WR[3]                          ;
U 0A54, BE89,95 ;7367            MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A55, 2F87,95 ;7368            CLR WR[3]                                     ;
                ;7369     ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7370     ;  ванному следующим образом:
                ;7371     ;
                ;7372     ;     7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7373     ;
U 0A56, 847C,0C ;7374            OPC2/DECODE,DEC.ADRS/7,IFUNC/6,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A57, CA51,95 ;7375            SUB WR[3] FROM LS[#100] TO Q                  ;
U 0A58, A180,15 ;7376            MOV Q TO WR[0]                                ; полученные данные = 100(H) (256 десятичн.)-WR3
U 0A59, 1813,95 ;7377            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]             ;
U 0A5A, B022,95 ;7378            MOV MEM.DATA TO WR[1]                         ; выборка ожидаемых данных из памяти
U 0A5B, 0869,3C ;7379            JSR [CHECK.RESULT]                            ; проверка
U 0A5C, 08A5,24 ;7380            JMP [LOOP.T4.7]                               ; зацикливание при ошибке, если разрешено
U 0A5D, FF12,15 ;7381            INC LS[T9]                                    ; увеличение указателя ответов
U 0A5E, 4319,15 ;7382            XOR LS[T12] TO WR[2]                          ; переключение битов 0-2 в WR3
U 0A5F, 3E15,15 ;7383            MOV WR[2] TO LS[T10]                          ;
U 0A60, 1910,75 ;7384            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
U 0A61, B214,15 ;7385            WRITE.MEM LS[T10]                             ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7386    LOOP2.T4.7:
U 0A62, 1B11,75 ;7387            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]            ; заполнение регистра предвыборки инструкций данными из
                ;7388                                                          ; ячейки памяти 94
U 0A63, 3615,95 ;7389            MOV LS[T10] TO WR[3]                          ;
U 0A64, BE89,95 ;7390            MOV WR[3] TO LS[ADDRESS.DATA]                 ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A65, 2F87,95 ;7391            CLR WR[3]                                     ;
U 0A66, 847C,0C ;7392            OPC2/DECODE,DEC.ADRS/7,IFUNC/6,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0A67, CA51,95 ;7393            SUB WR[3] FROM LS[#100] TO Q                  ;
U 0A68, A180,15 ;7394            MOV Q TO WR[0]                                ; выборка полученных данных
U 0A69, 1813,95 ;7395            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]             ;
U 0A6A, B022,95 ;7396            MOV MEM.DATA TO WR[1]                         ; выборка ожидаемых данных
U 0A6B, 0869,3C ;7397            JSR [CHECK.RESULT]                            ; проверка
U 0A6C, 08A6,24 ;7398            JMP [LOOP2.T4.7]                              ; зацикливание при ошибке, если разрешено
U 0A6D, FF12,15 ;7399            INC LS[T9]                                    ;
U 0A6E, C047,15 ;7400            ADD LS[#8] TO WR[2]                           ; добавление 8(H) к коду для пересылки на IB и переключение
U 0A6F, 4319,15 ;7401            XOR LS[T12] TO WR[2]                          ; битов 0-2
U 0A70, 3E15,15 ;7402            MOV WR[2] TO LS[T10]                          ;
U 0A71, 1910,75 ;7403            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]            ;
```

```
U 0A72, B214,15 ;7404            WRITE.MEM LS[T10]                        ; запись нового кода в ячейку памяти 94
U 0A73, 5816,15 ;7405            MOV LS[T11] TO Q                         ;
                ;7406              CMP WR[2] WITH Q,                      ;
U 0A74, A484,35 ;7407            DT(LONG)&SET.ALU.CC                      ; код = 107(H)?
U 0A75, 08A5,21 ;7408            JMP.IF[NEQ] TO [LOOP.T4.7]               ; зацикливание, если нет
                ;7409    T4.8:
U 0A76, FF82,15 ;7410            INC LS[ERROR.NUMBER]                     ; номер ошибки = 8
U 0A77, 3647,15 ;7411            MOV LS[#8] TO WR[2]                      ;
U 0A78, 2105,15 ;7412            DEC WR[2]                                ; восстановление кода для пересылки на IB до 7
U 0A79, 3E15,15 ;7413            MOV WR[2] TO LS[T10]                     ;
U 0A7A, 1910,75 ;7414            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0A7B, B214,15 ;7415            WRITE.MEM LS[T10]                        ; восстановление этого кода в памяти
                ;7416    LOOP.T4.8:
U 0A7C, 1B11,75 ;7417            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; заполнение регистра предвыборки инструкций данными из
                ;7418                                                     ; ячейки памяти 94
U 0A7D, 3615,95 ;7419            MOV LS[T10] TO WR[3]                     ;
U 0A7E, BE89,95 ;7420            MOV WR[3] TO LS[ADDRESS.DATA]            ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0A7F, 2F87,95 ;7421            CLR WR[3]                                ;
                ;7422     ;  Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7423     ;  ванному следующим образом:
                ;7424     ;
                ;7425     ;     7 в DEC.ADRS+ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ (IFUNC+IB<7:3>+"НЕ-И"(IB<2:0>))
                ;7426     ;
U 0AB0, 047E,0C ;7427            OPC2/DECODE,DEC.ADRS/7,IFUNC/7,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0AB1, CA51,95 ;7428            SUB WR[3] FROM LS[#100] TO Q             ;
U 0AB2, A180,15 ;7429            MOV Q TO WR[0]                           ; полученные данные = 100(н) (256 десятичн.)-WR3
U 0AB3, 1813,95 ;7430            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0AB4, B022,95 ;7431            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных из памяти
U 0AB5, 0869,3C ;7432            JSR [CHECK.RESULT]                       ; проверка
U 0AB6, 08A7,C4 ;7433            JMP [LOOP.T4.8]                          ; зацикливание при ошибке, если разрешено
U 0AB7, FF12,15 ;7434            INC LS[T9]                               ; увеличение указателя ответов
U 0AB8, 4319,15 ;7435            XOR LS[T12] TO WR[2]                     ; переключение битов 0-2 в WR2
U 0AB9, 3E15,15 ;7436            MOV WR[2] TO LS[T10]                     ;
U 0ABA, 1910,75 ;7437            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0ABB, B214,15 ;7438            WRITE.MEM LS[T10]                        ; запись нового кода для пересылки на IB в ячейку памяти 94
                ;7439    LOOP2.T4.8:
U 0ABC, 1B11,75 ;7440            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; заполнение регистра предвыборки инструкций данными из
                ;7441                                                     ; ячейки памяти 94
U 0ABD, 3615,95 ;7442            MOV LS[T10] TO WR[3]                     ;
U 0ABE, BE89,95 ;7443            MOV WR[3] TO LS[ADDRESS.DATA]            ; установка поля "ДРУГИЕ ДАННЫЕ"
U 0ABF, 2F87,95 ;7444            CLR WR[3]                                ;
U 0AC0, 047E,0C ;7445            OPC2/DECODE,DEC.ADRS/7,IFUNC/7,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0AC1, CA51,95 ;7446            SUB WR[3] FROM LS[#100] TO Q             ;
U 0AC2, A180,15 ;7447            MOV Q TO WR[0]                           ; выборка полученных данных
U 0AC3, 1813,95 ;7448            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0AC4, B022,95 ;7449            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных
U 0AC5, 0869,3C ;7450            JSR [CHECK.RESULT]                       ; проверка
U 0AC6, 08A8,C4 ;7451            JMP [LOOP2.T4.8]                         ; зацикливание при ошибке, если разрешено
U 0AC7, FF12,15 ;7452            INC LS[T9]                               ;
U 0AC8, C047,15 ;7453            ADD LS[#8] TO WR[2]                      ; добавление 8(H) к коду для пересылки на IB и переключение
U 0AC9, 4319,15 ;7454            XOR LS[T12] TO WR[2]                     ; битов 0-2
U 0ACA, 3E15,15 ;7455            MOV WR[2] TO LS[T10]                     ;
U 0ACB, 1910,75 ;7456            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0ACC, B214,15 ;7457            WRITE.MEM LS[T10]                        ; запись нового кода в ячейку памяти 94
U 0ACD, 5816,15 ;7458            MOV LS[T11] TO Q                         ;
```

```
                ;7459                 CMP WR[2] WITH Q,                          ;
U 0A9E, A4B4,35 ;7460                 DT(LONG)&SET.ALU.CC                        ; код = 107(H)?
U 0A9F, 08A7,C1 ;7461                 JMP.IF[NEQ] TO [LOOP.T4.8]                 ; зацикливание, если нет
U 0AA0, 0881,AC ;7462                 JSR [DEASSERT.OTHER]                       ;
U 0AA1, FF82,15 ;7463                 INC LS[ERROR.NUMBER]                       ; номер ошибки = 9
U 0AA2, 364D,15 ;7464                 MOV LS[#40] TO WR[2]                       ;
U 0AA3, C047,15 ;7465                 ADD LS[#8] TO WR[2]                        ;
U 0AA4, 4043,15 ;7466                 ADD LS[#2] TO WR[2]                        ;
U 0AA5, C041,15 ;7467                 ADD LS[#1] TO WR[2]                        ; WR2 = 4B
                ;7468         LOOP.T4.9:
U 0AA6, 3E15,15 ;7469                 MOV WR[2] TO LS[T10]                       ;
U 0AA7, 1910,75 ;7470                 MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]         ;
U 0AA8, B214,15 ;7471                 WRITE.MEM LS[T10]                          ; ячейка 94 памяти = 4B
U 0AA9, 1B11,75 ;7472                 MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]         ; IB = 4B
U 0AAA, 2F87,95 ;7473                 CLR WR[3]                                  ;
U 0AAB, 8470,0C ;7474                 OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0AAC, CA51,95 ;7475                 SUB WR[3] FROM LS[#100] TO Q               ;
U 0AAD, A180,15 ;7476                 MOV Q TO WR[0]                             ;
U 0AAE, B648,95 ;7477                 MOV LS[#10] TO WR[1]                       ;
U 0AAF, 4042,95 ;7478                 ADD LS[#2] TO WR[1]                        ;
U 0AB0, C040,95 ;7479                 ADD LS[#1] TO WR[1]                        ; WR1 =13
U 0AB1, 0869,3C ;7480                 JSR [CHECK.RESULT]                         ; проверка
U 0AB2, 08AA,6C ;7481                 JSR [LOOP.T4.9]                            ; зацикливание при ошибке, если разрешено
U 0AB3, FF82,15 ;7482                 INC LS[ERROR.NUMBER]                       ; номер ошибки = A
U 0AB4, 4043,15 ;7483                 ADD LS[#2] TO WR[2]                        ; WR2 = 4D
                ;7484         LOOP.T4.10:
U 0AB5, 3E15,15 ;7485                 MOV WR[2] TO LS[T10]                       ;
U 0AB6, 1910,75 ;7486                 MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]         ;
U 0AB7, B214,15 ;7487                 WRITE.MEM LS[T10]                          ; ячейка 94 памяти = 4D
U 0AB8, 1B11,75 ;7488                 MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]         ;
U 0AB9, 2F87,95 ;7489                 CLR WR[3]                                  ;
U 0ABA, 8470,0C ;7490                 OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0ABB, CA51,95 ;7491                 SUB WR[3] FROM LS[#100] TO Q               ;
U 0ABC, A180,15 ;7492                 MOV Q TO WR[0]                             ;
U 0ABD, B648,95 ;7493                 MOV LS[#10] TO WR[1]                       ;
U 0ABE, 4042,95 ;7494                 ADD LS[#2] TO WR[1]                        ;
U 0ABF, C040,95 ;7495                 ADD LS[#1] TO WR[1]                        ; WR1 = 13
U 0AC0, 0869,3C ;7496                 JSR [CHECK.RESULT]                         ; проверка
U 0AC1, 88A8,5C ;7497                 JSR [LOOP.T4.10]                           ; зацикливание при ошибке, если разрешено
U 0AC2, FF82,15 ;7498                 INC LS[ERROR.NUMBER]                       ; номер ошибки = B
U 0AC3, C041,15 ;7499                 ADD LS[#1] TO WR[2]                        ; WR2 = 4E
                ;7500         LOOP.T4.11:
U 0AC4, 3E15,15 ;7501                 MOV WR[2] TO LS[T10]                       ;
U 0AC5, 1910,75 ;7502                 MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]         ;
U 0AC6, B214,15 ;7503                 WRITE.MEM LS[T10]                          ; ячейка 94 памяти = 4E
U 0AC7, 1B11,75 ;7504                 MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]         ; IB = 4E
U 0AC8, 2F87,95 ;7505                 CLR WR[3]                                  ;
U 0AC9, 8470,0C ;7506                 OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/SPEC,JCTL/JSR ;
U 0ACA, CA51,95 ;7507                 SUB WR[3] FROM LS[#100] TO Q               ;
U 0ACB, A180,15 ;7508                 MOV Q TO WR[0]                             ;
U 0ACC, B648,95 ;7509                 MOV LS[#10] TO WR[1]                       ;
U 0ACD, 4042,95 ;7510                 ADD LS[#2] TO WR[1]                        ;
U 0ACE, C040,95 ;7511                 ADD LS[#1] TO WR[1]                        ; WR1 = 13
U 0ACF, 0869,3C ;7512                 JSR [CHECK.RESULT]                         ; проверка
U 0AD0, 88AC,4C ;7513                 JSR [LOOP.T4.11]                           ; зацикливание при ошибке, если разрешено
```

ТЕСТ 4 – тест ПЗУ ДЕШИФРАТОРА СПЕЦИФИКАТОРОВ (модуль DAP)

```
U 0AD1, 8724,15 ;7514          MOV LS[HI.IPL] TO WR[0]                ;
U 0AD2, BFFE,15 ;7515          MOV WR[0] TO LS[PSL.HW]                ; сброс режима совместимости
                ;7516  END.T4:
```

```
;7517      .PAGE "ТЕСТ 5 - тест ПЗУ ДЕШИФРАТОРА КОДОВ ОПЕРАЦИЙ (модуль DAP)"
;7518      ;**
;7519      ;
;7520      ;
;7521      ; ОПИСАНИЕ ТЕСТА:
;7522      ;
;7523      ;   Этот тест проверяет, что адресация и содержимое ПЗУ ДЕШ. КОДОВ ОПЕРАЦИЙ пра-
;7524      ;   вильные. Для ПЗУ ДЕШ. КОДОВ ОПЕРАЦИЙ частью адресных линий являются биты
;7525      ;   IFUNC, и в данном случае только биты CSR 9 и 10 являются входами, что озна-
;7526      ;   чает, что только четыре инструкции DECODE необходимы для выборки всей ПЗУ.
;7527      ;   Выходы ПЗУ являются микроадресом, поступающим на модуль WCS. Для определения,
;7528      ;   что инструкция DECODE выполняет переход на правильный микроадрес, содержится
;7529      ;   блок из 256 инструкции INC WR[3], каждая из которых соответствует одному из
;7530      ;   возможных 256 микроадресов, к которым может быть выполнен переход. В конце
;7531          блока содержится инструкция RETURN. WR[3] должен содержать 256 минус номер
;7532      ;   адреса перехода. Для определения, что переход выполнен по правильному адресу,
;7533      ;   WR[3] вычитается из 256 и загружается в WR[0]. Формируется ожидаемый адрес,
;7534      ;   загружается в WR[1] и вызывается программа проверки результата (CHECK RESULT).
;7535          Если имеется ошибка и разрешено зацикливание, тогда WR[2] не увеличивается,
;7536      ;   что обеспечивает выборку той же ячейки ПЗУ и формирование того же адреса
;7537      ;   перехода.
;7538      ;
;7539      ; ПРЕДПОЛОЖЕНИЯ:
;7540      ;
;7541      ;   Предполагается, что данные для проверки каждой ячейки ПЗУ имеются в памяти
;7542      ;   и их можно поместить на шину IB.
;7543      ;
;7544      ; ШАГИ ТЕСТА:
;7545      ;
;7546      ;   1)Начальная установка WR, ячеек местной памяти и ячеек оперативной памяти.
;7547      ;   2)Пересылка данных для IB в регистр предвыборки инструкций PFR.
;7548      ;   3)Выполнение инструкции DECODE С IFUNC=0 для перехода в блок из 256 инструк-
;7549      ;     ций INC WR[3].
;7550      ;   4)Вычитание WR[3] из 100(H) и запись в WR0.
;7551      ;   5)Выборка ожидаемых данных из памяти и вызов программы проверки результата.
;7552      ;   6)Увеличение данных для пересылки на IB.
;7553      ;   7)Повторение шагов 2-6 до получения данных на IB = 100(H).
;7554      ;   8)Восстановление начального значение данных на IB.
;7555      ;   9)Повторение шагов 2-8 для IFUNC=1,2 и 3.
;7556      ;
;7557      ; ОШИБКИ:
;7558      ;
;7559      ; ошибка 1 - схемы адресации или содержимое ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ неправильные.
;7560      ;
;7561      ; НАЛАДКА:
;7562      ;
;7563      ;  ОШИБКА 1: Появление ошибки возможно из-за трех неисправностей: адресации ПЗУ,
;7564      ;            содержимого ПЗУ или схемы управления ПЗУ. Вероятно, во всех случаях
;7565      ;            переход в результате инструкции DECODE выполнится в какое-то место
;7566      ;            блока инструкций INC WR[3]. Если имеется неисправность в схемах ад-
;7567      ;            ресации, переход выполнится к одному и тому же микроадресу, в зависи-
;7568      ;            мости от того, которые биты "ЗАСТРЕВАЮТ" в каком состоянии. Если не-
;7569      ;            правильно работают схемы управления, переход возможен в любое место
;7570      ;            в блоке или вообще в любое место, но схемы контроля будут проверяться
;7571      ;            позднее. Неправильное содержимое ПЗУ может быть как в одном, так и в
```

```
                ;7572     ;           нескольких местах. Входы ПЗУ можно определить вычитанием WR[2] из 512.
                ;7573     ;
                ;7574     ;
                ;7575     T.5:
U 0AD3, B65E,15 ;7576             MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;7577                                                      ; состояния
U 0AD4, 3E80,15 ;7578             MOV WR[0] TO LS[CONTROL.STATUS]          ; установка бита 15 в слове управления и состояния.Бит
                ;7579                                                      ; 15 указывает начало теста для конс.процессора
U 0AD5, 10E0,15 ;7580             MISC [SET.CP.ATTN]                       ; выдача сигнала CPU ATTN для конс.процессора
                ;7581     WAIT.T5.0:
U 0AD6, 0BAD,64 ;7582             JMP [WAIT.T5.0]                          ; зацикливание для ожидания ответа конс.процессора
U 0AD7, B724,15 ;7583             MOV LS[HI.IPL] TO WR[0]                  ;
U 0AD8, BFFE,15 ;7584             MOV WR[0] TO LS[PSL.HW]                  ; сброс режима совместимости
U 0AD9, DF26,15 ;7585             MCOM LS[#FF] TO WR[0]                    ;
U 0ADA, 3E8A,15 ;7586             MOV WR[0] TO LS[ERROR.MASK]              ; маска ошибки=FFFFFF00
U 0ADB, 65A0,15 ;7587             CLR LS[PREVIOUS.ERROR]                   ; сброс номера предыдущей ошибки
U 0ADC, B640,15 ;7588             MOV LS[#1] TO WR[0]                      ; установка 1 в WR0
U 0ADD, BE82,15 ;7589             MOV WR[0] TO LS[ERROR.NUMBER]            ; установка номера ошибки для первой ошибки
U 0ADE, A3C0,15 ;7590             ROL WR[0]                                ; установка 2 в WR0
U 0ADF, 3E8C,15 ;7591             MOV WR[0] TO LS[MODULE.NUM]              ; установка кода модуля для модуля DAP
U 0AE0, 3660,15 ;7592             MOV LS[INTERUPT.EN] TO WR[0]             ;
U 0AE1, 3E80,15 ;7593             MOV WR[0] TO LS[CONTROL.STATUS]          ; обеспечение отсутствия всех прерываний
U 0AE2, 10E0,15 ;7594             MISC [SET.CP.ATTN]                       ; выдача сигнала CPU ATTN для конс.процессора
                ;7595     WAIT.T5.1:
U 0AE3, 0BAE,34 ;7596             JMP [WAIT.T5.1]                          ; зацикливание для ожидания ответа конс.процессора
                ;7597     ;
                ;7598     ; T8 = 94(указатель ячейки памяти 94)
                ;7599     ; T9 = указатель ответов (начальное значение 298(H))
                ;7600     ; T10 = память общего назначения
                ;7601     ; WR2 = данные для передачи на IB
                ;7602     ; T11 = память ответов для первой половины
                ;7603     ; T12 = счетчик ответов для первой половины
                ;7604     ;
U 0AE4, 364E,15 ;7605             MOV LS[#80] TO WR[0]                     ;
U 0AE5, C048,15 ;7606             ADD LS[#10] TO WR[0]                     ;
U 0AE6, C044,15 ;7607             ADD LS[#4] TO WR[0]                      ;
U 0AE7, 3E10,15 ;7608             MOV WR[0] TO LS[T8]                      ; T8=94(H)
U 0AE8, 4052,15 ;7609             ADD LS[#200] TO WR[0]                    ;
U 0AE9, C044,15 ;7610             ADD LS[#4] TO WR[0]                      ;
U 0AEA, BE12,15 ;7611             MOV WR[0] TO LS[T9]                      ; T9=298(H)
U 0AEB, 2F85,15 ;7612             CLR WR[2]                                ; начальный данные для IB=0
U 0AEC, 3E15,15 ;7613             MOV WR[2] TO LS[T10]                     ;
U 0AED, 1910,75 ;7614             MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0AEE, B214,15 ;7615             WRITE.MEM LS[T10]                        ; запись начальных данных в ячейку 94
U 0AEF, E520,15 ;7616             CLR LS[PC]                               ;
U 0AF0, B724,15 ;7617             MOV LS[HI.IPL] TO WR[0]                  ;
U 0AF1, 477E,15 ;7618             BIS LS[BIT31] TO WR[0]                   ;
U 0AF2, BFFE,15 ;7619             MOV WR[0] TO LS[PSL.HW]                  ; установка режима совместимости, чтобы данные поступали
                ;7620                                                      ; из регистра предвыборки инструкций и установка значения iF
                ;7621     ;
                ;7622     ; Так как в первой половине ПЗУ содержимое повторяется, то ожидаемые данные мо-
                ;7623     ; гут быть получены методом счета ответа. Здесь устанавливаются начальные значе-
                ;7624     ; ния ответа и счетчика.
U 0AF3, 1813,95 ;7625             MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0AF4, B818,15 ;7626             MOV MEM.DATA TO LS[T12]                   ; T12=начальное значение счетчика
```

```
U 0AF5, FF12,15 ;7627            INC LS[T9]                                  ; увеличение указателя памяти
U 0AF6, 1813,95 ;7628            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]           ;
U 0AF7, 3816,15 ;7629            MOV MEM.DATA TO LS[T11]                     ; T11=первый ответ
                ;7630    LOOP.T5.1:
U 0AF8, 1B11,75 ;7631            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]          ; заполнение регистра предвыборки инструкций данными из
                ;7632                                                        ; ячейки памяти 94
U 0AF9, 2F87,95 ;7633            CLR WR[3]                                   ;
                ;7634     ; Следующая микроинструкция выполняет переход с возвратом по адресу, сформирова-
                ;7635     ; нему следующим образом:
                ;7636     ;   7 в DEC.ADRS+ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ (два младших бита IFUNC+IB)
                ;7637     ;
U 0AFA, 0470,1C ;7638            OPC2/DECODE,DEC.ADRS/7,IFUNC/0,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0AFB, CA51,95 ;7639            SUB WR[3] FROM LS[#100] TO Q                ;
U 0AFC, A180,15 ;7640            MOV Q TO WR[0]                              ; полученные данные=100(H)-WR3
U 0AFD, 3616,95 ;7641            MOV LS[T11] TO WR[1]                        ; выборка ожидаемых данных
U 0AFE, 0869,3C ;7642            JSR [CHECK.RESULT]                          ; проверка
U 0AFF, 08AF,B4 ;7643            JMP [LOOP.T5.1]                             ; зацикливание при ошибке, если разрешено
U 0B00, FC18,15 ;7644            DEC LS[T12]                                 ; уменьшение счетчика
U 0B01, 3618,15 ;7645            MOV LS[T12] TO WR[0]                        ;
                ;7646              TST WR[0],                                ;
U 0B02, 2000,35 ;7647            DT(LONG)&SET.ALU.CC                         ; уменьшено до нуля?
U 0B03, 08B0,A1 ;7648            JMP.IF[NEQ] TO [T5.1.SAME]                  ; переход, если нет
U 0B04, FF12,15 ;7649            INC LS[T9]                                  ;
U 0B05, 1813,95 ;7650            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]           ;
U 0B06, B818,15 ;7651            MOV MEM.DATA TO LS[T12]                     ; выборка следующего значения счетчика
U 0B07, FF12,15 ;7652            INC LS[T9]                                  ;
U 0B08, 1813,95 ;7653            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]           ;
U 0B09, 3816,15 ;7654            MOV MEM.DATA TO LS[T11]                     ; выборка следующего ответа
                ;7655    T5.1.SAME:
U 0B0A, 2045,15 ;7656            INC WR[2]                                   ; увеличение данных для IB
U 0B0B, 3E15,15 ;7657            MOV WR[2] TO LS[T10]                        ;
U 0B0C, 1910,75 ;7658            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]          ;
U 0B0D, B214,15 ;7659            WRITE.MEM LS[T10]                           ; запись этих данных в ячейку памяти 94
U 0B0E, D850,15 ;7660            MOV LS[#100] TO Q                           ;
                ;7661              CMP WR[2] WITH Q,                         ;
U 0B0F, A484,35 ;7662            DT(LONG)&SET.ALU.CC                         ; данные для IB=100(H)?
U 0B10, 08AF,81 ;7663            JMP.IF[NEQ] TO [LOOP.T5.1]                  ; возврат, если нет
                ;7664    T5.2:
U 0B11, 2F85,15 ;7665            CLR WR[2]                                   ; начальное значение данных для IB=0
U 0B12, 3E15,15 ;7666            MOV WR[2] TO LS[T10]                        ;
U 0B13, 1910,75 ;7667            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]          ;
U 0B14, B214,15 ;7668            WRITE.MEM LS[T10]                           ; запись начального значения данных в ячейку 94
                ;7669    LOOP.T5.2:
U 0B15, 1B11,75 ;7670            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]          ; заполнение регистра предвыборки инструкций данными
                ;7671                                                        ; из ячейки памяти 94
U 0B16, 2F87,95 ;7672            CLR WR[3]                                   ;
                ;7673     ; Следующая микроинструкция выполняет переход с возвратом по адресу, сформиро-
                ;7674     ; ванному следующим образом:
                ;7675     ;   7 в DEC.ADRS+ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ (два младших бита IFUNC+IB)
                ;7676     ;
U 0B17, 8472,1C ;7677            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B18, CA51,95 ;7678            SUB WR[3] FROM LS[#100] TO Q                ;
U 0B19, A180,15 ;7679            MOV Q TO WR[0]                              ; полученные данные=100(H)-WR3
U 0B1A, 3616,95 ;7680            MOV LS[T11] TO WR[1]                        ;
U 0B1B, 0869,3C ;7681            JSR [CHECK.RESULT]                          ; проверка
```

```
U 0B1C, 88B1,54 ;7682            JMP [LOOP.T5.2]                          ; зацикливание при ошибке, если разрешено
U 0B1D, FC18,15 ;7683            DEC LS[T12]                              ; уменьшение счетчика
U 0B1E, 3618,15 ;7684            MOV LS[T12] TO WR[0]                     ;
                ;7685              TST WR[0],                             ;
U 0B1F, 2000,35 ;7686            DT(LONG)&SET.ALU.CC                      ; уменьшено до нуля?
U 0B20, 08B2,71 ;7687            JMP.IF[NEQ] TO [T5.2.SAME]               ; переход, если нет
U 0B21, FF12,15 ;7688            INC LS[T9]                               ;
U 0B22, 1813,95 ;7689            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0B23, B818,15 ;7690            MOV MEM.DATA TO LS[T12]                  ; выборка следующего значения счетчика
U 0B24, FF12,15 ;7691            INC LS[T9]                               ;
U 0B25, 1813,95 ;7692            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0B26, 3816,15 ;7693            MOV MEM.DATA TO LS[T11]                  ; выборка следующего ответа
                ;7694    T5.2.SAME:
U 0B27, 2045,15 ;7695            INC WR[2]                                ; увеличение данных IB
U 0B28, 3E15,15 ;7696            MOV WR[2] TO LS[T10]                     ;
U 0B29, 1910,75 ;7697            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0B2A, B214,15 ;7698            WRITE.MEM LS[T10]                        ; запись в ячейку памяти 94
U 0B2B, D850,15 ;7699            MOV LS[#100] TO Q                        ;
                ;7700              CMP WR[2] WITH Q,                      ;
U 0B2C, A484,35 ;7701            DT(LONG)&SET.ALU.CC                      ; данные для IB=100(H)?
U 0B2D, 88B1,51 ;7702            JMP.IF[NEQ] TO [LOOP.T5.2]               ; возврат, если нет
                ;7703    T5.3:
U 0B2E, 2F85,15 ;7704            CLR WR[2]                                ; начальное значение данных IB=0
U 0B2F, 3E15,15 ;7705            MOV WR[2] TO LS[T10]                     ;
U 0B30, 1910,75 ;7706            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0B31, B214,15 ;7707            WRITE.MEM LS[T10]                        ; запись начального значения данных в ячейку 94
U 0B32, B72E,15 ;7708            MOV LS[#00F0] TO WR[0]                   ;
U 0B33, 4052,15 ;7709            ADD LS[#200] TO WR[0]                    ;
U 0B34, 4046,15 ;7710            ADD LS[#8] TO WR[0]                      ;
U 0B35, C042,15 ;7711            ADD LS[#2] TO WR[0]                      ;
U 0B36, BE12,15 ;7712            MOV WR[0] TO LS[T9]                      ; T9=2FA
                ;7713    LOOP.T5.3:
U 0B37, 1B11,75 ;7714            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; заполнение регистра предвыборки инструкций данными из
                ;7715                                                     ; ячейки памяти 94
U 0B38, 2F87,95 ;7716            CLR WR[3]                                ;
                ;7717     ; Следующая микроинструкция выполняет переход с возвратом по адресу, сформирова-
                ;7718     ; ному следующим образом:
                ;7719     ;    7 в DEC.ADRS+ПЗУ ДЕШ.КОДОВ ОПЕРАУИЙ (два младших бита IFUNC+IB)
                ;7720     ;
U 0B39, 8474,1C ;7721            OPC2/DECODE,DEC.ADRS/7,IFUNC/2,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B3A, CA51,95 ;7722            SUB WR[3] FROM LS[#100] TO Q             ;
U 0B3B, A180,15 ;7723            MOV Q TO WR[0]                           ; полученные данные=100(H)-WR3
U 0B3C, 1813,95 ;7724            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]        ;
U 0B3D, B022,95 ;7725            MOV MEM.DATA TO WR[1]                    ; выборка ожидаемых данных из памяти
U 0B3E, 0869,3C ;7726            JSR [CHECK.RESULT]                       ; проверка
U 0B3F, 88B3,74 ;7727            JMP [LOOP.T5.3]                          ; зацикливание при ошибке, если разрешено
U 0B40, FF12,15 ;7728            INC LS[T9]                               ; увеличение указателя ответа
U 0B41, 2045,15 ;7729            INC WR[2]                                ; увеличение данных IB
U 0B42, 3E15,15 ;7730            MOV WR[2] TO LS[T10]                     ;
U 0B43, 1910,75 ;7731            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0B44, B214,15 ;7732            WRITE.MEM LS[T10]                        ; запись этих данных в ячейку памяти 94
U 0B45, D850,15 ;7733            MOV LS[#100] TO Q                        ;
                ;7734              CMP WR[2] WITH Q,                      ;
U 0B46, A484,35 ;7735            DT(LONG)&SET.ALU.CC                      ; данные IB=100(H)?
U 0B47, 88B3,71 ;7736            JMP.IF[NEQ] TO [LOOP.T5.3]               ; возврат, если нет
```

```
                ;7737    T5.4:
U 0B4B, 2F85,15 ;7738            CLR WR[2]                              ; начальное значение данных на IB=0
U 0B49, 3E15,15 ;7739            MOV WR[2] TO LS[T10]                   ;
U 0B4A, 1910,75 ;7740            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0B4B, B214,15 ;7741            WRITE.MEM LS[T10]                      ; запись начального значения данных в ячейку 94
                ;7742    LOOP.T5.4:
U 0B4C, 1B11,75 ;7743            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение регистра предвыборки инструкции данными из
                ;7744                                                   ; ячейки памяти 94
U 0B4D, 2F87,95 ;7745            CLR WR[3]                              ;
                ;7746     ; Следующая микроинструкция выполняет переход с возвратом по адресу, сформирова-
                ;7747     ; ному следующим адресом:
                ;7748     ;    7 в DEC.ADRS+ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ (два младших бита IFUNC+IB)
                ;7749     ;
U 0B4E, 0476,1C ;7750            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B4F, CA51,95 ;7751            SUB WR[3] FROM LS[#100] TO Q           ;
U 0B50, A180,15 ;7752            MOV Q TO WR[0]                         ; полученные данные=100(H)-WR3
U 0B51, 1813,95 ;7753            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]      ;
U 0B52, B022,95 ;7754            MOV MEM.DATA TO WR[1]                  ; выборка ожидаемых данных из памяти
U 0B53, 0869,3C ;7755            JSR [CHECK.RESULT]                     ; проверка
U 0B54, B8B4,C4 ;7756            JMP [LOOP.T5.4]                        ; зацикливание при ошибке, если разрешено
U 0B55, FF12,15 ;7757            INC LS[T9]                             ; увеличение указателя ответа
U 0B56, 2045,15 ;7758            INC WR[2]                              ; увеличение данных IB
U 0B57, 3E15,15 ;7759            MOV WR[2] TO LS[T10]                   ;
U 0B58, 1910,75 ;7760            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0B59, B214,15 ;7761            WRITE.MEM LS[T10]                      ; запись этих данных в ячейку памяти 94
U 0B5A, D850,15 ;7762            MOV LS[#100] TO Q                      ;
                ;7763              CMP WR[2] WITH Q,                    ;
U 0B5B, A484,35 ;7764            D1(LONG)&SET.ALU.CC                    ; данные IB=100(H)?
U 0B5C, B8B4,C1 ;7765            JMP.IF[NEQ] TO [LOOP.T5.4]             ; возврат, если нет
U 0B5D, B724,15 ;7766            MOV LS[HI.IPL] TO WR[0]                ;
U 0B5E, BFFE,15 ;7767            MOV WR[0] TO LS[PSL.HW]                ; сброс режима совместимости
                ;7768    END.T5:
```

```
;7769      .PAGE "ТЕСТ 6 - тест схем управления дешифрацией инструкций (модуль DAP)"
;7770      ;
;7771      ;
;7772      ; ОПИСАНИЕ ТЕСТА:
;7773      ;
;7774      ;   Этот тест проверяет схемы управления ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ и ПЗУ ДЕШ.КО-
;7775      ;   ДОВ ОПЕРАЦИЙ. Схема управления работает таким образом, что для одной мик-
;7776      ;   росхемы ПЗУ работа разрешена, а для другой нет. Для разрешения одной из
;7777      ;   микросхем ПЗУ должна выполняться инструкция DECODE и поле JCTL должно
;7778      ;   определять переход или переход с возвратом. Кроме выше упомянутых усло-
;7779      ;   вий, имеются другие пять сигналов, которые участвуют в выборке одной
;7780      ;   из двух микросхем ПЗУ. Этими сигналами являются бит OPC/SPEC (CSR04),
;7781      ;   биты 9 и 10 CSR (IFUNC), INTERR REQ и MASK INTS. Для определения, ко-
;7782      ;   торая из микросхем ПЗУ будет выбрана, восемь битов шины IB будут содер-
;7783      ;   жать нули, а биты IFUNC будут содержать единицы, если не задано иначе.
;7784      ;   Инструкция DECODE, которая разрешает ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ или ПЗУ ДЕШ.
;7785      ;   СПЕЦИФИКАТОРОВ, вызывает переход к определенной ячейке в управляющей памяти
;7786      ;   Ячейка, к которой выполнен переход, определяется восемью младшими битами из
;7787      ;   ПЗУ и старшими битами из инструкции DECODE. Так как старшие биты одни и те
;7788      ;   же, имеется только 256 последовательных ячеек, к которым возможен переход,
;7789      ;   и каждая из них содержит INC WR[3]. Если выбрана микросхема ПЗУ ДЕШ.СПЕЦИ-
;7790      ;   ФИКАТОРОВ, тогда инструкция DECODE вызовет переход к инструкции INC WR[3],
;7791      ;   соответствующей выбранной ячейке ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ. Если выбрана микрос-
;7792      ;   хема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ, тогда будет выполнен переход к инструкции INC WR[3],
;7793      ;   соответствующей выбраной ячейке ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ. Так как ячейка пере-
;7794      ;   хода известна (из данных на IB, DEC.ADRS и битов IFUNC), тогда известно, сколь-
;7795      ;   ко раз будет увеличен WR[3], пока достигнет конца блока. Сравнивая значение WR[3]
;7796      ;   с ожидаемым значением, можно определить, правильно ли был выполнен переход. Для
;7797      ;   проверки застревания в состоянии "1" и в состоянии "0" этих пяти сигналов
;7798      ;   требуется 5 условий.
;7799      ;
;7800      ;   Номер условия  CSR4 IFUNC<0> IFUNC<1> INTERR REQ MASK INTS выбранная микросхема ПЗУ
;7801      ;
;7802      ;        1          1      1         1        1         1       ДЕШ.СПЕЦИФИКАТОРОВ
;7803      ;        2          1      1         1        1         0       ДЕШ.КОДОВ ОПЕРАЦИЙ
;7804      ;        3          1      0         1        1         1       ДЕШ.КОДОВ ОПЕРАЦИЙ
;7805      ;        4          1      1         0        1         1       ДЕШ.КОДОВ ОПЕРАЦИЙ
;7806      ;        5          1      1         1        0         1       ДЕШ.КОДОВ ОПЕРАЦИЙ
;7807      ;
;7808      ;   ПРИМЕЧАНИЕ: CSR4 был проверен в предыдущих двух тестах и должен быть уста-
;7809      ;               новлен во время этого теста.
;7810      ;
;7811      ; ПРЕДПОЛОЖЕНИЯ:
;7812      ;
;7813      ;   Предполагается, что память подготовлена так, что во время этого теста шина
;7814      ;   IB загружается нулями. Также предполагается, что тесты ПЗУ ДЕШ.СПЕЦИФИКАТО-
;7815      ;   РОВ и ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ выполнены успешно.
;7816      ;
;7817      ; ШАГИ ТЕСТА:
;7818      ;
;7819      ;   1) Выдача инструкции DECODE с условиями, установленными соответственно
;7820      ;      описанию условия 1.
;7821      ;   2) Вычитание WR3 из 256 (100(H)) и формирование ожидаемых данных.
;7822      ;   3) Вызов программы проверки.
;7823      ;   4) Повторение шагов 1-3 с условиями от 2 до 5.
```

```
                    ;7824     ;
                    ;7825     ;
                    ;7826     ; ОШИБКИ:
                    ;7827     ;
                    ;7828     ;    ошибка 1 - не выбрана микросхема ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ при условиях, уста-
                    ;7829     ;               новленных как описано для условия 1
                    ;7830     ;    ошибка 2 - не выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ при условиях, уста-
                    ;7831     ;               новленных как описано для условия 2
                    ;7832     ;    ошибка 3 - не выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ при условиях, уста-
                    ;7833     ;               новленных как описано для условия 3
                    ;7834     ;    ошибка 4 - не выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ при условиях, уста-
                    ;7835     ;               новленных как описано для условия 4
                    ;7836     ;    ошибка 5 - не выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ при условиях, уста-
                    ;7837     ;               новленных как описано для условия 5
                    ;7838     ;
                    ;7839     ; НАЛАДКА:
                    ;7840     ;
                    ;7841     ;    ОШИБКА 1 - Схема "4НЕ-ИЛИ" не формирует низкого уровня на входе схемы
                    ;7842     ;               "2НЕ-ИЛИ" и разрешает выбор ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ, или схема
                    ;7843     ;               "2НЕ-ИЛИ" не формирует высокого уровня для разрешения выбора ПЗУ
                    ;7844     ;               ДЕШ.СПЕЦИФИКАТОРОВ.
                    ;7845     ;    ОШИБКА 2 - Схема "4НЕ-ИЛИ" не формирует высокого уровня при низком уровне
                    ;7846     ;               сигнала MASK INTS и высоких уровнях на других входах.
                    ;7847     ;    ОШИБКА 3 - Схема "4НЕ-ИЛИ" не формирует высокого уровня при низком уровне
                    ;7848     ;               сигнала CSR 09 и высоких уровнях на других входах.
                    ;7849     ;    ОШИБКА 4 - Схема "4НЕ-ИЛИ" не формирует высокого уровня при низком уровне
                    ;7850     ;               сигнала CSR 10 и высоких уровнях на других входах.
                    ;7851     ;    ОШИБКА 5 - Схема "4НЕ-ИЛИ" не формирует высокого уровня при низком уровне
                    ;7852     ;               сигнала INTERR REQ и высоких уровнях на других входах.
                    ;7853     ;
                    ;7854     ;--
                    ;7855     T.6:
U 0B5F, B65E,15     ;7856          MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                    ;7857                                               ; состояния
U 0B60, 3E80,15     ;7858          MOV WR[0] TO LS[CONTROL.STATUS]       ; установка бита 15 в слове управления и состояния.
U 0B61, 10E0,15     ;7859          MISC [SET.CP.ATTN]                   ; Бит 15 указывает начало теста для консольного
                    ;7860                                               ; процессора
                    ;7861     WAIT.T6.0:
U 0B62, 88B6,24     ;7862          JMP [WAIT.T6.0]                      ; зацикливание для ожидания ответа консольного
                    ;7863                                               ; процессора
U 0B63, DF26,15     ;7864          MCOM LS[#FF] TO WR[0]                ;
U 0B64, 3EBA,15     ;7865          MOV WR[0] TO LS[ERROR.MASK]           ; маска ошибки = FFFFFF00
U 0B65, 65A0,15     ;7866          CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 0B66, B724,15     ;7867          MOV LS[HI.IPL] TO WR[0]              ;
U 0B67, BFFE,15     ;7868          MOV WR[0] TO LS[PSL.HW]               ; установка значения 1F в IPL и сброс режима
                    ;7869                                               ; совместимости
U 0B68, B640,15     ;7870          MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 0B69, BE82,15     ;7871          MOV WR[0] TO LS[ERROR.NUMBER]         ; установка номера ошибки для первой ошибки
U 0B6A, A3C0,15     ;7872          ROL WR[0]                            ; установка 2 в WR0
U 0B6B, 3E8C,15     ;7873          MOV WR[0] TO LS[MODULE.NUM]           ; установка кода модуля для модуля DAP
U 0B6C, B662,15     ;7874          MOV LS[CONSOLE.HALT] TO WR[0]        ;
U 0B6D, 4760,15     ;7875          BIS LS[INTERUPT.EN] TO WR[0]         ;
U 0B6E, 3E80,15     ;7876          MOV.WR[0] TO LS[CONTROL.STATUS]       ; втим формируется высокий уровень сигнала INTERR REQ
U 0B6F, 10E0,15     ;7877          MISC [SET.CP.ATTN]                   ; вызов консольного процессора
                    ;7878     WAIT.T6.1:
```

```
U 0B70, 8887,04 ;7879            JMP [WAIT.T6.1]                                  ; ожидание ответа консольного процессора
U 0B71, 364E,15 ;7880            MOV LS[#80] TO WR[0]                             ;
U 0B72, 4046,15 ;7881            ADD LS[#8] TO WR[0]                              ;
U 0B73, 3E10,15 ;7882            MOV WR[0] TO LS[T8]                              ; T8=88(H) (указатель слова, содержащего нули)
U 0B74, 1B11,75 ;7883            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]               ; заполнение регистра предвыборки инструкций нулями
U 0B75, B724,15 ;7884            MOV LS[HI.IPL] TO WR[0]                          ;
U 0B76, 477E,15 ;7885            BIS LS[BIT31] TO WR[0]                           ;
U 0B77, BFFE,15 ;7886            MOV WR[0] TO LS[PSL.HW]                          ; установка режима совместимости, чтобы данные поступали
                ;7887                                                             ; из регистра предвыборки инструкций и установка значения 1F
                ;7888    LOOP.T6.1:
U 0B78, 2F87,95 ;7889            CLR WR[3]                                        ;
U 0B79, 0476,1C ;7890            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B7A, CA51,95 ;7891            SUB WR[3] FROM LS[#100] TO Q                     ;
U 0B7B, A180,15 ;7892            MOV Q TO WR[0]                                   ; полученные данные = 100(H)-WR3
U 0B7C, B72C,95 ;7893            MOV LS[#000F] TO WR[1]                           ;
U 0B7D, 372F,15 ;7894            MOV LS[#00F0] TO WR[2]                           ;
U 0B7E, 2E04,95 ;7895            ADD WR[2] TO WR[1]                               ; ожидаемые данные=FF (ячейка 01 ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ)
U 0B7F, 0869,3C ;7896            JSR [CHECK.RESULT]                               ; проверка, что была выбрана микросхема ПЗУ
                ;7897                                                             ; ДЕШ.СПЕЦИФИКАТОРОВ
U 0B80, 0B87,84 ;7898            JMP [LOOP.T6.1]                                  ; зацикливание при ошибке, если разрешено
                ;7899    T6.2:
U 0B81, FF82,15 ;7900            INC LS[ERROR.NUMBER]                             ; увеличение номера ошибки до 2
                ;7901    LOOP.T6.2:
U 0B82, 2F87,95 ;7902            CLR WR[3]                                        ;
U 0B83, 10F2,15 ;7903            MISC2 [MASK.INTERRUPTS]                          ; установка низкого уровня сигнала MASK INTS на один
                ;7904                                                             ; цикл
U 0B84, 0476,1C ;7905            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B85, CA51,95 ;7906            SUB WR[3] FROM LS[#100] TO Q                     ;
U 0B86, A180,15 ;7907            MOV Q TO WR[0]                                   ; полученные данные = 100(H)-WR3
U 0B87, 372E,95 ;7908            MOV LS[#00F0] TO WR[1]                           ;
U 0B88, C54C,95 ;7909            BIC LS[BIT6] TO WR[1]                            ;
U 0B89, 4044,95 ;7910            ADD LS[#4] TO WR[1]                              ;
U 0B8A, C046,95 ;7911            ADD LS[#8] TO WR[1]                              ; ожидаемые данные = BC(H) (ячейка 300 ПЗУ ДЕШ.КОДОВ
                ;7912                                                             ; ОПЕРАЦИЙ)
U 0B8B, 0869,3C ;7913            JSR [CHECK.RESULT]                               ; проверка, что была выбрана микросхема ПЗУ ДЕШ.КОДОВ
                ;7914                                                             ; ОПЕРАЦИЙ
U 0B8C, 0B88,24 ;7915            JMP [LOOP.T6.2]                                  ;
                ;7916    T6.3:
U 0B8D, FF82,15 ;7917            INC LS[ERROR.NUMBER]                             ; увеличение номера ошибки до 3
                ;7918    LOOP.T6.3:
U 0B8E, 2F87,95 ;7919            CLR WR[3]                                        ;
U 0B8F, 8474,1C ;7920            OPC2/DECODE,DEC.ADRS/7,IFUNC/2,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR
U 0B90, CA51,95 ;7921            SUB WR[3] FROM LS[#100] TO Q                     ;
U 0B91, A180,15 ;7922            MOV Q TO WR[0]                                   ; полученные данные = 100(H)-WR3
U 0B92, 372E,95 ;7923            MOV LS[#00F0] TO WR[1]                           ;
U 0B93, C54A,95 ;7924            BIC LS[BIT5] TO WR[1]                            ;
U 0B94, C046,95 ;7925            ADD LS[#8] TO WR[1]                              ; ожидаемые данные = D8(H) (ячейка 200 ПЗУ ДЕШ.КОДОВ
                ;7926                                                             ; ОПЕРАЦИЙ)
U 0B95, 0869,3C ;7927            JSR [CHECK.RESULT]                               ; проверка, что была выбрана микросхема ПЗУ ДЕШ.КОДОВ
                ;7928                                                             ; ОПЕРАЦИЙ
U 0B96, 0B88,E4 ;7929            JMP [LOOP.T6.3]                                  ; зацикливание при ошибке, если разрешено
                ;7930    T6.4:
U 0B97, FF82,15 ;7931            INC LS[ERROR.NUMBER]                             ; увеличение номера ошибки до 4
                ;7932    LOOP.T6.4:
U 0B98, 2F87,95 ;7933            CLR WR[3]                                        ;
```

```
U 0B99, B472,1C ;7934            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0B9A, CA51,95 ;7935            SUB WR[3] FROM LS[#100] TO Q                    ;
U 0B9B, A180,15 ;7936            MOV Q TO WR[0]                                  ; полученные данные = 100(H)-WR3
U 0B9C, B72C,95 ;7937            MOV LS[#000F] TO WR[1]                          ;
U 0B9D, 4542,95 ;7938            BIC LS[BIT1] TO WR[1]                           ;
U 0B9E, 404E,95 ;7939            ADD LS[#80] TO WR[1]                            ; ожидаемые данные = 8D(H) (ячейка 100 ПЗУ ДЕШ.КОДОВ
                ;7940                                                            ; ОПЕРАЦИЙ)
U 0B9F, 0869,3C ;7941            JSR [CHECK.RESULT]                              ; проверка, что была выбрана микросхема ПЗУ ДЕШ.КОДОВ
                ;7942                                                            ; ОПЕРАЦИЙ
U 0BA0, B8B9,84 ;7943            JMP [LOOP.T6.4]                                 ; зацикливание при ошибке, если разрешено
                ;7944    T6.5:
U 0BA1, FF82,15 ;7945            INC LS[ERROR.NUMBER]                            ; увеличение номера ошибки до 5
U 0BA2, B724,15 ;7946            MOV LS[HI.IPL] TO WR[0]                         ;
U 0BA3, BFFE,15 ;7947            MOV WR[0] TO LS[PSL.HW]                         ; сброс режима совместимости, так что прерывание может
                ;7948                                                            ; быть переключено на низкий уровень
U 0BA4, 3660,15 ;7949            MOV LS[BIT16] TO WR[0]                          ;
U 0BA5, 3E80,15 ;7950            MOV WR[0] TO LS[CONTROL.STATUS]                 ; очистка прерываний
U 0BA6, 10E0,15 ;7951            MISC [SET.CP.ATTN]                              ; вызов консольного процессора
                ;7952    WAIT.T6.5:
U 0BA7, B8BA,74 ;7953            JMP [WAIT.T6.5]                                 ; ожидание ответа консольного процессора
U 0BA8, B724,15 ;7954            MOV LS[HI.IPL] TO WR[0]                         ;
U 0BA9, 477E,15 ;7955            BIS LS[BIT31] TO WR[0]                          ;
U 0BAA, BFFE,15 ;7956            MOV WR[0] TO LS[PSL.HW]                         ; восстановление режима совместимости
                ;7957    LOOP.T6.5:
U 0BAB, 2FB7,95 ;7958            CLR WR[3]                                       ;
U 0BAC, 0476,1C ;7959            OPC2/DECODE,DEC.ADRS/7,IFUNC/3,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/JSR ;
U 0BAD, CA51,95 ;7960            SUB WR[3] FROM LS[#100] TO Q                    ;
U 0BAE, A180,15 ;7961            MOV Q TO WR[0]                                  ; полученные данные = 100(H)-WR3
U 0BAF, 372E,95 ;7962            MOV LS[#00F0] TO WR[1]                          ;
U 0BB0, C54C,95 ;7963            BIC LS[BIT6] TO WR[1]                           ;
U 0BB1, 4044,95 ;7964            ADD LS[#4] TO WR[1]                             ;
U 0BB2, C046,95 ;7965            ADD LS[#8] TO WR[1]                             ; ожидаемые данные = BC(H) (ячейка 300 ПЗУ ДЕШ.КОДОВ
                ;7966                                                            ; ОПЕРАЦИЙ)
U 0BB3, 0869,3C ;7967            JSR [CHECK.RESULT]                              ; проверка, что выбрана микросхема ПЗУ ДЕШ.КОДОВ
                ;7968                                                            ; ОПЕРАЦИЙ
U 0BB4, B8BA,B4 ;7969            JMP [LOOP.T6.5]                                 ; зацикливание при ошибке, если разрешено
                ;7970    T6.CLEANUP:
U 0BB5, B724,15 ;7971            MOV LS[HI.IPL] TO WR[0]                         ;
U 0BB6, BFFE,15 ;7972            MOV WR[0] TO LS[PSL.HW]                         ; сброс режима совм тимости и восстановление значения
                ;7973                                                            ; 1F в IPL
                ;7974    END.T6:
```

```
;7975      .PAGE "ТЕСТ 7 - тест ПЗУ ТИП ДАННЫХ, КЛАСС КОДОВ УСЛОВИЙ (модуль DAP)"
;7976      ;
;7977      ;
;7978      ;  ОПИСАНИЕ ТЕСТА:
;7979      ;
;7980      ;       Этот тест проверяет, что содержимое и адресация ПЗУ ТИП ДАННЫХ,
;7981      ;       КЛАСС КОДОВ УСЛОВИЙ правильные. Входами ПЗУ ТИП ДАННЫХ, КЛАСС
;7982      ;       КОДОВ УСЛОВИЙ являются восемь линий шины IB и сигнал COMPAT MODE
;7983      ;       (режим совместимости). Линии шины IB могут меняться инструкцией
;7984      ;       DECODE, а сигнал COMPAT MODE устанавливается низким для первой
;7985      ;       половины теста и высоким для другой половины теста. При проверке
;7986      ;       ПЗУ ТИП ДАННЫХ, КЛАСС КОДОВ УСЛОВИЙ необходимо проверить другие
;7987      ;       сигналы и ПМЛ. Выходами ПЗУ являются сигналы DT CLASS 0, DT CLASS 1,
;7988      ;       CC CLASS 0 и CC CLASS 1. Для проверки сигналов CC CLASS 0 и CC CLASS 1
;7989      ;       они поступают на ПМЛ УСЛ.ПЕРЕХОДА, КЛАСС КОДОВ УСЛОВИЙ и на выходе
;7990      ;       формируют сигналы OPC TYPE 1 и OPC TYPE 0. Сигналы OPC TYPE 0 и OPC TYPE 1
;7991      ;       являются входами ПМЛ КОДЫ УСЛОВ. PSL[3:0]. Выходы ПМЛ КОДЫ УСЛОВ.
;7992      ;       PSL [3:0] могут быть переданы на шину D. Другие выходы ПЗУ ТИП ДАННЫХ,
;7993      ;       КЛАСС КОДОВ УСЛОВИЙ поступают на ПМЛ УПР.ТИПОМ ДАННЫХ, выходы которой
;7994      ;       поступают на шину D и регистр размера (SIZE). Когда выходы ПЗУ
;7995      ;       ТИП ДАННЫХ, КЛАСС КОДОВ УСЛОВИЙ устанавливаются путем варьирования
;7996      ;       значений IB и COMPAT MODE, тогда выдается инструкция MOV из LS[7E]
;7997      ;       (со значением 2 в поле CC) для передачи содержимого регистра размера
;7998      ;       на биты 0 и 1 шины D для проверки. Содержимое битов 0 и 1 должно быть
;7999      ;       таким же, как DT CLASS 0 и DT CLASS 1. Программа проверки результата
;8000      ;       сравнивает ожидаемое значение и фактическое значение и зацикливает
;8001      ;       при ошибке, если необходимо. Инструкция MOV из LS[FE] со значением 3
;8002      ;       в поле CC устанавливает коды условий АЛУ. Инструкция DECODE с уста-
;8003      ;       новленными битами 9 и 4 разрешает выдачу IRD STATE L для установки
;8004      ;       битов OPC и посредством этого обеспечивает входы для управления бита-
;8005      ;       ми PSL CC. Тогда биты PSL CC можно проверить путем выдачи инструкций
;8006      ;       MOV со значением 3 в поле CC и инструкции MOV из LS[7F] в какой-то
;8007      ;       WR. Для того, чтобы проверить сигналы OPC 0 и OPC 1 полностью, должны
;8008      ;       проверяться все случаи, где они участвуют в установке битов PSL CC.
;8009      ;       В большинстве случаев это можно сделать установкой битов АЛУ на все
;8010      ;       нули или на все единицы.
;8011      ;
;8012      ;       Каждый раз должен быть корректно установлен режим совместимости,
;8013      ;       так как инструкция MOV меняет его состояние. Инструкции DECODE
;8014      ;       загружают регистр кода операции, что позволяет содержимое регистра
;8015      ;       кода операции передать на шину IB, когда выполняется зацикл ание
;8016      ;       при ошибке.
;8017      ;
;8018      ;  ПРЕДПОЛОЖЕНИЯ:
;8019      ;
;8020      ;       предполагается, что схемы загрузки шины IB и регистра кода операции
;8021      ;       были проверены и работают правильно. также предполагается, что основ-
;8022      ;       ная память содержит коды от 0 до 256, так что ПЗУ ТИП ДАННЫХ, КЛАСС
;8023      ;       КОДОВ УСЛОВИЙ может быть проверена полностью.
;8024      ;
;8025      ;  ШАГИ ТЕСТА:
;8026      ;
;8027      ;       1.Начальная установка ячеек местной памяти, основной памяти и
;8028      ;         рабочих регистров (WR), которые будут использоваться.
;8029      ;       2.Выдача MEM.REQ для заполнения PFR с нарастающими данными.
```

```
                ;8030      ;        3.При низком уровне сигнала COMPAT MODE (адреса ПЗУ 0-FF) загрузка
                ;8031      ;          регистра кода операции данными из регистра предвыборки инструкций.
                ;8032      ;        4.Выдача инструкции DECODE с нечетным полем IFUNC и выборка
                ;8033      ;          ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ (это приводит к установке сигнала IRD STATE).
                ;8034      ;        5.Выборка значений DT CLASS из регистра SIZE, выборка ожидаемых
                ;8035      ;          данных и проверка.
                ;8036      ;        6.Установка кодов условий АЛУ в единицы.
                ;8037      ;        7.Пересылка кодов условий в PSL CC.
                ;8038      ;        8.Формирование ожидаемых данных в соответствии с ожидаемыми значе-
                ;8039      ;          ниями CC CLASS и проверка.
                ;8040      ;        9.Повторение шагов 6-8 для кодов условий АЛУ, установленных в нули.
                ;8041      ;       10.Повторение шагов 2-9 для всех адресов ПЗУ от 0 до FF.
                ;8042      ;       11.Установка режима совместимости и повторение шагов 2-9 для всех
                ;8043      ;          адресов ПЗУ от 100 до 1FF.
                ;8044      ;
                ;8045      ;  ОШИБКИ:
                ;8046      ;
                ;8047      ;       ДРУГИЕ ДАННЫЕ: адрес ПЗУ (ПРИМЕЧАНИЕ: если поле "ДРУГИЕ ДАННЫЕ"
                ;8048      ;       больше или равно 100(H), устанавливается режим совместимости и
                ;8049      ;       выполняется вторая часть теста).
                ;8050      ;
                ;8051      ;       ошибка 1 - значения DT CLASS не соответствуют ожидаемым.
                ;8052      ;       ошибка 2 - значения CC CLASS не соответствуют ожидаемым.
                ;8053      ;                  коды условий АЛУ установлены.
                ;8054      ;       ошибка 3 - значения CC CLASS не соответствуют ожидаемым.
                ;8055      ;                  коды условий АЛУ сброшены.
                ;8056      ;
                ;8057      ;  НАЛАДКА:
                ;8058      ;
                ;8059      ;       ОШИБКА 1:  Ошибка может появиться при неисправности в адресации
                ;8060      ;                  ПЗУ в результате двойной адресации или из-за неправиль-
                ;8061      ;                  ного содержимого ПЗУ, или из-за неисправности в остав-
                ;8062      ;                  шейся части схемы. Сигналы DT CLASS поступают на входы
                ;8063      ;                  ПМЛ УПР.ТИПОМ ДАННЫХ и передаются на шину D, если выпол-
                ;8064      ;                  няется инструкция MOV с ячейкой 7E местной памяти. Они
                ;8065      ;                  также загружаются в регистр SIZE. Следует проверить ПМЛ
                ;8066      ;                  УПР.ТИПОМ ДАННЫХ и упомянутые сигналы.
                ;8067      ;       ОШИБКА 2-3: Эти ошибки могут появиться при неисправности в адресации
                ;8068      ;                  ПЗУ в результате двойной адресации или в результате не-
                ;8069      ;                  правильного содержимого ПЗУ. Также эта ошибка может по-
                ;8070      ;                  явиться при неисправности в схемах выхода ПЗУ тип данных,
                ;8071      ;                  класс кодов УСЛ. при передаче этой информации на шину D.
                ;8072      ;                  Необходимо проверить сигналы OPC TYPE 1, OPC TYPE 0,
                ;8073      ;                  CC CLASS 1, CC CLASS 0 и выходы из ПМЛ коды УСЛ.PSL. Также
                ;8074      ;                  источником ошибки может быть ПМЛ УСЛОВИЯ ПЕРЕХОДА, КЛАСС КОДОВ
                ;8075      ;                  УСЛОВИЙ. Правильность битов CC CLASS определяется по битам PSL.
                ;8076      ;                  Биты PSL являются результатом типа кода операции и битов АЛУ.
                ;8077      ;
                ;8078      ;
                ;8079      T.7:
U 0887, 865E,15 ;8080              MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;8081                                                       ; состояния
U 0888, 3E80,15 ;8082              MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
                ;8083                                                       ; 15 указывает начало теста для консольного процессора
U 0889, 10E0,15 ;8084              MISC [SET.CP.ATTN]                       ; выдача сигнала CPU ATTN для консольного процессора
```

```
                ;8085   WAIT.T7.0:
U 0BBA, 8BBB,A4 ;8086           JMP [WAIT.T7.0]                 ; зацикливание для ожидания ответа консольного
                ;8087                                           ; процессора
U 0BBB, B724,15 ;8088           MOV LS[HI.IPL] TO WR[0]         ;
U 0BBC, BFFE,15 ;8089           MOV WR[0] TO LS[PSL.HW]         ; сброс режима совместимости
U 0BBD, E50A,15 ;8090           CLR LS[ERROR.MASK]              ; очистка маски ошибки
U 0BBE, 650A,15 ;8091           CLR LS[PREVIOUS.ERROR]          ; очистка предыдущего номера ошибки
U 0BBF, B640,15 ;8092           MOV LS[#1] TO WR[0]             ; установка 1 в WR0
U 0BC0, 8002,15 ;8093           MOV WR[0] TO LS[ERROR.NUMBER]   ; установка номера ошибки для первой ошибки
U 0BC1, A000,15 ;8094           ROL WR[0]                       ; установка 2 в WR0
U 0BC2, 300C,15 ;8095           MOV WR[0] TO LS[MODULE.NUM]     ; установка кода модуля для модуля DAP
                ;8096    ;
                ;8097    ;T8 =94 (указатель ячейки эталона)
                ;8098    ;T9 =указатель ответа (начальное значение 500)
                ;8099    ;T10=ячейка местной памяти для эталона
                ;8100    ;T11=счетчик ответа
                ;8101    ;T12=текущий ответ
                ;8102    ;T13=FFFFFFFC маска для битов 1 и 0
                ;8103    ;T14=FFFFFFF0 маска для битов 3-0
                ;8104    ;WR2=рабочий регистр для эталона
                ;8105    ;
U 0BC3, 8881,7C ;8106           JSR [ASSERT.OTHER]              ;
U 0BC4, 6588,15 ;8107           CLR LS[ADDRESS.DATA]            ;
U 0BC5, 364E,15 ;8108           MOV LS[#80] TO WR[0]            ;
U 0BC6, C048,15 ;8109           ADD LS[#10] TO WR[0]            ;
U 0BC7, C044,15 ;8110           ADD LS[#4] TO WR[0]             ;
U 0BC8, 3E10,15 ;8111           MOV WR[0] TO LS[T8]             ; T8=94
U 0BC9, B654,15 ;8112           MOV LS[#400] TO WR[0]           ;
U 0BCA, C050,15 ;8113           ADD LS[#100] TO WR[0]           ;
U 0BCB, BE12,15 ;8114           MOV WR[0] TO LS[T9]             ; T9=500(H
U 0BCC, 2F85,15 ;8115           CLR WR[2]                       ; начальное значение данных на IB=(
U 0BCD, 3E15,15 ;8116           MOV WR[2] TO LS[T10]            ;
U 0BCE, 1910,75 ;8117           MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]  ;
U 0BCF, B214,15 ;8118           WRITE.MEM LS[T10]               ; запись начального значения данных в ячейку памяти 94
U 0BD0, E520,15 ;8119           CLR LS[PC]                      ;
U 0BD1, 372A,15 ;8120           MOV LS[#FFFFFFFC] TO WR[0]      ;
U 0BD2, 3E1A,15 ;8121           MOV WR[0] TO LS[T13]            ; T13=FFFFFFFC
U 0BD3, 372C,15 ;8122           MOV LS[#000F] TO WR[0]          ;
U 0BD4, A0C0,15 ;8123           COM WR[0]                       ;
U 0BD5, 3E1C,15 ;8124           MOV WR[0] TO LS[T14]            ; T14=FFFFFFF0
U 0BD6, B724,15 ;8125           MOV LS[HI.IPL] TO WR[0]         ;
U 0BD7, BFFE,15 ;8126           MOV WR[0] TO LS[PSL.HW]         ; сброс режима совместимости (ПРИМЕЧАНИЕ: данные
                ;8127                                           ; поступают из регистра кода операции)
U 0BD8, 1813,95 ;8128           MEM.REQ[READ.P] ADRS[T9] DT[BYTE]   ;
U 0BD9, 3816,15 ;8129           MOV MEM.DATA TO LS[T11]         ; T11=начальное значение счетчика
U 0BDA, FF12,15 ;8130           INC LS[T9]                      ; увеличение указателя памяти
U 0BDB, 1813,95 ;8131           MEM.REQ[READ.P] ADRS[T9] DT[BYTE]   ;
U 0BDC, B818,15 ;8132           MOV MEM.DATA TO LS[T12]         ; T12=первый ответ
                ;8133   LOOP.T7.1:
U 0BDD, 1B11,75 ;8134           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]  ; заполнение PFR
                ;8135             DECODE.OPC1 ADRS[T7.JUST.IN.CASE.1],  ; заполнение регистра кода операции
U 0BDE, 0487,1E ;8136           SKIP.IF(IB VALID)               ;
                ;8137   T7.JUST.IN.CASE.1:
U 0BDF, DB00,15 ;8138           NOP
                ;8139    ;
```

```
                    ;8140     ;  Следующая инструкция DECODE устанавливает линии DT CLASS и CC CLASS из ПЗУ
                    ;8141     ;  ТИП ДАННЫХ, КЛАСС КОДОВ УСЛОВИЙ, так как сигнал IRD STATE возбужден (значение
                    ;8142     ;  IFUNC нечетное и выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ)
                    ;8143     ;
U 0BE0, 8402,1E ;8144               OPC2/DE ODE,DEC.ADRS/0,IFUNC/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0BE1, DB00,15 ;8145               NOP                                     ;
U 0BE2, 36FC,15 ;8146               MOV LS[SIZE] TO WR[0]                   ; выборка DT CLASS 0 и DT CLASS 1
U 0BE3, 451A,15 ;8147               BIC LS[T13] TO WR[0]                    ; проверяются только биты 1 и 0
U 0BE4, 08C3,3C ;8148               JSR [T7.GET.EXPECTED.DT]                ; выборка ожидаемых данных
U 0BE5, B641,95 ;8149               MOV LS[#1] TO WR[3]                     ;
U 0BE6, BE83,95 ;8150               MOV WR[3] TO LS[ERROR.NUMBER]           ; номер ошибки=1
U 0BE7, 0869,3C ;8151               JSR [CHECK.RESULT]                      ; проверка DT CLASS 0 и 1
U 0BE8, 08BD,D4 ;8152               JMP [LOOP.T7.1]                         ; зацикливание при ошибке, если разрешено
                    ;8153     T7.1.1111:
U 0BE9, FF82,15 ;8154               INC LS[ERROR.NUMBER]                    ; номер ошибки=2
U 0BEA, 372D,95 ;8155               MOV LS[#000F] TO WR[3]                  ;
U 0BEB, 3FFD,95 ;8156               MOV WR[3] TO LS[ALU.CC]                 ; установка всех кодов условий АЛУ
U 0BEC, DB00,75 ;8157               DT(SIZE)&MAP.ALU.CC.TO.PSL              ; преобразование кодов условий АЛУ в коды условий PSL в
                    ;8158                                                   ; зависимости от CC CLASS
U 0BED, B6FE,15 ;8159               MOV LS[PSL.CC] TO WR[0]                 ; выборка преобразованных кодов условий
U 0BEE, 451C,15 ;8160               BIC LS[T14] TO WR[0]                    ; проверка только битов 3-0
U 0BEF, 08C4,7C ;8161               JSR [T7.GET.EXPECTED.CC1111]            ; выборка ожидаемых преобразованных кодов условий
U 0BF0, 0869,3C ;8162               JSR [CHECK.RESULT]                      ; проверка CC CLASS 0 и 1 с данными 1111(B)
U 0BF1, 08BD,D4 ;8163               JMP [LOOP.T7.1]                         ; зацикливание при ошибке, если разрешено
                    ;8164     T7.1.0000:
U 0BF2, FF82,15 ;8165               INC LS[ERROR.NUMBER]                    ; номер ошибки=3
U 0BF3, 2F80,15 ;8166               CLR WR[0]                               ;
U 0BF4, 3FFC,15 ;8167               MOV WR[0] TO LS[ALU.CC]                 ; очистка всех кодов условий АЛУ
U 0BF5, DB00,75 ;8168               DT(SIZE)&MAP.ALU.CC.TO.PSL              ; преобразование кодов условий АЛУ в коды условий PSL в
                    ;8169                                                   ; зависимости от CC CLASS
U 0BF6, B6FE,15 ;8170               MOV LS[PSL.CC] TO WR[0]                 ; выборка преобразованных кодов условий
U 0BF7, 451C,15 ;8171               BIC LS[T14] TO WR[0]                    ; проверка только битов 3-0
U 0BF8, 08C6,0C ;8172               JSR [T7.GET.EXPECTED.CC0000]            ; выборка ожидаемых преобразованных кодов условий
U 0BF9, 0869,3C ;8173               JSR [CHECK.RESULT]                      ; проверка CC CLASS 0 и 1 с данными 0000(B)
U 0BFA, 08BD,D4 ;8174               JMP [LOOP.T7.1]                         ; зацикливание при ошибке, если разрешено
U 0BFB, 7C16,15 ;8175               DEC LS[T11]                             ; уменьшение счетчика ответа
U 0BFC, 08C3,CC ;8176               JSR [T7.CHECK.COUNTER]                  ; проверка счетчика и выборка нового ответа, если 0
U 0BFD, 2045,15 ;8177               INC WR[2]                               ;
U 0BFE, 3E15,15 ;8178               MOV WR[2] TO LS[T10]                    ; увеличение данных для IB
U 0BFF, 1910,75 ;8179               MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0C00, B214,15 ;8180               WRITE.MEM LS[T10]                       ; запись новых данных в ячейку памяти 94
U 0C01, FF88,15 ;8181               INC LS[ADDRESS.DATA]                    ;
U 0C02, D850,15 ;8182               MOV LS[#100] TO Q                       ;
                    ;8183                 CMP WR[2] WITH Q,                 ;
U 0C03, A484,35 ;8184               DT(LONG)&SET.ALU.CC                     ; данные для IB=100(H)?
U 0C04, 08BD,D1 ;8185               JMP.IF[NEQ] TO [LOOP.T7.1]              ; возврат, если нет
                    ;8186     T7.2:
U 0C05, 2F85,15 ;8187               CLR WR[2]                               ;
U 0C06, 6514,15 ;8188               CLR LS[T10]                             ; восстановление данных для IB
U 0C07, 1910,75 ;8189               MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0C08, B214,15 ;8190               WRITE.MEM LS[T10]                       ; запись новых данных в ячейку памяти 94
U 0C09, B724,15 ;8191               MOV LS[HI.IPL] TO WR[0]                 ;
U 0C0A, 477E,15 ;8192               BIS LS[BIT31] TO WR[0]                  ;
U 0C0B, BFFE,15 ;8193               MOV WR[0] TO LS[PSL.HW]                 ; установка режима совместимости
                    ;8194     LOOP.T7.2:
```

```
U 0C0C, 1B11,75 ;8195            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; заполнение PFR
                ;8196     ;
                ;8197     ;  Следующая инструкция DECODE устанавливает линии DT CLASS и CC CLASS из ПЗУ
                ;8198     ;  ТИП ДАННЫХ, КЛАСС КОДОВ УСЛОВИЙ, так как сигнал IRD STATE возбужден (значение
                ;8199     ;  IFUNC нечетное и выбрана микросхема ПЗУ ДЕШ.КОДОВ ОПЕРАЦИЙ)
                ;8200     ;
U 0C0D, 8402,1E ;8201            OPC2/DECODE,DEC.ADRS/0,IFUNC/1,BFC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0C0E, DB00,15 ;8202            NOP                                     ;
U 0C0F, 36FC,15 ;8203            MOV LS[SIZE] TO WR[0]                   ; выборка DT CLASS 0 и DT CLASS 1
U 0C10, 451A,15 ;8204            BIC LS[T13] TO WR[0]                    ; проверка только битов 1 и 0
U 0C11, 08C3,3C ;8205            JSR [T7.GET.EXPECTED.DT]                ; выборка ожидаемых данных
U 0C12, B641,95 ;8206            MOV LS[#1] TO WR[3]                     ;
U 0C13, BEB3,95 ;8207            MOV WR[3] TO LS[ERROR.NUMBER]           ; номер ошибки=1
U 0C14, 0869,3C ;8208            JSR [CHECK.RESULT]                      ; проверка DT CLASS 0 и 1
U 0C15, 88C0,C4 ;8209            JMP [LOOP.T7.2]                         ; зацикливание при ошибке, если разрешено
                ;8210     T7.2.1111:
U 0C16, FF82,15 ;8211            INC LS[ERROR.NUMBER]                    ; номер ошибки=2
U 0C17, 372D,95 ;8212            MOV LS[#000F] TO WR[3]                  ;
U 0C18, 3FFD,95 ;8213            MOV WR[3] TO LS[ALU.CC]                 ; установка всех кодов условий АЛУ
U 0C19, DB00,75 ;8214            DT(SIZE)&MAP.ALU.CC.TO.PSL              ; преобразование кодов условий АЛУ в коды условий PSL в
                ;8215                                                    ; зависимости от CC CLASS
U 0C1A, B6FE,15 ;8216            MOV LS[PSL.CC] TO WR[0]                 ; выборка преобразованных кодов условий
U 0C1B, 451C,15 ;8217            BIC LS[T14] TO WR[0]                    ; проверка только битов 3-0
U 0C1C, 08C4,7C ;8218            JSR [T7.GET.EXPECTED.CC1111]            ; выборка ожидаемых преобразованных кодов условий
U 0C1D, 0869,3C ;8219            JSR [CHECK.RESULT]                      ; проверка CC CLASS 0 и 1 с данными 1111(B)
U 0C1E, 88C0,C4 ;8220            JMP [LOOP.T7.2]                         ; зацикливание при ошибке, если разрешено
                ;8221     T7.2.0000:
U 0C1F, FF82,15 ;8222            INC LS[ERROR.NUMBER]                    ; номер ошибки=3
U 0C20, 2F80,15 ;8223            CLR WR[0]                               ;
U 0C21, 3FFC,15 ;8224            MOV WR[0] TO LS[ALU.CC]                 ; очистка всех кодов условий АЛУ
U 0C22, DB00,75 ;8225            DT(SIZE)&MAP.ALU.CC.TO.PSL              ; преобразование кодов условий АЛУ в коды условий PSL в
                ;8226                                                    ; зависимости от CC CLASS
U 0C23, B6FE,15 ;8227            MOV LS[PSL.CC] TO WR[0]                 ; выборка преобразованных кодов условий
U 0C24, 451C,15 ;8228            BIC LS[T14] TO WR[0]                    ; проверка только битов 3-0
U 0C25, 08C6,0C ;8229            JSR [T7.GET.EXPECTED.CC0000]            ; выборка ожидаемых преобразованных кодов условий
U 0C26, 0869,3C ;8230            JSR [CHECK.RESULT]                      ; проверка CC CLASS 0 и 1 с данными 0000(B)
U 0C27, 88C0,C4 ;8231            JMP [LOOP.T7.2]                         ; зацикливание при ошибке, если разрешено
U 0C28, 7C16,15 ;8232            DEC LS[T11]                             ; уменьшение счетчика ответа
U 0C29, 08C3,CC ;8233            JSR [T7.CHECK.COUNTER]                  ; проверка счетчика и выборка нового ответа, если 0
U 0C2A, 2045,15 ;8234            INC WR[2]                               ;
U 0C2B, 3E15,15 ;8235            MOV WR[2] TO LS[T10]                    ; увеличение данных для IB
U 0C2C, 1910,75 ;8236            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0C2D, B214,15 ;8237            WRITE.MEM LS[T10]                       ; запись новых данных в ячейку памяти 94
U 0C2E, FF88,15 ;8238            INC LS[ADDRESS.DATA]                    ;
U 0C2F, D850,15 ;8239            MOV LS[#100] TO Q                       ;
                ;8240              CMP WR[2] WITH Q,                     ;
U 0C30, A484,35 ;8241            DT(LONG)&SET.ALU.CC                     ; данные для IB=100(H)?
U 0C31, 88C0,C1 ;8242            JMP.IF[NEQ] TO [LOOP.T7.2]              ; возврат, если нет
U 0C32, 08C7,54 ;8243            JMP [T7.CLEANUP]                        ; обход всех подпрограмм
                ;8244     ;
                ;8245     ;  Эта подпрограмма принимает биты 2 и 3 ответа и помещает их в биты 0 и
                ;8246     ;  1 WR1 (соответственно)
                ;8247     ;
                ;8248     T7.GET.EXPECTED.DT:
U 0C33, 2F82,95 ;8249            CLR WR[1]                               ; очистка ожидаемых данных
```

```
U 0C34, 3619,95 ;8250            MOV LS[T12] TO WR[3]                    ; запоминание ответа в WR3
                ;8251              BIT LS[BIT2] WITH WR[3],              ;
U 0C35, D945,B5 ;8252            DT(LONG)&SET.ALU.CC                     ; бит 2 установлен?
U 0C36, 5B00,09 ;8253            SKIP.IF[EQL]                            ; пропуск, если нет
U 0C37, 4740,95 ;8254            BIS LS[BIT0] TO WR[1]                   ; установка бита 0
                ;8255              BIT LS[BIT3] WITH WR[3],              ;
U 0C38, 5947,B5 ;8256            DT(LONG)&SET.ALU.CC                     ; бит 3 установлен?
U 0C39, 5B00,09 ;8257            SKIP.IF[EQL]                            ; пропуск, если нет
U 0C3A, C742,95 ;8258            BIS LS[BIT1] TO WR[1]                   ; установка бита 1
U 0C3B, 5B00,14 ;8259            RETURN                                  ;
                ;8260     ;
                ;8261     ;  Эта подпрограмма проверяет наличие нуля в счетчике ответа и при наличии
                ;8262     ;  нуля выбирает из памяти новое значение счетчика (помещает в LS[T11]) и
                ;8263     ;  новый ответ (LS[T12]).
                ;8264     ;
                ;8265     T7.CHECK.COUNTER:
U 0C3C, B617,95 ;8266            MOV LS[T11] TO WR[3]                    ; пересылка счетчика в WR3
                ;8267              TST WR[3],                            ;
U 0C3D, 2007,B5 ;8268            DT(LONG)&SET.ALU.CC                     ; счетчик=0?
U 0C3E, 5B00,09 ;8269            SKIP.IF[EQL]                            ; пропуск, если да
U 0C3F, 5B00,14 ;8270            RETURN                                  ; возврат, если нет
U 0C40, FF12,15 ;8271            INC LS[T9]                              ; увеличение указателя
U 0C41, 1813,95 ;8272            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 0C42, 3816,15 ;8273            MOV MEM.DATA TO LS[T11]                 ; выборка нового счетчика
U 0C43, FF12,15 ;8274            INC LS[T9]                              ; увеличение указателя
U 0C44, 1813,95 ;8275            MEM.REQ[READ.P] ADRS[T9] DT[BYTE]       ;
U 0C45, 8818,15 ;8276            MOV MEM.DATA TO LS[T12]                 ; выборка нового ответа
U 0C46, 5B00,14 ;8277            RETURN                                  ;
                ;8278     ;
                ;8279     ;  Эта подпрограмма формирует в WR1 один из возможных преобразованных эталонов
                ;8280     ;  кодов условий АЛУ, когда все биты кодов условий установлены.
                ;8281     ;
                ;8282     T7.GET.EXPECTED.CC1111:
U 0C47, 3619,95 ;8283            MOV LS[T12] TO WR[3]                    ; выборка ответа
U 0C48, B72A,95 ;8284            MOV LS[#FFFFFFFC] TO WR[1]              ;
U 0C49, AF43,95 ;8285            BIC WR[1] TO WR[3]                      ; очистка всех битов, кроме 1 и 0
U 0C4A, D89C,15 ;8286            MOV LS[ZERO] TO Q                       ;
                ;8287              CMP WR[3] WITH Q,                     ;
U 0C4B, 2486,35 ;8288            DT(LONG)&SET.ALU.CC                     ; WR3=0?
U 0C4C, 08C4,F1 ;8289            JMP.IF[NEQ] TO [T7.1111.TRY.1]          ; переход, если нет
U 0C4D, B72C,95 ;8290            MOV LS[#000F] TO WR[1]                  ; ожидаемые данные=все биты кодов условий установлены
U 0C4E, 5B00,14 ;8291            RETURN                                  ;
                ;8292     T7.1111.TRY.1:
U 0C4F, 5840,15 ;8293            MOV LS[#1] TO Q                         ;
                ;8294              CMP WR[3] WITH Q,                     ;
U 0C50, 2486,35 ;8295            DT(LONG)&SET.ALU.CC                     ; WR3=1?
U 0C51, 88C5,51 ;8296            JMP.IF[NEQ] TO [T7.1111.TRY.2]          ; переход, если нет
U 0C52, B72C,95 ;8297            MOV LS[#000F] TO WR[1]                  ;
U 0C53, C540,95 ;8298            BIC LS[BIT0] TO WR[1]                   ; ожидаемые данные=все биты установлены, кроме C
U 0C54, 5B00,14 ;8299            RETURN                                  ;
                ;8300     T7.1111.TRY.2:
U 0C55, D842,15 ;8301            MOV LS[#2] TO Q                         ;
                ;8302              CMP WR[3] WITH Q,                     ;
U 0C56, 2486,35 ;8303            DT(LONG)&SET.ALU.CC                     ; WR3=2?
U 0C57, 08C5,B1 ;8304            JMP.IF[NEQ] TO [T7.1111.MUST.BE.3]      ; переход, если нет
```

```
U 0C58, 2F82,95 ;8305            CLR WR[1]                              ;
U 0C59, C744,95 ;8306            BIS LS[BIT2] TO WR[1]                  ; ожидаемые данные=все биты сброшены, кроме Z
U 0C5A, 5B00,14 ;8307            RETURN                                 ;
                ;8308    T7.1111.MUST.BE.3:
U 0C5B, B72C,95 ;8309            MOV LS[#000F] TO WR[1]                 ;
U 0C5C, 4542,95 ;8310            BIC LS[BIT1] TO WR[1]                  ; ожидаемые данные=все биты установлены, кроме V
U 0C5D, 4540,15 ;8311            BIC LS[BIT0] TO WR[0]                  ; бит C не проверяется
U 0C5E, C540,95 ;8312            BIC LS[BIT0] TO WR[1]                  ; бит C не проверяется
U 0C5F, 5B00,14 ;8313            RETURN                                 ;
                ;8314    ;
                ;8315    ;  Эта подпрограмма формирует в WR1 один из возможных четырех преобразованных
                ;8316    ;  эталонов кодов условий АЛУ, когда все биты кодов условий сброшены.
                ;8317    ;
                ;8318    T7.GET.EXPECTED.CC0000:
U 0C60, 3619,95 ;8319            MOV LS[T12] TO WR[3]                   ; выборка ответа
U 0C61, B72A,95 ;8320            MOV LS[#FFFFFFFC] TO WR[1]             ;
U 0C62, AF43,95 ;8321            BIC WR[1] TO WR[3]                     ; очистка всех битов, кроме 1 и 0
U 0C63, D89C,15 ;8322            MOV LS[ZERO] TO Q                      ;
                ;8323              CMP WR[3] WITH Q,                    ;
U 0C64, 2486,35 ;8324            DT(LONG)&SET.ALU.CC                    ; WR3=0?
U 0C65, 08C6,81 ;8325            JMP.IF[NEQ] TO [T7.0000.TRY.1]         ; переход, если нет
U 0C66, 2F82,95 ;8326            CLR WR[1]                              ; ожидаемые данные=все биты кодов условий сброшены
U 0C67, 5B00,14 ;8327            RETURN                                 ;
                ;8328    T7.0000.TRY.1:
U 0C68, 5840,15 ;8329            MOV LS[#1] TO Q                        ;
                ;8330              CMP WR[3] WITH Q,                    ;
U 0C69, 2486,35 ;8331            DT(LONG)&SET.ALU.CC                    ; WR3=1?
U 0C6A, 08C6,D1 ;8332            JMP.IF[NEQ] TO [T7.0000.TRY.2]         ; переход, если нет
U 0C6B, 3640,95 ;8333            MOV LS[BIT0] TO WR[1]                  ; ожидаемые данные=все биты сброшены, кроме C
U 0C6C, 5B00,14 ;8334            RETURN                                 ;
                ;8335    T7.0000.TRY.2:
U 0C6D, D842,15 ;8336            MOV LS[#2] TO Q                        ;
                ;8337              CMP WR[3] WITH Q,                    ;
U 0C6E, 2486,35 ;8338            DT(LONG)&SET.ALU.CC                    ; WR3=2?
U 0C6F, 88C7,21 ;8339            JMP.IF[NEQ] TO [T7.0000.MUST.BE.3]     ; переход, если нет
U 0C70, 3640,95 ;8340            MOV LS[BIT0] TO WR[1]                  ; ожидаемые данные=все биты сброшены, кроме C
U 0C71, 5B00,14 ;8341            RETURN                                 ;
                ;8342    T7.0000.MUST.BE.3:
U 0C72, 2F82,95 ;8343            CLR WR[1]                              ; ожидаемые данные=все биты сброшены
U 0C73, 4540,15 ;8344            BIC LS[BIT0] TO WR[0]                  ; бит C не проверяется
U 0C74, 5B00,14 ;8345            RETURN                                 ;
                ;8346    T7.CLEANUP:
U 0C75, B724,15 ;8347            MOV LS[HI.IPL] TO WR[0]                ;
U 0C76, BFFE,15 ;8348            MOV WR[0] TO LS[PSL.HW]                ; сброс режима совместимости
                ;8349    END.T7:
```

```
                ;8350      .PAGE "ТЕСТ 8 - тест останова синхронизатора (CLOCK STALL) (модуль DAP)"
                ;8351      ;
                ;8352      ;
                ;8353      ; ОПИСАНИЕ ТЕСТА:
                ;8354      ;
                ;8355      ;   Этот тест проверяет формирование сигнала CLOCK STALL (останов синхронизатора).
                ;8356      ;   Сигнал CLOCK STALL появляется, если выдается инструкция DECODE при PC=3, ко-
                ;8357      ;   торая формирует запрос памяти, а за ней следует другая инструкция DECODE.
                ;8358      ;   Этот сигнал останова появляется потому, что первая инструкция DECODE вызывает
                ;8359      ;   процедуру запроса памяти, которая не заканчивается до выдачи следующей инст-
                ;8360      ;   рукции DECODE. Ошибка может быть обнаружена, если разные данные содержатся в
                ;8361      ;   регистре предвыборки инструкций и в следующей ячейке памяти.
                ;8362      ;
                ;8363      ; ПРЕДПОЛОЖЕНИЯ:
                ;8364      ;
                ;8365      ;   Предполагается, что регистр предвыборки инструкций и регистр OS проверены.
                ;8366      ;   Также предполАгается, что инструкция DECODE выполняется и инструкция DECODE
                ;8367      ;   при PC=3 формирует запрос памяти. Предполагается, что регистр предвыборки
                ;8368      ;   инструкций содержит нули и оперативная память содержит единицы.
                ;8369      ;
                ;8370      ; ШАГИ ТЕСТА:
                ;8371      ;
                ;8372      ;   1) Выдача инструкции DECODE при PC=3 с установленным IB REQ, которая дол-
                ;8373      ;      жна выставить запрос памяти. Учитывается время работы памяти.
                ;8374      ;   2) Выдача следующей инструкции DECODE для загрузки регистра OS.
                ;8375      ;   3) Загрузка в WR[0] регистра OS и единиц в WR[1]. Вызов программы проверки
                ;8376      ;      результата.
                ;8377      ;   4) Выдача инструкции DECODE при PC=3 с установленным IB REQ, которая
                ;8378      ;      должна выставить запрос памяти. Учитывается время работы памяти.
                ;8379      ;   5) Выдача следующей инструкции DECODE для загрузки регистра OS.
                ;8380      ;   6) Загрузка в WR[0] регистра OS и единиц в WR[1]. Вызов программы проверки
                ;8381      ;      результата.
                ;8382      ;
                ;8383      ; ОШИБКИ:
                ;8384      ;
                ;8385      ;   ошибка 1 - сигнал IB VALID не снимается во время запроса памяти.
                ;8386      ;   ошибка 2 - не формируется разрешение сигнала CLOCK STALL.
                ;8387      ;
                ;8388      ; НАЛАДКА:
                ;8389      ;
                ;8390      ;   ОШИБКА 1: Если IB VALID не снимается во время запроса памяти, необходимо
                ;8391      ;                проверить входы ПМЛ УПР.IB, РЕГ.ПРЕДВЫБ.ИНСТРУКЦИЙ. В частности,
                ;8392      ;                биты 0 и 1 шины Y, сигналы DECODE INSTR и CPU P2.
                ;8393      ;   ОШИБКА 2: Если неправильно работает CLOCK STALL, два участка схем могут
                ;8394      ;                быть источниками ошибок. первым участком являются схемы, формиру-
                ;8395      ;                ющие CLOCK STALL. На этом участке, вероятнее всего, неправильно
                ;8396      ;                работает вентиль, пропускающий сигнал STALL ON IB. Вторым возмож-
                ;8397      ;                ным источником ошибки может быть ПМЛ УПР.IB, РЕГ.ПРЕДВЫБ.ИНСТ-
                ;8398      ;                РУКЦИЙ.
                ;8399      ;
                ;8400      T.8:
U 0C77, B65E,15 ;8401              MOV LS[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;8402                                                       ; состояния
U 0C78, 3E80,15 ;8403              MOV WR[0] TO LS[CONTROL.STATUS]           ; установка бита 15 в слове управления и состояния. Бит
                ;8404                                                       ; 15 указывает начало теста для конс. процессора
```

```
U 0C79, 10E0,15 ;8405            MISC [SET.CP.ATTN]                    ; выдача сигнала CPU ATTN для конс.процессора
                ;8406    WAIT.T8.0:
U 0C7A, 08C7,A4 ;8407            JMP [WAIT.T8.0]                       ; зацикливание для ожидания ответа конс.процессора
U 0C7B, B724,15 ;8408            MOV LS[HI.IPL] TO WR[0]               ;
U 0C7C, BFFE,15 ;8409            MOV WR[0] TO LS[PSL.HW]               ; сброс режима совместимости
U 0C7D, 372C,15 ;8410            MOV LS[#000F] TO WR[0]                ;
U 0C7E, 372F,15 ;8411            MOV LS[#00F0] TO WR[2]                ;
U 0C7F, AE04,15 ;8412            ADD WR[2] TO WR[0]                    ;
U 0C80, A0C0,15 ;8413            COM WR[0]                             ;
U 0C81, 3E8A,15 ;8414            MOV WR[0] TO LS[ERROR.MASK]           ; маска=FFFFFF00
U 0C82, 65A0,15 ;8415            CLR LS[PREVIOUS.ERROR]                ; очистка предыдущего номера ошибки
U 0C83, B640,15 ;8416            MOV LS[#1] TO WR[0]                   ; установка 1 в WR0
U 0C84, BE82,15 ;8417            MOV WR[0] TO LS[ERROR.NUMBER]         ; установка номера ошибки на первую ошибку
U 0C85, A3C0,15 ;8418            ROL WR[0]                             ; установка 2 в WR0
U 0C86, 3E8C,15 ;8419            MOV WR[0] TO LS[MODULE.NUM]           ; установка кода модуля для модуля DAP
U 0C87, 364E,15 ;8420            MOV LS[#80] TO WR[0]                  ;
U 0C88, 4046,15 ;8421            ADD LS[#8] TO WR[0]                   ;
U 0C89, 3E10,15 ;8422            MOV WR[0] TO LS[T8]                   ; T8=88 (указатель слова, содержащего нули)
U 0C8A, C042,15 ;8423            ADD LS[#2] TO WR[0]                   ;
U 0C8B, BE12,15 ;8424            MOV WR[0] TO LS[T9]                   ; T9=8A
                ;8425    LOOP.T8:
U 0C8C, 3612,15 ;8426            MOV LS[T9] TO WR[0]                   ;
U 0C8D, 3E20,15 ;8427            MOV WR[0] TO LS[PC]                   ; PC=8A
U 0C8E, 1B11,75 ;8428            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]    ; заполнение PFR нулями
U 0C8F, DB00,15 ;8429            NOP                                   ; учитывается время работы памяти
U 0C90, DB00,15 ;8430            NOP                                   ;
U 0C91, DB00,15 ;8431            NOP                                   ;
U 0C92, DB00,15 ;8432            NOP                                   ;
U 0C93, DB00,15 ;8433            NOP                                   ;
U 0C94, 8401,4E ;8434            MOV IB.DATA TO OS                     ; пересылка данных IB в OS и увеличение PC
U 0C95, DB00,15 ;8435            NOP                                   ;
U 0C96, 8401,4E ;8436            MOV IB.DATA TO OS                     ; пересылка данных IB в OS и увеличение PC
U 0C97, DB00,15 ;8437            NOP                                   ;
                ;8438    T8.1:
U 0C98, 8401,4E ;8439            MOV IB.DATA TO OS                     ; выборка следующего байта (запущенная выше операция
                ;8440                                                  ; закончилась)
U 0C99, 08C9,84 ;8441            JMP [T8.1]                            ; если нет IB VALID, переход назад
U 0C9A, B6F8,15 ;8442            MOV LS[OS] TO WR[0]                   ; выборка записанных данных
U 0C9B, B72C,95 ;8443            MOV LS[#000F] TO WR[1]                ;
U 0C9C, 372F,15 ;8444            MOV LS[#00F0] TO WR[2]                ;
U 0C9D, 2E04,95 ;8445            ADD WR[2] TO WR[1]                    ; ожидаемые данные=000000FF
U 0C9E, 0869,3C ;8446            JSR [CHECK.RESULT]                    ; проверка, что сигнал CLOCK STALL сработал
U 0C9F, 08C8,C4 ;8447            JMP [LOOP.T8]                         ;
U 0CA0, FF82,15 ;8448            INC LS[ERROR.NUMBER]                  ; увеличение номера ошибки до 2
                ;8449    LOOP.T8.2:
U 0CA1, 3612,15 ;8450            MOV LS[T9] TO WR[0]
U 0CA2, 3E20,15 ;8451            MOV WR[0] TO LS[PC]                   ; PC=8A
U 0CA3, 1B11,75 ;8452            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]    ; заполнение PFR нулями
U 0CA4, 8401,4E ;8453            MOV IB.DATA TO OS                     ; пересылка данных IB в OS и увеличение PC (должен быт
                ;8454                                                  ; приостанов, пока память не закончит работу)
U 0CA5, DB00,15 ;8455            NOP                                   ;
U 0CA6, 8401,4E ;8456            MOV IB.DATA TO OS                     ; пересылка данных IB в OS и увеличение PC (должен быт
                ;8457                                                  ; приостанов, пока память не закончит работу)
U 0CA7, DB00,15 ;8458            NOP
                ;8459    T8.2:
```

TECT 8 - тест останова синхронизатора (CLOCK STALL) (модуль DAP)

```
U 0CA8, 8401,4E ;8460            MOV IB.DATA TO OS                    ; выборка следующего байта (запущенная выше операция
                ;8461                                                 ; закончилась)
U 0CA9, 08CA,84 ;8462            JMP [T8.2]                           ; если нет IB VALID, переход назад
U 0CAA, B6F8,15 ;8463            MOV LS[OS] TO WR[0]                  ; выборка записанных данных
U 0CAB, B72C,95 ;8464            MOV LS[#000F] TO WR[1]               ;
U 0CAC, 372F,15 ;8465            MOV LS[#00F0] TO WR[2]               ;
U 0CAD, 2E04,95 ;8466            ADD WR[2] TO WR[1]                   ; ожидаемые данные=000000FF
U 0CAE, 0869,3C ;8467            JSR [CHECK.RESULT]                   ; проверка, что сигнал CLOCK STALL сработал
U 0CAF, 08CA,14 ;8468            JMP [LOOP.T8.2]                      ;
                ;8469    END.T8:
```

```
;8470      .PAGE "ТЕСТ 9 - тест регистрового режима адресации (модуль DAP)"
;8471      ;
;8472      ; ОПИСАНИЕ ТЕСТА:
;8473      ;
;8474      ;    Этот тест проверяет, что схемы, формирующие сигналы REGISTER MODE и GPR
;8475      ;   DEST, работают правильно. Сигналы REGISTER MODE и GPR DEST формируются при
;8476      ;   одном из двух условий. Они формируются, если биты IB 4-7 содержАт 0101(B) и
;8477      ;   выполняется инструкция DECODE с установленными битами 5 и 6. Вторым случаем
;8478      ;   является инструкция DECODE с установленными битами 5 и 6 при режиме совмес-
;8479      ;   тимости, когда биты шины IB 3,4 и 5 сброшены. Для полной проверки этих слу-
;8480      ;   чаев подбираются тестовые данные, аналогичные кодам, которые вызывают уста-
;8481      ;   новку сигнала REGISTER MODE и проверяется, что только правильный код вызыва-
;8482      ;   ет установку сигнала REGISTER MODE. коды, которые должны быть проверены в
;8483      ;   собственном режиме, следующие: 0101,1101,0001,0111 и 0100(B). В режиме сов-
;8484      ;   местимости должны быть проверены коды в битах 3-5 101, 000, 001, 010 и
;8485      ;   100(B).
;8486      ;
;8487      ; ПРЕДПОЛОЖЕНИЯ:
;8488      ;
;8489      ;   Предполагается, что шина IB проверена, а также разрешение и работа ПЗУ ДЕШ
;8490      ;   спецификаторов проверены и работают правильно.
;8491      ;
;8492      ; ШАГИ ТЕСТА:
;8493      ;
;8494      ;   1. Выдача инструкции DECODE с установленными битами 5 и 7 и значением 50(H)
;8495      ;      на шине IB.
;8496      ;   2. Выдача NOP с пропуском при RDEST в поле управления пропуском. Если пропуск
;8497      ;      не выполняется, тогда печатается ошибка 1.
;8498      ;   3. Снова выдается инструкция DECODE с установленными битами 5 и 7 для проверки
;8499      ;      одного из четырех неправильных кодов на шине IB.
;8500      ;   4. Выдача NOP с пропуском при RDEST в поле управления пропуском. Если пропуск
;8501      ;      выполняется, тогда печатается ошибка 2.
;8502      ;   5. Если не все неправильные коды проверены, тогда переход к шагу 3.
;8503      ;   6. Выдача инструкции MOV для установки режима совместимости.
;8504      ;   7. Выдача инструкции DECODE с установленными битами 5 и 7 и одним из четы-
;8505      ;      рех неправильных кодов на шине IB.
;8506      ;   8. Выдача инструкции NOP с пропуском при RDEST в поле управления пропуском
;8507      ;      Если пропуск выполняется, печатается ошибка 3, иначе ошибки нет
;8508      ;   9. Если ошибки нет, и не все неправильные коды проверены, тогда переход к
;8509      ;      шагу 7.
;8510      ;  10. Выдача инструкции DECODE с установленными битами 5 и 6 и нулями на IB.
;8511      ;  11. Выдача инструкции NOP с пропуском при RDEST в поле управления пропуском.
;8512      ;      Если пропуск не выполняется, тогда печатается ошибка 4.
;8513      ;
;8514      ; ОШИБКИ:
;8515      ;
;8516      ;   ошибка 1 - не возбуждается сигнал REGISTER MODE в собственном режиме.
;8517      ;   ошибка 2 - сигнал REGISTER MODE возбуждается в собственном режиме, когда
;8518      ;              это не должно произойти
;8519      ;   ошибка 3 - сигнал REGISTER MODE возбуждается в режиме совместимости, когда
;8520      ;              это не должно произойти.
;8521      ;   ошибка 4 - не возбуждается сигнал REGISTER MODE в режиме совместимости.
;8522      ;
;8523      ; НАЛАДКА:
;8524      ;
```

```
                    ;8525      ;    ОШИБКА 1: Если имеется эта ошибка, GPR DEST не был выставлен, когда это долж-
                    ;8526      ;              но было произойти. Может быть несколько источников ошибки. Первым
                    ;8527      ;              источником может быть ПМЛ УПР.МИАСС ЕВЕНЕСЕРОМ и мультиплексор уп-
                    ;8528      ;              равления микросеквенсером. Если сигнал GPR DEST сформирован, а эти
                    ;8529      ;              схемы неисправны, тогда  пропуск не произойдет. Другим возможным ис-
                    ;8530      ;              точником ошибки может быть ПМЛ РЕЖИМ СОВМЕСТ. РЕГ.АДРЕСАЦИЯ,
                    ;8531      ;              которая выдает сигнал GPR DEST. Из входов, которые определяют
                    ;8532      ;              сигнал GPR DEST, скорее всего только один сигнал REGISTER MODE может
                    ;8533      ;              быть неправильным. Сигнал REGISTER MODE формируется в ПМЛ УСЛОВИЯ ПЕРЕ-
                    ;8534      ;              ХОДОВ, КЛАСС КОДОВ УСЛОВИЙ и определяется сигналами RMODE A и RMODE B.
                    ;8535      ;              Сигнал RMODE A формируется в ПМЛ OS[7:1] НЕЧЕТН. а сигнал RMODE B - ПМЛ
                    ;8536      ;              OS[6:0] ЧЕТН. Сигналы RMODE A и RMODE B формируются только в том
                    ;8537      ;              случае, если на шине IB имеется определенный код.
                    ;8538      ;    ОШИБКА 2: Если появляется эта ошибка, тогда код, который не должен возбуждать
                    ;8539      ;              сигнала GPR DEST, возбуждает его. Этот тип ошибки, как правило, ука-
                    ;8540      ;              зывает закорачивание между контактами, хотя любые схемы, упомянутые
                    ;8541      ;              в описании ошибки 1 могут также работать неправильно.
                    ;8542      ;    ОШИБКА 3: В этом случае сигнал GPR DEST не должен возбуждаться в результате не-
                    ;8543      ;              правильного кода, загруженного на шину IB, когда процессор работает
                    ;8544      ;              в режиме совместимости. Как и в случае ошибки 2, эта ошибка может быть
                    ;8545      ;              в результате закорачивания между контактами, но также может появиться
                    ;8546      ;              в результате неправильной работы любой схемы, упомянутой в описании
                    ;8547      ;              ошибки 1.
                    ;8548      ;    ОШИБКА 4: Если появляется эта ошибка, тогда сигнал GPR DEST должен быть возбуж-
                    ;8549      ;              ден, процессор должен быть в режиме совместимости и должны быть нули
                    ;8550      ;              в битах 3-5 шины IB. Возможными источниками ошибки в этом случае яв-
                    ;8551      ;              ляются те же, которые упомянуты в описании ошибки 1 и при ошибке дол-
                    ;8552      ;              ны проверяться.
                    ;8553      T.9:
U 0CB0, B65E,15     ;8554              MOV LS[BEGIN.TEST] TO WR[0]                ; установка бита 15 в WR0 для слова управления и
                    ;8555                                                         ; состояния
U 0CB1, 3EB0,15     ;8556              MOV WR[0] TO LS[CONTROL.STATUS]             ; установка бита 15 в слове управления и состояния. Бит
                    ;8557                                                         ; 15 указывает начало теста для консольного процессора
U 0CB2, 10E0,15     ;8558              MISC [SET:CP.ATTN]                         ; выдача сигнала CPU ATTN для консольного процессора
                    ;8559      WAIT.T9.0:
U 0CB3, 08CB,34     ;8560              JMP [WAIT.T9.0]                            ; зацикливание для ожидания ответа консольного
                    ;8561                                                         ; процессора
U 0CB4, B724,15     ;8562              MOV LS[HI.IPL] TO WR[0]                    ;
U 0CB5, BFFE,15     ;8563              MOV WR[0] TO LS[PSL.HW]                     ; сброс режима совместимости
U 0CB6, E58A,15     ;8564              CLR LS[ERROR.MASK]                         ; очистка маски ошибок
U 0CB7, 65A0,15     ;8565              CLR LS[PREVIOUS.ERROR]                     ; очистка предыдущего номера ошибки
U 0CB8, B640,15     ;8566              MOV LS[#1] TO WR[0]                        ; установка 1 в WR0
U 0CB9, BE82,15     ;8567              MOV WR[0] TO LS[ERROR.NUMBER]              ; установка номера ошибки для первой ошибки
U 0CBA, A3C0,15     ;8568              ROL WR[0]                                  ; установка 2 в WR0 .
U 0CBB, 3E8C,15     ;8569              MOV WR[0] TO LS[MODULE.NUM]                 ; установка номера модуля для модуля DAP
U 0CBC, 364E,15     ;8570              MOV LS[#80] TO WR[0]                       ;
U 0CBD, C048,15     ;8571              ADD LS[#10] TO WR[0]                       ;
U 0CBE, C044,15     ;8572              ADD LS[#4] TO WR[0]                        ;
U 0CBF, 3E10,15     ;8573              MOV WR[0] TO LS[TB]                        ; TB=94 (указатель ячейки кода на IB)
U 0CC0, 6514,15     ;8574              CLR LS[T10]                                ; T10=код на IB
U 0CC1, 0881,0C     ;8575              JSR [ASSERT.NA]                            ; ожидаемые и полученные данные не применяются
                    ;8576      LOOP.T9.1:
U 0CC2, E520,15     ;8577              CLR LS[PC]                                 ;
U 0CC3, B64C,15     ;8578              MOV LS[BIT6] TO WR[0]                      ;
U 0CC4, 4748,15     ;8579              BIS LS[BIT4] TO WR[0]
```

```
U 0CC5, BE14,15 ;8580            MOV WR[0] TO LS[T10]                    ; код=00000050(H)
U 0CC6, 1910,75 ;8581            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0CC7, B214,15 ;8582            WRITE.MEM LS[T10]                       ; запись кода в ячейку 94 памяти
U 0CC8, 1B11,75 ;8583            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
                ;8584            OPC2/DECODE,BPC/NOP,IFUNC/0,IB.REQ/1,DEC.ADRS/0,LD.OS/1,R.DST/1,
U 0CC9, 0401,6E ;8585   OPC.SPEC/0,JCTL/NO.JUMP.TST  ;
U 0CCA, DB00,15 ;8586            NOP                                     ;
U 0CCB, 5B00,06 ;8587            SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0CCC, 5B00,1E ;8588            SKIP                                    ;
U 0CCD, 88CD,24 ;8589            JMP [T9.2]                              ; продолжение, если правильно
U 0CCE, 2F80,15 ;8590            CLR WR[0]                               ;
U 0CCF, 369E,95 ;8591            MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0CD0, 0869,3C ;8592            JSR [CHECK.RESULT]                      ; сообщение об ошибке 1
U 0CD1, 08CC,24 ;8593            JMP [LOOP.T9.1]                         ; зацикливание при ошибке, если разрешено
                ;8594    T9.2:
U 0CD2, FF82,15 ;8595            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 2
                ;8596    LOOP.T9.2.0001:
U 0CD3, 3648,15 ;8597            MOV LS[BIT4] TO WR[0]                   ;
U 0CD4, BE14,15 ;8598            MOV WR[0] TO LS[T10]                    ; код=00000010(H)
U 0CD5, 1910,75 ;8599            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0CD6, B214,15 ;8600            WRITE.MEM LS[T10]                       ; запись кода в ячейку памяти 94
U 0CD7, 1B11,75 ;8601            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0CD8, 8400,6E ;8602            OPC2/DECODE,BPC/NOP,DEC.ADRS/0,IFUNC/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0CD9, DB00,15 ;8603            NOP                                     ;
U 0CDA, 5B00,06 ;8604            SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0CDB, 08CE,04 ;8605            JMP [LOOP.T9.2.0100]                    ; продолжение, если правильно
U 0CDC, 2F80,15 ;8606            CLR WR[0]                               ;
U 0CDD, 369E,95 ;8607            MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0CDE, 0869,3C ;8608            JSR [CHECK.RESULT]                      ; сообщение об ошибке 2
U 0CDF, 08CD,34 ;8609            JMP [LOOP.T9.2.0001]                    ; зацикливание при ошибке, если разрешено
                ;8610    LOOP.T9.2.0100:
U 0CE0, B64C,15 ;8611            MOV LS[BIT6] TO WR[0]                   ;
U 0CE1, BE14,15 ;8612            MOV WR[0] TO LS[T10]                    ; код=00000040(H)
U 0CE2, 1910,75 ;8613            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0CE3, B214,15 ;8614            WRITE.MEM LS[T10]                       ; запись кода в ячейку паммяти 94
U 0CE4, 1B11,75 ;8615            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0CE5, 8400,6E ;8616            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST
U 0CE6, DB00,15 ;8617            NOP                                     ;
U 0CE7, 5B00,06 ;8618            SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0CE8, 88CE,D4 ;8619            JMP [LOOP.T9.2.0111]                    ; продолжение, если правильно
U 0CE9, 2F80,15 ;8620            CLR WR[0]                               ;
U 0CEA, 369E,95 ;8621            MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0CEB, 0869,3C ;8622            JSR [CHECK.RESULT]                      ; сообщение об ошибке 2
U 0CEC, 08CD,34 ;8623            JMP [LOOP.T9.2.0001]                    ; зацикливание при ошибке, если разрешено
                ;8624    LOOP.T9.2.0111:
U 0CED, B64C,15 ;8625            MOV LS[BIT6] TO WR[0]                   ;
U 0CEE, C74A,15 ;8626            BIS LS[BIT5] TO WR[0]                   ;
U 0CEF, 4748,15 ;8627            BIS LS[BIT4] TO WR[0]                   ;
U 0CF0, BE14,15 ;8628            MOV WR[0] TO LS[T10]                    ; код=00000070(H)
U 0CF1, 1910,75 ;8629            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0CF2, B214,15 ;8630            WRITE.MEM LS[T10]                       ; запись кода в ячейку памяти 94
U 0CF3, 1B11,75 ;8631            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0CF4, 8400,6E ;8632            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0CF5, DB00,15 ;8633            NOP                                     ;
U 0CF6, 5B00,06 ;8634            SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
```

```
U 0CF7, 88CF,C4 ;8635           JMP [LOOP.T9.2.1101]                    ; продолжение, если правильно
U 0CF8, 2F80,15 ;8636           CLR WR[0]                               ;
U 0CF9, 369E,95 ;8637           MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0CFA, 0869,3C ;8638           JSR [CHECK.RESULT]                      ; сообщение об ошибке 2
U 0CFB, 08CD,34 ;8639           JMP [LOOP.T9.2.0001]                    ; зацикливание при ошибке, если разрешено
                ;8640    LOOP.T9.2.1101:
U 0CFC, B64C,15 ;8641           MOV LS[BIT6] TO WR[0]                   ;
U 0CFD, 474E,15 ;8642           BIS LS[BIT7] TO WR[0]                   ;
U 0CFE, 4748,15 ;8643           BIS LS[BIT4] TO WR[0]                   ;
U 0CFF, BE14,15 ;8644           MOV WR[0] TO LS[T10]                    ; код=00000000
U 0D00, 1910,75 ;8645           MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0D01, B214,15 ;8646           WRITE.MEM LS[T10]                       ; запись кода в ячейку памяти 94
U 0D02, 1B11,75 ;8647           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0D03, 8400,6E ;8648           OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0D04, DB00,15 ;8649           NOP                                     ;
U 0D05, 5B00,06 ;8650           SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0D06, 88D0,B4 ;8651           JMP [T9.3]                              ; продолжение, если правильно
U 0D07, 2F80,15 ;8652           CLR WR[0]                               ;
U 0D08, 369E,95 ;8653           MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0D09, 0869,3C ;8654           JSR [CHECK.RESULT]                      ; сообщение об ошибке 2
U 0D0A, 08CD,34 ;8655           JMP [LOOP.T9.2.0001]                    ; зацикливание при ошибке, если разрешено
                ;8656    T9.3:
U 0D0B, FF82,15 ;8657           INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 3
U 0D0C, B724,15 ;8658           MOV LS[HI.IPL] TO WR[0]                 ;
U 0D0D, 477E,15 ;8659           BIS LS[BIT31] TO WR[0]                  ;
U 0D0E, BFFE,15 ;8660           MOV WR[0] TO LS[PSL.HW]                 ; установка режима совместимости
                ;8661    LOOP.T9.3.001:
U 0D0F, B646,15 ;8662           MOV LS[BIT3] TO WR[0]                   ;
U 0D10, BE14,15 ;8663           MOV WR[0] TO LS[T10]                    ; код=0000000B(H)
U 0D11, 1910,75 ;8664           MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0D12, B214,15 ;8665           WRITE.MEM LS[T10]                       ; запись кода в ячейку памяти 94
U 0D13, 1B11,75 ;8666           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0D14, 8400,6E ;8667           OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0D15, DB00,15 ;8668           NOP                                     ;
U 0D16, 5B00,06 ;8669           SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0D17, 88D1,C4 ;8670           JMP [LOOP.T9.3.010]                     ; продолжение, если правильно
U 0D18, 2F80,15 ;8671           CLR WR[0]                               ;
U 0D19, 369E,95 ;8672           MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0D1A, 0869,3C ;8673           JSR [CHECK.RESULT]                      ; сообщение об ошибке 3
U 0D1B, 08D0,F4 ;8674           JMP [LOOP.T9.3.001]                     ; зацикливание при ошибке, если разрешено
                ;8675    LOOP.T9.3.010:
U 0D1C, 3648,15 ;8676           MOV LS[BIT4] TO WR[0]                   ;
U 0D1D, BE14,15 ;8677           MOV WR[0] TO LS[T10]                    ; код=00000010(H)
U 0D1E, 1910,75 ;8678           MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0D1F, B214,15 ;8679           WRITE.MEM LS[T10]                       ; запись кода в ячейку памяти 94
U 0D20, 1B11,75 ;8680           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; пересылка кода на шину IB
U 0D21, 8400,6E ;8681           OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0D22, DB00,15 ;8682           NOP                                     ;
U 0D23, 5B00,06 ;8683           SKIP.IF[R.DST]                          ; пропуск, если RDEST разрешен
U 0D24, 88D2,94 ;8684           JMP [LOOP.T9.3.100]                     ; продолжение, если правильно
U 0D25, 2F80,15 ;8685           CLR WR[0]                               ;
U 0D26, 369E,95 ;8686           MOV LS[ONES] TO WR[1]                   ; установка ошибочных данных для индикации ошибки
U 0D27, 0869,3C ;8687           JSR [CHECK.RESULT]                      ; сообщение об ошибке 3
U 0D28, 88D1,C4 ;8688           JMP [LOOP.T9.3.010]                     ; зацикливание при ошибке, если разрешено
                ;8689    LOOP.T9.3.100:
```

```
U 0D29,  B646,15 ;8690            MOV LS[BIT3] TO WR[0]                    ;
U 0D2A,  BE14,15 ;8691            MOV WR[0] TO LS[T10]                     ; код=00000020(H)
U 0D2B,  1910,75 ;8692            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0D2C,  B214,15 ;8693            WRITE.MEM LS[T10]                        ; запись кода в ячейку памяти 94
U 0D2D,  1B11,75 ;8694            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; пересылка кода на шину IB
U 0D2E,  8400,6E ;8695            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0D2F,  DB00,15 ;8696            NOP                                      ;
U 0D30,  5B00,06 ;8697            SKIP.IF[R.DST]                           ; пропуск, если RDEST разрешен
U 0D31,  08D3,64 ;8698            JMP [T9.4]                               ; продолжение, если разрешено
U 0D32,  2FB0,15 ;8699            CLR WR[0]                                ;
U 0D33,  369E,95 ;8700            MOV LS[ONES] TO WR[1]                    ; установка ошибочных данных для индикации ошибки
U 0D34,  0869,3C ;8701            JSR [CHECK.RESULT]                       ; сообщение об ошибке 3
U 0D35,  88D2,94 ;8702            JMP [LOOP.T9.3.100]                      ; зацикливание при ошибке, если разрешено
                 ;8703    T9.4:
U 0D36,  FFB2,15 ;8704            INC LS[ERROR.NUMBER]                     ; номер ошибки=4
                 ;8705    LOOP.T9.4:
U 0D37,  2FB0,15 ;8706            CLR WR[0]                                ;
U 0D38,  BE14,15 ;8707            MOV WR[0] TO LS[T10]                     ; код=00000000
U 0D39,  1910,75 ;8708            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]       ;
U 0D3A,  B214,15 ;8709            WRITE.MEM LS[T10]                        ; запись кода в ячейку памяти 94
U 0D3B,  1B11,75 ;8710            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]       ; пересылка кода на шину IB
U 0D3C,  8400,6E ;8711            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/0,LD.OS/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0D3D,  DB00,15 ;8712            NOP                                      ;
U 0D3E,  5B00,06 ;8713            SKIP.IF[R.DST]                           ; пропуск, если RDEST разрешен
U 0D3F,  5B00,1E ;8714            SKIP                                     ;
U 0D40,  88D4,54 ;8715            JMP [END.T9]                             ; продолжение, если правильно
U 0D41,  2FB0,15 ;8716            CLR WR[0]                                ;
U 0D42,  369E,95 ;8717            MOV LS[ONES] TO WR[1]                    ; установка ошибочных данных для индикации ошибки
U 0D43,  0869,3C ;8718            JSR [CHECK.RESULT]                       ; сообщение об ошибке 4
U 0D44,  88D3,74 ;8719            JMP [LOOP.T9.4]                          ; зацикливание при ошибке, если разрешено
                 ;8720    END.T9:
```

```
;8721      .PAGE "ТЕСТ А - тест управления регистром OS (модуль DAP)"
;8722      ;
;8723      ; ОПИСАНИЕ ТЕСТА:
;8724      ;
;8725      ;   Это тест проверяет, что сигналы OS CTL 1 и OS CTL 0 не застревают в состо-
;8726      ;   янии единицы или нуля. Инструкция DECODE с установленным битом LOAD OS и
;8727      ;   сброшенным битом LOAD RDEST устанавливает низкий уровень сигналов OS CTL 1
;8728      ;   и OS CTL 0. При этом регистр OS должен загружаться с шины IB. Если регистр
;8729      ;   OS содержит то, что было в регистре предвыборки инструкций, это означает,
;8730      ;   что сигналы OS CTL 1 и OS CTL 0 выполняют ожидаемые функции при низком уров-
;8731      ;   не. Операция MOV WR[0] TO LS[OS] вызывает установку сигналов OS CTL 1 и OS
;8732      ;   CTL 0 в единичное состояние. Это вызывает загрузку регистра OS содержимым
;8733      ;   WR[0]. Инструкция DECODE с установленными RDEST и LOAD OS устанавливает сиг-
;8734      ;   нал OS CTL 0 в нулевое состояние, а OS CTL 1 в единичное состояние, и ре-
;8735      ;   гистр OS будет загружен с шины IB. Для установки сигнала OS CTL 1 в нулевое
;8736      ;   состояние, а OS CTL 0 в единичное состояние, выполняется инструкция MOV с
;8737      ;   установленным битом 31 для установки высокого уровня сигнала COMPAT MODE.
;8738      ;   Следующая инструкция DECODE устанавливает высокий уровень сигнала CM IRD.
;8739      ;   Следующая инструкция MOV загружает регистр OS содержимым WR[0] и устанавли-
;8740      ;   вает сигнал OS CTL 1 в нулевое состояние, а OS CTL 0 в единичное состояние.
;8741      ;   Если все инструкции выполняются как ожидается, то сигналы OS CTL 0 и OS CTL
;8742      ;   1 можно считать проверенными на застревание в состоянии единицы и в состоя-
;8743      ;   нии нуля.
;8744      ;
;8745      ; ПРЕДПОЛОЖЕНИЯ:
;8746      ;
;8747      ;   Предполагается, что регистр OS, шина IB и регистр предвыборки инструкций
;8748      ;   были проверены и работают правильно. Также основная память должна содер-
;8749      ;   жать единицы и WR[2] должен содержать нули.
;8750      ;
;8751      ; ШАГИ ТЕСТА:
;8752      ;
;8753      ;   1) Инструкция DECODE со сброшенным RDEST и установленным LOAD OS вызывает
;8754      ;      установку сигналов OS CTL 0 и OS CTL 1 в нулевое состояние и регистр OS
;8755      ;      загружается с шины IB.
;8756      ;   2) Регистр OS переписывается в WR[0], WR[1] загружается единицами и вызы-
;8757      ;      вается программа проверки результата.
;8758      ;   3) Если имеется ошибка и разрешено зацикливание при ошибке, тогда выполняет-
;8759      ;      ся возврат к шагу 1.
;8760      ;   4) Иначе выдается инструкция MOV WR[2] TO LS[OS], которая вызывает установку
;8761      ;      сигналов OS CTL 1 и OS CTL 0 в единичное состояние и регистр OS загружа-
;8762      ;      ется из WR[2].
;8763      ;   5) Регистр OS переписывается в WR[0], а WR[1] загружается нулями и вызывает-
;8764      ;      ся программа проверки результата.
;8765      ;   6) Если имеется ошибка, и разрешено зацикливание при ошибке, тогда выполня-
;8766      ;      ется возврат к шагу 4.
;8767      ;   7) Иначе выдается инструкция DECODE с установленными RDEST и LOAD OS и вызы-
;8768      ;      вает установку сигнала OS CTL 1 в единичное состояние, OS CTL 0 в нулевое
;8769      ;      состояние, а регистр OS загружается с шины IB.
;8770      ;   8) Регистр OS переписывается в WR[0], а WR[1] загружается единицами и вызы-
;8771      ;      вается программа проверки результата.
;8772      ;   9) Если имеется ошибка, и разрешено зацикливание при ошибке, тогда выполня-
;8773      ;      ется возврат к шагу 7.
;8774      ;  10) Иначе выдается инструкция MOV с установленным битом 31 для установки вы-
;8775      ;      сокого уровня сигнала COMPAT MODE, затем инструкция DECODE устанавливает
```

```
                ;8776       ;       высокий уровень сигнала CM IRD для следующего цикла и тогда инструкция
                ;8777       ;       MOV WR[2] TO LS[OS], которая устанавливает сигнал OS CTL 1 в нулевое сос-
                ;8778       ;       тояние, а OS CTL 0 - в единичное состояние, загружает регистр OS содержи-
                ;8779       ;       мым WR[2].
                ;8780       ;    11) Регистр OS переписывается в WR[0], а WR[1] загружается нулями и вызывает-
                ;8781       ;       ся программа проверки результата.
                ;8782       ;    12) Если имеется ошибка и зацикливание при ошибке, тогда выполняется возврат
                ;8783       ;       к шагу 10.
                ;8784       ;
                ;8785       ; ОШИБКИ:
                ;8786       ;
                ;8787       ;   ошибка 1 - или сигнал OS CTL 1=1, или сигнал OS CTL 0=1.
                ;8788       ;   ошибка 2 - или сигнал OS CTL 1=0, или сигнал OS CTL 0=0.
                ;8789       ;   ошибка 3 - или сигнал OS CTL 1=0, или сигнал OS CTL 0=1.
                ;8790       ;   ошибка 4 - или сигнал OS CTL 1=1, или сигнал OS CTL 0=0.
                ;8791       ;
                ;8792       ; НАЛАДКА:
                ;8793       ;
                ;8794       ;   ОШИБКА 1: Если ПМЛ РЕЖ.СОВМЕСТ.,РЕГ. АДРЕСАЦИЯ проверена, то либо
                ;8795       ;             сигнал OS CTL 1 либо OS CTL 0 застряли в единичном состоянии, а
                ;8796       ;             выделение неисправности возможно при проверке ошибок 2-4 этого
                ;8797       ;             теста.
                ;8798       ;   ОШИБКА 2: Если предполагается, что ПМЛ РЕЖ.СОВМЕСТ., РЕГ. АДРЕСАЦИЯ
                ;8799       ;             проверена, то либо сигнал OS CTL 1, либо OS CTL 0 застряли
                ;8800       ;             в нулевом состоянии.
                ;8801       ;   ОШИБКА 3: Снова, если ПМЛ РЕЖ.СОВМЕСТ.,РЕГ.АДРЕСАЦИЯ проверена,
                ;8802       ;             то или сигнал OS CTL 1 застревает в нулевом состоянии, или сигнал
                ;8803       ;             OS CTL 0 застревает в единичном состоянии.
                ;8804       ;   ОШИБКА 4: Следует убедиться, что уровень сигнала CM IRD высокий, а если нет,
                ;8805       ;             то необходимо проверить ПМЛ РЕЖ. СОВМЕСТ.,РЕГ. АДРЕСАЦИЯ
                ;8806       ;             В противном случае, или сигнал OS CTL 1 застрял в единице,
                ;8807       ;             или сигнал OS CTL 0 застрял в нуле.
                ;8808       ;
                ;8809       T.A:
U 0D45, B65E,15 ;8810               MOV LS[BEGIN.TEST] TO WR[0]          ; установка бита 15 в WR0 для слова управления и
                ;8811                                                    ; состояния
U 0D46, 3E80,15 ;8812               MOV WR[0] TO LS[CONTROL.STATUS]      ; установка бита 15 в слове управления и состояния. Бит
                ;8813                                                    ; 15 указывает начало теста для конс. процессора
U 0D47, 10E0,15 ;8814               MISC [SET.CP.ATTN]                   ; выдача сигнала CPU ATTN для конс.процессора
                ;8815       WAIT.TA.0:
U 0D48, 08D4,84 ;8816               JMP [WAIT.TA.0]
                ;8817                                                    ; зацикливание для ожидания ответа от консольного
                ;8818                                                    ; процессора
U 0D49, B724,15 ;8819               MOV LS[HI.IPL] TO WR[0]              ;
U 0D4A, BFFE,15 ;8820               MOV WR[0] TO LS[PSL.HW]               ; сброс режима совместимости
U 0D4B, E58A,15 ;8821               CLR LS[ERROR.MASK]                   ; очистка маски ошибок
U 0D4C, 65A0,15 ;8822               CLR LS[PREVIOUS.ERROR]               ; очистка предыдущего номера ошибки
U 0D4D, B640,15 ;8823               MOV LS[#1] TO WR[0]                  ; установка 1 в WR0
U 0D4E, BE82,15 ;8824               MOV WR[0] TO LS[ERROR.NUMBER]        ; установка номера ошибки на 1
U 0D4F, A3C0,15 ;8825               ROL WR[0]                            ; установка 2 в WR0
U 0D50, 3EBC,15 ;8826               MOV WR[0] TO LS[MODULE.NUM]          ; установка кода модуля для модуля DAP
U 0D51, E520,15 ;8827               CLR LS[PC]                           ;
U 0D52, 364E,15 ;8828               MOV LS[#80] TO WR[0]                 ;
U 0D53, C048,15 ;8829               ADD LS[#10] TO WR[0]                 ;
U 0D54, 3E10,15 ;8830               MOV WR[0] TO LS[T8]                  ; T8=90 (указатель слова памяти с установленным битом 0 в
```

```
                ;8831                                                   ; каждом байте (01010101(H))
U 0D55, 1B11,75 ;8832           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; установка бита 0 шины IB
                ;8833   LOOP.TA.1:
U 0D56, E520,15 ;8834           CLR LS[PC]                              ;
U 0D57, B69E,15 ;8835           MOV LS[ONES] TO WR[0]                   ;
U 0D58, 3EF8,15 ;8836           MOV WR[0] TO LS[OS]                     ; предварительная загрузка регистра OS единицами
U 0D59, 8401,4E ;8837           OPC2/DECODE,IB.REQ/1,DEC.ADRS/0,OPC.SPEC/0,LD.OS/1,R.DST/0,JCTL/NO.JUMP.TST ;
U 0D5A, DB00,15 ;8838           NOP                                     ;
U 0D5B, B6F8,15 ;8839           MOV LS[OS] TO WR[0]                     ; чтение из регистра OS
U 0D5C, 3640,95 ;8840           MOV LS[BIT0] TO WR[1]                   ; ожидаемые данные = бит 0
U 0D5D, 0869,3C ;8841           JSR [CHECK.RESULT]                      ; проверка, что инструкция DECODE загружает регистр OS
                ;8842                                                   ; из шины IB
U 0D5E, 08D5,64 ;8843           JMP [LOOP.TA.1]                         ; зацикливание при ошибке, если разрешено
                ;8844   TA.2:
U 0D5F, FF82,15 ;8845           INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 2
                ;8846   LOOP.TA.2:
U 0D60, E520,15 ;8847           CLR LS[PC]                              ;
U 0D61, B69E,15 ;8848           MOV LS[ONES] TO WR[0]                   ;
U 0D62, 3EF8,15 ;8849           MOV WR[0] TO LS[OS]                     ; предварительная загрузка регистра OS единицами
U 0D63, B643,15 ;8850           MOV LS[BIT1] TO WR[2]                   ; установка бита 1 в WR2
U 0D64, BEF9,15 ;8851           MOV WR[2] TO LS[OS]                     ; загрузка регистра OS
U 0D65, B6F8,15 ;8852           MOV LS[OS] TO WR[0]                     ;
U 0D66, B642,95 ;8853           MOV LS[BIT1] TO WR[1]                   ; ожидаемые данные=бит 1 установлен
U 0D67, 0869,3C ;8854           JSR [CHECK.RESULT]                      ; проверка, что регистр OS не загружается с шины IB
U 0D68, 08D6,04 ;8855           JMP [LOOP.TA.2]                         ; зацикливание при ошибке, если разрешено
                ;8856   TA.3:
U 0D69, FF82,15 ;8857           INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 3
                ;8858   LOOP.TA.3:
U 0D6A, E520,15 ;8859           CLR LS[PC]                              ;
U 0D6B, B69E,15 ;8860           MOV LS[ONES] TO WR[0]                   ;
U 0D6C, 3EF8,15 ;8861           MOV WR[0] TO LS[OS]                     ; предварительная загрузка регистра OS единицами
U 0D6D, 0401,6E ;8862           OPC2/DECODE,IB.REQ/1,DEC.ADRS/0,OPC.SPEC/0,LD.OS/1,R.DST/1,JCTL/NO.JUMP.TST ;
U 0D6E, DB00,15 ;8863           NOP                                     ;
U 0D6F, B6F8,15 ;8864           MOV LS[OS] TO WR[0]                     ; выборка регистра OS
U 0D70, 3640,95 ;8865           MOV LS[BIT0] TO WR[1]                   ; ожидаемые данные=бит 0 установлен
U 0D71, 0869,3C ;8866           JSR [CHECK.RESULT]                      ; проверка, что регистр OS загружен из шины IB
U 0D72, 08D6,A4 ;8867           JMP [LOOP.TA.3]                         ; зацикливание при ош.  ке, если разрешено
                ;8868   TA.4:
U 0D73, FF82,15 ;8869           INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 4
U 0D74, B724,15 ;8870           MOV LS[HI.IPL] TO WR[0]                 ;
U 0D75, 477E,15 ;8871           BIS LS[BIT31] TO WR[0]                  ;
U 0D76, BFFE,15 ;8872           MOV WR[0] TO LS[PSL.HW]                 ; установка режима совместимости
                ;8873   LOOP.TA.4:
U 0D77, E520,15 ;8874           CLR LS[PC]                              ;
U 0D78, B69E,15 ;8875           MOV LS[ONES] TO WR[0]                   ;
U 0D79, 3EF8,15 ;8876           MOV WR[0] TO LS[OS]                     ; предварительная загрузка регистра OS единицами
                ;8877    ; Эта инструкция DECODE устанавливает IRD на один цикл
U 0D7A, 0403,1E ;8878           OPC2/DECODE,DEC.ADRS/0,IFUNC/1,IB.REQ/1,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0D7B, DB00,15 ;8879           NOP                                     ;
U 0D7C, BEF9,15 ;8880           MOV WR[2] TO LS[OS]                     ; загрузка регистра OS
U 0D7D, B6F8,15 ;8881           MOV LS[OS] TO WR[0]                     ;
U 0D7E, 2F82,95 ;8882           CLR WR[1]                               ; ожидаемые данные = очищены (WR2 не был загружен в OS)
U 0D7F, 0869,3C ;8883           JSR [CHECK.RESULT]                      ; проверка, что регистр OS не был загружен из шины IB
U 0D80, 08D7,74 ;8884           JMP [LOOP.TA.4]                         ; зацикливание при ошибке, если разрешено
                ;8885   CLEANUP.TA:
```

```
U 0D81, B724,15 ;8886            MOV LS[HI.IPL] TO WR[0]                ;
U 0D82, BFFE,15 ;8887            MOV WR[0] TO LS[PSL.HW]                ; сброс режима совместимости
                ;8888    END.TA:
```

```
;8889      .PAGE "ТЕСТ В - тест битов регистра OS (модуль DAP)"
;8890      ;**
;8891      ;
;8892      ; ОПИСАНИЕ ТЕСТА:
;8893      ;
;8894      ;   Целью этого теста является проверка содержимого регистра OS при различных сиг-
;8895      ;   налах OS CTL и различных данных на шине IB. Когда OS CTL=0, тогда регистр OS
;8896      ;   загружается из шины IB. Это уже было проверено распространением единицы в поле
;8897      ;   нулей. Когда OS CTL=1, тогда OS 0 принимает значение IB 6, OS 1 принимает IB 7,
;8898      ;   OS 2 принимает OS 0, OS 3 принимает 0, OS 4 принимает OS 2, OS 5 принимает OS 3,
;8899      ;   OS 6 принимает OS 4 и OS 7 принимает OS 5. Для проверки регистра OS при этих обстоя-
;8900      ;   тельствах задаются сдвигаемые данные на шине IB. В случае, когда OS CTL=2, ис-
;8901      ;   пользуются те же наборы, как и для проверки регистра OS при OS CTL=1.
;8902      ;
;8903      ; ПРЕДПОЛОЖЕНИЯ:
;8904      ;
;8905      ;   Предполагается, что тест управления OS выполнен успешно, и что шина IB, ре-
;8906      ;   гистр OS и регистр предвыборки инструкций были проверены и работают правильно.
;8907      ;   Память должна быть загружена данными со сдвигаемой единицей, которые загружа-
;8908      ;   ются в регистр предвыборки инструкций.
;8909      ;
;8910      ; ШАГИ ТЕСТА:
;8911      ;
;8912      ;   1)Выдача инструкции MOV с установленным битом 31 для установки высокого уро-
;8913      ;     вня сигнала COMPAT MODE.
;8914      ;   2)Загрузка шины IB данными со сдвигаемой единицей, затем выполнение инструк-
;8915      ;     ции DECODE для установки CM IRD, что приводит к установке OS CTL=1.
;8916      ;   3)Чтение регистра OS в WR0, загрузка ожидаемых данных в WR1 и вызов подпрог-
;8917      ;     раммы проверки результата. Если все сдвигаемые наборы проверены, тогда пе-
;8918      ;     реход к шагу 4, иначе к шагу 1.
;8919      ;   4)Загрузка шины IB данными со сдвигаемой единицей и предварительная загрузка
;8920      ;     регистра OS всеми единицами.
;8921      ;   5)Выдача инструкции DECODE с установленными RDEST и LOAD OS, которые устанав-
;8922      ;     ливают OS CTL=2 и регистр OS загружается из шины IB.
;8923      ;   6)Пересылка содержимого регистра OS в WR0, загрузка ожидаемых данных в WR1
;8924      ;     и вызов программы для проверки результата. Если все сдвигаемые наборы прове-
;8925      ;     рены, тогда переход к концу теста, иначе выборка следующего набора и переход
;8926      ;     к шагу 4.
;8927      ;
;8928      ; ОШИБКИ:
;8929      ;
;8930      ;   ошибка 1 - неправильно работает регистр OS, когда OS CTL=1.
;8931      ;   ошибка 2 - неправильно работает регистр OS, когда OS CTL=2.
;8932      ;
;8933      ; НАЛАДКА:
;8934      ;
;8935      ;   ОШИБКА 1: Если появляется эта ошибка, она означает, что не прошел один из ис-
;8936      ;             пользуемых наборов для проверки регистра OS при OS CTL=1. Исследо-
;8937      ;             ванием ожидаемых и полученных данных можно определить набор, при ко-
;8938      ;             тором схемы не срабатывают, и где имеется отказ. Так как схемы OS
;8939      ;             CTL были проверены раньше, скорее всего, источником ошибки может быть
;8940      ;             ПМЛ OS[7:1] НЕЧЕТН , если отказ в нечетном бите, или ПМЛ OS[6:0] ЧЕТН,
;8941      ;             если отказ в четном бите.
;8942      ;   ОШИБКА 2: Если появляется эта ошибка, она означает, что не прошел один из ис-
;8943      ;             пользуемых наборов для проверки регистра OS при OS CTL=2. Исследова-
```

```
                    ;8944     ;             нием ожидаемых и полученых данных можно определить набор, при кото-
                    ;8945     ;             ром выявлен отказ, и где он имеется. Так как схемы OS CTL были про-
                    ;8946     ;             верены раньше, скорее всего, источником ошибки может быть ПМЛ OS
                    ;8947     ;             [7:1] НЕЧЕТН, если отказ в нечетном бите, или ПМЛ OS[6:0] ЧЕТН., если
                    ;8948     ;             отказ в четном бите.
                    ;8949     ;
                    ;8950     ;
                    ;8951     T.B:
U 0D83, B65E,15     ;8952             MOV LS[BEGIN.TEST] TO WR[0]                   ; установка бита 15 в WR0 для слова управления и
                    ;8953                                                           ; состояния
U 0D84, 3E80,15     ;8954             MOV WR[0] TO LS[CONTROL.STATUS]               ; установка бита 15 в слове управления и состояния. Бит 15
                    ;8955                                                           ; указывает начало теста для консольного процессора
U 0D85, 10E0,15     ;8956             MISC [SET.CP.ATTN]                            ; выдача сигнала CPU ATTN для конс. процессора
                    ;8957     WAIT.TB.0:
U 0D86, 88D8,64     ;8958             JMP [WAIT.TB.0]                               ; зацикливание для ожидания ответа от конс. процессора
U 0D87, DF26,15     ;8959             MCOM LS[#FF] TO WR[0]                         ;
U 0D88, 3E8A,15     ;8960             MOV WR[0] TO LS[ERROR.MASK]                   ; маска ошибки=FFFFFF00
U 0D89, 65A0,15     ;8961             CLR LS[PREVIOUS.ERROR]                        ; очистка предыдущего номера ошибки
U 0D8A, B640,15     ;8962             MOV LS[#1] TO WR[0]                           ; установка 1 в WR0
U 0D8B, BEB2,15     ;8963             MOV WR[0] TO LS[ERROR.NUMBER]                 ; установка номера ошибки для первой ошибки
U 0D8C, A3C0,15     ;8964             ROL WR[0]                                     ; установка 2 в WR0
U 0D8D, 3E8C,15     ;8965             MOV WR[0] TO LS[MODULE.NUM]                   ; установка кода модуля для модуля DAP
U 0D8E, 364E,15     ;8966             MOV LS[#80] TO WR[0]                          ;
U 0D8F, C048,15     ;8967             ADD LS[#10] TO WR[0]                          ;
U 0D90, C044,15     ;8968             ADD LS[#4] TO WR[0]                           ;
U 0D91, 3E10,15     ;8969             MOV WR[0] TO LS[TB]                           ; TB=94 (указатель слов памяти)
U 0D92, B651,15     ;8970             MOV LS[BITB] TO WR[2]                         ; установка начального набора в WR2
U 0D93, B724,15     ;8971             MOV LS[HI.IPL] TO WR[0]                       ;
U 0D94, 477E,15     ;8972             BIS LS[BIT31] TO WR[0]                        ;
U 0D95, BFFE,15     ;8973             MOV WR[0] TO LS[PSL.HW]                       ; установка режима совместимости
                    ;8974     LOOP.TB.1:
U 0D96, E520,15     ;8975             CLR LS[PC]                                    ;
U 0D97, 3E15,15     ;8976             MOV WR[2] TO LS[T10]                          ; запись набора в местную память
U 0D98, 1910,75     ;8977             MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]            ;
U 0D99, B214,15     ;8978             WRITE.MEM LS[T10]                             ; запись этого набора в ячейку памяти 94
U 0D9A, 1B11,75     ;8979             MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]            ; пересылка набора на шину IB
U 0D9B, B69E,15     ;8980             MOV LS[ONES] TO WR[0]                         ;
U 0D9C, 3EF8,15     ;8981             MOV WR[0] TO LS[OS]                           ; предварительное заполнение регистра OS
                    ;8982     ;
                    ;8983     ;  Следующая инструкция DECODE устанавливает OS CTL=01(B)
                    ;8984     ;
U 0D9D, 0403,1E     ;8985             OPC2/DECODE,IB.REQ/1,BPC/NOP,DEC.ADRS/0,IFUNC/1,OPC.SPEC/1,JCTL/NO.JUMP.TST ;
U 0D9E, DB00,15     ;8986             NOP                                           ;
U 0D9F, DB00,15     ;8987             NOP                                           ; выдержка времени для выборки измененного содержимого
                    ;8988                                                           ; OS
U 0DA0, B6F8,15     ;8989             MOV LS[OS] TO WR[0]                           ; выборка содержимого OS
U 0DA1, 88D8,DC     ;8990             JSR [TB.GEN.EX]                               ; формирование ожидаемых данных
U 0DA2, 0869,3C     ;8991             JSR [CHECK.RESULT]                            ; проверка OS
U 0DA3, 08D9,64     ;8992             JMP [LOOP.TB.1]                               ; зацикливание при ошибке, если разрешено
U 0DA4, 585E,15     ;8993             MOV LS[BIT15] TO Q                            ;
                    ;8994               CMP WR[2] WITH Q,                           ;
U 0DA5, A484,35     ;8995             DT(LONG)&SET.ALU.CC                           ; набор=00008000(H) ?
U 0DA6, 88DA,99     ;8996             JMP.IF[EQL] TO [TB.2]                         ; продолжение, если да
U 0DA7, 23C1,15     ;8997             ROL WR[2]                                     ; сдвиг единицы в наборе
U 0DA8, 08D9,64     ;8998             JMP [LOOP.TB.1]                               ; возврат к началу цикла
```

```
                ;8999    TB.2:
U 0DA9, FFB2,15 ;9000            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 2
U 0DAA, 3641,15 ;9001            MOV LS[BIT0] TO WR[2]                   ; восстановление набора в WR2
                ;9002    LOOP.TB.2:
U 0DAB, 3E15,15 ;9003            MOV WR[2] TO LS[T10]                    ; запись набора в местную память
U 0DAC, 1910,75 ;9004            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]      ;
U 0DAD, B214,15 ;9005            WRITE.MEM LS[T10]                       ; запись этого набора в ячейку памяти 94
U 0DAE, 1B11,75 ;9006            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]      ; пересылка набора на шину IB
U 0DAF, B69E,15 ;9007            MOV LS[ONES] TO WR[0]                   ;
U 0DB0, 3EF8,15 ;9008            MOV WR[0] TO LS[OS]                     ; предварительное заполнение регистра OS
                ;9009     ;
                ;9010     ; Следующая инструкция DECODE устанавливает OS CTL=10(B)
                ;9011     ;
U 0DB1, 8400,6E ;9012            OPC2/DECODE,DEC.ADRS/0,IFUNC/0,LD.OS/i,R.DST/i,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0DB2, DB00,15 ;9013            NOP                                     ;
U 0DB3, B6F8,15 ;9014            MOV LS[OS] TO WR[0]                     ; выборка OS
U 0DB4, A004,95 ;9015            MOV WR[2] TO WR[1]                      ;
U 0DB5, C546,95 ;9016            BIC LS[BIT3] TO WR[1]                   ; ожидаемые данные=набор со сдвигаемой единицей,
                ;9017                                                    ; за исключением сброшенного
                ;9018                                                    ; бита 3
U 0DB6, 0869,3C ;9019            JSR [CHECK.RESULT]                      ; проверка OS
U 0DB7, 88DA,B4 ;9020            JMP [LOOP.TB.2]                         ; зацикливание при ошибке, если разрешено
U 0DB8, 23C1,15 ;9021            ROL WR[2]                               ; сдвиг единицы в наборе
U 0DB9, DB50,15 ;9022            MOV LS[BIT8] TO Q                       ;
                ;9023              CMP WR[2] WITH Q,                     ;
U 0DBA, A484,35 ;9024            DT(LONG)&SET.ALU.CC                     ; набор=00000100(H) ?
U 0DBB, 88DA,B1 ;9025            JMP.IF[NEQ] TO [LOOP.TB.2]              ; возврат, если нет
U 0DBC, 08DC,94 ;9026            JMP [CLEANUP.TB]                        ;
                ;9027    TB.GEN.EX:
U 0DBD, 2005,95 ;9028            MOV WR[2] TO WR[3]                      ; WR3=текущий набор
U 0DBE, 369E,95 ;9029            MOV LS[ONES] TO WR[1]                   ;
U 0DBF, C540,95 ;9030            BIC LS[BIT0] TO WR[1]                   ;
U 0DC0, 4542,95 ;9031            BIC LS[BIT1] TO WR[1]                   ;
U 0DC1, C546,95 ;9032            BIC LS[BIT3] TO WR[1]                   ; очистка битов 3,1 и 0
                ;9033              BIT LS[BIT14] WITH WR[3],             ;
U 0DC2, D95D,B5 ;9034            DT(LONG)&SET.ALU.CC                     ; бит 14 в WR3 установлен?
U 0DC3, 5B00,09 ;9035            SKIP.IF[EQL]                            ; пропуск, если нет
U 0DC4, 4740,95 ;9036            BIS LS[BIT0] TO WR[1]                   ; установка бита 0 в WR1, если да
                ;9037              BIT LS[BIT15] WITH WR[3],             ;
U 0DC5, 595F,B5 ;9038            DT(LONG)&SET.ALU.CC                     ; бит 15 в WR3 установлен?
U 0DC6, 5B00,09 ;9039            SKIP.IF[EQL]                            ; пропуск, если нет
U 0DC7, C742,95 ;9040            BIS LS[BIT1] TO WR[1]                   ; установка бита 1 в WR1, если да
U 0DC8, 5B00,14 ;9041            RETURN                                  ;
                ;9042    CLEANUP.TB:
U 0DC9, B724,15 ;9043            MOV LS[HI.IPL] TO WR[0]                 ;
U 0DCA, BFFE,15 ;9044            MOV WR[0] TO LS[PSL.HW]                 ; сброс режима совместимости
                ;9045    END.TB:
```

```
                ;9046      .PAGE "ТЕСТ С - тест сигнала GPR DEST (модуль DAP)"
                ;9047      ;
                ;9048      ; ОПИСАНИЕ ТЕСТА:
                ;9049      ;
                ;9050      ;   Этот тест проверяет застревание сигнала GPR DEST (условия перехода или про-
                ;9051      ;   пуска микросекванссра) в логическом состоянии нуля или в логическом состо-
                ;9052      ;   янии единицы. Проверка сигнала состоит из двух частей: в первой части выпол-
                ;9053      ;   няется установка сигнала GPR DEST в единичное состояние, а во второй части
                ;9054      ;   проверяется, не застрял ли сигнал GPR DEST в состоянии логической единицы.
                ;9055      ;   Инструкцией для создания низкого уровня сигнала GPR DEST (возбужден) должна
                ;9056      ;   быть инструкция DECODE с установленным битом RDEST (CSR05) и должны быть ус-
                ;9057      ;   тановлены биты 4 и 6 или сброшены биты 5 и 7 шины IB. В данном случае ис-
                ;9058      ;   пользуется IB=0, те биты 5 и 7 сброшены. после того, как выдана инструкция
                ;9059      ;   DECODE для установки низкого сигнала GPR DEST, должно проверяться предпола-
                ;9060      ;   гаемое состояние сигнала GPR DEST. Инструкция JUMP содержит поле управления
                ;9061      ;   переходом, которое вызывает переход в определенное место управляющей памяти,
                ;9062      ;   если условие удовлетворено. Одним из таких условий является условие RDEST,
                ;9063      ;   которое вызывает переход, если сигнал GPR DEST установлен.
                ;9064      ;
                ;9065      ; ПРЕДПОЛОЖЕНИЯ:
                ;9066      ;
                ;9067      ;   Предполагается, что ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ, ПЗУ ТИП ДАННЫХ, КЛАСС КОДОВ УС-
                ;9068      ;   ЛОВИЙ и последующие схемы, которые используются для разрешения сигнала
                ;9069      ;   GPR DEST, проверены.
                ;9070      ;
                ;9071      ; ШАГИ ТЕСТА:
                ;9072      ;
                ;9073      ;   1.Установка данных 00(H) на шине IB.
                ;9074      ;   2.Выдача инструкции DECODE с установленным битом RDEST.
                ;9075      ;   3.Выдача инструкции JUMP со значением 6 (переход по RDEST) в поле управления
                ;9076      ;     переходом. Если перехода не было, выдается сообщение об ошибке 1.
                ;9077      ;   4.Установка новых данных на шине IB (30(H)).
                ;9078      ;   5.Выдача инструкции DECODE с установленным битом RDEST.
                ;9079      ;   6.Выдача инструкци JUMP со значением 6 в поле управления переходом. Переходе
                ;9080      ;     не должно быть, иначе выдается сообщение об ошибке 2.
                ;9081      ;
                ;9082      ; ОШИБКИ:
                ;9083      ;
                ;9084      ;   ошибка 1 - сингал GPR DEST застрял в логическом нуле или неправильно работает
                ;9085      ;              управление переходом.
                ;9086      ;   ошибка 2 - сигнал GPR DEST застрял в логической единице или неправильно рабо-
                ;9087      ;              тает управление переходом по RDEST.                6+
                ;9088      ;
                ;9089      ; НАЛАДКА:
                ;9090      ;
                ;9091      ;   ОШИБКИ 1-2: Схемы, формирующие сигнал REGISTER MODE, проверены тестом 9. Повто-
                ;9092      ;                му ошибки 1 и 2 втого теста могут произойти из-за неисправности мик-
                ;9093      ;                росхемы РЕЖ.СОВМЕСТ., РЕГ.АДРЕСАЦИЯ. Проверьте сигнал GPR DEST L
                ;9094      ;                на контакте 16.
                ;9095      T.C:
U 0DCB, B65E,15 ;9096              MOV LSI[BEGIN.TEST] TO WR[0]              ; установка бита 15 в WR0 для слова управления и
                ;9097                                                        ; состояния
U 0DCC, 3EB0,15 ;9098              MOV WR[0] TO LSI[CONTROL.STATUS]          ; установка бита 15 в слове управления и состояния. Бит
                ;9099                                                        ; 15 указывает начало теста для консольного процессора
U 0DCD, 10E0,15 ;9100              MISC [SET.CP.ATTN]                        ; выдача сигнала CPU ATTN для конс. процессора
```

```
                ;9101    WAIT.TC.0:
U 0DCE, 88DC,E4 ;9102            JMP [WAIT.TC.0]                    ; зацикливание для ожидания ответа от консольного
                ;9103                                               ; процессора
U 0DCF, B724,15 ;9104            MOV LS[HI.IPL] TO WR[0]            ;
U 0DD0, BFFE,15 ;9105            MOV WR[0] TO LS[PSL.HW]            ; сброс режима совместимости
U 0DD1, E58A,15 ;9106            CLR LS[ERROR.MASK]                 ; очистка маски ошибки
U 0DD2, 65A0,15 ;9107            CLR LS[PREVIOUS.ERROR]             ; очистка предыдущего номера ошибки
U 0DD3, B640,15 ;9108            MOV LS[#1] TO WR[0]                ; установка 1 в WR0
U 0DD4, BE82,15 ;9109            MOV WR[0] TO LS[ERROR.NUMBER]      ; установка номера ошибки для первой ошибки
U 0DD5, A3C0,15 ;9110            ROL WR[0]                          ; установка 2 в WR0
U 0DD6, 3EBC,15 ;9111            MOV WR[0] TO LS[MODULE.NUM]        ; установка кода модуля для модуля DAP
U 0DD7, 08B1,0C ;9112            JSR [ASSERT.NA]                    ;
U 0DD8, 364E,15 ;9113            MOV LS[#80] TO WR[0]               ;
U 0DD9, C048,15 ;9114            ADD LS[#10] TO WR[0]               ;
U 0DDA, C044,15 ;9115            ADD LS[#4] TO WR[0]                ;
U 0DDB, 3E1C,15 ;9116            MOV WR[0] TO LS[T8]                ; T8=94 (указатель ячейки памяти)
U 0DDC, 6514,15 ;9117            CLR LS[T10]                        ; очистка ячейки местной памяти
U 0DDD, 1910,75 ;9118            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG] ;
U 0DDE, B214,15 ;9119            WRITE.MEM LS[T10]                  ; очистка ячейки памяти 94
                ;9120    LOOP.TC.1:
U 0DDF, E520,15 ;9121            CLR LS[PC]                         ;
U 0DE0, 1B11,75 ;9122            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG] ; очистка IB
U 0DE1, 0471,2E ;9123            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/7,IB.REQ/1,R.DST/1,OPC.SPEC/0,JCTL/ 0.JUMP.TST ;
U 0DE2, DB00,15 ;9124            NOP                                ;
U 0DE3, 88DE,B6 ;9125            JMP.IF[R.DST] TO [TC.2]            ; переход, если RDEST установлен
U 0DE4, 2F80,15 ;9126            CLR WR[0]                          ; сообщение об ошибке 1, если нет
U 0DE5, 369E,95 ;9127            MOV LS[ONES] TO WR[1]              ;
U 0DE6, 0869,3C ;9128            JSR [CHECK.RESULT]                 ;
U 0DE7, 88DD,F4 ;9129            JMP [LOOP.TC.1]                    ;
                ;9130    TC.2:
U 0DE8, FF82,15 ;9131            INC LS[ERROR.NUMBER]               ; увеличение номера ошибки до 2
U 0DE9, 3648,15 ;9132            MOV LS[#10] TO WR[0]               ;
U 0DEA, 404A,15 ;9133            ADD LS[#20] TO WR[0]               ;
U 0DEB, 8E14,15 ;9134            MOV WR[0] TO LS[T10]               ; T10=30(H)
                ;9135    LOOP.TC.2:
U 0DEC, 1910,75 ;9136            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG] ;
U 0DED, B214,15 ;9137            WRITE.MEM LS[T10]                  ; ячейка 94 памяти=30(H
U 0DEE, E520,15 ;9138            CLR LS[PC]                         ;
U 0DEF, 1B11,75 ;9139            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG] ; IB=30(H)
U 0DF0, 0471,2E ;9140            OPC2/DECODE,BPC/NOP,IFUNC/0,DEC.ADRS/7,IB.REQ/1,R.DST/1,OPC.SPEC/0,JCTL/NO.JUMP.TST ;
U 0DF1, DB00,15 ;9141            NOP                                ;
U 0DF2, 08DF,46 ;9142            JMP.IF[R.DST] TO [TC.2.ERR]        ; переход, если RDEST все еще имеется
U 0DF3, 88DF,B4 ;9143            JMP [END.TC]                       ; переход по концу, если нет
                ;9144    TC.2.ERR:
U 0DF4, 2F80,15 ;9145            CLR WR[0]                          ;
U 0DF5, 369E,95 ;9146            MOV LS[ONES] TO WR[1]              ;
U 0DF6, 0869,3C ;9147            JSR [CHECK.RESULT]                 , сообщение об ошибке 2
U 0DF7, 88DE,C4 ;9148            JMP [LOOP.TC.2]                    ;
                ;9149    END.TC:
```

```
;9150      .PAGE "ТЕСТ D - тест признака маски резервной копии регистров (модуль DAP)"
;9151      ;**
;9152      ;
;9153      ;
;9154      ; ОПИСАНИЕ ТЕСТА:
;9155      ;
;9156      ;   Этот тест проверяет застревание сигнала RBKUP FLAG (признак маски резервной
;9157      ;   копии регистров) в состоянии логического нуля или единицы. Инструкция MISC
;9158      ;   устанавливает бит RBKUP FLAG. Состояние бита RBKUP FLAG может быть проверено
;9159      ;   функцией пропуска по признаку маски резервной копии регистра. Если пропуск
;9160      ;   выполняется, значит бит установлен так, как ожидается, и сигнал RBKUP FLAG не
;9161      ;   застрял в состоянии нуля. Бит RBKUP FLAG может быть сброшен инструкцией DECODE
;9162      ;   с установленными в 0 битами CSR 9 и 4. Для проверки сброшенного состояния сиг-
;9163      ;   нала RBKUP FLAG выполняется инструкция NOP с пропуском по признаку маски ре-
;9164      ;   зервной копии регистров. Если пропуск выполняется, значит сигнал RBKUP FLAG
;9165      ;   застрял в состоянии логической единицы.
;9166      ;
;9167      ; ПРЕДПОЛОЖЕНИЯ:
;9168      ;
;9169      ;   Предполагается, что шина IB проверена и работает.
;9170      ;
;9171      ; ШАГИ ТЕСТА:
;9172      ;
;9173      ;   1) Начальная установка сигнала RBKUP FLAG в нулевое состояние инструкцией
;9174      ;      DECODE с установленными битами 9 и 4.
;9175      ;   2) Выдача инструкции MISC с полем MF1=7 для установки сигнала RBKUP FLAG.
;9176      ;   3) Используя условие пропуска по признаку маски резервной копии регистров и
;9177      ;      инструкцию NOP, можно определить состояние сигнала RBKUP FLAG (высокий
;9178      ;      или низкий уровень).
;9179      ;   4) Если пропуск произошел, тогда ошибки нет, поэтому будет проверяться низкий
;9180      ;      уровень сигнала RBKUP FLAG посредством выдачи инструкции DECODE с устано-
;9181      ;      вленными в CSR битами 9 и 4.
;9182      ;   5) Низкий уровень сигнала RBKUP FLAG может быть проверен посредством выдачи
;9183      ;      инструкции NOP с условием пропуска по признаку маски резервной копии ре-
;9184      ;      гистров. Проверяется отсутствие пропуска.
;9185      ;
;9186      ; ОШИБКИ:
;9187      ;
;9188      ;   ошибка 1 - неисправность в схемах формирования высокого уровня сигнала
;9189      ;              RBKUP FLAG или в схемах пропуска.
;9190      ;   ошибка 2 - неисправность в схемах формирования низкого уровня сигнала
;9191      ;              RBKUP FLAG или в схемах пропуска.
;9192      ;
;9193      ; НАЛАДКА:
;9194      ;
;9195      ;   ОШИБКА 1: Если сигнал RBKUP FLAG не становится высоким, тогда или сам сигнал
;9196      ;             застрял на низком уровне, или неисправна схема установки сигнала,
;9197      ;             или неисправна схема управления пропуском. Вероятнее всего, источ-
;9198      ;             никами ошибки являются застревание сигнала RBKUP FLAG на низком
;9199      ;             уровне или неисправность в схеме формирования этого сигнала. Полез-
;9200      ;             но проверить ПМЛ РЕЖ.СОВМЕСТ., РЕГ. АДРЕСАЦИЯ и сигнал
;9201      ;             SET RBKUP. Если это место исправно, тогда необходимо проверить сиг-
;9202      ;             нал RBKUP FLAG, поступающий на вход ПМЛ УПР.МУЛЬТИПЛ. и ИНКРЕМЕНТ.
;9203      ;             Этот вход должен иметь высокий уровень, а на входах CSR 00 - 04
;9204      ;             должно быть 1B(H). Сигнал JUMP INSTR должен иметь высокий уровень,
```

```
                    ;9205     ;              а выход на контакте 12 должен иметь низкий уровень.
                    ;9206     ;    ОШИБКА 2: Если сигнал RBKUP FLAG не становится низким, тогда или сам сигнал
                    ;9207     ;              застрял в высоком уровне, или неисправна схема формирования сигнала,
                    ;9208     ;              или неправильно работает схема управления пропуском. Необходимо про-
                    ;9209     ;              верить ПМЛ РЕЖ.СОВМЕСТИМОСТИ, РЕГ. АДРЕСАЦИЯ и сигнал SET
                    ;9210     ;              RBKUP. Если это место исправно, тогда, повидимому, сигнал RBKUP FLAG
                    ;9211     ;              застрял в состоянии нуля. Если нет, необходимо проверить ПМЛ УПР.
                    ;9212     ;              МУЛЬТИПЛ. и ИНКРЕМЕНТ. Выходы должны быть такими, какие описаны для
                    ;9213     ;              ошибки 1, за исключением того, что сигнал RBKUP FLAG должен быть низ-
                    ;9214     ;              ким и выходы 12 и 19 должны быть высокими.
                    ;9215     ;
                    ;9216     ;--
                    ;9217     T.D:
U 0DF8,  B65E,15    ;9218             MOV LS[BEGIN.TEST] TO WR[0]                    ; установка бита 15 в WR0 для слова управления и
                    ;9219                                                            ; состояния
U 0DF9,  3E80,15    ;9220             MOV WR[0] TO LS[CONTROL.STATUS]                ; установка бита 15 в слове управления и состояния. Бит
                    ;9221                                                            ; 15 указывает начало теста для консольного процессора
U 0DFA,  10E0,15    ;9222             MISC [SET.CP.ATTN]                             ; выдача сигнала CPU ATTN для консольного процессора
                    ;9223     WAIT.TD.0:
U 0DFB,  88DF,B4    ;9224             JMP [WAIT.TD.0]                                ; зацикливание для ожидания ответа от консольного
                    ;9225                                                            ; процессора
U 0DFC,  B724,15    ;9226             MOV LS[HI.IPL] TO WR[0]                        ;
U 0DFD,  BFFE,15    ;9227             MOV WR[0] TO LS[PSL.HW]                        ; сброс режима совместимости
U 0DFE,  E58A,15    ;9228             CLR LS[ERROR.MASK]                             ; очистка маски ошибки
U 0DFF,  65A0,15    ;9229             CLR LS[PREVIOUS.ERROR]                         ; очистка предыдущего номера ошибки
U 0E00,  B640,15    ;9230             MOV LS[#1] TO WR[0]                            ; установка 1 в WR0
U 0E01,  BEB2,15    ;9231             MOV WR[0] TO LS[ERROR.NUMBER]                  ; установка номера ошибки для первой ошибки
U 0E02,  A3C0,15    ;9232             ROL WR[0]                                      ; установка 2 в WR0
U 0E03,  3E8C,15    ;9233             MOV WR[0] TO LS[MODULE.NUM]                    ; установка кода для модуля DAP
  0E04,  0881,0C    ;9234             JSR [ASSERT.NA]                                ; установка признака, что ожидаемые и полученные данные
                    ;9235                                                            ; не применяются
                    ;9236     LOOP.TD.1:
                    ;9237      ;
                    ;9238      ; Следующая инструкция DECODE сбрасывает сигнал RBKUP FLAG
                    ;9239      ;
U 0E05,  8402,1E    ;9240             OPC2/DECODE,BPC/NOP,DEC.ADRS/0,IFUNC/1,OPC.SPEC/1,JCTL/NO.JUMP TST ;
U 0E06,  DB00,15    ;9241             NOP                                            ;
U 0E07,  1070,15    ;9242             MISC [SET.BACKUP.MASK.VALID]                   ; установка признака
U 0E08,  5B00,1B    ;9243             SKIP.IF[BACKUP.MASK.VALID]                     ; пропуск, если признак установлен
U 0E09,  5B00,1E    ;9244             SKIP                                           ;
U 0E0A,  08E0,F4    ;9245             JMP [TD.2]                                     ; продолжение, если правильно
U 0E0B,  B69E,15    ;9246             MOV LS[ONES] TO WR[0]                          ;
U 0E0C,  2F82,95    ;9247             CLR WR[1]                                      ; установка ошибочных данных для индикации ошибки
U 0E0D,  0869,3C    ;9248             JSR [CHECK.RESULT]                             ; сообщение об ошибке 1
U 0E0E,  08E0,54    ;9249             JMP [LOOP.TD.1]                                ; зацикливание при ошибке, если разрешено
                    ;9250     TD.2:
U 0E0F,  FF82,15    ;9251             INC LS[ERROR.NUMBER]                           ; увеличение номера ошибки до 2
                    ;9252     LOOP.TD.2:
                    ;9253      ;
                    ;9254      ; Следующая инструкция DECODE сбрасывает сигнал RBKUP FLAG
                    ;9255      ;
U 0E10,  8402,1E    ;9256             OPC2/DECODE,BPC/NOP,DEC.ADRS/0,IFUNC/1,OPC.SPEC/1,JCTL/NO.JUMP.TST ;
U 0E11,  DB00,15    ;9257             NOP                                            ;
U 0E12,  5B00,1B    ;9258             SKIP.IF[BACKUP.MASK.VALID]                     ; пропуск, если признак установлен
U 0E13,  08E1,84    ;9259             JMP [END.TD]                                   ; продолжение, если правильно
```

ТЕСТ D - тест признака маски резервной копии регистров (модуль DAP)

```
U 0E14, B69E,15 ;9260           MOV LS[ONES] TO WR[0]                ;
U 0E15, 2F82,95 ;9261           CLR WR[1]                            ; установка ошибочных данных для индикации ошибки
U 0E16, 0869,3C ;9262           JSR [CHECK.RESULT]                   ; сообщение об ошибке
U 0E17, 88E1,04 ;9263           JMP [LOOP.TD.2]                      ; зацикливание при ошибке, если разрешено
                ;9264   END.TD:
```

```
;9265      .PAGE "ТЕСТ Е - тест сигнала BRANCH FALSE (ветвление ложно) (модуль DAP)"
;9266      ;
;9267      ;   ОПИСАНИЕ ТЕСТА:
;9268      ;
;9269      ;       Этот тест проверяет условие "ВЕТВЛЕНИЕ ЛОЖНО" (BRANCH FALSE).
;9270      ;       Сигнал BRANCH FALSE возбуждается, когда возбужден бит 0 шины IB, а
;9271      ;       PRETEST не возбужден, или когда PRETEST возбужден, а IB 0 нет. Для то-
;9272      ;       го, чтобы проверить установку сигнала BRANCH FALSE, Если сигнал PRETEST
;9273      ;       не выставлен, шина IB должна быть загружена значением 01(H). При этом
;9274      ;       на IB 0 помещается 1, а сигнал PRETEST не возбуждается. Если сигнал
;9275      ;       BRANCH FALSE возбужден, формируется сигнал ENABLE IR ROM и таким обра-
;9276      ;       зом возможно проверить, был ли выставлен сигнал BRANCH FALSE. Сигнал
;9277      ;       ENABLE IR ROM выбирает ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ, если выполняемой инст-
;9278      ;       рукцией является DECODE и бит OPC/SPEC сброшен. Инструкция DECODE уста-
;9279      ;       навливает также сигналы DT CLASS и CC CLASS в соответствии со значением
;9280      ;       на шине IB. Следующая инструкция выполняет переход или пропуск по приз-
;9281      ;       наку BR FALSE согласно заданным кодам условий и далее продолжается вы-
;9282      ;       полнение теста или выводятся сообщения при ошибках. Как отмечалось рань-
;9283      ;       ше, сигнал BRANCH FALSE может быть возбужден, если сигнал IB 0 SAVE вы-
;9284      ;       сокий, а сигнал PRETEST низкий. Для того, чтобы добиться выполнения этих
;9285      ;       условий, IB 0 должен быть загружен значением 1, а коды условий могут быть
;9286      ;       определены выполнением инструкции MOV для загрузки LSL7F] такими кодами
;9287      ;       условий, которые вызовут установку BRANCH FALSE. Инструкция DECODE со
;9288      ;       значением 00(H) на шине IB должна возбудить сигнал BRANCH FALSE. Поле уп-
;9289      ;       равления переходом должно быть установлено на условие BRANCH FALSE. Если
;9290      ;       переход выполняется, тогда в LSL7E] загружается другой набор кодов усло-
;9291      ;       вий, который заведомо формирует BRANCH FALSE и выполняется инструкция JUMP
;9292      ;       или NOP с полем управления пропуском, установленным для проверки условия
;9293      ;       BRANCH FALSE. Эта последовательность проверки продолжается до тех пор, по-
;9294      ;       ка не будут проверены все условия, возбуждающие сигнал PRETEST.
;9295      ;
;9296      ;   ПРЕДПОЛОЖЕНИЯ:
;9297      ;
;9298      ;       Предполагается, что ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ и все сигналы разрешения
;9299      ;       проверены, а также шина IB проверена и работает.
;9300      ;
;9301      ;   ШАГИ ТЕСТА:
;9302      ;
;9303      ;     1.Очистка регистра кодов условий PSL.CC. Установка на шине IB значения
;9304      ;       01(H).
;9305      ;     2.Выдача инструкции DECODE для установки DATA TYPE=0, CC CLASS=3 (проверка
;9306      ;       условия C).
;9307      ;     3.Переход, если BR FALSE установлен. Если перехода не было, сообщение
;9308      ;       об ошибке 1.
;9309      ;     4.Установка на шине IB данных 00(H).
;9310      ;     5.Выдача инструкции DECODE для установки DATA TYPE=0, CC CLASS=3. Если
;9311      ;       BR FALSE установлен, выполняется пропуск инструкции и выдается сообще-
;9312      ;       ние об ошибке 2.
;9313      ;     6.Выдача инструкции MOV для установки PSL Z.
;9314      ;     7.Выдача инструкции DECODE для установки DATA TYPE=0 и CC CLASS=0.
;9315      ;       если перехода не было, выдается сообщение об ошибке 3.
;9316      ;     8.Выдача инструкции MOV для установки PSL C. Выдача инструкции DECODE
;9317      ;       для установки DATA TYPE=0, CC CLASS=3.
;9318      ;     9.Выдача инструкции JUMP для перехода по BRANCH FALSE. Если перехода нет,
;9319      ;       выдается сообщение об ошибке 4.
```

```
;9320      ;    10.Выдача инструкции MOV для установки PSL N. Выдача инструкции DECODE
;9321      ;       для установки DATA TYPE=1, CC CLASS=0.
;9322      ;    11.Выдача инструкции NOP С пропуском по BRANCH FALSE.
;9323      ;       Если пропуска нет, выдается сообщение об ошибке 5.
;9324      ;    12.Выдача инструкции MOV для установки PSL V. Выдача инструкции DECODE
;9325      ;       для установки DATA TYPE=0, CC CLASS=2.
;9326      ;    13.Выдача инструкции JUMP для перехода по BRANCH FALSE. Если перехода нет,
;9327      ;       выдается сообщение об ошибке 6.
;9328      ;    14.Выдача инструкции MOV для установки PSL N и сброса PSL V.
;9329      ;    15.Выдача инструкции DECODE для установки DATA TYPE=1 и CC CLASS=3. Пере-
;9330      ;       ход по BRANCH FALSE. Если перехода нет, выдается сообщение об ошибке 7.
;9331      ;    16.Выдача инструкции MOV для установки PSL V и сброса PSL N.
;9332      ;    17.Выдача инструкции DECODE для установки DATA TYPE=1 и CC CLASS=3. Пере-
;9333      ;       ход по BRANCH FALSE. Если перехода нет, выдается сообщение об ошибке 8.
;9334      ;    18.Выдача инструкции MOV для установки PSL Z.
;9335      ;    19.Выдача инструкции DECODE для установки DATA TYPE=0 и CC CLASS=0. Про-
;9336      ;       пуск по BRANCH FALSE. Если пропуск выполняется, выдается сообщение об
;9337      ;       ошибке 9.
;9338      ;    20.Выдача инструкции MOV для очистки кодов условий PSL.
;9339      ;    21.Выдача инструкции DECODE для установки DATA TYPE=0 и CC CLASS=0. Про-
;9340      ;       пуск по BRANCH FALSE. Если пропуск выполняется, выдается сообщение об
;9341      ;       ошибке A.
;9342      ;
;9343      ; ОШИБКИ:
;9344      ;
;9345      ;    ошибка 1 - неправильный сигнал BRANCH FALSE, когда должен быть возбуждён.
;9346      ;    ошибка 2 - неправильный сигнал BRANCH FALSE, когда не должен быть возбужден.
;9347      ;    ошибка 3 - неправильный сигнал PRETEST.
;9348      ;    ошибка 4 - неправильный сигнал PRETEST.
;9349      ;    ошибка 5 - неправильный сигнал PRETEST.
;9350      ;    ошибка 6 - неправильный сигнал PRETEST.
;9351      ;    ошибка 7 - неправильный сигнал PRETEST.
;9352      ;    ошибка 8 - неправильный сигнал PRETEST.
;9353      ;    ошибка 9 - был возбужден сигнал BRANCH FALSE.
;9354      ;    ошибка A - был возбужден сигнал BRANCH FALSE.
;9355      ;
;9356      ; НАЛАДКА:
;9357      ;
;9358      ;    ОШИБКА 1: Ошибка появляется из-за низкого уровня сигнала BRANCH FALSE, ког-
;9359      ;              да ожидается высокий уровень. Источниками ошибки скорее всего яв-
;9360      ;              ляются: застревание сигнала BRANCH FALSE на низком уровне или не-
;9361      ;              исправная ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ.
;9362      ;    ОШИБКА 2: Ошибка появляется из-за высокого уровня сигнала BRANCH FALSE,
;9363      ;              когда ожидается низкий уровень. Источниками ошибки скорее всего
;9364      ;              являются: застревание сигнала BRANCH FALSE на высоком уровне или
;9365      ;              неисправная ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ.
;9366      ;    ОШИБКА 3: Неправильно формируется сигнал BRANCH FALSE, так как не формиру-
;9367      ;              ется сигнал PRETEST. Сигнал PRETEST может быть неправильным из-за
;9368      ;              неправильной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ или ПМЛ КОДЫ
;9369      ;              УСЛОВ.PSL[3:0]. Необходимо проверить сигнал PSL Z.
;9370      ;    ОШИБКА 4: Неправильно формируется сигнал BRANCH FALSE, так как неправильно
;9371      ;              формируется сигнал PRETEST. Сигнал PRETEST, по видимому, формиру-
;9372      ;              ется неправильно из-за неисправной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ
;9373      ;              УСЛОВИЙ или ПМЛ КОДЫ УСЛОВ.PSL[3:0]. Здесь необходимо проверить сигнал
;9374      ;              PSL C.
```

```
                    ;9375      ;     ОШИБКА 5: Неправильно формируется сигнал BRANCH FALSE, так как неправильно
                    ;9376      ;               формируется сигнал PRETEST. Сигнал PRETEST, повидимому, формиру-
                    ;9377      ;               ется неправильно из-за неисправной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ
                    ;9378      ;               УСЛОВИЙ или ПМЛ КОДЫ УСЛОВ.PSL[3:0]. Необходимо проверить сигнал PSL N.
                    ;9379      ;     ОШИБКА 6: Неправильно формируется сигнал BRANCH FALSE, так как неправильно
                    ;9380      ;               формируется сигнал PRETEST. Сигнал PRETEST, повидимому, формиру-
                    ;9381      ;               ется неправильно из-за неисправной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ
                    ;9382      ;               УСЛОВИЙ или ПМЛ КОДЫ УСЛОВ.PSL[3:0]. Необходимо проверить сигнал PSL V.
                    ;9383      ;     ОШИБКА 7: Неправильно формируется сигнал BRANCH FALSE, так как неправильно
                    ;9384      ;               формируется сигнал PRETEST. Сигнал PRETEST, повидимому, формиру-
                    ;9385      ;               ется неправильно из-за неисправной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ
                    ;9386      ;               УСЛОВИЙ или ПМЛ КОДЫ УСЛОВ.PSL[3:0]. Необходимо проверить сигналы PSL N
                    ;9387      ;               и PSL V.
                    ;9388      ;     ОШИБКА 8: Неправильно формируется сигнал BRANCH FALSE, так как неправильно
                    ;9389      ;               формируется сигнал PRETEST. Сигнал PRETEST формируется неправиль-
                    ;9390      ;               но из-за неисправной ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ или ПМЛ
                    ;9391      ;               КОДЫ УСЛОВ.PSL[3:0]. Необходимо проверить сигналы PSL V и PSL N.
                    ;9392      ;     ОШИБКА 9: Ошибка появляется из-за низкого уровня сигнала BRANCH FALSE, ког-
                    ;9393      ;               да ожидается высокий уровень. Источниками ошибки скорее всего яв-
                    ;9394      ;               ляются: застревание сигнала BRANCH FALSE на низком уровне или не-
                    ;9395      ;               исправная ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ. Необходимо проверить
                    ;9396      ;               застревание сигнала PSL Z на низком уровне.
                    ;9397      ;     ОШИБКА A: Сигнал BRANCH FALSE был низким (возбужден), когда ожидается высо-
                    ;9398      ;               кий уровень (не возбужден). Вероятным источником этой ошибки мо-
                    ;9399      ;               жет быть ПМЛ УСЛОВИЯ ПЕРЕХОДОВ, КЛАСС КОДОВ УСЛОВИЙ, где сигнал PSL Z за-
                    ;9400      ;               стрял на высоком уровне.
                    ;9401      ;
                    ;9402      ;--
                    ;9403      T.E:
U 0E1B, B65E,15     ;9404                MOV LS[BEGIN.TEST] TO WR[0]                ; установка бита 15 в WR0 для слова управления и
                    ;9405                                                           ; состояния
U 0E19, 3EB0,15     ;9406                MOV WR[0] TO LS[CONTROL.STATUS]             ; установка бита 15 в слове управления и состояния. Бит
                    ;9407                                                           ; 15 указывает начало теста для конс.процессора
U 0E1A, 10E0,15     ;9408                MISC [SET.CP.ATTN]                         ; выдача сигнала CPU ATTN для конс.процессора
                    ;9409      WAIT.TE.0:
U 0E1B, 0BE1,B4     ;9410                JMP [WAIT.TE.0]                            ; зацикливание для ожидания ответа конс.процессора
U 0E1C, B724,15     ;9411                MOV LS[HI.IPL] TO WR[0]                    ;
U 0E1D, BFFE,15     ;9412                MOV WR[0] TO LS[PSL.HW]                    ; сброс режима совместимости
U 0E1E, E58A,15     ;9413                CLR LS[ERROR.MASK]                         ; очистка маски ошибки
U 0E1F, 65A0,15     ;9414                CLR LS[PREVIOUS.ERROR]                     ; очистка предыдущего номера ошибки
U 0E20, 0881,0C     ;9415                JSR [ASSERT.NA]                            ; установка признака, что ожидаемые и полученные данные
                    ;9416                                                           ; не применимы
U 0E21, B640,15     ;9417                MOV LS[#1] TO WR[0]                        ; установка 1 в WR0
U 0E22, BEB2,15     ;9418                MOV WR[0] TO LS[ERROR.NUMBER]              ; установка номера ошибки для первой ошибки
U 0E23, A3C0,15     ;9419                ROL WR[0]                                  ; установка 2 в WR0
U 0E24, 3E8C,15     ;9420                MOV WR[0] TO LS[MODULE.NUM]                ; установка кода модуля для модуля DAP
U 0E25, 364E,15     ;9421                MOV LS[#80] TO WR[0]                       ;
U 0E26, C048,15     ;9422                ADD LS[#10] TO WR[0]                       ;
U 0E27, C044,15     ;9423                ADD LS[#4] TO WR[0]                        ;
U 0E28, 3E10,15     ;9424                MOV WR[0] TO LS[T8]                        ; T8=94
U 0E29, B640,15     ;9425                MOV LS[#1] TO WR[0]                        ;
U 0E2A, BE14,15     ;9426                MOV WR[0] TO LS[T10]                       ; T10=1
                    ;9427      LOOP.TE.1:
U 0E2B, 1910,75     ;9428                MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]         ;
U 0E2C, B214,15     ;9429                WRITE.MEM LS[T10]
```

```
U 0E2D, E520,15 ;9430            CLR LS[PC]                                      ;
U 0E2E, 1B11,75 ;9431            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]              ; заполнение IB значением 01(H)
U 0E2F, E5FE,15 ;9432            CLR LS[PSL.CC]                                  ; очистка кодов условий
                ;9433     ;
                ;9434     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=3
                ;9435     ;
U 0E30, 8473,1E ;9436            OPC2/DECODE,IB.REQ/1,BPC/NOP,DEC.ADRS/7,IFUNC/1,OPC.SPEC/1,JCTL/NO.JUMP.TST ;
U 0E31, DB00,15 ;9437            NOP                                             ;
U 0E32, 08E3,75 ;9438            JMP.IF[BR.FALSE] TO [TE.2]                      ; переход, если правильно
U 0E33, 2FB0,15 ;9439            CLR WR[0]                                       ;
U 0E34, 369E,95 ;9440            MOV LS[ONES] TO WR[1]                           ;
U 0E35, 0869,3C ;9441            JSR [CHECK.RESULT]                              ; сообщение об ошибке 1
U 0E36, 08E2,B4 ;9442            JMP [LOOP.TE.1]                                 ; зацикливание при ошибке, если разрешено
                ;9443     TE.2:
U 0E37, FF82,15 ;9444            INC LS[ERROR.NUMBER]                            ; увеличение номера ошибки до 2
                ;9445     LOOP.TE.2:
U 0E38, 2FB0,15 ;9446            CLR WR[0]                                       ;
U 0E39, BE14,15 ;9447            MOV WR[0] TO LS[T10]                            ; T10=0
U 0E3A, 1910,75 ;9448            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]              ;
U 0E3B, B214,15 ;9449            WRITE.MEM LS[T10]                               ;
U 0E3C, E520,15 ;9450            CLR LS[PC]                                      ;
U 0E3D, 1B11,75 ;9451            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]              ; заполнение IB значением 0
U 0E3E, E5FE,15 ;9452            CLR LS[PSL.CC]                                  ; очистка кодов условий
                ;9453     ;
                ;9454     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=3
                ;9455     ;
U 0E3F, 8473,1E ;9456            OPC2/DECODE,IB.REQ/1,BPC/NOP,DEC.ADRS/7,IFUNC/1,OPC.SPEC/1,JCTL/NO.JUMP.TST
U 0E40, DB00,15 ;9457            NOP                                             ;
U 0E41, 5B00,05 ;9458            SKIP.IF[BR.FALSE]                               ; пропуск, если ошибка
U 0E42, 08E4,74 ;9459            JMP [TE.3]                                      ;
U 0E43, 2FB0,15 ;9460            CLR WR[0]                                       ;
U 0E44, 369E,95 ;9461            MOV LS[ONES] TO WR[1]                           ;
U 0E45, 0869,3C ;9462            JSR [CHECK.RESULT]                              ; проверка, что инструкция DECODE не выполняет перехода
U 0E46, 88E3,84 ;9463            JMP [LOOP.TE.2]                                 ; зацикливание при ошибке, если разрешено
                ;9464     TE.3:
U 0E47, FF82,15 ;9465            INC LS[ERROR.NUMBER]                            ; увеличение номера ошибки до 3
U 0E48, 3648,15 ;9466            MOV LS[#10] TO WR[0]                            ;
U 0E49, BE14,15 ;9467            MOV WR[0] TO LS[T10]                            ; T10=10(H)
                ;9468     LOOP.TE.3:
U 0E4A, 1910,75 ;9469            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]              ;
U 0E4B, B214,15 ;9470            WRITE.MEM LS[T10]                               ;
U 0E4C, 1B11,75 ;9471            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]              ; заполнение IB значением 10(H)
U 0E4D, 3644,15 ;9472            MOV LS[BIT2] TO WR[0]                           ;
U 0E4E, 3EFE,15 ;9473            MOV WR[0] TO LS[PSL.CC]                         ; установка PSL Z
                ;9474     ;
                ;9475     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=0
                ;9476     ;
U 0E4F, 8473,1E ;9477            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0E50, DB00,15 ;9478            NOP                                             ;
U 0E51, 88E5,65 ;9479            JMP.IF[BR.FALSE] TO [TE.4]                      ; переход, если правильно
U 0E52, 2FB0,15 ;9480            CLR WR[0]                                       ;
U 0E53, 369E,95 ;9481            MOV LS[ONES] TO WR[1]                           ;
U 0E54, 0869,3C ;9482            JSR [CHECK.RESULT]                              ; сообщение об ошибке 3
U 0E55, 88E4,A4 ;9483            JMP [LOOP.TE.3]                                 ; зацикливание при ошибке, если разрешено
                ;9484     TE.4:
```

```
U 0E56, FF82,15 ;9485            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 4
U 0E57, 3644,15 ;9486            MOV LS[#4] TO WR[0]                     ;
U 0E58, BE14,15 ;9487            MOV WR[0] TO LS[T10]                    ; T10=4(H)
                ;9488    LOOP.TE.4:
U 0E59, 1910,75 ;9489            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]      ;
U 0E5A, B214,15 ;9490            WRITE.MEM LS[T10]                       ;
U 0E5B, 1B11,75 ;9491            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]      ; заполнение IB значением 4(H)
U 0E5C, B640,15 ;9492            MOV LS[BIT0] TO WR[0]                   ;
U 0E5D, 3EFE,15 ;9493            MOV WR[0] TO LS[PSL.CC]                 ; установка PSL C
                ;9494     ,
                ;9495     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=3
                ;9496     ;
U 0E5E, 8473,1E ;9497            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0E5F, DB00,15 ;9498            NOP                                     ;
U 0E60, 88E6,55 ;9499            JMP.IF[BR.FALSE] TO [TE.5]              ; переход, если правильно
U 0E61, 2F80,15 ;9500            CLR WR[0]                               ;
U 0E62, 369E,95 ;9501            MOV LS[ONES] TO WR[1]                   ,
U 0E63, 0869,3C ;9502            JSR [CHECK.RESULT]                      ; сообщение об ошибке 4
U 0E64, 08E5,94 ;9503            JMP [LOOP.TE.4]                         ; зацикливание при ошибке, если разрешено
                ;9504    TE.5:
U 0E65, FF82,15 ;9505            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 5
U 0E66, 3644,15 ;9506            MOV LS[#4] TO WR[0]                     ;
U 0E67, C048,15 ;9507            ADD LS[#10] TO WR[0]                    ;
U 0E68, BE14,15 ;9508            MOV WR[0] TO LS[T10]                    ; T10=14(H)
                ;9509    LOOP.TE.5:
U 0E69, 1910,75 ;9510            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]
U 0E6A, B214,15 ;9511            WRITE.MEM LS[T10]                       ,
U 0E6B, 1B11,75 ;9512            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]      ; заполнение IB значением 14(H)
U 0E6C, B646,15 ;9513            MOV LS[BIT3] TO WR[0]                   ;
U 0E6D, 3EFE,15 ;9514            MOV WR[0] TO LS[PSL.CC]                 ; установка PSL N
                ;9515     ;
                ;9516     ; Эта инструкция DECODE устанавливает DT=1 и CC CLASS=0
                ;9517     ;
U 0E6E, 8473,1E ;9518            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0E6F, DB00,15 ;9519            NOP                                     ;
U 0E70, 08E7,55 ;9520            JMP.IF[BR.FALSE] TO [TE.6]              ; переход, если правильно
U 0E71, 2F80,15 ;9521            CLR WR[0]                               ;
U 0E72, 369E,95 ;9522            MOV LS[ONES] TO WR[1]                   ;
U 0E73, 0869,3C ;9523            JSR [CHECK.RESULT]                      ; сообщение об ошибке 5
U 0E74, 08E6,94 ;9524            JMP [LOOP.TE.5]                         ; зацикливание при ошибке, если разрешено
                ;9525    TE.6:
U 0E75, FF82,15 ;9526            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 6
U 0E76, 3648,15 ;9527            MOV LS[#10] TO WR[0]                    ;
U 0E77, 4142,15 ;9528            SUB LS[#2] FROM WR[0]                   ;
U 0E78, BE14,15 ;9529            MOV WR[0] TO LS[T10]                    ; T10=E(H)
                ;9530    LOOP.TE.6:
U 0E79, 1910,75 ;9531            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]      ;
U 0E7A, B214,15 ;9532            WRITE.MEM LS[T10]                       ;
U 0E7B, 1B11,75 ;9533            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]      ; заполнение IB значением E(H)
U 0E7C, 3642,15 ;9534            MOV LS[BIT1] TO WR[0]                   ;
U 0E7D, 3EFE,15 ;9535            MOV WR[0] TO LS[PSL.CC]                 ; установка PSL V
                ;9536     ;
                ;9537     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=2
                ;9538     ;
U 0E7E, 8473,1E ;9539            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
```

```
U 0E7F, DB00,15 ;9540            NOP                                    ;
U 0E80, 08E8,55 ;9541            JMP.IF[BR.FALSE] TO [TE.7]             ; переход, если правильно
U 0E81, 2FB0,15 ;9542            CLR WR[0]                              ;
U 0E82, 369E,95 ;9543            MOV LS[ONES] TO WR[1]                  ;
U 0E83, 0869,3C ;9544            JSR [CHECK.RESULT]                     ; сообщение об ошибке 6
U 0E84, 88E7,94 ;9545            JMP [LOOP.TE.6]                        ; зацикливание при ошибке, если разрешено
                ;9546    TE.7:
U 0E85, FF82,15 ;9547            INC LS[ERROR.NUMBER]                   ; увеличение номера ошибки до 7
U 0E86, 8646,15 ;9548            MOV LS[#8] TO WR[0]                    ;
U 0E87, BE14,15 ;9549            MOV WR[0] TO LS[T10]                   ; T10=8(H)
                ;9550    LOOP.TE.7:
U 0E88, 1910,75 ;9551            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0E89, B214,15 ;9552            WRITE.MEM LS[T10]                      ;
U 0E8A, 1B11,75 ;9553            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение IB значением 8(H)
U 0E8B, 8646,15 ;9554            MOV LS[BIT3] TO WR[0]                  ;
U 0E8C, 3EFE,15 ;9555            MOV WR[0] TO LS[PSL.CC]                ; установка PSL N (сброс V)
                ;9556     ;
                ;9557     ; Эта инструкция DECODE устанавливает DT=1 и CC CLASS=3
                ;9558     ;
U 0E8D, 8473,1E ;9559            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0E8E, DB00,15 ;9560            NOP                                    ;
U 0E8F, 08E9,45 ;9561            JMP.IF[BR.FALSE] TO [TE.8]             ; переход, если правильно
U 0E90, 2FB0,15 ;9562            CLR WR[0]                              ;
U 0E91, 369E,95 ;9563            MOV LS[ONES] TO WR[1]                  ;
U 0E92, 0869,3C ;9564            JSR [CHECK.RESULT]                     ; сообщение об ошибке 7
U 0E93, 08E8,84 ;9565            JMP [LOOP.TE.7]                        ; зацикливание при ошибке, если разрешено
                ;9566    TE.8:
U 0E94, FF82,15 ;9567            INC LS[ERROR.NUMBER]                   ; увеличение номера ошибки до 8
U 0E95, B646,15 ;9568            MOV LS[#8] TO WR[0]                    ;
U 0E96, BE14,15 ;9569            MOV WR[0] TO LS[T10]                   ; T10=8(H)
                ;9570    LOOP.TE.8:
U 0E97, 1910,75 ;9571            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0E98, B214,15 ;9572            WRITE.MEM LS[T10]                      ;
U 0E99, 1B11,75 ;9573            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение IB значением 8(H)
U 0E9A, 3642,15 ;9574            MOV LS[BIT1] TO WR[0]                  ;
U 0E9B, 3EFE,15 ;9575            MOV WR[0] TO LS[PSL.CC]                ; установка PSL V (сброс N)
                ;9576     ;
                ;9577     ; эта инструкция DECODE устанавливает DT=1 и CC CLASS=3
                ;9578     ;
U 0E9C, 8473,1E ;9579            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0E9D, DB00,15 ;9580            NOP                                    ;
U 0E9E, 88EA,35 ;9581            JMP.IF[BR.FALSE] TO [TE.9]             ; переход, если правильно
U 0E9F, 2FB0,15 ;9582            CLR WR[0]                              ;
U 0EA0, 369E,95 ;9583            MOV LS[ONES] TO WR[1]                  ;
U 0EA1, 0869,3C ;9584            JSR [CHECK.RESULT]                     ; сообщение об ошибке 8
U 0EA2, 88E9,74 ;9585            JMP [LOOP TE.8]                        ; зацикливание при ошибке, если разрешено
                ;9586    TE.9:
U 0EA3, FF82,15 ;9587            INC LS[ERROR.NUMBER]                   ; увеличение номера ошибки до 9
U 0EA4, 3648,15 ;9588            MOV LS[#10] TO WR[0]                   ;
U 0EA5, 4040,15 ;9589            ADD LS[#1] TO WR[0]                    ;
U 0EA6, BE14,15 ;9590            MOV WR[0] TO LS[T10]                   ; T10=11(H)
                ;9591    LOOP.TE.9:
U 0EA7, 1910,75 ;9592            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]     ;
U 0EA8, B214,15 ;9593            WRITE.MEM LS[T10]                      ;
U 0EA9, 1B11,75 ;9594            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]     ; заполнение IB значением 11(H)
```

```
U 0EAA, 3644,15 ;9595            MOV LS[BIT2] TO WR[0]                          ;
U 0EAB, 3EFE,15 ;9596            MOV WR[0] TO LS[PSL.CC]                        ; установка PSL Z
                ;9597     ;
                ;9598     ; Эта инструкция DECODE устанавливает DT=0 и CC CLASS=0
                ;9599     ;
U 0EAC, 8473,1E ;9600            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0EAD, DB00,15 ;9601            NOP                                            ;
U 0EAE, 5B00,05 ;9602            SKIP.IF[BR.FALSE]                              ; пропуск, если ошибка
U 0EAF, 08EB,44 ;9603            JMP [TE.A]                                     ;
U 0EB0, 2F80,15 ;9604            CLR WR[0]                                      ;
U 0EB1, 369E,95 ;9605            MOV LS[ONES] TO WR[1]                          ;
U 0EB2, 0869,3C ;9606            JSR [CHECK.RESULT]                             ; сообщение об ошибке 9
U 0EB3, 88EA,74 ;9607            JMP [LOOP.TE.9]                                ; зацикливание при ошибке, если разрешено
                ;9608     TE.A:
U 0EB4, FF82,15 ;9609            INC LS[ERROR.NUMBER]                           ; увеличение номера ошибки до A
U 0EB5, 3648,15 ;9610            MOV LS[#10] TO WR[0]                           ;
U 0EB6, BE14,15 ;9611            MOV WR[0] TO LS[T10]                           ; T10=10(H)
                ;9612     LOOP.TE.A:
U 0EB7, 1910,75 ;9613            MEM.REQ[WRITE.P] ADRS[TB] DT[LONG]             ;
U 0EB8, B214,15 ;9614            WRITE.MEM LS[T10]                              ;
U 0EB9, 1B11,75 ;9615            MEM.REQ[IB.FILL] ADRS[TB] DT[LONG]             ; заполнение IB значением 10(H)
U 0EBA, 2F80,15 ;9616            CLR WR[0]                                      ;
U 0EBB, 3EFE,15 ;9617            MOV WR[0] TO LS[PSL.CC]                        ; очистка кодов условий
                ;9618     ;
                ;9619     ; Эжа инструкция DECODE устанавливает DT=0 и CC CLASS=0
                ;9620     ;
U 0EBC, 8473,1E ;9621            OPC2/DECODE,DEC.ADRS/7,IFUNC/1,IB.REQ/1,BPC/NOP,OPC.SPEC/CM.IRD,JCTL/NO.JUMP.TST ;
U 0EBD, DB00,15 ;9622            NOP                                            ;
U 0EBE, 5B00,05 ;9623            SKIP.IF[BR.FALSE]                              ; не должно быть пропуска
U 0EBF, 88EC,44 ;9624            JMP [END.TE]                                   ; конец теста, если был переход сюда
U 0EC0, 2F80,15 ;9625            CLR WR[0]                                      ;
U 0EC1, 369E,95 ;9626            MOV LS[ONES] TO WR[1]                          ;
U 0EC2, 0869,3C ;9627            JSR [CHECK.RESULT]                             ; сообщение об ошибке A
U 0EC3, 08EB,74 ;9628            JMP [LOOP.TE.A]                                ; зацикливание при ошибке, если разрешено
                ;9629     END.TE:
```

```
;9630      .PAGE "ТЕСТ F - тест условных переходов (модуль DAP)"
;9631      ;
;9632      ; ОПИСАНИЕ ТЕСТА:
;9633      ;
;9634      ;    Этот тест проверяет разные условия перехода, которые еще не были
;9635      ;    проверены. Условия перехода можно проверить путем использования
;9636      ;    инструкции JUMP или инструкции DECODE с условием в поле JCTL.
;9637      ;    Условиями, которые должны быть проверены, являются IB VALID
;9638      ;    (IB истинный) и JUMP (переход) в инструкции DECODE, JSR IF IB
;9639      ;    VALID (переход с возвратом, если IB истинный) и NO JUMP, SKIP IF IB
;9640      ;    VALID (нет перехода, пропуск микроинструкции, если IB истинный).
;9641      ;    Сигнал IB VALID может быть возбужден инструкцией DECODE, которая
;9642      ;    вызывает запрос памяти. Когда сигнал IB VALID возбужден, могут
;9643      ;    проверяться три условия, зависящие от сигнала IB VALID и затем могут
;9644      ;    проверяться другие условия безусловных переходов.
;9645      ;
;9646      ;       ПРЕДПОЛОЖЕНИЯ:
;9647      ;
;9648      ;    Предполагается, что шина IB и ПЗУ ДЕШ.СПЕЦИФИКАТОРОВ проверены и
;9649      ;    работают. Сигнал IB VALID должен быть проверен раньше.
;9650      ;
;9651      ; ШАГИ ТЕСТА:
;9652      ;
;9653      ;    1. Выдача инструкции DECODE для возбуждения сигнала IB VALID с полем
;9654      ;       JCTL, установленным на IB VALID. Если переход не выполняется, выдается
;9655      ;       сообщение об ошибке 1.
;9656      ;    2. Выдача инструкции JUMP для перехода по IB VALID. Если перехода нет,
;9657      ;       выдается сообщение об ошибке 2.
;9658      ;    3. Выдача инструкции JUMP для JSR по IB VALID. Если не выполняется JSR
;9659      ;       (переход с возвратом), выдается сообщение об ошибке 3. Если JSR
;9660      ;       выполняется, тогда выдается возврат.
;9661      ;    4. Выдача инструкции JUMP для NO JUMP, SKIP IF IB VALID (нет перехода,
;9662      ;       пропуск если IB истинный). Если пропуска нет, выдается сообщение об
;9663      ;       ошибке 4.
;9664      ;    5. Выдача инструкции DECODE с полем JCTL, установленным для перехода.
;9665      ;       Если перехода нет, выдается сообщение об ошибке 5.
;9666      ;    6. Выдача инструкции DECODE, которая должна установить сигнал IB VALID
;9667      ;       с полем JCTL, установленным для JSR IF IB VALID (переход с возвратом,
;9668      ;       если IB истинный). Если JSR не выполняется, выдается сообщение об
;9669      ;       ошибке 6. Если JSR выполняется, тогда выдается возврат.
;9670      ;
;9671      ; ОШИБКИ:
;9672      ;
;9673      ;    ошибка 1 - не было перехода по IB VALID с инструкцией DECODE.
;9674      ;    ошибка 2 - не было перехода по IB VALID с инструкцией JUMP.
;9675      ;    ошибка 3 - не было перехода с возвратом по IB VALID с инструкцией JUMP.
;9676      ;    ошибка 4 - инструкция JUMP не выполнила функции: нет перехода, пропуск,
;9677      ;               если IB истинный.
;9678      ;    ошибка 5 - не было перехода с инструкцией DECODE.
;9679      ;    ошибка 6 - не было перехода с возвратом по IB VALID с инструкцией DECODE.
;9680      ;
;9681      ; НАЛАДКА:
;9682      ;
;9683      ;    ОШИБКА 1: Самым вероятным источником ошибки, если переход по IB VALID ра-
;9684      ;              ботает неправильно, является ПМЛ УПР.МИАСС ЕКВЕНСЕРОМ. Схемы,
```

```
                    ;9685     ;              связанные с сигналом IB VALID, были проверены раньше и должны
                    ;9686     ;              работать.
                    ;9687     ;     ОШИБКА 2: Самым вероятным источником ошибки является ПМЛ УПР.МИАСС ЕКВЕН-
                    ;9688     ;              СЕРОМ.
                    ;9689     ;     ОШИБКА 3: Самым вероятным источником ошибки является ПМЛ УПР.МИАСС ЕКВЕН-
                    ;9690     ;              СЕРОМ.
                    ;9691     ;     ОШИБКА 4: Самым вероятным источником ошибки является ПМЛ УПР.МИАСС ЕКВЕН-
                    ;9692     ;              СЕРОМ.
                    ;9693     ;     ОШИБКА 5: Самым вероятным источником ошибки является ПМЛ УПР.МИАСС ЕКВЕН-
                    ;9694     ;              СЕРОМ.
                    ;9695     ;     ОШИБКА 6: Самым вероятным источником ошибки является ПМЛ УПР.МИАСС ЕКВЕН-
                    ;9696     ;              СЕРОМ.
                    ;9697     T.F:
U 0EC4,  B65E,15    ;9698               MOV LS[BEGIN.TEST] TO WR[0]                 ; установка бита 15 в WR0 для слова управления и
                    ;9699                                                           ; состояния
U 0EC5,  3E80,15    ;9700               MOV WR[0] TO LS[CONTROL.STATUS]              ; установка бита 15 в слове управления и состояния. Бит
                    ;9701                                                           ; 15 указывает начало теста для консольного процессора
U 0EC6,  10E0,15    ;9702               MISC [SET.CP.ATTN]                          ; выдача сигнала CPU ATTN для консольного процессора
                    ;9703     WAIT.TF.0:
U 0EC7,  88EC,74    ;9704               JMP [WAIT.TF.0]                             ; зацикливание для ожидания ответа от консольного
                    ;9705                                                           ; процессора
U 0EC8,  B724,15    ;9706               MOV LS[HI.IPL] TO WR[0]                     ;
U 0EC9,  BFFE,15    ;9707               MOV WR[0] TO LS[PSL.HW]                      ; сброс режима совместимости
U 0ECA,  E58A,15    ;9708               CLR LS[ERROR.MASK]                          ; очистка маски ошибки
U 0ECB,  65A0,15    ;9709               CLR LS[PREVIOUS.ERROR]                      ; очистка предыдущего номера ошибки
U 0ECC,  B640,15    ;9710               MOV LS[#1] TO WR[0]                         ; установка 1 в WR0
U 0ECD,  BE82,15    ;9711               MOV WR[0] TO LS[ERROR.NUMBER]               ; установка номера ошибки для первой ошибки
U 0ECE,  A3C0,15    ;9712               ROL WR[0]                                   ; установка 2 в WR0
U 0ECF,  3E8C,15    ;9713               MOV WR[0] TO LS[MODULE.NUM]                  ; установка кода модуля для модуля DAP
                    ;9714
U 0ED0,  0881,0C    ;9715               JSR [ASSERT.NA]                             ;
U 0ED1,  6514,15    ;9716               CLR LS[T10]
U 0ED2,  364E,15    ;9717               MOV LS[#80] TO WR[0]                        ;
U 0ED3,  C048,15    ;9718               ADD LS[#10] TO WR[0]                        ;
U 0ED4,  C044,15    ;9719               ADD LS[#4] TO WR[0]                         ;
U 0ED5,  3E10,15    ;9720               MOV WR[0] TO LS[T8]                         ; T8=94
                    ;9721     LOOP.TF.1:
U 0ED6,  1910,75    ;9722               MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]          ;
U 0ED7,  B214,15    ;9723               WRITE.MEM LS[T10]                           ; очистка ячейки памяти 94
U 0ED8,  1B11,75    ;9724               MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]          ; очистка PFR
U 0ED9,  88F2,3C    ;9725               JSR [TF.5.NOP]                              ;
U 0EDA,  08ED,CC    ;9726               JSR [TF.DECODE.1]                           ;
U 0EDB,  08EE,24    ;9727               JMP [TF.2]                                  ; продолжение, если возврат
                    ;9728     TF.DECODE.1:
U 0EDC,  0470,1F    ;9729               OPC2/DECODE,BPC/NOP,DEC.ADRS/7,IFUNC/0,OPC.SPEC/1,JCTL/JUMP.VALID ;
U 0EDD,  DB00,15    ;9730               NOP                                         ;
U 0EDE,  2F80,15    ;9731               CLR WR[0]                                   ;
U 0EDF,  369E,95    ;9732               MOV LS[ONES] TO WR[1]                       ;
U 0EE0,  0869,3C    ;9733               JSR [CHECK.RESULT]                          ; сообщение об ошибке 1
U 0EE1,  88ED,64    ;9734               JMP [LOOP.TF.1]                             ; зацикливание при ошибке, если разрешено
                    ;9735     TF.2:
U 0EE2,  FF82,15    ;9736               INC LS[ERROR.NUMBER]                        ; увеличение номера ошибки до 2
                    ;9737     LOOP.TF.2:
U 0EE3,  1910,75    ;9738               MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]          ;
U 0EE4,  B214,15    ;9739               WRITE.MEM LS[T10]                           ; очистка ячейки памяти 94
```

```
U 0EE5, 1B11,75 ;9740            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; очистка PFR
U 0EE6, 88F2,3C ;9741            JSR [TF.5.NOP]                          ;
U 0EE7, 08EE,CF ;9742            JMP.IF[JUMP.VALID] TO [TF.3]            ; переход, если правильно
U 0EE8, 2F80,15 ;9743            CLR WR[0]                               ;
U 0EE9, 369E,95 ;9744            MOV LS[ONES] TO WR[1]                   ;
U 0EEA, 0869,3C ;9745            JSR [CHECK.RESULT]                      ; сообщение об ошибке 2
U 0EEB, 88EE,34 ;9746            JMP [LOOP.TF.2]                         ; зацикливание при ошибке, если разрешено
                ;9747    TF.3:
U 0EEC, FF82,15 ;9748            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 3
                ;9749    LOOP.TF.3:
U 0EED, 1910,75 ;9750            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0EEE, B214,15 ;9751            WRITE.MEM LS[T10]                       ; очистка ячейки памяти 94
U 0EEF, 1B11,75 ;9752            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; очистка PFR
U 0EF0, 88F2,3C ;9753            JSR [TF.5.NOP]                          ;
U 0EF1, 2F87,95 ;9754            CLR WR[3]                               ;
U 0EF2, 88F2,1D ;9755            JMP.IF[JSR.VALID] TO [TF.SUB3]          ;
                ;9756              TST WR[3],                            ;
U 0EF3, 2007,B5 ;9757            DT(LONG)&SET.ALU.CC                     ; WR3 был увеличен
U 0EF4, 08EF,91 ;9758            JMP.IF[NEQ] TO [TF.4]                   ; правильно, если да
U 0EF5, 2F80,15 ;9759            CLR WR[0]                               ;
U 0EF6, 369E,95 ;9760            MOV LS[ONES] TO WR[1]                   ;
U 0EF7, 0869,3C ;9761            JSR [CHECK.RESULT]                      ; сообщение об ошибке 3
U 0EF8, 08EE,D4 ;9762            JMP [LOOP.TF.3]                         ; зацикливание при ошибке, если разрешено
                ;9763    TF.4:
U 0EF9, FF82,15 ;9764            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 4
                ;9765    LOOP.TF.4:
U 0EFA, 1910,75 ;9766            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0EFB, B214,15 ;9767            WRITE.MEM LS[T10]                       ; очистка ячейки памяти 94
U 0EFC, 1B11,75 ;9768            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; очистка PFR
U 0EFD, 88F2,3C ;9769            JSR [TF.5.NOP]                          ;
U 0EFE, 08F0,1E ;9770            JMP.IF[NO.JUMP.TST] TO [TF.4.ERR]       ; переход, если правильно
U 0EFF, 5B00,1E ;9771            SKIP                                    ; пропуск, если не было пропуска
U 0F00, 88F0,54 ;9772            JMP [TF.5]                              ; переход, если был пропуск
                ;9773    TF.4.ERR:
U 0F01, 2F80,15 ;9774            CLR WR[0]                               ;
U 0F02, 369E,95 ;9775            MOV LS[ONES] TO WR[1]                   ;
U 0F03, 0869,3C ;9776            JSR [CHECK.RESULT]                      ; сообщение об ошибке 4
U 0F04, 08EF,A4 ;9777            JMP [LOOP.TF.4]                         ; зацикливание при ошибке, если разрешено
                ;9778    TF.5:
U 0F05, FF82,15 ;9779            INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 5
                ;9780    LOOP.TF.5:
U 0F06, 1910,75 ;9781            MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0F07, B214,15 ;9782            WRITE.MEM LS[T10]                       ;
U 0F08, 1B11,75 ;9783            MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; очистка ячейки памяти 94
U 0F09, 88F2,3C ;9784            JSR [TF.5.NOP]                          ;
U 0F0A, 08F0,CC ;9785            JSR [TF.DECODE.5]                       ;
U 0F0B, 88F1,24 ;9786            JMP [TF.6]                              ; продолжение, если возврат
                ;9787    TF.DECODE.5:
U 0F0C, 8470,14 ;9788            OPC2/DECODE,BPC/NOP,DEC.ADRS/7,IFUNC/0,OPC.SPEC/1,JCTL/JUMP ;
U 0F0D, DB00,15 ;9789            NOP                                     ;
U 0F0E, 2F80,15 ;9790            CLR WR[0]                               ;
U 0F0F, 369E,95 ;9791            MOV LS[ONES] TO WR[1]                   ;
U 0F10, 0869,3C ;9792            JSR [CHECK.RESULT]                      ; сообщение об ошибке 5
U 0F11, 88F0,64 ;9793            JMP [LOOP.TF.5]                         ; зацикливание при ошибке, если разрешено
                ;9794    TF.6:
```

```
                        ;  ТЕСТ F - тест условных переходов (модуль DAP)

U 0F12, FF82,15 ;9795           INC LS[ERROR.NUMBER]                    ; увеличение номера ошибки до 6
                ;9796   LOOP.TF.6:
U 0F13, 1910,75 ;9797           MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 0F14, B214,15 ;9798           WRITE.MEM LS[T10]                       ; очистка ячейки памяти 94
U 0F15, 1B11,75 ;9799           MEM.REQ[IB.FILL] ADRS[T8] DT[LONG]      ; очистка PFR
U 0F16, 88F2,3C ;9800           JSR [TF.5.NOP]                          ;
U 0F17, 2F87,95 ;9801           CLR WR[3]                               ;
U 0F18, 8470,1D ;9802           OPC2/DECODE,BPC/NOP,DEC.ADRS/7,IFUNC/0,OPC.SPEC/1,JCTL/JSR.VALID ;
U 0F19, DB00,15 ;9803           NOP                                     ;
                ;9804             TST WR[3],                            ;
U 0F1A, 2007,B5 ;9805           DT(LONG)&SET.ALU.CC                     ; WR3 был увеличен?
U 0F1B, 08F2,91 ;9806           JMP.IF[NEQ] TO [END.TF]                 ; правильно, если да
U 0F1C, 2F80,15 ;9807           CLR WR[0]                               ;
U 0F1D, 369E,95 ;9808           MOV LS[ONES] TO WR[1]                   ;
U 0F1E, 0869,3C ;9809           JSR [CHECK.RESULT]                      ; сообщение об ошибке 6
U 0F1F, 08F1,34 ;9810           JMP [LOOP.TF.6]                         ; зацикливание при ошибке, если разрешено
U 0F20, 08F2,94 ;9811           JMP [END.TF]                            ;
                ;9812   TF.SUB3:
U 0F21, 2047,95 ;9813           INC WR[3]                               ;
U 0F22, 5B00,14 ;9814           RETURN                                  ;
                ;9815   TF.5.NOP:
U 0F23, DB00,15 ;9816           NOP                                     ; учитывается время работы памяти
U 0F24, DB00,15 ;9817           NOP                                     ;
U 0F25, DB00,15 ;9818           NOP                                     ;
U 0F26, DB00,15 ;9819           NOP                                     ;
U 0F27, DB00,15 ;9820           NOP                                     ;
U 0F28, 5B00,14 ;9821           RETURN                                  ;
                ;9822   END.TF:
```

```
                ;9823     .PAGE "ТЕСТ 10 - тест ПЗУ УПР.ФУНКЦИЕЙ АЛУ и ПМЛ УПР.ИСТ.ПРИЕМН. АЛУ (модуль DAP)"
                ;9824
                ;9825     ;**
                ;9826     ;
                ;9827     ; ОПИСАНИЕ ТЕСТА:
                ;9828     ;
                ;9829     ;   Этот тест проверяет каждую ячейку ПЗУ УПР.ФУНКЦИЕЙ АЛУ и все возможные входы
                ;9830     ;   ПМЛ УПР.ИСТ.ПРИЕМН. АЛУ, которые не были проверены в тесте F второй
                ;9831     ;   части (ENKCB). Имеется ввиду проверка, что любой микрокод инструкции управляет
                ;9832     ;   микросхемой К1804ВС1 так, как ожидается, даже если эти инструкции не были
                ;9833     ;   проверены в предыдущих тестах. Некоторые инструкции, уже проверенные в этом
                ;9834     ;   тесте, не проверяются. Тест разделен на секции в зависимости от типа прове-
                ;9835     ;   ряемой инструкции, т.е. MEM.REQ, DECODE и MOVE. В большинстве сигналов в этом
                ;9836     ;   тесте обращение к памяти будет запрещено. Будут проверяться только режимы
                ;9837     ;   К1804ВС1. Необходимо обращаться к комментариям на листинге.
                ;9838     ;
                ;9839     ; ПРЕДПОЛОЖЕНИЯ:
                ;9840     ;
                ;9841     ;   Предполагается, что все предыдущие тесты выполнены успешно.
                ;9842     ;
                ;9843     ; ШАГИ ТЕСТА:
                ;9844     ;
                ;9845     ;   См. комментарий листинга
                ;9846     ;
                ;9847     ; ОШИБКИ:
                ;9848     ;
                ;9849     ;   ДРУГИЕ ДАННЫЕ: Адрес ПЗУ УПР.ФУНКЦИЕЙ АЛУ
                ;9850     ;
                ;9851     ;   ошибка 1 - ошибка микроинструкции DECODE (BPC разрешен)
                ;9852     ;   ошибка 2 - ошибка микроинструкции DECODE (BPC запрешен)
                ;9853     ;   ошибка 3 - ошибка микроинструкции MOVE
                ;9854     ;   ошибка 4 - ошибка микроинструкции MEM.REQ
                ;9855     ;
                ;9856     ; НАЛАДКА:
                ;9857     ;
                ;9858     ;    Все ошибки, за исключением ошибки 3 с полем "ДРУГИЕ ДАННЫЕ" = C0, за-
                ;9859     ;    трагивают ПЗУ УПР.ФУНКЦИЕЙ АЛУ или ПМЛ УПР.ИСТ.ПРИЕМН. АЛУ. Предпо-
                ;9860     ;    лагаемые выходы могут быть определены из таблицы для линий управления
                ;9861     ;    АЛУ и режима (см. содержимое управляющей памяти путей данных). Таким образом,
                ;9862     ;    неисправность может быть определена до одной из двух микросхем, упомя-
                ;9863     ;    нутых выше.
                ;9864     ;
                ;9865     ;    ОШИБКА 3, поле "ДРУГИЕ ДАННЫЕ" = C0: эта ошибка указывает на неисправ-
                ;9866     ;    ности ПМЛ УПР.МЕСТНОЙ ПАМЯТИ или входов CSR 17, 18 или 19. Необходимо прове-
                ;9867     ;    рить низкий уровень этих входов CSR во время инструкции MOV MEM.DATA TO
                ;9868     ;    WR[1] XCHG TO LS [T9]. Если эти выходы правильные, вероятно, неисправна ПМЛ
                ;9869     ;    УПР.Местной памяти.
                ;9870     ;
                ;9871     ;--
                ;9872     T.10:
U 0F29, B65E,15 ;9873           MOV LS[BEGIN.TEST] TO WR[0]             ; установка бита 15 в WR0 для слова управления и
                ;9874                                                   ; состояния
U 0F2A, 3E80,15 ;9875           MOV WR[0] TO LS[CONTROL.STATUS]         ; установка бита 15 в слове управления и состояния. Бит
                ;9876                                                   ; 15 указывает начало теста для консольного процессора
U 0F2B, 10E0,15 ;9877           MISC [SET.CP.ATTN]                      ; выдача сигнала CPU ATTN для консольного процессора
```

```
                 ;9878    WAIT.T10.0:
U 0F2C, 08F2,C4 ;9879            JMP [WAIT.T10.0]                          ; зацикливание для ожидания ответа от консольного
                 ;9880                                                      ; процессора
U 0F2D, E58A,15 ;9881            CLR LS[ERROR.MASK]                        ; очистка маски ошибки
U 0F2E, 65A0,15 ;9882            CLR LS[PREVIOUS.ERROR]                    ; очистка предыдущего номера ошибки
U 0F2F, B640,15 ;9883            MOV LS[#1] TO WR[0]                       ; установка 1 в WR0
U 0F30, BE82,15 ;9884            MOV WR[0] TO LS[ERROR.NUMBER]             ; установка номера ошибки для первой ошибки
U 0F31, A3C0,15 ;9885            ROL WR[0]                                 ; установка 2 в WR0
U 0F32, 3E8C,15 ;9886            MOV WR[0] TO LS[MODULE.NUM]               ; установка кода модуля для модуля DAP
U 0F33, 8881,7C ;9887            JSR [ASSERT.OTHER]                        ; поле "ДРУГИЕ ДАННЫЕ" будет использовано
U 0F34, 6588,15 ;9888            CLR LS[ADDRESS.DATA]                      ; начальная установка "ДРУГИХ ДАННЫХ" в 0
                 ;9889     ;
                 ;9890     ; Тесты инструкции DECODE
                 ;9891     ;
                 ;9892    LOOP.T10.1.00:
U 0F35, E520,15 ;9893            CLR LS[PC]                                ; установка PC в 0
U 0F36, 8002,0E ;9894            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/00,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F37, DB00,15 ;9895            NOP                                       ;
U 0F38, 891C,8C ;9896            JSR [T10.SUB1]                            ; переход к подпрограмме
U 0F39, 0869,3C ;9897            JSR [CHECK.RESULT]                        ; проверка, что увеличенный PC записан в WR4
U 0F3A, 88F3,54 ;9898            JMP [LOOP.T10.1.00]                       ; зацикливание при ошибке, если разрешено
U 0F3B, FF88,15 ;9899            INC LS[ADDRESS.DATA]                      ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;9900    LOOP.T10.1.01:
U 0F3C, E520,15 ;9901            CLR LS[PC]                                ; установка PC в 0
U 0F3D, 0042,0E ;9902            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/04,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F3E, DB00,15 ;9903            NOP                                       ;
U 0F3F, 891C,8C ;9904            JSR [T10.SUB1]                            ; переход к подпрограмме
U 0F40, 0869,3C ;9905            JSR [CHECK.RESULT]                        ; проверка, что увеличенный PC записан в WR4
U 0F41, 88F3,C4 ;9906            JMP [LOOP.T10.1.01]                       ; зацикливание при ошибке, если разрешено
U 0F42, FF88,15 ;9907            INC LS[ADDRESS.DATA]                      ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;9908    LOOP.T10.1.02:
U 0F43, E520,15 ;9909            CLR LS[PC]                                ; установка PC в 0
U 0F44, 0082,0E ;9910            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/08,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F45, DB00,15 ;9911            NOP                                       ;
U 0F46, 891C,8C ;9912            JSR [T10.SUB1]                            ; переход к подпрограмме
U 0F47, 0869,3C ;9913            JSR [CHECK.RESULT]                        ; проверка, что увеличенный PC записан в WR4
U 0F48, 08F4,34 ;9914            JMP [LOOP.T10.1.02]                       ; зацикливание при ошибке, если разрешено
U 0F49, FF88,15 ;9915            INC LS[ADDRESS.DATA]                      ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;9916    LOOP.T10.1.03:
U 0F4A, E520,15 ;9917            CLR LS[PC]                                ; установка PC в 0
U 0F4B, 80C2,0E ;9918            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/0C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F4C, DB00,15 ;9919            NOP                                       ;
U 0F4D, 891C,8C ;9920            JSR [T10.SUB1]                            ; переход к подпрограмме
U 0F4E, 0869,3C ;9921            JSR [CHECK.RESULT]                        ; проверка, что увеличенный PC записан в WR4
U 0F4F, 08F4,A4 ;9922            JMP [LOOP.T10.1.03]                       ; зацикливание при ошибке, если разрешено
U 0F50, FF88,15 ;9923            INC LS[ADDRESS.DATA]                      ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;9924    LOOP.T10.1.04:
U 0F51, E520,15 ;9925            CLR LS[PC]                                ; установка PC в 0
U 0F52, 0102,0E ;9926            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/10,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F53, DB00,15 ;9927            NOP                                       ;
U 0F54, 891C,8C ;9928            JSR [T10.SUB1]                            ; переход к подпрограмме
U 0F55, 0869,3C ;9929            JSR [CHECK.RESULT]                        ; проверка, что увеличенный PC записан в WR4
U 0F56, 08F5,14 ;9930            JMP [LOOP.T10.1.04]                       ; зацикливание при ошибке, если разрешено
U 0F57, FF88,15 ;9931            INC LS[ADDRESS.DATA]                      ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;9932    LOOP.T10.1.05:
```

```
U 0F58, E520,15 ;9933            CLR LS[PC]                                  ; установка PC в 0
U 0F59, 8142,0E ;9934            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/14,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F5A, DB00,15 ;9935            NOP                                         ;
U 0F5B, 891C,8C ;9936            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F5C, 0869,3C ;9937            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F5D, 08F5,B4 ;9938            JMP [LOOP.T10.1.05]                         ; зацикливание при ошибке, если разрешено
U 0F5E, FF88,15 ;9939            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9940    LOOP.T10.1.06:
U 0F5F, E520,15 ;9941            CLR LS[PC]                                  ; установка PC в 0
U 0F60, 8182,0E ;9942            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/18,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F61, DB00,15 ;9943            NOP                                         ;
U 0F62, 891C,8C ;9944            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F63, 0869,3C ;9945            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F64, 88F5,F4 ;9946            JMP [LOOP.T10.1.06]                         ; зацикливание при ошибке, если разрешено
U 0F65, FF88,15 ;9947            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9948    LOOP.T10.1.07:
U 0F66, E520,15 ;9949            CLR LS[PC]                                  ; установка PC в 0
U 0F67, 01C2,0E ;9950            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/1C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F68, DB00,15 ;9951            NOP                                         ;
U 0F69, 891C,8C ;9952            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F6A, 0869,3C ;9953            JSR [CHECK.RESULT]                          ; проверка,что увеличенный PC записан в WR4
U 0F6B, 88F6,64 ;9954            JMP [LOOP.T10.1.07]                         ; зацикливание при ошибке, если разрешено
U 0F6C, FF88,15 ;9955            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9956    LOOP.T10.1.08:
U 0F6D, E520,15 ;9957            CLR LS[PC]                                  ; установка PC в 0
U 0F6E, 0202,0E ;9958            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/20,IFUNC/1,JCTL/NO.JUMP.TST
U 0F6F, DB00,15 ;9959            NOP                                         ;
U 0F70, 891C,8C ;9960            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F71, 0869,3C ;9961            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F72, 08F6,D4 ;9962            JMP [LOOP.T10.1.08]                         ; зацикливание при ошибке, если разрешено
U 0F73, FF88,15 ;9963            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9964    LOOP.T10.1.09:
U 0F74, E520,15 ;9965            CLR LS[PC]                                  ; установка PC в 0
U 0F75, 8242,0E ;9966            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/24,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F76, DB00,15 ;9967            NOP                                         ;
U 0F77, 891C,8C ;9968            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F78, 0869,3C ;9969            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F79, 88F7,44 ;9970            JMP [LOOP.T10.1.09]                         ; зацикливание при ошибке, если разрешено
U 0F7A, FF88,15 ;9971            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9972    LOOP.T10.1.0A:
U 0F7B, E520,15 ;9973            CLR LS[PC]                                  ; установка PC в 0
U 0F7C, 8282,0E ;9974            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/28,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F7D, DB00,15 ;9975            NOP                                         ;
U 0F7E, 891C,8C ;9976            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F7F, 0869,3C ;9977            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F80, 88F7,B4 ;9978            JMP [LOOP.T10.1.0A]                         ; зацикливание при ошибке, если разрешено
U 0F81, FF88,15 ;9979            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9980    LOOP.T10.1.0B:
U 0F82, E520,15 ;9981            CLR LS[PC]                                  ; установка PC в 0
U 0F83, 02C2,0E ;9982            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/2C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F84, DB00,15 ;9983            NOP                                         ;
U 0F85, 891C,8C ;9984            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F86, 0869,3C ;9985            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F87, 88F8,24 ;9986            JMP [LOOP.T10.1.0B]                         ; зацикливание при ошибке, если разрешено
U 0F88, FF88,15 ;9987            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
```

```
                ;9988    LOOP.T10.1.0C:
U 0F89, E520,15 ;9989            CLR LS[PC]                                  ; установка PC в 0
U 0F8A, 8302,0E ;9990            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/30,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F8B, DB00,15 ;9991            NOP                                         ;
U 0F8C, 891C,8C ;9992            JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F8D, 0869,3C ;9993            JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F8E, 08F8,94 ;9994            JMP [LOOP.T10.1.0C]                         ; зацикливание при ошибке, если разрешено
U 0F8F, FF88,15 ;9995            INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;9996    LOOP.T10.1.0D:
U 0F90, E520,15 ;9997            CLR LS[PC]                                  ; установка PC в 0
U 0F91, 0342,0E ;9998            OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/34,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F92, DB00,15 ;9999            NOP                                         ;
U 0F93, 891C,8C ;10000           JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F94, 0869,3C ;10001           JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F95, 08F9,04 ;10002           JMP [LOOP.T10.1.0D]                         ; зацикливание при ошибке, если разрешено
U 0F96, FF88,15 ;10003           INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10004   LOOP.T10.1.0E:
U 0F97, E520,15 ;10005           CLR LS[PC]                                  ; установка PC в 0
U 0F98, 0382,0E ;10006           OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/38,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0F99, DB00,15 ;10007           NOP                                         ;
U 0F9A, 891C,8C ;10008           JSR [T10.SUB1]                              ; переход к подпрограмме
U 0F9B, 0869,3C ;10009           JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0F9C, 08F9,74 ;10010           JMP [LOOP.T10.1.0E]                         ; зацикливание при ошибке, если разрешено
U 0F9D, FF88,15 ;10011           INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10012   LOOP.T10.1.0F:
U 0F9E, E520,15 ;10013           CLR LS[PC]                                  ; установка PC в 0
U 0F9F, 83C2,0E ;10014           OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/3C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FA0, DB00,15 ;10015           NOP                                         ;
U 0FA1, 891C,8C ;10016           JSR [T10.SUB1]                              ; переход к подпрограмме
U 0FA2, 0869,3C ;10017           JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC записан в WR4
U 0FA3, 08F9,E4 ;10018           JMP [LOOP.T10.1.0F]                         ; зацикливание при ошибке, если разрешено
U 0FA4, FF88,15 ;10019           INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10020   T10.2:
U 0FA5, B646,15 ;10021           MOV LS[#8] TO WR[0]                         ;
U 0FA6, 2100,15 ;10022           DEC WR[0]                                   ;
U 0FA7, 3E20,15 ;10023           MOV WR[0] TO LS[PC]                         ; PC=7
                ;10024    ;
                ;10025    ; Запись 8 (PC+1) в WR4
                ;10026    ;
U 0FA8, 83C2,0E ;10027           OPC2/DECODE,BPC/SAVE.PC,DEC.ADRS/3C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FA9, DB00,15 ;10028           NOP                                         ;
U 0FAA, FFB2,15 ;10029           INC LS[ERROR.NUMBER]                        ; номер ошибки = 2
                ;10030   LOOP.T10.2.10:
U 0FAB, E520,15 ;10031           CLR LS[PC]                                  ; установка PC в 0
U 0FAC, 0402,0E ;10032           OPC2/DECODE,BPC/NOP,DEC.ADRS/00,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FAD, DB00,15 ;10033           NOP                                         ;
U 0FAE, 091C,CC ;10034           JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FAF, 0869,3C ;10035           JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FB0, 08FA,B4 ;10036           JMP [LOOP.T10.2.10]                         ; зацикливание при ошибке, если разрешено
U 0FB1, FF88,15 ;10037           INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10038   LOOP.T10.2.11:
U 0FB2, E520,15 ;10039           CLR LS[PC]                                  ; установка PC в 0
U 0FB3, 8442,0E ;10040           OPC2/DECODE,BPC/NOP,DEC.ADRS/04,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FB4, DB00,15 ;10041           NOP                                         ;
U 0FB5, 091C,CC ;10042           JSR [T10.SUB2]                              ; переход к подпрограмме
```

```
U 0FB6,  0869,3C ;10043          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FB7,  88FB,24 ;10044          JMP [LOOP.T10.2.11]                         ; зацикливание при ошибке, если разрешено
U 0FB8,  FF88,15 ;10045          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10046  LOOP.T10.2.12:
U 0FB9,  E520,15 ;10047          CLR LS[PC]                                  ; установка PC в 0
U 0FBA,  8482,0E ;10048          OPC2/DECODE,BPC/NOP,DEC.ADRS/08,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FBB,  DB00,15 ;10049          NOP                                         ;
U 0FBC,  091C,CC ;10050          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FBD,  0869,3C ;10051          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FBE,  08FB,94 ;10052          JMP [LOOP.T10.2.12]                         ; зацикливание при ошибке, если разрешено
U 0FBF,  FF88,15 ;10053          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10054  LOOP.T10.2.13:
U 0FC0,  E520,15 ;10055          CLR LS[PC]                                  ; установка PC в 0
U 0FC1,  04C2,0E ;10056          OPC2/DECODE,BPC/NOP,DEC.ADRS/0C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FC2,  DB00,15 ;10057          NOP                                         ;
U 0FC3,  091C,CC ;10058          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FC4,  0869,3C ;10059          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FC5,  88FC,04 ;10060          JMP [LOOP.T10.2.13]                         ; зацикливание при ошибке, если разрешено
U 0FC6,  FF88,15 ;10061          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10062  LOOP.T10.2.14:
U 0FC7,  E520,15 ;10063          CLR LS[PC]                                  ; установка PC в 0
U 0FC8,  8502,0E ;10064          OPC2/DECODE,BPC/NOP,DEC.ADRS/10,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FC9,  DB00,15 ;10065          NOP                                         ;
U 0FCA,  091C,CC ;10066          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FCB,  0869,3C ;10067          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FCC,  08FC,74 ;10068          JMP [LOOP.T10.2.14]                         ; зацикливание при ошибке, если разрешено
U 0FCD,  FF88,15 ;10069          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10070  LOOP.T10.2.15:
U 0FCE,  E520,15 ;10071          CLR LS[PC]                                  ; установка PC в 0
U 0FCF,  0542,0E ;10072          OPC2/DECODE,BPC/NOP,DEC.ADRS/14,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FD0,  DB00,15 ;10073          NOP                                         ;
U 0FD1,  091C,CC ;10074          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FD2,  0869,3C ;10075          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FD3,  08FC,E4 ;10076          JMP [LOOP.T10.2.15]                         ; зацикливание при ошибке, если разрешено
U 0FD4,  FF88,15 ;10077          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10078  LOOP.T10.2.16:
U 0FD5,  E520,15 ;10079          CLR LS[PC]                                  ; установка PC в 0
U 0FD6,  0582,0E ;10080          OPC2/DECODE,BPC/NOP,DEC.ADRS/18,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FD7,  DB00,15 ;10081          NOP                                         ;
U 0FD8,  091C,CC ;10082          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FD9,  0869,3C ;10083          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FDA,  08FD,54 ;10084          JMP [LOOP.T10.2.16]                         ; зацикливание при ошибке, если разрешено
U 0FDB,  FF88,15 ;10085          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10086  LOOP.T10.2.17:
U 0FDC,  E520,15 ;10087          CLR LS[PC]                                  ; установка PC в 0
U 0FDD,  85C2,0E ;10088          OPC2/DECODE,BPC/NOP,DEC.ADRS/1C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FDE,  DB00,15 ;10089          NOP                                         ;
U 0FDF,  091C,CC ;10090          JSR [T10.SUB2]                              ; переход к подпрограмме
U 0FE0,  0869,3C ;10091          JSR [CHECK.RESULT]                          ; проверка, что увеличенный PC не записан в WR4
U 0FE1,  08FD,C4 ;10092          JMP [LOOP.T10.2.17]                         ; зацикливание при ошибке, если разрешено
U 0FE2,  FF88,15 ;10093          INC LS[ADDRESS.DATA]                        ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                 ;10094  LOOP.T10.2.18:
U 0FE3,  E520,15 ;10095          CLR LS[PC]                                  ; установка PC в 0
U 0FE4,  8602,0E ;10096          OPC2/DECODE,BPC/NOP,DEC.ADRS/20,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FE5,  DB00,15 ;10097          NOP                                         ;
```

```
U 0FE6, 091C,CC ;10098          JSR [T10.SUB2]                          ; переход к подпрограмме
U 0FE7, 0869,3C ;10099          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан в в WR4
U 0FE8, 08FE,34 ;10100          JMP [LOOP.T10.2.18]                     ; зацикливание при ошибке, если разрешено
U 0FE9, FF88,15 ;10101          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10102  LOOP.T10.2.19:
U 0FEA, E520,15 ;10103          CLR LS[PC]                              ; установка PC в 0
U 0FEB, 0642,0E ;10104          OPC2/DECODE,BPC/NOP,DEC.ADRS/24,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FEC, DB00,15 ;10105          NOP                                     ;
U 0FED, 091C,CC ;10106          JSR [T10.SUB2]                          ; переход к подпрограмме
U 0FEE, 0869,3C ;10107          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 0FEF, 08FE,A4 ;10108          JMP [LOOP.T10.2.19]                     ; зацикливание при.ошибке, если разрешено
U 0FF0, FF88,15 ;10109          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10110  LOOP.T10.2.1A:
U 0FF1, E520,15 ;10111          CLR LS[PC]                              ; установка PC в 0
U 0FF2, 0682,0E ;10112          OPC2/DECODE,BPC/NOP,DEC.ADRS/28,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FF3, DB00,15 ;10113          NOP                                     ;
U 0FF4, 091C,CC ;10114          JSR [T10.SUB2]                          ; переход к подпрограмме
U 0FF5, 0869,3C ;10115          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 0FF6, 08FF,14 ;10116          JMP [LOOP.T10.2.1A]                     ; зацикливание при ошибке, если разрешено
U 0FF7, FF88,15 ;10117          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10118  LOOP.T10.2.1B:
U 0FF8, E520,15 ;10119          CLR LS[PC]                              ; установка PC в 0
U 0FF9, 86C2,0E ;10120          OPC2/DECODE,BPC/NOP,DEC.ADRS/2C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 0FFA, DB00,15 ;10121          NOP                                     ;
U 0FFB, 091C,CC ;10122          JSR [T10.SUB2]                          ; преход к подпрограмме
U 0FFC, 0869,3C ;10123          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан в WR4
U 0FFD, 08FF,84 ;10124          JMP [LOOP.T10.2.1B]                     ; зацикливание при ошибке, если разрешено
U 0FFE, FF88,15 ;10125          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10126  LOOP.T10.2.1C:
U 0FFF, E520,15 ;10127          CLR LS[PC]                              ; установка PC в 0
U 1000, 0702,0E ;10128          OPC2/DECODE,BPC/NOP,DEC.ADRS/30,IFUNC/1,JCTL/NO.JUMP.TST ;
U 1001, DB00,15 ;10129          NOP                                     ;
U 1002, 091C,CC ;10130          JSR [T10.SUB2]                          ; переход к подпрограмме
U 1003, DB00,15 ;10131          NOP                                     ;
U 1004, DB00,15 ;10132          NOP                                     ;
U 1005, 0869,3C ;10133          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 1006, 88FF,F4 ;10134          JMP [LOOP.T10.2.1C]                     ; зацикливание при ошибке, если разрешено
U 1007, FF88,15 ;10135          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10136  LOOP.T10.2.1D:
U 1008, E520,15 ;10137          CLR LS[PC]                              ; установка PC в 0
U 1009, 8742,0E ;10138          OPC2/DECODE,BPC/NOP,DEC.ADRS/34,IFUNC/1,JCTL/NO.JUMP.TST ;
U 100A, DB00,15 ;10139          NOP                                     ;
U 100B, 091C,CC ;10140          JSR [T10.SUB2]                          ; переход к подпрограмме
U 100C, 0869,3C ;10141          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 100D, 8900,84 ;10142          JMP [LOOP.T10.2.1D]                     ; зацикливание при ошибке, если разрешено
U 100E, FF88,15 ;10143          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10144  LOOP.T10.2.1E:
U 100F, E520,15 ;10145          CLR LS[PC]                              ; установка PC в 0
U 1010, 8782,0E ;10146          OPC2/DECODE,BPC/NOP,DEC.ADRS/38,IFUNC/1,JCTL/NO.JUMP.TST ;
U 1011, DB00,15 ;10147          NOP                                     ;
U 1012, 091C,CC ;10148          JSR [T10.SUB2]                          ; переход к подпрограмме
U 1013, 0869,3C ;10149          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 1014, 0900,F4 ;10150          JMP [LOOP.T10.2.1E]                     ; зацикливание при ошибке, если разрешено
U 1015, FF88,15 ;10151          INC LS[ADDRESS.DATA]                    ; увеличение поля "ДРУГИЕ ДАННЫЕ"
                ;10152  LOOP.T10.2.1F:
```

```
U 1016, E520,15 ;10153          CLR LS[PC]                              ; установка PC в 0
U 1017, 07C2,0E ;10154          OPC2/DECODE,BPC/NOP,DEC.ADRS/3C,IFUNC/1,JCTL/NO.JUMP.TST ;
U 1018, DB00,15 ;10155          NOP                                     ;
U 1019, 091C,CC ;10156          JSR [T10.SUB2]                          ; переход к подпрограмме
U 101A, 0869,3C ;10157          JSR [CHECK.RESULT]                      ; проверка, что увеличенный PC не записан В WR4
U 101B, 8901,64 ;10158          JMP [LOOP.T10.2.1F]                     ; зацикливание при ошибке, если разрешено
                ;10159   ;
                ;10160   ; Тесты инструкции MOV
                ;10161   ;
                ;10162   T10.3:
U 101C, FF82,15 ;10163          INC LS[ERROR.NUMBER]                    ; номер ошибки = 3
U 101D, 364E,15 ;10164          MOV LS[#80] TO WR[0]                    ;
U 101E, 404C,15 ;10165          ADD LS[#40] TO WR[0]                    ;
U 101F, BE88,15 ;10166          MOV WR[0] TO LS[ADDRESS.DATA]           ; начальная установка значения C0 в поле "ДРУГИЕ ДАННЫЕ"
                ;10167   LOOP.T10.3.C0:
U 1020, 6522,15 ;10168          CLR LS[WR.BACKUP]                       ; начальная очистка ячейки местной памяти WR.BACKUP
U 1021, B698,15 ;10169          MOV LS[ALTER.ODD] TO WR[0]              ; восстановление тестового набора в WR0
U 1022, 364B,15 ;10170          MOV LS[#20] TO WR[2]                    ;
U 1023, BE11,15 ,10171          MOV WR[2] TO LS[T8]                     ; T8 = 20(H)
U 1024, 9811,F5 ;10172          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1025, 3022,15 ;10173          MOV MEM.DATA TO WR[0]                   ; запись В WR0
U 1026, 364C,95 ;10174          MOV LS[#40] TO WR[1]                    ; ожидаемые данные = AAAAAAAA
U 1027, 0869,3C ;10175          JSR [CHECK.RESULT]                      ; проверка операции MOV.MEM.WR
U 1028, 8902,04 ;10176          JMP [LOOP.T10.3.C0]                     ; зацикливание при ошибке, если разрешено
U 1029, 3622,15 ;10177          MOV LS[WR.BACKUP] TO WR[0]              ; выборка резервной копии
U 102A, 3698,95 ;10178          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = оригинал WR0
U 102B, 0869,3C ;10179          JSR [CHECK.RESULT]                      ; проверка, что WR0 был сохранен в ячейке 11 местной
                ;10180                                                  ; памяти
U 102C, 8902,04 ;10181          JMP [LOOP.T10.3.C0]                     ; зацикливание при ошибке, если разрешено
U 102D, 3688,15 ;10182          MOV LS[ADDRESS.DATA] TO WR[0]           ; обновление поля "ДРУГИЕ ДАННЫЕ"
U 102E, 4046,15 ;10183          ADD LS[#8] TO WR[0]                     ;
U 102F, BE88,15 ;10184          MOV WR[0] TO LS[ADDRESS.DATA]           ; установка значения C8 в поле "ДРУГИЕ ДАННЫЕ"
U 1030, 364E,15 ;10185          MOV LS[#80] TO WR[0]                    ;
U 1031, C049,15 ;10186          ADD LS[#10] TO WR[0]                    ;
U 1032, C044,15 ;10187          ADD LS[#4] TO WR[0]                     ;
U 1033, 3E10,15 ;10188          MOV WR[0] TO LS[T8]                     ; T8 = 94
                ;10189   LOOP.T10.3.C8:
U 1034, B698,15 ;10190          MOV LS[ALTER.ODD] TO WR[0]              ;
U 1035, BE14,15 ;10191          MOV WR[0] TO LS[T10]                    ; запись тестового набора в ячейку местной памяти
U 1036, 1910,75 ;10192          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 1037, B214,15 ;10193          WRITE.MEM LS[T10]                       ; запись набора в ячейку памяти 94
U 1038, 9811,F5 ;10194          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1039, 3022,15 ;10195          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 103A, 3698,95 ;10196          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 103B, 0869,3C ;10197          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 103C, 8903,44 ;10198          JMP [LOOP.T10.3.C8]                     ; зацикливание при ошибке, если разрешено
U 103D, FF88,15 ;10199          INC LS[ADDRESS.DATA]                    ;
                ;10200   LOOP.T10.3.C9:
U 103E, B698,15 ;10201          MOV LS[ALTER.ODD] TO WR[0]              ;
U 103F, 3E62,15 ;10202          MOV WR[0] TO LS[L31]                    ; запись тестового набора в ячейку местной памяти
U 1040, 1910,75 ;10203          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 1041, 3262,15 ;10204          WRITE.MEM LS[L31]                       ; запись набора в ячейку памяти 94
U 1042, 9811,F5 ;10205          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1043, 3022,15 ;10206          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 1044, 3698,95 ;10207          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
```

```
U 1045, 0869,3C ;10208          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 1046, 8903,E4 ;10209          JMP [LOOP.T10.3.C9]                     ; зацикливание при ошибке, если разрешено
U 1047, FF88,15 ;10210          INC LS[ADDRESS.DATA]                    ;
                ;10211  LOOP.T10.3.CA:
U 1048, B698,15 ;10212          MOV LS[ALTER.ODD] TO WR[0]              ;
U 1049, BEA6,15 ;10213          MOV WR[0] TO LS[L53]                    ; запись тестового набора в ячейку местной памяти
U 104A, 1910,75 ;10214          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 104B, B2A6,15 ;10215          WRITE.MEM LS[L53]                       ; запись набора в ячейку памяти 94
U 104C, 9811,F5 ;10216          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 104D, 3022,15 ;10217          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 104E, 3698,95 ;10218          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 104F, 0869,3C ;10219          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 1050, 0904,84 ;10220          JMP [LOOP.T10.3.CA]                     ; зацикливание при ошибке, если разрешено
U 1051, FF88,15 ;10221          INC LS[ADDRESS.DATA]                    ;
                ;10222  LOOP.T10.3.CB:
U 1052, B698,15 ;10223          MOV LS[ALTER.ODD] TO WR[0]              ;
U 1053, 3EE6,15 ;10224          MOV WR[0] TO LS[L73]                    ; запись тестового набора в ячейку местной памяти
U 1054, 1910,75 ;10225          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 1055, 32E6,15 ;10226          WRITE.MEM LS[L73]                       ; запись набора в ячейку памяти 94
U 1056, 9811,F5 ;10227          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1057, 3022,15 ;10228          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 1058, 3698,95 ;10229          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 1059, 0869,3C ;10230          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 105A, 8905,24 ;10231          JMP [LOOP.T10.3.CB]                     ; зацикливание при ошибке, если разрешено
U 105B, FF88,15 ;10232          INC LS[ADDRESS.DATA]                    ;
                ;10233  LOOP.T10.3.CC:
U 105C, B698,15 ;10234          MOV LS[ALTER.ODD] TO WR[0]              ;
U 105D, BF20,15 ;10235          MOV WR[0] TO LS[L90]                    ; запись тестового набора в ячейку местной памяти
U 105E, 1910,75 ;10236          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 105F, B320,15 ;10237          WRITE.MEM LS[L90]                       ; запись набора в ячейку памяти 94
U 1060, 9811,F5 ;10238          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1061, 3022,15 ;10239          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 1062, 3698,95 ;10240          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 1063, 0869,3C ;10241          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 1064, 0905,C4 ;10242          JMP [LOOP.T10.3.CC]                     ; зацикливание при ошибке, если разрешено
U 1065, FF88,15 ;10243          INC LS[ADDRESS.DATA]                    ;
                ;10244  LOOP.T10.3.CD:
U 1066, B698,15 ;10245          MOV LS[ALTER.ODD] TO WR[0]              ;
U 1067, 3F60,15 ;10246          MOV WR[0] TO LS[LB0]                    ; запись тестового набора в ячейку местной памяти
U 1068, 1910,75 ;10247          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 1069, 3360,15 ;10248          WRITE.MEM LS[LB0]                       ; запись набора в ячейку памяти 94
U 106A, 9811,F5 ;10249          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 106B, 3022,15 ;10250          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 106C, 3698,95 ;10251          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
U 106D, 0869,3C ;10252          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 106E, 0906,64 ;10253          JMP [LOOP.T10.3.CD]                     ; зацикливание при ошибке, если разрешено
U 106F, FF88,15 ;10254          INC LS[ADDRESS.DATA]                    ;
                ;10255  LOOP.T10.3.CE:
U 1070, B698,15 ;10256          MOV LS[ALTER.ODD] TO WR[0]              ;
U 1071, 3FA0,15 ;10257          MOV WR[0] TO LS[LD0]                    ; запись тестового набора в ячейку местной памяти
U 1072, 1910,75 ;10258          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 1073, 33A0,15 ;10259          WRITE.MEM LS[LD0]                       ; запись набора в ячейку памяти 94
U 1074, 9811,F5 ;10260          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1075, 3022,15 ;10261          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 1076, 3698,95 ;10262          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = AAAAAAAA
```

```
U 1077, 0869,3C ;10263          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 1078, 8907,04 ;10264          JMP [LOOP.T10.3.CE]                     ; зацикливание при ошибке, если разрешено
U 1079, FF88,15 ;10265          INC LS[ADDRESS.DATA]                    ;
                ;10266  LOOP.T10.3.CF:
U 107A, B698,15 ;10267          MOV LS[ALTER.ODD] TO WR[0]              ;
U 107B, BFE0,15 ;10268          MOV WR[0] TO LS[LF0]                    ; запись тестового набора в ячейку местной памяти
U 107C, 1910,75 ;10269          MEM.REQ[WRITE.P] ADRS[T8] DT[LONG]      ;
U 107D, B3E0,15 ;10270          WRITE.MEM LS[LF0]                       ; запись набора в ячейку памяти 94
U 107E, 9811,F5 ;10271          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 107F, 3022,15 ;10272          MOV MEM.DATA TO WR[0]                   ; чтение ячейки памяти 94
U 1080, 3698,95 ;10273          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные .= AAAAAAAA
U 1081, 0869,3C ;10274          JSR [CHECK.RESULT]                      ; проверка операции MOV.LS.MEM
U 1082, 8907,A4 ;10275          JMP [LOOP.T10.3.CF]                     ; зацикливание при ошибке, если разрешено
U 1083, FF88,15 ;10276          INC LS[ADDRESS.DATA]                    ;
U 1084, 3688,15 ;10277          MOV LS[ADDRESS.DATA] TO WR[0]           ;
U 1085, C048,15 ;10278          ADD LS[#10] TO WR[0]                    ;
U 1086, BE88,15 ;10279          MOV WR[0] TO LS[ADDRESS.DATA]           ; обновление поля "ДРУГИЕ ДАННЫЕ" значением E0
U 1087, 3644,15 ;10280          MOV LS[#4] TO WR[0]                     ;
U 1088, 3E10,15 ;10281          MOV WR[0] TO LS[T8]                     ; T8 = 4
U 1089, 9811,F5 ;10282          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 108A, 3816,15 ;10283          MOV MEM.DATA TO LS[T11]                 ;
                ;10284  LOOP.T10.3.E0:
U 108B, 9811,F5 ;10285          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 108C, B814,15 ;10286          MOV MEM.DATA TO LS[T10]                 ; запись в ячейку местной памяти
U 108D, 3614,15 ;10287          MOV LS[T10] TO WR[0]                    ; выборка полученных данных
U 108E, 3616,95 ;10288          MOV LS[T11] TO WR[1]                    ; ожидаемые данные = 00FF00FF
U 108F, 0869,3C ;10289          JSR [CHECK.RESULT]                      ; проверка операции MOV.MEM.LS
U 1090, 0908,B4 ;10290          JMP [LOOP.T10.3.E0]                     ; зацикливание при ошибке, если разрешено
U 1091, FF88,15 ;10291          INC LS[ADDRESS.DATA]                    ;
                ;10292  LOOP.T10.3.E1:
U 1092, 9811,F5 ;10293          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 1093, 3862,15 ;10294          MOV MEM.DATA TO LS[L31]                 ; запись в ячейку местной памяти
U 1094, B662,15 ;10295          MOV LS[L31] TO WR[0]                    ; выборка полученных данных
U 1095, 3616,95 ;10296          MOV LS[T11] TO WR[1]                    ; ожидаемые данные = 00FF00FF
U 1096, 0869,3C ;10297          JSR [CHECK.RESULT]                      ; проверка операции MOV.MEM.LS
U 1097, 8909,24 ;10298          JMP [LOOP.T10.3.E1]                     ; зацикливание при ошибке, если разрешено
U 1098, FF88,15 ;10299          INC LS[ADDRESS.DATA]                    ;
                ;10300  LOOP.T10.3.E2:
U 1099, 9811,F5 ;10301          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 109A, B8A6,15 ;10302          MOV MEM.DATA TO LS[L53]                 ; запись в ячейку местной памяти
U 109B, 36A6,15 ;10303          MOV LS[L53] TO WR[0]                    ; выборка полученных данных
U 109C, 3616,95 ;10304          MOV LS[T11] TO WR[1]                    ; ожидаемые данные = 00FF00FF
U 109D, 0869,3C ;10305          JSR [CHECK.RESULT]                      ; проверка операции MOV.MEM.LS
U 109E, 0909,94 ;10306          JMP [LOOP.T10.3.E2]                     ; зацикливание при ошибке, если разрешено
U 109F, FF88,15 ;10307          INC LS[ADDRESS.DATA]                    ;
                ;10308  LOOP.T10.3.E3:
U 10A0, 9811,F5 ;10309          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
U 10A1, 38E6,15 ;10310          MOV MEM.DATA TO LS[L73]                 ; запись в ячейку местной памяти
U 10A2, 86E6,15 ;10311          MOV LS[L73] TO WR[0]                    ; выборка полученных данных
U 10A3, 3616,95 ;10312          MOV LS[T11] TO WR[1]                    ; ожидаемые данные = 00FF00FF
U 10A4, 0869,3C ;10313          JSR [CHECK.RESULT]                      ; проверка операции MOV.MEM.LS
U 10A5, 090A,04 ;10314          JMP [LOOP.T10.3.E3]                     ; зацикливание при ошибке, если разрешено
U 10A6, FF88,15 ;10315          INC LS[ADDRESS.DATA]                    ;
                ;10316  LOOP.T10.3.E4:
U 10A7, 9811,F5 ;10317          MEM.REQ[READ.P] ADRS[T8] DT[LONG]       ;
```

```
U 10A8, B920,15 ;10318          MOV MEM.DATA TO LS[L90]              ; запись в ячейку местной памяти
U 10A9, 3720,15 ;10319          MOV LS[L90] TO WR[0]                 ; выборка полученных данных
U 10AA, 3616,95 ;10320          MOV LS[T11] TO WR[1]                 ; ожидаемые данные = 00FF00FF
U 10AB, 0869,3C ;10321          JSR [CHECK.RESULT]                   ; проверка операции MOV.MEM.LS
U 10AC, 890A,74 ;10322          JMP [LOOP.T10.3.E4]                  ; зацикливание при ошибке, если разрешено
U 10AD, FF88,15 ;10323          INC LS[ADDRESS.DATA]                 ;
                ;10324  LOOP.T10.3.E5:
U 10AE, 9811,F5 ;10325          MEM.REQ[READ.P] ADRS[T8] DT[LONG]    ;
U 10AF, 3960,15 ;10326          MOV MEM.DATA TO LS[LB0]              ; запись в ячейку местной памяти
U 10B0, B760,15 ;10327          MOV LS[LB0] TO WR[0]                 ; выборка полученных данных
U 10B1, 3616,95 ;10328          MOV LS[T11] TO WR[1]                 ; ожидаемые данные = 00FF00FF
U 10B2, 0869,3C ;10329          JSR [CHECK.RESULT]                   ; проверка операции MOV.MEM.LS
U 10B3, 890A,E4 ;10330          JMP [LOOP.T10.3.E5]                  ; зацикливание при ошибке, если разрешено
U 10B4, FF88,15 ;10331          INC LS[ADDRESS.DATA]                 ;
                ;10332  LOOP.T10.3.E6:
U 10B5, 9811,F5 ;10333          MEM.REQ[READ.P] ADRS[T8] DT[LONG]    ;
U 10B6, 39A0,15 ;10334          MOV MEM.DATA TO LS[LD0]              ; запись в ячейку местной памяти
U 10B7, B7A0,15 ;10335          MOV LS[LD0] TO WR[0]                 ; выборка полученных данных
U 10B8, 3616,95 ;10336          MOV LS[T11] TO WR[1]                 ; ожидаемые данные = 00FF00FF
U 10B9, 0869,3C ;10337          JSR [CHECK.RESULT]                   ; проверка операции MOV.MEM.LS
U 10BA, 890B,54 ;10338          JMP [LOOP.T10.3.E6]                  ; зацикливание при ошибке, если разрешено
U 10BB, FF88,15 ;10339          INC LS[ADDRESS.DATA]                 ;
                ;10340  LOOP.T10.3.E7:
U 10BC, 9811,F5 ;10341          MEM.REQ[READ.P] ADRS[T8] DT[LONG]    ;
U 10BD, B9E0,15 ;10342          MOV MEM.DATA TO LS[LF0]              ; запись в ячейку местной памяти
U 10BE, 37E0,15 ;10343          MOV LS[LF0] TO WR[0]                 ; выборка полученных данных
U 10BF, 3616,95 ;10344          MOV LS[T11] TO WR[1]                 ; ожидаемые данные = 00FF00FF
U 10C0, 0869,3C ;10345          JSR [CHECK.RESULT]                   ; проверка операции MOV.MEM.LS
U 10C1, 890B,C4 ;10346          JMP [LOOP.T10.3.E7]                  ; зацикливание при ошибке, если разрешено
U 10C2, FF88,15 ;10347          INC LS[ADDRESS.DATA]                 ;
                ;10348  LOOP.T10.3.E8:
U 10C3, B698,15 ;10349          MOV LS[ALTER.ODD] TO WR[0]           ; запись тестового набора в WR0
U 10C4, 3A14,15 ;10350          MOV FPA TO LS[T10]                   ;
U 10C5, 3698,95 ;10351          MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные = WR0 не изменен
U 10C6, 0869,3C ;10352          JSR [CHECK.RESULT]                   ; проверка операции MOV.ACC.LS
U 10C7, 090C,34 ;10353          JMP [LOOP.T10.3.E8]                  ; зацикливание при ошибке, если разрешено
U 10C8, FF88,15 ;10354          INC LS[ADDRESS.DATA]                 ;
                ;10355  LOOP.T10.3.E9:
U 10C9, B698,15 ;10356          MOV LS[ALTER.ODD] TO WR[0]           ; запись тестового набора в WR0
U 10CA, BA62,15 ;10357          MOV FPA TO LS[L31]                   ;
U 10CB, 3698,95 ;10358          MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные = WR0 не изменен
U 10CC, 0869,3C ;10359          JSR [CHECK.RESULT]                   ; проверка операции MOV.ACC.LS
U 10CD, 090C,94 ;10360          JMP [LOOP.T10.3.E9]                  ; зацикливание при ошибке, если разрешено
U 10CE, FF88,15 ;10361          INC LS[ADDRESS.DATA]                 ;
                ;10362  LOOP.T10.3.EA:
U 10CF, B698,15 ;10363          MOV LS[ALTER.ODD] TO WR[0]           ; запись тестового набора в WR0
U 10D0, 3AA6,15 ;10364          MOV FPA TO LS[L53]                   ;
U 10D1, 3698,95 ;10365          MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные = WR0 не изменен
U 10D2, 0869,3C ;10366          JSR [CHECK.RESULT]                   ; проверка операции MOV.ACC.LS
U 10D3, 090C,F4 ;10367          JMP [LOOP.T10.3.EA]                  ; зацикливание при ошибке, если разрешено
U 10D4, FF88,15 ;10368          INC LS[ADDRESS.DATA]                 ;
                ;10369  LOOP.T10.3.EB:
U 10D5, B698,15 ;10370          MOV LS[ALTER.ODD] TO WR[0]           ; запись тестового набора в WR0
U 10D6, BAE6,15 ;10371          MOV FPA TO LS[L73]                   ;
U 10D7, 3698,95 ;10372          MOV LS[ALTER.ODD] TO WR[1]           ; ожидаемые данные = WR0 не изменен
```

```
U 10D8, 0869,3C ;10373          JSR [CHECK.RESULT]                      ; проверка операции MOV.ACC.LS
U 10D9, 890D,54 ;10374          JMP [LOOP.T10.3.EB]                     ; зацикливание при ошибке, если разрешено
U 10DA, FF88,15 ;10375          INC LS[ADDRESS.DATA]                    ;
                ;10376  LOOP.T10.3.EC:
U 10DB, B698,15 ;10377          MOV LS[ALTER.ODD] TO WR[0]              ; запись тестового набора в WR0
U 10DC, 3B20,15 ;10378          MOV FPA TO LS[L90]                      ;
U 10DD, 3698,95 ;10379          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные  = WR0 не изменен
U 10DE, 0869,3C ;10380          JSR [CHECK.RESULT]                      ; проверка операции MOV.ACC.LS
U 10DF, 090D,B4 ;10381          JMP [LOOP.T10.3.EC]                     ; зацикливание при ошибке, если разрешено
U 10E0, FF88,15 ;10382          INC LS[ADDRESS.DATA]                    ;
                ;10383  LOOP.T10.3.ED:
U 10E1, B698,15 ;10384          MOV LS[ALTER.ODD] TO WR[0]              ; запись тестового набора в WR0
U 10E2, BB60,15 ;10385          MOV FPA TO LS[LB0]                      ;
U 10E3, 3698,95 ;10386          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = WR0 не изменен
U 10E4, 0869,3C ;10387          JSR [CHECK.RESULT]                      ; проверка операции MOV.ACC.LS
U 10E5, 090E,14 ;10388          JMP [LOOP.T10.3.ED]                     ; зацикливание при ошибке, если разрешено
U 10E6, FF88,15 ;10389          INC LS[ADDRESS.DATA]                    ;
                ;10390  LOOP.T10.3.EE:
U 10E7, B698,15 ;10391          MOV LS[ALTER.ODD] TO WR[0]              ; запись тестового набора в WR0
U 10E8, BBA0,15 ;10392          MOV FPA TO LS[LD0]                      ;
U 10E9, 3698,95 ;10393          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = WR0 не изменен
U 10EA, 0869,3C ;10394          JSR [CHECK.RESULT]                      ; проверка операции MOV.ACC.LS
U 10EB, 090E,74 ;10395          JMP [LOOP.T10.3.EE]                     ; зацикливание при ошибке, если разрешено
U 10EC, FF88,15 ;10396          INC LS[ADDRESS.DATA]                    ;
                ;10397  LOOP.T10.3.EF:
U 10ED, B698,15 ;10398          MOV LS[ALTER.ODD] TO WR[0]              ; запись тестового набора в WR0
U 10EE, 3BE0,15 ;10399          MOV FPA TO LS[LF0]                      ;
U 10EF, 3698,95 ;10400          MOV LS[ALTER.ODD] TO WR[1]              ; ожидаемые данные = WR0 не изменен
U 10F0, 0869,3C ;10401          JSR [CHECK.RESULT]                      ; проверка операции MOV.ACC.LS
U 10F1, 090E,D4 ;10402          JMP [LOOP.T10.3.EF]                     ; зацикливание при ошибке, если разрешено
                ;10403   ;
                ;10404   ; Тесты инструкции MEM.REQ
                ;10405   ;
                ;10406  T10.4:
U 10F2, 3660,15 ;10407          MOV LS[BIT16] TO WR[0]                  ;
U 10F3, 476C,15 ;10408          BIS LS[BIT22] TO WR[0]                  ; установка слова управления и состояния для запрета
                ;10409                                                  ; обращений к памяти
U 10F4, 3E80,15 ;10410          MOV WR[0] TO LS[CONTROL.STATUS]         ; запрет обращений к памяти
U 10F5, 10E0,15 ;10411          MISC [SET.CP.ATTN]                      ; вызов консольного процессора
                ;10412  WAIT.T10.4:
U 10F6, 090F,64 ;10413          JMP [WAIT.T10.4]                        ; ожидание ответа консольного процессора
U 10F7, FF82,15 ;10414          INC LS[ERROR.NUMBER]                    ; номер ошибки = 4
U 10F8, B64C,15 ;10415          MOV LS[#40] TO WR[0]                    ;
U 10F9, 404A,15 ;10416          ADD LS[#20] TO WR[0]                    ;
U 10FA, BE88,15 ;10417          MOV WR[0] TO LS[ADDRESS.DATA]           ; "ДРУГИЕ ДАННЫЕ" = 60(H)
U 10FB, 3650,15 ;10418          MOV LS[#100] TO WR[0]                   ;
U 10FC, BE14,15 ;10419          MOV WR[0] TO LS[T10]                    ; восстановление ячеек местной памяти
U 10FD, 3E62,15 ;10420          MOV WR[0] TO LS[L31]                    ;
U 10FE, BEA6,15 ;10421          MOV WR[0] TO LS[L53]                    ;
U 10FF, 3EE6,15 ;10422          MOV WR[0] TO LS[L73]                    ;
U 1100, B650,95 ;10423          MOV LS[#100] TO WR[1]                   ; ожидаемые данные = 100(H)
                ;10424  LOOP.T10.4.60:
U 1101, 1815,F5 ;10425          MEM.REQ[READ.P] ADRS[T10] DT[LONG]      ;
U 1102, 3400,95 ;10426          MOV XWR[VAR] TO Q                       ;
U 1103, A180,15 ;10427          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
```

```
U 1104, 0869,3C ;10428          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1105, 0910,14 ;10429          JMP [LOOP.T10.4.60]                     ; зацикливание при ошибке, если разрешено
U 1106, FF88,15 ;10430          INC LS[ADDRESS.DATA]                    ;
                ;10431  LOOP.T10.4.61:
U 1107, 9863,F5 ;10432          MEM.REQ[READ.P] ADRS[L31] DT[LONG]      ;
U 1108, 3400,95 ;10433          MOV XWR[VAR] TO Q                       ;
U 1109, A180,15 ;10434          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 110A, 0869,3C ;10435          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 110B, 0910,74 ;10436          JMP [LOOP.T10.4.61]                     ; зацикливание при ошибке, если разрешено
U 110C, FF88,15 ;10437          INC LS[ADDRESS.DATA]                    ;
                ;10438  LOOP.T10.4.62:
U 110D, 18A7,F5 ;10439          MEM.REQ[READ.P] ADRS[L53] DT[LONG]      ;
U 110E, 3400,95 ;10440          MOV XWR[VAR] TO Q                       ;
U 110F, A180,15 ;10441          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1110, 0869,3C ;10442          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1111, 0910,D4 ;10443          JMP [LOOP.T10.4.62]                     ; зацикливание при ошибке, если разрешено
U 1112, FF88,15 ;10444          INC LS[ADDRESS.DATA]                    ;
                ;10445  LOOP.T10.4.63:
U 1113, 98E7,F5 ;10446          MEM.REQ[READ.P] ADRS[L73] DT[LONG]      ;
U 1114, 3400,95 ;10447          MOV XWR[VAR] TO Q                       ;
U 1115, A180,15 ;10448          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1116, 0869,3C ;10449          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1117, 0911,34 ;10450          JMP [LOOP.T10.4.63]                     ; зацикливание при ошибке, если разрешено
U 1118, FF88,15 ;10451          INC LS[ADDRESS.DATA]                    ;
                ;10452  LOOP.T10.4.64:
U 1119, 9914,75 ;10453          MEM.REQ[WRITE.P] ADRS[T10] DT[LONG]     ;
U 111A, 3400,95 ;10454          MOV XWR[VAR] TO Q                       ;
U 111B, A180,15 ;10455          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 111C, 0869,3C ;10456          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 111D, 0911,94 ;10457          JMP [LOOP.T10.4.64]                     ; зацикливание при ошибке, если разрешено
U 111E, FF88,15 ;10458          INC LS[ADDRESS.DATA]                    ;
                ;10459  LOOP.T10.4.65:
U 111F, 1962,75 ;10460          MEM.REQ[WRITE.P] ADRS[L31] DT[LONG]     ;
U 1120, 3400,95 ;10461          MOV XWR[VAR] TO Q                       ;
U 1121, A180,15 ;10462          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1122, 0869,3C ;10463          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1123, 0911,F4 ;10464          JMP [LOOP.T10.4.65]                     ; зацикливание при ошибке, если разрешено
U 1124, FF88,15 ;10465          INC LS[ADDRESS.DATA]                    ;
                ;10466  LOOP.T10.4.66:
U 1125, 99A6,75 ;10467          MEM.REQ[WRITE.P] ADRS[L53] DT[LONG]     ;
U 1126, 3400,95 ;10468          MOV XWR[VAR] TO Q                       ;
U 1127, A180,15 ;10469          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1128, 0869,3C ;10470          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1129, 0912,54 ;10471          JMP [LOOP.T10.4.66]                     ; зацикливание при ошибке, если разрешено
U 112A, FF88,15 ;10472          INC LS[ADDRESS.DATA]                    ;
                ;10473  LOOP.T10.4.67:
U 1128, 19E6,75 ;10474          MEM.REQ[WRITE.P] ADRS[L73] DT[LONG]     ;
U 112C, 3400,95 ;10475          MOV XWR[VAR] TO Q                       ;
U 112D, A180,15 ;10476          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 112E, 0869,3C ;10477          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 112F, 8912,B4 ;10478          JMP [LOOP.T10.4.67]                     ; зацикливание при ошибке, если разрешено
U 1130, FF88,15 ;10479          INC LS[ADDRESS.DATA]                    ;
                ;10480  LOOP.T10.4.68:
U 1131, 9A14,75 ;10481          MEM.REQ[READ.V.NOCHK] ADRS[T10] DT[LONG] ;
U 1132, 3400,95 ;10482          MOV XWR[VAR] TO Q                       ;
```

```
U 1133, A180,15 ;10483          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1134, 0869,3C ;10484          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1135, 0913,14 ;10485          JMP [LOOP.T10.4.6B]                     ; зацикливание при ошибке, если разрешено
U 1136, FF88,15 ;10486          INC LS[ADDRESS.DATA]                    ;
                ;10487  LOOP.T10.4.69:
U 1137, 1A62,75 ;10488          MEM.REQ[READ.V.NOCHK] ADRS[L31] DT[LONG] ;
U 1138, 3400,95 ;10489          MOV XWR[VAR] TO Q                       ;
U 1139, A180,15 ;10490          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 113A, 0869,3C ;10491          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 113B, 0913,74 ;10492          JMP [LOOP.T10.4.69]                     ; зацикливание при ошибке, если разрешено
U 113C, FF88,15 ;10493          INC LS[ADDRESS.DATA]                    ;
                ;10494  LOOP.T10.4.6A:
U 113D, 9AA6,75 ;10495          MEM.REQ[READ.V.NOCHK] ADRS[L53] DT[LONG] ;
U 113E, 3400,95 ;10496          MOV XWR[VAR] TO Q                       ;
U 113F, A180,15 ;10497          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1140, 0869,3C ;10498          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1141, 0913,D4 ;10499          JMP [LOOP.T10.4.6A]                     ; зацикливание при ошибке, если разрешено
U 1142, FF88,15 ;10500          INC LS[ADDRESS.DATA]                    ;
                ;10501  LOOP.T10.4.6B:
U 1143, 1AE6,75 ;10502          MEM.REQ[READ.V.NOCHK] ADRS[L73] DT[LONG] ;
U 1144, 3400,95 ;10503          MOV XWR[VAR] TO Q                       ;
U 1145, A180,15 ;10504          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1146, 0869,3C ;10505          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1147, 0914,34 ;10506          JMP [LOOP.T10.4.6B]                     ; зацикливание при ошибке, если разрешено
U 1148, FF88,15 ;10507          INC LS[ADDRESS.DATA]                    ;
                ;10508  LOOP.T10.4.6C:
U 1149, 1B14,75 ;10509          MEM.REQ[WRITE.V.NOCHK] ADRS[T10] DT[LONG] ;
U 114A, 3400,95 ;10510          MOV XWR[VAR] TO Q                       ;
U 114B, A180,15 ;10511          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 114C, 0869,3C ;10512          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 114D, 0914,94 ;10513          JMP [LOOP.T10.4.6C]                     ; зацикливание при ошибке, если разрешено
U 114E, FF88,15 ;10514          INC LS[ADDRESS.DATA]                    ;
                ;10515  LOOP.T10.4.6D:
U 114F, 9B62,75 ;10516          MEM.REQ[WRITE.V.NOCHK] ADRS[L31] DT[LONG] ;
U 1150, 3400,95 ;10517          MOV XWR[VAR] TO Q                       ;
U 1151, A180,15 ;10518          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1152, 0869,3C ;10519          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1153, 0914,F4 ;10520          JMP [LOOP.T10.4.6D]                     ; зацикливание при ошибке, если разрешено
U 1154, FF88,15 ;10521          INC LS[ADDRESS.DATA]                    ;
                ;10522  LOOP.T10.4.6E:
U 1155, 1BA6,75 ;10523          MEM.REQ[WRITE.V.NOCHK] ADRS[L53] DT[LONG] ;
U 1156, 3400,95 ;10524          MOV XWR[VAR] TO Q                       ;
U 1157, A180,15 ;10525          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1158, 0869,3C ;10526          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1159, 8915,54 ;10527          JMP [LOOP.T10.4.6E]                     ; зацикливание при ошибке, если разрешено
U 115A, FF88,15 ;10528          INC LS[ADDRESS.DATA]                    ;
                ;10529  LOOP.T10.4.6F:
U 115B, 9BE6,75 ;10530          MEM.REQ[WRITE.V.NOCHK] ADRS[L73] DT[LONG] ;
U 115C, 3400,95 ;10531          MOV XWR[VAR] TO Q                       ;
U 115D, A180,15 ;10532          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 115E, 0869,3C ;10533          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 115F, 0915,B4 ;10534          JMP [LOOP.T10.4.6F]                     ; зацикливание при ошибке, если разрешено
U 1160, FF88,15 ;10535          INC LS[ADDRESS.DATA]                    ;
                ;10536  LOOP.T10.4.70:
U 1161, 9C14,75 ;10537          MEM.REQ[OCTA.WRITE.P] ADRS[T10] DT[LONG] ;
```

```
U 1162, 3400,95 ;10538          MOV XWR[VAR] TO Q                       ;
U 1163, A180,15 ;10539          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1164, 0869,3C ;10540          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1165, 0916,14 ;10541          JMP [LOOP.T10.4.70]                     ; зацикливание при ошибке, если разрешено
U 1166, FF88,15 ;10542          INC LS[ADDRESS.DATA]                    ;
                ;10543  LOOP.T10.4.71:
U 1167, 1C62,75 ;10544          MEM.REQ[OCTA.WRITE.P] ADRS[L31] DT[LONG] ;
U 1168, 3400,95 ;10545          MOV XWR[VAR] TO Q                       ;
U 1169, A180,15 ;10546          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 116A, 0869,3C ;10547          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 116B, 0916,74 ;10548          JMP [LOOP.T10.4.71]                     ; зацикливание при ошибке, если разрешено
U 116C, FF88,15 ;10549          INC LS[ADDRESS.DATA]                    ;
                ;10550  LOOP.T10.4.72:
U 116D, 9CA6,75 ;10551          MEM.REQ[OCTA.WRITE.P] ADRS[L53] DT[LONG] ;
U 116E, 3400,95 ;10552          MOV XWR[VAR] TO Q                       ;
U 116F, A180,15 ;10553          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1170, 0869,3C ;10554          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1171, 0916,D4 ;10555          JMP [LOOP.T10.4.72]                     ; зацикливание при ошибке, если разрешено
U 1172, FF88,15 ;10556          INC LS[ADDRESS.DATA]                    ;
                ;10557  LOOP.T10.4.73:
U 1173, 1CE6,75 ;10558          MEM.REQ[OCTA.WRITE.P] ADRS[L73] DT[LONG] ;
U 1174, 3400,95 ;10559          MOV XWR[VAR] TO Q                       ;
U 1175, A180,15 ;10560          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1176, 0869,3C ;10561          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1177, 0917,34 ;10562          JMP [LOOP.T10.4.73]                     ; зацикливание при ошибке, если разрешено
U 1178, FF88,15 ;10563          INC LS[ADDRESS.DATA]                    ;
                ;10564  LOOP.T10.4.74:
U 1179, 1D14,75 ;10565          MEM.REQ[READ.MAINT.ADRS] ADRS[T10] DT[LONG] ;
U 117A, 3400,95 ;10566          MOV XWR[VAR] TO Q                       ;
U 117B, A180,15 ;10567          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 117C, 0869,3C ;10568          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 117D, 0917,94 ;10569          JMP [LOOP.T10.4.74]                     ; зацикливание при ошибке, если разрешено
U 117E, FF88,15 ;10570          INC LS[ADDRESS.DATA]                    ;
                ;10571  LOOP.T10.4.75:
U 117F, 9D62,75 ;10572          MEM.REQ[READ.MAINT.ADRS] ADRS[L31] DT[LONG] ;
U 1180, 3400,95 ;10573          MOV XWR[VAR] TO Q                       ;
U 1181, A180,15 ;10574          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1182, 0869,3C ;10575          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1183, 0917,F4 ;10576          JMP [LOOP.T10.4.75]                     ; зацикливание при ошибке, если разрешено
U 1184, FF88,15 ;10577          INC LS[ADDRESS.DATA]                    ;
                ;10578  LOOP.T10.4.76:
U 1185, 1DA6,75 ;10579          MEM.REQ[READ.MAINT.ADRS] ADRS[L53] DT[LONG] ;
U 1186, 3400,95 ;10580          MOV XWR[VAR] TO Q                       ;
U 1187, A180,15 ;10581          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 1188, 0869,3C ;10582          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 1189, 0918,54 ;10583          JMP [LOOP.T10.4.76]                     ; зацикливание при ошибке, если разрешено
U 118A, FF88,15 ;10584          INC LS[ADDRESS.DATA]                    ;
                ;10585  LOOP.T10.4.77:
U 118B, 9DE6,75 ;10586          MEM.REQ[READ.MAINT.ADRS] ADRS[L73] DT[LONG] ;
U 118C, 3400,95 ;10587          MOV XWR[VAR] TO Q                       ;
U 118D, A180,15 ;10588          MOV Q TO WR[0]                          ; выборка виртуального адреса из WR5
U 118E, 0869,3C ;10589          JSR [CHECK.RESULT]                      ; проверка, что виртуальный адрес был записан в WR5
U 118F, 8918,B4 ;10590          JMP [LOOP.T10.4.77]                     ; зацикливание при ошибке, если разрешено
U 1190, FF88,15 ;10591          INC LS[ADDRESS.DATA]                    ;
                ;10592  LOOP.T10.4.78:
```

```
U 1191, 1E14,75 ;10593          MEM.REQ[OCTA.READ.P] ADRS[T10] DT[LONG];
U 1192, 3400,95 ;10594          MOV XWR[VAR] TO Q                     ;
U 1193, A180,15 ;10595          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 1194, 0869,3C ;10596          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 1195, 0919,14 ;10597          JMP [LOOP.T10.4.78]                   ; зацикливание при ошибке, если разрешено
U 1196, FF88,15 ;10598          INC LS[ADDRESS.DATA]                  ;
                ;10599  LOOP.T10.4.79:
U 1197, 9E62,75 ;10600          MEM.REQ[OCTA.READ.P] ADRS[L31] DT[LONG];
U 1198, 3400,95 ;10601          MOV XWR[VAR] TO Q                     ;
U 1199, A180,15 ;10602          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 119A, 0869,3C ;10603          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 119B, 0919,74 ;10604          JMP [LOOP.T10.4.79]                   ; зацикливание при ошибке, если разрешено
U 119C, FF88,15 ;10605          INC LS[ADDRESS.DATA]                  ;
                ;10606  LOOP.T10.4.7A:
U 119D, 1EA6,75 ;10607          MEM.REQ[OCTA.READ.P] ADRS[L53] DT[LONG];
U 119E, 3400,95 ;10608          MOV XWR[VAR] TO Q                     ;
U 119F, A180,15 ;10609          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11A0, 0869,3C ;10610          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11A1, 0919,D4 ;10611          JMP [LOOP.T10.4.7A]                   ; зацикливание при ошибке, если разрешено
U 11A2, FF88,15 ;10612          INC LS[ADDRESS.DATA]                  ;
                ;10613  LOOP.T10.4.7B:
U 11A3, 9EE6,75 ;10614          MEM.REQ[OCTA.READ.P] ADRS[L73] DT[LONG];
U 11A4, 3400,95 ;10615          MOV XWR[VAR] TO Q                     ;
U 11A5, A180,15 ;10616          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11A6, 0869,3C ;10617          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11A7, 891A,34 ;10618          JMP [LOOP.T10.4.7B]                   ; зацикливание при ошибке, если разрешено
U 11A8, FF88,15 ;10619          INC LS[ADDRESS.DATA]                  ;
                ;10620  LOOP.T10.4.7C:
U 11A9, 9F14,75 ;10621          MEM.REQ[ISSUE.BG] ADRS[T10] DT[LONG]  ;
U 11AA, 3400,95 ;10622          MOV XWR[VAR] TO Q                     ;
U 11AB, A180,15 ;10623          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11AC, 0869,3C ;10624          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11AD, 891A,94 ;10625          JMP [LOOP.T10.4.7C]                   ; зацикливание при ошибке, если разрешено
U 11AE, FF88,15 ;10626          INC LS[ADDRESS.DATA]                  ;
                ;10627  LOOP.T10.4.7D:
U 11AF, 1F62,75 ;10628          MEM.REQ[ISSUE.BG] ADRS[L31] DT[LONG]  ;
U 11B0, 3400,95 ;10629          MOV XWR[VAR] TO Q                     ;
U 11B1, A180,15 ;10630          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11B2, 0869,3C ;10631          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11B3, 891A,F4 ;10632          JMP [LOOP.T10.4.7D]                   ; зацикливание при ошибке, если разрешено
U 11B4, FF88,15 ;10633          INC LS[ADDRESS.DATA]                  ;
                ;10634  LOOP.T10.4.7E:
U 11B5, 9FA6,75 ;10635          MEM.REQ[ISSUE.BG] ADRS[L53] DT[LONG]  ;
U 11B6, 3400,95 ;10636          MOV XWR[VAR] TO Q                     ;
U 11B7, A180,15 ;10637          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11B8, 0869,3C ;10638          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11B9, 891B,54 ;10639          JMP [LOOP.T10.4.7E]                   ; зацикливание при ошибке, если разрешено
U 11BA, FF88,15 ;10640          INC LS[ADDRESS.DATA]                  ;
                ;10641  LOOP.T10.4.7F:
U 11BB, 1FE6,75 ;10642          MEM.REQ[ISSUE.BG] ADRS[L73] DT[LONG]  ;
U 11BC, 3400,95 ;10643          MOV XWR[VAR] TO Q                     ;
U 11BD, A180,15 ;10644          MOV Q TO WR[0]                        ; выборка виртуального адреса из WR5
U 11BE, 0869,3C ;10645          JSR [CHECK.RESULT]                    ; проверка, что виртуальный адрес был записан в WR5
U 11BF, 891B,B4 ;10646          JMP [LOOP.T10.4.7F]                   ; зацикливание при ошибке, если разрешено
                ;10647  CLEANUP.T10:
```

```
U 11C0, 3660,15 ;10648          MOV LS[BIT16] TO WR[0]                  ;
U 11C1, A3C0,15 ;10649          ROL WR[0]                               ;
U 11C2, 3E62,15 ;10650          MOV WR[0] TO LS[L31]                    ; восстановление ячейки 31
U 11C3, 3660,15 ;10651          MOV LS[BIT16] TO WR[0]                  ;
U 11C4, 3E80,15 ;10652          MOV WR[0] TO LS[CONTROL.STATUS]         ; восстановление разрешения запросов памяти
U 11C5, 10E0,15 ;10653          MISC [SET.CP.ATTN]                      ; вызов консольного процессора
                ;10654  WAIT.T10.CLNUP:
U 11C6, 891C,64 ;10655          JMP [WAIT.T10.CLNUP]                    ; ожидание ответа консольного процессора
U 11C7, 091D,04 ;10656          JMP [END.T10]                           ; конец теста
                ;10657  T10.SUB1:
U 11C8, B400,15 ;10658          MOV XWR[BPC] TO Q                       ; выборка WR4
U 11C9, A180,15 ;10659          MOV Q TO WR[0]                          ;
U 11CA, 3640,95 ;10660          MOV LS[#1] TO WR[1]                     ; ожидаемые данные = увеличенный PC
U 11CB, 5B00,14 ;10661          RETURN                                  ;
                ;10662  T10.SUB2:
U 11CC, B400,15 ;10663          MOV XWR[BPC] TO Q                       ; выборка WR4
U 11CD, A180,15 ;10664          MOV Q TO WR[0]                          ;
U 11CE, 3646,95 ;10665          MOV LS[#8] TO WR[1]                     ; ожидаемые данные = WR4 не записан
U 11CF, 5B00,14 ;10666          RETURN                                  ;
                ;10667  END.T10:
```

```
                ;10668     .PAGE
                ;10669    END.SECTION:
U 11D0, 365C,15 ;10670            MOV LS[BIT14] TO WR[0]                ; установка WR0 для конца секции
U 11D1, 3EB0,15 ;10671            MOV WR[0] TO LS[CONTROL.STATUS]       ; установка слова управления и состояния для сообщения
                ;10672                                                  ; конца секции.
U 11D2, 10E0,15 ;10673            MISC [SET.CP.ATTN]                    ; выдача сигнала CPU ATTN для консольного процессора
                ;10674    WAIT.EOS:
U 11D3, 091D,34 ;10675            JMP [WAIT.EOS]                        ; зацикливание для ожидания ответа от консольного
                ;10676                                                  ; процессора
                ;10677
                ;10678
```

Лист регистрации изменений

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