

ПРОЦЕССОР ПЛАВАЮЩЕЙ ЗАПЯТОЙ" СМ 2700.2008

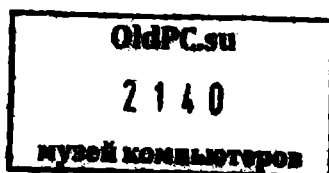
Руководство по эксплуатации

Часть 1

3.050.001РЭ

Магнитная лента

Листов 81



1988

Перв. примен.
3.050.001

Литера

Инв. N подл.	Подп. и дата	Взам. инв. N	Инв. N дубл.	Подпись и дата

3.050.001РЭ

СОСТАВ ДОКУМЕНТА

Настоящий документ состоит из двух частей:

часть 1 - 3.050.001РЭ "Процессор плавающей запятой СМ 2700.2008. Руководство по эксплуатации";

часть 2 - 3.050.001РЭ1 "Процессор плавающей запятой СМ 2700.2008. Руководство по эксплуатации". Приложение.

Часть первая содержит описание устройства и работы процессора плавающей запятой.

Во второй части даны рисунки, поясняющие текст первой части.

СОДЕРЖАНИЕ

Лист

Часть 1

1.	ВВЕДЕНИЕ.....	6
2.	НАЗНАЧЕНИЕ.....	11
3.	ТЕХНИЧЕСКИЕ ДАННЫЕ.....	12
4.	УКАЗАНИЯ МЕР БЕЗОПАСНОСТИ.....	13
5.	УСТРОЙСТВО И РАБОТА ПРОЦЕССОРА ОПЕРАЦИЙ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ.....	14
5.1.	Описание упрощенной блок-схемы процессора с плавающей запятой.....	14
5.2.	Конструктивное исполнение.....	15
5.3.	Диагностические характеристики.....	16
5.4.	Арифметика и числа с плавающей запятой....	16
5.4.1.	Целые числа.....	16
5.4.2.	Числа с плавающей запятой.....	17
5.4.3.	Нормализация.....	19
5.4.4.	Запись чисел с плавающей запятой.....	20
5.4.5.	Сложение и вычитание чисел с плавающей запятой.....	21
5.4.6.	Умножение и деление чисел с плавающей запятой.....	21
5.5.	Форматы данных.....	23
5.5.1.	Форматы чисел с плавающей запятой.....	23
5.5.2.	Мантисса.....	23
5.5.3.	Порядок.....	24
5.5.4.	Форматы целых чисел.....	27
5.5.5.	Особые случаи при операции над числами с плавающей запятой.....	28
5.5.6.	Переполнение.....	28
5.5.7.	Потеря порядка.....	28

3.050.001P3

5.5.3.	Деление на ноль.....	28
5.5.9.	Ошибка при получении неправильного операнда.....	29
5.6.	Сопряжение и взаимодействие с процессором.....	29
5.6.1.	Сигналы сопряжения.....	29
5.6.2.	Операции по взаимодействию с процессором.....	31
5.6.2.1.	Декодирование кода операции.....	31
5.6.2.2.	Загрузка операндов.....	31
5.6.2.3.	Запоминание результата.....	32
5.6.2.4.	Управление записью микроадреса.....	32
5.6.2.5.	Управление чтением микроадреса.....	33
5.6.2.6.	Ошибки по паритету и коды условий.....	33
5.7.	Команды и их алгоритмы.....	33
5.7.1.	Арифметические команды.....	35
5.7.1.1.	Сложение/вычитание.....	35
5.7.1.2.	Команда сравнения (CMP).....	37
5.7.1.3.	Команда вычисления полинома (POLY)...	38
5.7.1.4.	Команда деления (DIV).....	40
5.7.1.5.	Команда умножения (MUL).....	41
5.7.2.	Команды преобразования чисел.....	44
5.8.	Работа процессора с плавающей запятой.....	48
5.8.1.	Функциональное описание.....	48
5.8.1.1.	Прохождение данных.....	49
5.8.1.2.	Синхронизация процессора плавающей запятой.....	53
5.8.1.3.	Логика декодирования команд.....	54
5.8.1.4.	Формирование микроадреса.....	56
5.8.1.5.	Управляющая память.....	60
5.8.2.	Обработка данных.....	66

3.050.00123

5.8.3.	4-разрядная микропроцессорная секция KM18048C1	67
5.8.4.	Пути данных для обработки порядка.....	67
5.8.5.	Пути данных для обработки мантиссы.....	69
5.8.6.	Логика формирования знака.....	70
5.9.	Средства наладки FPA.....	71
5.9.1.	Установка микроадреса.....	72
5.9.2.	Чтение микроадреса.....	72
5.9.3.	Логика формирования паритета.....	72
6.	ПОДГОТОВКА К РАБОТЕ.....	74
6.1.	Порядок установки.....	74
6.2.	Первоначальное включение.....	74
7.	ИЗМЕРЕНИЕ ПАРАМЕТРОВ, РЕГУЛИРОВАНИЕ И НАСТРОЙКА.....	75
8.	ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ.....	76
9.	ХАРАКТЕРНЫЕ НЕИСПРАВНОСТИ, МЕТОДЫ ИХ ОБНАРУЖЕНИЯ.....	78
10.	ТРАНСПОРТИРОВАНИЕ И ХРАНЕНИЕ.....	80

ЧАСТЬ 2

ПРИЛОЖЕНИЕ. Рисунки, поясняющие текст части 1.

1. ВВЕДЕНИЕ

1.1 Настоящее руководство по эксплуатации предназначено для ознакомления с принципом работы, техническими характеристиками и условиями эксплуатации процессора операций с плавающей запятой СМ 2700.2008 (в дальнейшем - FPA).

1.2 При изучении, эксплуатации и техническом обслуживании необходимо пользоваться следующими документами:

3.700.013РЭ - "Комплекс вычислительный СМ 1700";

3.039.006РЭ - "Вычислительная машина СМ 2700";

00076-01-13-01 - "Система микродиагностического обеспечения ВК СМ 1700. Описание программы".

1.3 Сокращения и термины, применяемые в настоящем РЭ.

Алгоритм	забор действий (процедур), которые FPA выполняет для реализации команд плавающей арифметики за конечное число шагов.
ACC	ускоритель.
ACC SYNC	Разряд CSL47] микрокоманды, идентифицирующий работу FPA. Он устанавливается, когда поле управления ветвлением CSL14:10] равно 2, 3 или 16. ACC SYNC H указывает центральному процессору, что FPA готов.
ALU	Арифметико-логическое устройство, которое содержит логику путей данных и микросеквенсер.
Избыток (BIAS)	избыток порядка (от его истинного значения).
Поле управления ветвлением	пятиразрядное поле CSL14:10], используемое для выбора сигналов состояния (сигналов изменения микроадреса), которые, подключаясь по схеме "ИЛИ" к младшим двум разрядам поля микроадреса (UPF), могут его модифицировать. При определенных значениях полей MOD и CLK CTL поле управления ветвлением может использоваться для модификации пяти разрядов поля UPF.
BUS FPA	внутренняя 32-разрядная шина FPA.
BUS NUA	шина следующего адреса микрокоманды. Является выходом микросеквенсера.
Синхронизация (CLOCK)	когда FPA работает автономно, длительность его цикла - 130 ns. При синхронизации своей работы с

3.050.001P3

Поле управления синхронизацией	процессором, длительность цикла - 270 ns. Трехразрядное поле CS[17:15], которое используется для управления синхронизацией, а также для выполнения специальных функций.
CMR	Команда сраамения.
CSR	Регистр микрокоманд.
CVT	Команда преобразования, используемая для преобразования одного типа данных в другой.
D	Формат данных двойной точности (64 разряда).
Деление на ноль	Особый (ошибочный) случай, который происходит, когда делитель равен нулю. Для этого случая приемник не изменяется, а значение кодов условия не предсказуемо.
DIVL	Команда деления длинных слов.
EMOD	Команда умножения увеличенной точности и выделения целого числа.
Особый случай (EXCEPTION)	Условие ошибки, которое происходит в процессе обработки операндов; об этом сообщается процессору через шину BUS Y.
FPA	Процессор плавающей запятой
Запись с избытком (EXCESS NOTATION)	Избыток порядка, который в зависимости от формата данных может принимать значения 80, 400 или 4000; используется для запоминания и обработки порядка в числах с плавающей запятой.
Порядок (EXPONENT)	Представляет собой показатель степени, в которую возводится основание степени (число 2), записывается с избытком. Занимает 8 разрядов в формате числа для форматов F и D, 11 разрядов - для формата G и 15 разрядов - для формата H.
Поле EXP CTL	Поле в микрокоманде - CS[44:39].
Поле EXP DST	Поле управления приемником порядка.
Пути данных порядка	Пути данных, имеющие размер 16 разрядов.
Расширенный код операции	Код операции, разный FD; используется для расширения нормального кода операции команды, который обычно равен 8 разрядам, передаваемым по шине BUS IR.
Поле FALU	Поле функции ALU в путях данных мантиссы.
Установка микроадреса (FORCE)	Процессор запрещает формирование микроадреса со стороны микросеквенсера FPA и затем сам устанавливает следующий адрес микрокоманды к управляющей памяти через шину BUS Y.
F	Формат данных одинарной точности (32 разряда).
FPAА-FPAN	Обозначение листов принципиальных электрических схем для FPA.
Пути данных мантиссы	Пути данных, имеющие размер 64 разряда.
Поле FRAC	Поле управления мантиссой.
Мантисса (FRACTION)	Нормализованная, в двоичном представлении, содержащая знак и величину.
Поле FRSC	Поле операнда источника для ALU в путях

3.050.001PЭ

	данных мантииссы.
Поле FSHF.	Поле управления приемником от ALU в путях данных мантииссы.
G	Формат данных с расширенным порядком.
Формат	Формат 64-разрядного длинного слова.
(GRAND FORMAT)	
Дополнительные	Разряды, которые используются для сохранения самого младшего разряда (LSB) операнда и которые выдвигаются из мантииссы.
разряды	Требуются для выполнения точных вычислений.
(GIARD BITS)	
Скрытый бит	Это самый старший разряд мантииссы, который у нормализованной мантииссы всегда единица и поэтому в оперативной памяти он не запоминается. После получения операнда от процессора, FPA устанавливает эту единицу в соответствующее место. После получения результата и перед записью его в память эта единица исключается.
(HIDDEN BIT)	
H	Формат данных учетверенной точности с расширенным порядком.
Формат Ч	Формат данных, имеющих размер в 128
(HUGE FORMAT)	разрядов.
BUS IB	шина, используемая процессором для передачи кода операции от FPA.
Пути данных	Часть путей данных мантииссы, разряды [47:16].
целого	
(integer DATA PATH)	
IRD	Состояние декодирования команды.
Поле	8-разрядное поле микрокоманды CS[7:0].
литерала (LIT)	
Загрузка	Передача операндов со стороны процессора процессору FPA.
(LOAD)	
LSB	Самый младший значащий разряд.
Микроадрес	10-разрядное поле (шина), значение которого вырабатывается микросеквенсером FPA (или устанавливается процессором) и является адресом микрокоманды.
Поле микро-	10-разрядное поле микрокоманды CS[9:0],
адреса	которое является базовым адресом следующей микрокоманды для микросеквенсера.
Микрослово	микроманданда, которая хранится в управляющей памяти и к которой осуществляется доступ 10-разрядным микроадресом.
MIER	Множитель.
Поле MOD	4-разрядное поле модификации CS[27:20], используемое для расширения функций других полей, а также для выполнения специальных функций.
MSB	Самый старший значащий разряд.
MUL	Команда умножения коротких слов.
Нормализация	Процедура выполнения сдвигов с целью получения в самом старшем разряде мантииссы единицы.
Код операции	8-разрядное поле кода операции, которое указывает FPA, какая операция
(OP CODE)	(команда) должна быть выполнена над

Операнд	операндами, получаемыми по шине BUS Y. Данные, получаемые по шине BUS Y для выполнения над ними указанной команды.
Переполнение	Особый случай (ошибка), который происходит, когда порядок числа с плавающей запятой оказался больше, чем самый большой представимый для данного типа данных порядок после того, как была выполнена нормализация и округление.
ПМЛ	Программируемая матричная логика.
Поле паритета	2-разрядное поле микрокоманды CS[46:45], используемое для контроля управляющей памяти.
POLY	Команда вычисления полинома.
Выравнивание порядков (PREALIGNMENT)	Процедура выравнивания порядков (то есть получение равноценных порядков), необходимая перед выполнением сложения или вычитания двух чисел с плавающей запятой.
Проба (PROBING)	Процесс определения, является ли адрес доступным.
PRUM	Программируемое постоянное запоминающее устройство.
Поле адреса "А" памяти RAM	4-разрядное поле микрокоманды CS[29:26], предназначенное для указания адреса памяти RAM путей данных как порядка, так и мантиссы.
Поле адреса "В" памяти RAM	4-разрядное поле микрокоманды CS[25:22], предназначенное для указания адреса памяти RAM (по порту "В") путей данных как порядка, так и мантиссы.
Проверка порядков (RANGE TEST)	Проверка порядков, выполняемая перед сложением и вычитанием двух чисел с плавающей запятой с целью определить, нужно ли выравнивать порядки.
ROM	Постоянное запоминающее устройство.
Округление (ROUNDING)	Прибавление единицы к старшему дополнительному разряду.
RTOL	Чтение справа налево.
Сохранение (SAVE)	Суффикс в названии сигнала, который указывает, что данный сигнал был сгенерирован в предыдущем цикле (например, EXPRD SAVE H).
Поле сдвига (SHF)	2-разрядное поле микрокоманды CS[19:18], которое управляет функциями сдвига.
Поле размера (SIZE FIELD)	2-разрядный выход схемы декодирования команд. значение на этом выходе указывает тип операндов (F, D, G или H), который будет передаваться от центрального процессора по шине BUS Y.
Регистр состояния	Регистр логики состояния, который получает сигналы состояния от логики путей данных.
Запоминание (STORE)	FPA посылает свой результат процессору.
SUB	Команда вычитания.
Суммирование (SUMMATION)	Сложение двух чисел, если знак обоих операндов одинаковый.
Прерывание (TRAP)	Процессор прекращает работу процессора FPA на текущем адресе микрокоманды с тем, чтобы он мог прочитать его с

3.050.001P3

Потеря порядка (UNDERFLOW)	шины BUS Y. Особый случай (ошибка), который происходит, когда порядок числа с плавающей запятой оказался меньше, чем самый маленький представимый для данного типа данных порядок после того, как были выполнены нормализация и округление.
UPF BUS Y	Поле микроадреса. 32-разрядная шина, которая связывает процессор и FPA и по которой передаются данные

2. НАЗНАЧЕНИЕ

Процессор операций с плавающей запятой СМ 2700.2008 предназначен для ускоренного аппаратного исполнения всех команд арифметики с плавающей запятой, а также для преобразования чисел из форматов целых чисел в форматы чисел с плавающей запятой и наоборот.

FPA является аппаратным расширением процессора. Он использует те же режимы адресации операндов и те же имеющиеся в процессоре средства диспетчера памяти. Команды FPA могут обращаться как к общим регистрам процессора, так и к любым ячейкам оперативной памяти.

По устойчивости к внешним факторам относится к изделиям категории 3б обыкновенного исполнения по ГОСТ 20397-82.

В процессе эксплуатации FPA выдерживает воздействие следующих климатических и механических факторов:

температура окружающего воздуха от +5 ° до +60 градусов С;

относительная влажность воздуха от 40 до 90% при температуре +30 градусов С;

атмосферное давление от 84 до 107 кПа;

вибрация частотой 25 Нз с амплитудой не более 0,1 мм.

Нормальные условия эксплуатации агрегатного модуля:

температура окружающего воздуха от +15 до +25 градусов С;

относительная влажность воздуха от 45% до 75%;

атмосферное давление от 84 до 107 кПа.

3. ТЕХНИЧЕСКИЕ ДАННЫЕ

Процессор плавающей запятой имеет следующие технические характеристики:

управление аппаратурой FPA реализовано на основе собственного микропрограммного аппарата;

используются два основных цикла выполнения микрокоманд 180 и 270 нс;

разрядность слова микрокоманд - 48;

объем памяти микрокоманд - 1024 слов;

форматы обрабатываемых операндов имеют следующие длины: одинарной точности F - 32 разряда, двойной точности D - 64 разряда и с расширенным порядком G - 64 разряда, учетверенной точности H - 128 разрядов.

Электропитание осуществляется от источника питания СМ 1700.0304 2.087.065. Напряжение источника питания +5V.

Конструктивно FPA представляет собой многослойную печатную плату типа ЕЗ с размещенными на ней интегральными микросхемами и радиоэлектронными элементами.

Соединение элементов платы выполняется двухслойным печатным монтажом. Питание и земля разводятся двумя внутренними слоями, которые являются также экранирующими.

Максимальные габаритные размеры платы 426x285x19мм.

Тип разъемов СПС9-64/94x11В-23-1-В. На плате установлено 4 разъема, с использованием в каждом из них двух рядов контактов.

Масса - не более 1кг.

4. УКАЗАНИЯ МЕР БЕЗОПАСНОСТИ

Обслуживание ФРА должно выполняться инженерно-техническим персоналом, имеющим удостоверение на право обслуживания ЗК СМ 1700.

Обслуживающий инженерно-технический персонал должен регулярно проходить инструктаж и выполнять все требования электробезопасности.

Категорически запрещается вставлять и извлекать блоки элементов, производить замену электроэлементов при включенном электропитании ЗК СМ 1700.

Монтажные работы следует производить паяльником 36V, имеющим исправную изоляцию токоведущих частей от корпуса.

При обслуживании ФРА вне частичного монтажного блока, следует надеть на руку заземляющий браслет.

5. УСТРОЙСТВО И РАБОТА ПРОЦЕССОРА ОПЕРАЦИИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

5.1. Описание упрощенной блок-схемы процессора с плавающей запятой

Процессор операций над числами с плавающей запятой FPA представляет собой дополнительные технические средства, которые могут подключаться к вычислительному комплексу СМ 1700. Он может выполнять команды сложения, вычитания, умножения и деления над числами с плавающей запятой.

FPA работает совместно с процессором и с высоким быстродействием выполняет команды арифметики с плавающей запятой. Оба процессора могут работать параллельно, при этом процессор будет выполнять такие операции для команд арифметики с плавающей запятой, как вычисление адреса приемника, в то время как сам FPA непосредственно исполняет эти команды. Процессор не может одновременно с работой FPA выполнять другие команды (следующие по порядку за выполняемой командой), он должен ждать выполнения текущей команды арифметики с плавающей запятой. Одновременная работа процессора и FPA при выполнении команд арифметики с плавающей запятой позволяет реализовать их с высокой скоростью.

FPA может также увеличить скорость исполнения некоторых команд целочисленной арифметики. Операции FPA "прозрачны" для программного обеспечения. При его подключении к процессору необходимо только перезагрузить микропрограммы.

FPA может обрабатывать широкий диапазон чисел. В нем могут быть представлены числа с плавающей запятой от $1,5 * 10^{**}-39$ до $3,4 * 10^{**}38$. Формат с одинарной точностью обрабатывает числа до 7 десятичных цифр, а формат с двойной точностью - до 16 десятичных цифр. Диапазон формата с расширенным порядком: от $8,9 * 10^{**}-307$ до $1,11 * 10^{**}303$. Диапазон формата учетверенной точности с расширенным порядком: от $5,94 * 10^{**}4951$ до $8,40 * 10^{**}-4932$. FPA может также обрабатывать 32-разрядные целые числа со знаком в диапазоне от $-2\ 147\ 483\ 648$ до $2\ 147\ 483\ 647$.

Поскольку FPA представляет собой расширение процессора, он сам не может обращаться в оперативную память. Функции вычисления адреса оперативной памяти, чтения

операндов из памяти и передача их в FPA выполняется процессором. Он же выполняет чтение результатов работы из FPA и запоминает их в оперативной памяти. Сам FPA выполняет непосредственно команды над целыми числами или числами с плавающей запятой, которые переданы ему процессором.

Как показано на рис.1 в приложении FPA содержит схемы путей данных и управляющую память (память микрокоманд), которая генерирует управляющие сигналы, используемые для обработки данных. Схема путей данных содержит сигналы, используемые для обработки данных. Схема путей данных содержит 20 4-разрядных микропроцессорных секций KM18048C1.

Первоначально процессор передает FPA код операции, который после его дешифрации формирует соответствующий начальный адрес микрокоманды. Непосредственно начальный микроадрес управляющей памяти PROM формируется секвенсором FPA. Затем микрокоманды, считываемые из нее, генерируют сигналы для управления схемами путей данных. Под управлением микропрограммы эти схемы получают входные данные по шине BUS Y.

Процессор передает FPA данные с плавающей запятой в упакованной и нормализованной форме со знаком длиной по 32 разряда. Эти данные принимаются в буферный регистр и затем используются схемами путей данных, которые разделяют числа (операнды) на части (распаковывают их) и над каждой из этих частей выполняют необходимые операции. После получения результата операции схемы путей данных под управлением микропрограммы нормализуют и упаковывают его, затем запоминают в буферном регистре и пересылают его в процессор порциями по 32 разряда по шине BUS Y.

Когда FPA выполняет вычисления, поле микроадреса в управляющей памяти FPA выбирает адрес следующей микрокоманды, которая должна быть выполнена. Этот адрес затем принимается в микросеквенсер KM1804BУ1, в котором он может быть модифицирован путем наложения на него по схеме "ИЛИ" сигналов изменения адреса. В результате микросеквенсер формирует следующий микроадрес для управляющей памяти.

3.2. Конструктивное исполнение

Процессор с плавающей запятой реализован на стандартной плате размером 426*285*19, содержащей главным образом логику ИТЛ Шотки. Он не имеет устройств для калибровки, каких-либо переключателей или других средств настройки.

5.3. Диагностические характеристики

Средства диагностики FPA включают в себя функции чтения и записи микроадреса, то есть процессор может установить (записать) требуемый адрес микрокоманды для управляющей памяти или прочитать адрес следующей микрокоманды. Диагностика проверяет работу схем декодирования команд, работу микросеквенсера, управляющей памяти и схем путей данных. Для контроля работы управляющей памяти вводятся два бита контроля по паритету. Если происходит ошибка паритета, то FPA переходит на подпрограмму обработки этой ошибки.

5.4. Арифметика и числа с плавающей запятой

5.4.1. Целые числа

Все данные, обрабатываемые процессором, могут быть представлены в 32-разрядной сетке машины, имеют величины от 00000000 до FFFFFFFF (в шестнадцатеричной системе счисления) то есть от 0 до 4 294 967 295 (в десятичной системе счисления), однако форма представления в виде целых чисел имеет свои ограничения, поскольку не могут быть представлены числа с дробными частями. Эта форма имеет ограничения в точности представления. Кроме того, числа большие чем 4 294 967 295 в этой форме не могут быть представлены.

Целые числа представляются с фиксированной запятой таким образом, что справа от запятой не имеется никаких цифр. По этой причине целые числа называются числами с фиксированной запятой.

Данные, представленные в виде чисел с фиксированной запятой, могут быть преобразованы таким образом, чтобы увеличить диапазон представления и исключить ограничения в точности их представления, которые накладываются форматом с фиксированной запятой. Это делается путем представления данных в формате чисел с плавающей запятой. Эта форма представления дает возможность при той же разрядной сетке реализовать более широкий диапазон чисел, чем в целочисленном формате с фиксированной запятой.

5.4.2. Числа с плавающей запятой

Числа с плавающей запятой в отличие от целых чисел не имеют таких ограничений в точности их представлений. Наиболее популярной формой представления чисел с плавающей запятой является форма их представления для научных расчетов. В этой форме число представляется произведением некоторого базового значения на степень числа, основанием которой является принятая система счисления.

Пример 1.

$$1\ 000\ 000 = 1 \cdot 10^{**6},$$

где 1 - базовое значение,
10 - основание степени (системы счисления),
6 - показатель степени (порядок).

Существует много способов отображения одного и того же числа в форме представления чисел для научных расчетов, как это показано в примере 2.

Пример 2.

Правый сдвиг

Левый сдвиг

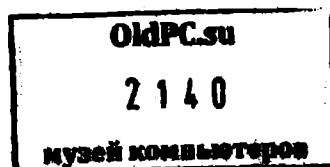
$$\begin{aligned} 512 &= 512, \quad * 10^{**0} \\ &= 51,2 \quad * 10^{**1} \\ &= 5,12 \quad * 10^{**2} \\ &= ,512 \quad * 10^{**3} \end{aligned}$$

$$\begin{aligned} 512 &= 512 * 10^{**0} \\ &= 5120 * 10^{**-1} \\ &= 51200 * 10^{**-2} \\ &= 512000 * 10^{**-3} \end{aligned}$$

Правило, принятое в ФРА для представления чисел с плавающей запятой, требует, чтобы запятая была расположена левее самой старшей значащей цифры в базовом значении (в выше приведенном примере этому правилу удовлетворяет строка „512 * 10^{**3}“). Такое модифицированное базовое значение называется нормализованной мантиссой.

Заметим, что для каждого правого сдвига базового значения порядок увеличивается, а для каждого левого - уменьшается на единицу. Величина самого числа остается без изменений, если для каждого сдвига базового значения соответствующим образом изменится и порядок.

Дополнительные примеры формы представления чисел для научных расчетов приведены в примере 3.



5.050.001P3

Пример 3.

Десятичное представление:

64
53
1/2 (,5)
5/32 (,09375)

Десятичное представление для научных расчетов:

64 * 10**2
53 * 10**2
.5 * 10**0
.9375 * 10**-1

Двоичное представление:

1000000,
100001,
0,1
0,00011

Шестнадцатеричное представление:

40 (в 16-ной с/с)
21 (в 16-ной с/с)
.8 (в 16-ной с/с)
.18 (в 16-ной с/с)

Шестнадцатеричное представление для научных расчетов:

.4 * 16**2
.21 * 16**2
.8 * 16**0
.18 * 16**0

5.4.3. Нормализация

Имеется много способов записи числа с плавающей запятой, используя форму представления чисел для научных расчетов. Правила, принятые в вычислительном комплексе SM 1700, а также в процессоре операций с плавающей запятой FPA, требуют такой формы представления, при которой плавающая запятая располагается сразу же перед самым старшим значащим разрядом базового значения, как это показано в примере 1.

Пример 1.

Форма записи с плавающей запятой:

29(10-ная с/с)=11101(2-ная с/с)=

3.050.001PЭ

= 1 1101,	*2**0	1 1101,*2**0
1110,1	*2**1=	11 1010,*2**1
111,01	*2**2=	111 0100,*2**2
11,101	*2**3=	1110 1000,*2**3
1,1101	*2**4=	1 1101 0000,*2**4
выбранная форма записи--> ,11101	*2**5=	11 1010 0000,*2**5
(,11101-ман- тисса, 5-по- рядок)	,011101 *2**6=	111 0100 0000,*2**6
	,0011101*2**7=	1110 1000 0000,*2**7

Процесс сдвига числа до такого положения, при котором самый первый значащий разряд размещается сразу же после двоичной запятой, называется нормализацией. Если данное число равно единице или имеет большее значение, то оно должно сдвигаться вправо с одновременным увеличением порядка до того момента, пока самый старший разряд (MSB) не расположится сразу же за двоичной запятой. Если данное число является дробью с нулями после запятой, то оно должно сдвигаться влево с одновременным уменьшением порядка при выполнении процедуры нормализации. В примерах 2 и 3 рассматриваются процессы преобразования чисел в нормализованную форму.

Пример 2.

Преобразовать число 75 (10-ная система счисления) в нормализованную двоичную форму.

Преобразование целого числа в двоичную форму:

$$75(10\text{-ная с/с}) = 100\ 1011(2\text{-ная с/с})$$

Преобразование в форму чисел с плавающей запятой:

$$100\ 1011(2\text{-ная с/с}) = 100\ 1011(2\text{-ная с/с}) * 2^{**0}$$

Процедура нормализации числа (сдвиг вправо мантиссы на 7 разрядов и увеличение порядка на 7):

$$100\ 1011(2\text{-ная с/с}) * 2^{**0} = ,100\ 1011 * 2^{**7}$$

$$\text{мантисса} = ,100\ 1011$$

$$\text{порядок} = 7$$

Пример 3.

Преобразовать число 3/16 (,1875) к нормализованной двоичной форме.

3.050.007P3

Преобразование числа в двоичную форму:

$$,1875(10\text{-ная с/с}) = ,0011(2\text{-ная с/с})$$

Преобразование в форму числа с плавающей запятой:

$$,0011(2\text{-ная с/с}) = ,0011(2\text{-ная с/с}) * 2^{**0}$$

Процедура нормализации числа: сдвиг влево мантиссы на 2 разряда и уменьшение порядка на 2:

$$,0011(2\text{-ная с/с}) * 2^{**0} = ,11 * 2^{**-2}$$

мантисса = ,11

порядок = -2

5.4.4. Запись чисел с плавающей запятой

При хранении чисел с плавающей запятой необходимо добиваться по возможности экономии пространства памяти без потери точности представления чисел. Поэтому первым правилом при записи чисел с плавающей запятой в память является исключение так называемого скрытого бита (HIDDEN BIT). Все числа, передаваемые между процессором и FPA, а также записываемые в память, должны быть нормализованными. Это означает, что самый первый значащий разряд (который всегда 1) располагается сразу же справа за двоичной запятой. Следовательно, этот разряд может не запоминаться в памяти и не передаваться в процессор и обратно. Например, мантисса нормализованного двоичного числа $,1100\dots * 2^{**-2}$ передается и запоминается в памяти (и, наоборот, из памяти в FPA) как $100\dots$. Нормализованная мантисса числа $1/2$ ($,100\dots * 2^{**0}$) передается и запоминается как $000\dots$. В обоих случаях перед обработкой этих чисел в FPA к ним будет добавлена аппаратурой первая после запятой единица (скрытый бит). Когда же FPA передает нормализованный результат назад в процессор, этот скрытый бит исключается из передачи.

Второе правило при представлении чисел с плавающей запятой заключается в том, что порядок записывается со смещением. Это смещение может иметь величину 8U(16-ная с/с), 40U(16-ная с/с) или 400U(16-ная с/с). Такое представление порядка называется представлением с избытком и дает возможность упростить аппаратуру, которая обрабатывает порядок во время выполнения арифметических команд с плавающей запятой. Порядок с избытком 8U(16-ная с/с) получается путем добавления 10000000(2-ная с/с) (20U(8-ная с/с), 8U(16-ная с/с) или 128(10-ная с/с)) к дополнительному коду порядка. Эта форма порядка дает возможность записывать его как положительное значение.

5.4.5. Сложение и вычитание чисел с плавающей запятой

Чтобы выполнить сложение или вычитание чисел с плавающей запятой, их порядки должны быть выравнены или равны. Если порядки исходных операндов не равны, то мантисса числа с меньшим порядком сдвигается вправо, пока порядки не выравняются. Каждый сдвиг вправо сопровождается инкрементированием соответствующего порядка. Когда порядки чисел равны, то можно производить сложение или вычитание мантисс. Величина порядка указывает на число разрядов, на которое двоичная запятая должна быть перемещена, чтобы получить истинное значение числа.

В примере 1 число 7(10-ная с/с) прибавляется к числу 40(10-ная с/с), используя формат чисел с плавающей запятой. Заметим, что сначала выравниваются порядки, а затем складываются мантиссы. Результирующее значение порядка определяет окончательную позицию двоичной запятой.

Пример 1.

Сложение чисел с плавающей запятой:

0,1010 0000 0000 000 * 2**6=28(16-ная с/с)=40(10-ная с/с)
+ 0,1110 0000 0000 000 * 2**3=7(16-ная с/с)=7(10-ная с/с)

Последовательность выполнения сложения:

1) чтобы выравнивать порядки, необходимо сдвинуть мантиссу с меньшим порядком на три разряда вправо и увеличить порядок на 3;

2) сложить обе мантиссы

0,1010 0000 0000 000 * 2**6=28(16-ная с/с)=40(16-ная с/с)
+ 0,0001 1100 0000 000 * 2**6=7(16-ная с/с)=7(10-ная с/с)

0,1011 1100 0000 000 * 2**6=2F(16-ная с/с)=47(10-ная с/с)

3) чтобы найти целую величину результата, необходимо сдвинуть двоичную запятую на шесть позиций вправо.

010 1111,0000 0000 0
----->

5.4.6. Умножение и деление чисел с плавающей запятой

При умножении чисел с плавающей запятой мантиссы

3.0511.001P3

перемножаются, а порядки складываются. При делении чисел с плавающей запятой мантиссы делятся, а порядки вычитаются. При умножении и делении не требуется выравнивание порядков. В примере 2 рассматривается умножение с плавающей запятой, а в примере 3 - деление.

Пример 2.

Умножить 7 (10-ная с/с) на 40 (10-ная с/с).

0,1110000 * 2**3=7(16-ная с/с)=7(10-ная с/с)

* 0,1010000 * 2**6=28(16-ная с/с)=40(10-ная с/с)

1110000
0000000
1110000

1000110000 * 2**9

(полученный результат в нормализованной форме)

Для получения истинного значения результата переместите двоичную запятую на девять позиций (разрядов) вправо:

10001,1000 00000=118(16-ная с/с)=280(10-ная с/с)
----->

Пример 3.

Деление. Разделить 15 (10-ная с/с) на 5 (10-ная с/с).

1111000 * 2**4

1010000 * 2**3

1111000 000000 : 1010000
1010000 :-----
----- 1,10000
101000
101000

0

Вычисление порядка: 4 - 3 = 1

Результат: 1,100000 * 2**1

Нормализованный результат: 1100000 * 2**2

где 1100000 - нормализованная мантисса

2**2 - нормализованный порядок

Для получения окончательного значения числа пере-

местите двоичную запятую на два разряда вправо.

$$11,000000 = 3(16\text{-ная с/с}) = 3(10\text{-ная с/с})$$

5.5. Форматы данных

5.5.1. Форматы чисел с плавающей запятой

Процессор операций с плавающей запятой должен получать свои входные данные (операнды) определенных форматов. Существует пять различных форматов данных, которые может обрабатывать FPA: одинарной точности (F) - FLOATING, двойной точности (D) - DOUBLE, формат данных с расширенным порядком (G) - GRAND, формат данных учетверенной точности с расширенным порядком (H) - HUGE, а также формат целого числа. Результирующий операнд может быть выдан FPA в форматах F, D, G, H или в формате целого числа.

Форматы чисел с плавающей запятой (рис. 2-5 в приложении) имеют следующие длины: одинарной точности F - 32 разряда, двойной точности D и двойной точности с расширенным порядком G - 64 разряда, учетверенной точности с расширенным порядком H - 128 разрядов. Все эти числа содержат поля мантиссы и порядка, а также разряд знака. На указанных выше рисунках показано, как числа, получаемые FPA, перепозиционируются в нем перед их обработкой.

5.5.2. Мантисса

Мантисса числа имеет двоичное представление и хранится в нормализованном виде. В табл.1 дается запись мантиссы в прямом и дополнительном коде. При записи числа в прямом коде со знаком для изменения знака числа требуется изменить только знаковый разряд. Заметим, что представление положительного числа в прямом и дополнительном коде одинаково (см. табл.1).

Таблица 1

Запись величины мантиссы и ее знака

	! Дополнительный код	! Прямой код (со знаком)
+2	! 000010	! 000010
-2	! 111110	! 100010

3.050.001PЭ

Мантисса представляет собой двоичное число следующего вида:

0,1xxxxx...

Первый разряд мантиссы всегда единица, поскольку мантисса всегда нормализуется после получения результата во всех командах арифметики с плавающей запятой. Нормализация заключается в сдвиге числа до тех пор, пока в старшем разряде мантиссы не появится единица, причем при каждом сдвиге производится соответствующее изменение порядка. Например:

$$[0,1 * 2^{**1}] * [0,1 * 2^{**3}] = 0,01 * 2^{**4}$$

Нормализованный результат = 0,1 * 2^{**3}, где знаком ** обозначается операция возведения в степень.

Поскольку самый старший разряд нормализованной мантиссы всегда равен единице, он не запоминается в оперативной памяти. Этот разряд называется "скрытым" битом. ФРА всякий раз, когда он получает число из оперативной памяти, вставляет этот недостающий разряд.

5.5.3. Порядок

Формат числа одинарной точности F имеет 8-разрядный порядок, формат двойной точности D также 8-разрядный порядок, формат числа с расширенным порядком G использует 11-разрядный порядок и формат числа учетверенной точности с расширенным порядком H - 15-разрядный порядок.

Порядок представляет собой показатель степени, в которую возводится число 2, а в самой машине он представляется с избытком 80, 400, 4000 в зависимости от формата. Таким образом, порядок записываемый в формат данных, образуется путем сложения истинного значения порядка (показателя степени двух) с одним из указанных выше избытков (см. табл. 2).

Таблица 2

Представление порядка для различных форматов данных	
Избыток (в шестнадцатеричной форме)	Формат данных
80	F, D
400	G
4000	H

Запись с избытком 80/400/4000 используется как для запоминания, так и обработки порядка, как части числа с плавающей запятой. Порядки с указанными избытками обрабатываются одинаковыми способами, например, запись с избытком 80 представляет собой значение порядка в дополнительном коде плюс 128(10-ная с/с) или 80(16-ная с/с).

Удобно порядок числа с плавающей запятой обрабатывать в дополнительном коде. Это дает возможность представления широкого диапазона как положительных, так и отрицательных порядков. Однако при использовании дополнительного кода должно происходить переполнение при переходе от самого маленького отрицательного числа к нулю. Чтобы этого избежать, к дополнительному коду числа прибавляется 128(10-ная с/с) или 80(16-ная с/с).

При выполнении операций умножения или деления над числами с плавающей запятой, содержащими порядок с избытком 80 (или 400 или 4000, когда требуется), результирующий порядок должен быть откорректирован так, чтобы он тоже имел избыток 80(16-ная с/с). При выполнении умножения порядки складываются, и результирующий порядок будет иметь двойной избыток, поэтому из него должно быть вычтено 80(16-ная с/с), чтобы получить в итоге нормальный избыток 80. Ниже приводится пример, показывающий необходимость вычитания 80(16-ная с/с) из результирующего порядка при умножении.

$$\text{Порядок } A + 80(16\text{-ная с/с})$$

$$\text{Порядок } B + 80(16\text{-ная с/с})$$

$$\text{Порядок } A + \text{порядок } B + 100(16\text{-ная с/с})$$

где 80(16-ная с/с) - запись порядка с избытком 80(16-ная с/с)

Как видим, результирующий порядок имеет избыток 100(16-ная с/с), поэтому он должен быть откорректирован так, чтобы иметь избыток 80(16-ная с/с).

Пример 1. Умножение.

$$2 * 3 = 6$$

Мантисса

Порядок

$$2 = 0,100 *$$

$$82(16 \text{ с/с})$$

$$3 = 0,110 *$$

$$82(16 \text{ с/с})$$

Вычисление мантиссы

Вычисление порядка

$$2 = 0,100$$

$$82(16 \text{ с/с})$$

$$3 = 0,110$$

$$+ 82(16 \text{ с/с})$$

$$\text{-----}$$

$$100$$

$$\text{-----}$$

$$104(16 \text{ с/с})$$

$$100$$

$$- 80(16 \text{ с/с})$$

$$\text{-----}$$

$$6=0,011000$$

*

$$\text{-----}$$

$$84(16 \text{ с/с})$$

3.050.001PЭ

Нормализация результирующей мантиссы осуществляется путем сдвига влево ее на один разряд с одновременным уменьшением порядка на 1.

Мантисса		Порядок
0,11000	*	33 = 6

При выполнении операции деления порядки вычитаются, и к результирующему порядку должно быть прибавлено 80(16 с/с) (если в данном формате используется избыток 80), чтобы в итоге порядок имел избыток 30. Чтобы это было понятно, рассмотрим следующий пример:

$$\begin{array}{r} \text{Порядок A} + 30 \\ - \text{Порядок B} + 80 \\ \hline \end{array}$$

Порядок A - порядок B + 30 - 30 = порядок A - порядок B + 0.

Однако, так как окончательный порядок должен иметь избыток 30, к полученному выше результату необходимо прибавить 80(16 с/с): порядок A - порядок B + 30.

Пример 2. Деление.

16/4 = 4

Мантисса		Порядок
16 = ,10000	*	33
4 = ,10000	*	33

Вычисление мантиссы	Вычисление порядка
0,10000 : 0,10000	33
!-----	- 33
1,000	-----
	2
	+ 80

	82

Нормализация полученного результата заключается в сдвиге на один разряд вправо мантиссы и увеличение на единицу порядка:

Мантисса		Порядок
,10000	*	84 = 4

5.050.001PЭ

последних в оперативной памяти.

5.5.5. Особые случаи при операции над числами с плавающей запятой

FPA реагирует на различные особые ситуации, которые могут возникнуть в процессе его работы. Когда FPA фиксирует такие ситуации, он информирует об этом процессор путем установки на BUS Y различных разрядов или комбинации этих разрядов. После этого один из процессоров или оба вместе выполняют специальные процедуры, которые обрабатывают эти случаи. В ряде случаев текущая операция FPA останавливается, и он возвращается в состояние декодирования команды (IRD), в котором все схемы приводятся в исходное состояние, и FPA ожидает получения новой команды.

5.5.6. Переполнение

Переполнение возникает, когда порядок получился больше, чем самый большой порядок, который может быть представлен для данного формата данных. Переполнение должно проверяться после нормализации и округления. Операнд приемника в этом случае не изменяется, а коды условия не предсказуемы.

5.5.7. Потеря порядка

Потеря порядка возникает тогда, когда порядок оказался меньше чем самый маленький представленный для данного типа данных порядок, после проведения нормализации и округления. Если бит потери порядка (FU) установлен, операнд приемника не изменяется, а коды условия (CC) не предсказуемы, иначе результат будет нулевой.

5.5.8. Деление на ноль

При делении на 0 (делитель равен нулю) операнд приемника не изменяется, а коды условия (CC) не предсказуемы.

5.5.9. Ошибка при получении неправильного операнда

Эта ошибка возникает, когда один из операндов имеет неправильный формат. Неправильный формат имеет значение отрицательного нуля (знаковый разряд = 1, а порядок = 0).

5.6. Сопряжение и взаимодействие с процессором

5.6.1. Сигналы сопряжения

Связь между процессором и FPA осуществляется следующим образом. Центральный процессор посылает FPA команду, в которой указывается операция и форматы обрабатываемых операндов (F, D, G или H). FPA потом настраивает соответствующим способом свои схемы путей данных, чтобы выполнить требуемую операцию. Процессор посылает данные для обработки (порциями по 32 разряда), и FPA запускает заданную операцию. После получения результата он возвращает его процессору, с помощью которого он запоминается в памяти.

Сигналы сопряжения между процессором и FPA показаны на рис.7 в приложении и описаны в табл.3. Сигналы синхронизации CPU PZ H и PORT CLOCK L постоянно подаются в FPA. Процессор управляет работой FPA с помощью следующих сигналов: READ PORT L, SEL ACC IN H, READ ACC UPC L, TRAP ACC L, IRD START E L и CPU DATA AVAIL L. Единственный сигнал управления, который FPA посылает процессору, является ACC SYNC H (причем он не является частью результата, который FPA передает по шинам BUSY).

Таблица 3

Сигналы сопряжения	
Сигнал	Описание
BUS Y	32-разрядная шина, используемая для передачи данных между процессором и FPA в обе стороны
CPU PZ H	Импульс длительностью 90 ns, используемый процессором для синхронизации FPA. Длина микрокоманды для этого импульса - 270 ns.
PORT CLOCK L	Основной импульс длительностью 90 ns.

Продолжение табл. 3

Сигнал	Описание
READ PORT L	! Управляющий сигнал, посылаемый ! процессором FPA для ! включения в FPA выходного буфера с ! тремя состояниями
SEL ACC IN H	! Сигнал, используемый центральным ! процессором для выбора FPA. Когда он ! установлен, то дает возможность FPA ! управлять BUS Y для передачи результата ! выполненной операции
READ ACC UPC L	! Сигнал генерируемый процес- ! сором. В конце микрокоманды, в которой ! этот сигнал был установлен, FPA ! приостанавливает свои синхросигналы, ! тем самым запуская изменение следующего ! микроадреса (NUA). В следующий момент ! времени FPA установит сигнал CPU RCV ! DATA L и выдаст на BUS Y свой следующий ! микроадрес. После этого микросигналы ! будут вновь запущены
TRAP ACC L	! По этому сигналу процессор ! устанавливает на BUS YLY:UJ ! микроадрес, который будет следующим ! для FPA. Это происходит при необходи- ! мости прервать работу FPA при ошибках ! диспетчера памяти, различных прерыва- ! ниях. Этот сигнал используется также ! для запуска микродиагностической ! подпрограммы в FPA
BUS IB	! 8-разрядная шина для передачи кода ! операции
IRD STATE L	! Сигнал, идентифицируемый для ! FPA, что на BUS IB установлен код ! операции
CPU DATA AVAIL L	! Сигнал процессора, ! указывающий, что по BUS Y передается ! операнд процессору FPA
ACC SYNC H	! Сигнал, генерируемый FPA, который ука- ! зывает процессору, что FPA ! готов т.е. свободен от выполнения ! операций. Он также используется для ! синхронизации работы при передаче ! результата (с целью его запоминания) ! от FPA к процессору и при ! передаче операнда от ! процессора во время выполнения команды ! POLY

5.6.2. Операции по взаимодействию с процессором

5.6.2.1. Декодирование кода операции

На рис. 8 в приложении показана временная диаграмма работы FPA при декодировании кода операции, посылаемого процессором по шинам команды (BUS I6) в сопровождении сигнала IRD STATE L. Внутри FPA схема декодирования команды, в зависимости от кода операции, вырабатывает сигналы, которые микропроцессор использует для формирования начального адреса микрокоманды. Этот начальный микроадрес подается на управляющую память. Из управляющей памяти считываются микрокоманды, вырабатывающие сигналы, которые микросеквенсер использует для формирования начального адреса микрокоманды. Этот начальный микроадрес подается на управляющую память. Из управляющей памяти считываются микрокоманды, которые вырабатывают сигналы для управления логикой путей данных. С помощью этой логики операнды будут загружаться в FPA с шин BUS Y.

5.6.2.2. Загрузка операндов

На рис. 9 в приложении показана временная диаграмма работы при загрузке в FPA операндов из процессора. Первоначально центральный процессор устанавливает сигнал CPU DATA AVAIL L, который является синхронизирующим и указывает FPA, что данные находятся на шине BUS Y. Внутри FPA сигнал CPU DATA AVAIL L используется схемой ветвления.

Сигнал CPU DATA AVAIL L изменяет адрес следующей микрокоманды в FPA путем добавления по функции "ИЛИ" единицы в самый младший разряд (L65). Это заставляет микросеквенсер выйти из петли, в которой он находится. В этой петле (в которой происходит загрузка путей данных FPA) устанавливается сигнал ACC SYNC H FPA. Процессор игнорирует этот сигнал при передаче им данных FPA, кроме

случаев, когда передаются полиномиальные коэффициенты (во время команды POLY).

5.6.2.3. Запоминание результата

На рис.10 в приложении показана временная диаграмма передачи FPA результата процессору. Процессор сначала выбирает FPA путем установки сигнала SEL ACC IN H (поскольку могут быть другие устройства, связанные с этой шиной порта (BUS PORT)). После этого процессор устанавливает сигнал READ PORT L для инициирования получения результата от FPA.

FPA выполняет функцию "НЕ-И" над сигналом SEL ACC IN H и инверсным значением сигнала READ PORT L. Когда результат этой функции имеет низкий уровень, схема ветвления, выполняя функцию "ИЛИ" добавляет единицу в самый младший разряд следующего адреса микрокоманды. Это заставит FPA выйти из петли, в которой он находится (в которой результат постоянно подавался в процессор и устанавливался сигнал ACC SYL H). FPA никогда не может передавать данные процессору по шинам BUS Y, если сигналы SEL ACC IN и READ PORT не установлены.

5.6.2.4. Управление записью микроадреса

Процессор может запретить нормальную работу микросеквенсера FPA и загрузить в него свой микроадрес для очередного обращения к управляющей памяти. Это происходит, когда процессор вынужден прекратить выполнение команды арифметики с плавающей запятой из-за ошибок в диспетчере памяти или каких-либо прерываний. Процессор может также прочитать текущий адрес микрокоманды, предназначенный для управляющей памяти.

На рис.11 в приложении представлена временная диаграмма работы процессора при загрузке микроадреса для управляющей памяти FPA. Когда процессор устанавливает сигнал TRAP ACC L, запрещается поступление очередного микроадреса на управляющую память FPA с выхода микросеквенсера, FPA замедляет свой цикл (со 130 на 270 ns) и синхронизирует свою работу со стороны процессора. Последний устанавливает адрес микрокоманды на шину BUS Y, который затем поступает на шины следующего адреса микрокоманды BUS NUA L08:00J и тем самым выбирает следующую микрокоманду из управляющей памяти.

3.6.2.5. Управление чтением микроадреса

Процессор может прочесть текущий адрес микрокоманды, предназначенный для управляющей памяти. На рис. 12 в приложении показана временная диаграмма работы при чтении процессором текущего микроадреса управляющей памяти FPA. Первоначально процессор устанавливает сигнал READ ACC UPC, затем READ PORT L. Наличие этих сигналов настраивает схему управления FPA таким образом, чтобы подать выход микросеквенсера на шину BUS Y (предварительно зафиксировав его в буфере).

3.6.2.6. Ошибки по паритету и коды условий

FPA содержит схему, обнаруживающую ошибки паритета микрокоманды, и логику формирования кодов условия, с помощью которых процессору сообщается о состоянии и ошибках FPA.

Каждая микрокоманда в FPA дополняется до нечетности, и имеются схемы, обнаруживающие нарушение этой нечетности. Если обнаружена ошибка, то для указания, какая была ошибка(и) используется 3-разрядное поле (через BUS Y).

Микросхема программируемой матричной логики для кодов условия (ПМЛ) используется (помимо других целей) для сообщений об ошибках, когда операнды обрабатываются схемой путей данных. К этим ошибкам относятся:

- 1) неправильный операнд - отрицательный ноль;
- 2) деление на ноль;
- 3) переполнение в операции с плавающей запятой;
- 4) потеря порядка в операции с плавающей запятой;
- 5) ошибка паритета.

3.7. Команды и их алгоритмы

В табл. 4 приводится список команд, выполняемых FPA. Все арифметические команды используют два операнда, которые запоминаются в ячейках местной памяти процессора FPA TEMP 0 и TEMP 2. TEMP 0 содержит знак первого операнда (OP1), содержимое рабочего регистра порядка (EWR) ET0 и рабочий регистр мантиссы (FWR) FT0. TEMP 2 содержит знак OP2, EWR ET2 и FWR FT2.

Для арифметических команд с плавающей запятой, ис-

3.050.001P3

пользующих формат операндов учетверенной точности с расширенным порядком (формат H), для хранения мантиссы требуется два рабочих регистра. Для первого операнда OP1 используется FWR FT0 и FWR FT1, а для второго операнда OP2 - FWR FT2 и FWR FT3.

Для арифметических команд, использующих целые числа, операнды запоминаются в FT0 DL47:16J и в FT2 DL47:16J.

Таблица 4

Команды процессора операций с плавающей запятой FPA

Команда	Тип	Описание
ADD	Арифметический	Сложение
CMP	Арифметический	Сравнение
SUB	Арифметический	Вычитание
POLY	Арифметический	Вычисление полинома
DIV	Арифметический	Деление
MUL	Арифметический	Умножение
EMOD	Арифметический	Умножение с повышенной точностью с выделением целого числа
MULL	Арифметический	Умножение длинных слов
DIVL	Арифметический	Деление длинных слов
CVT F, D, G, H --> B	Преобразование	Преобразование из плавающего формата в байт
CVT F, D, G, H --> W	Преобразование	Преобразование из плавающего формата в слово
CVT F, D, G, H --> LW	Преобразование	Преобразование из плавающего формата в длинное слово
CVT F, D, G, H --> LW ROUNDED	Преобразование	Преобразование из плавающего формата в длинное слово с округлением

Продолжение табл. 4

Команда	Тип	Описание
CTV в F из D, G или H	Преобразование повышенной точности	Преобразование из форматов D, G, H в формат F
CTV в D из F или H	Преобразование повышенной точности	Преобразование из форматов F или H в формат D
CVT в G из H или F	Преобразование повышенной точности	Преобразование из форматов H или F в G
CVT в H из F, D или G	Преобразование повышенной точности	Преобразование из форматов F, D или G в формат H
CVT BYTE --> F, D, G, H	Преобразование	Преобразовать байт в плавающий формат
CVT WORD --> F, D, G, H	Преобразование	Преобразовать слово в плавающий формат
CVT L WORD --> F, D, G, H	Преобразование	Преобразовать длинное слово в плавающий формат

5.7.1. Арифметические команды

5.7.1.1. Сложение/вычитание

Алгоритм выполнения команды сложения приведен на рис. 13-18 в приложении. Перед тем, как два числа с плавающей запятой могут быть сложены или вычтены, их порядки должны быть выражены (если они не были равны). При выравнивании мантисса с меньшим порядком сдвигается вправо до тех пор, пока порядки не выравняются. При каждом правом сдвиге мантиссы порядок увеличивается на единицу.

Алгоритм команды сложения:

1) порядки не выражены

$$(,123 * 10^{**+5}) + (,456 * 10^{**+2});$$

3.050.001PЭ

2) выравнивание меньшего порядка /000456 *10**5

3) сложение чисел

/123	* 10**5
/000456	* 10**5

4) результат /123456 * 10**5

В начале операций сложения или вычитания FPA определяет, не равны ли порядки обоих операндов. Если они не равны, то определяется операнд с большим порядком. Для этого порядок операнда OP2 вычитается из порядка OP1. Если порядки не равны, то в этом случае FPA сначала проверяет, имеет ли смысл делать выравнивание порядков, т.е. не является ли больший порядок настолько большим, что выравнивание бессмысленно. Это произойдет тогда, когда число шагов выравнивания равно числу разрядов в мантиссе плюс единица. (например, для команд, обрабатывающих операнды одинарной точности - формата F - поле мантиссы содержит 24 разряда. Если при этом разность порядков окажется больше, чем 25, то выравнивание не имеет смысла).

Перед выравниванием порядка FPA определяет, какая должна выполняться операция: суммирование или вычитание. Суммирование выполняется, когда для команд ADD операнды имеют одинаковые знаки, а для команды SUB - разные. В том случае, если выполняемая операция - вычитание, то перед выравниванием порядков от операнда с меньшим абсолютным значением берется инверсное значение.

Чтобы выравнять мантиссу с меньшим порядком, разность порядков помещается в Q-регистр порядка (EQ), а меньшая мантисса - в Q-регистр мантиссы (FQ). Затем содержимое FQ сдвигается вправо с одновременным уменьшением на одну единицу при каждом сдвиге содержимого FQ до тех пор, пока FQ не обнулится. При каждом сдвиге порядок меньшего числа увеличивается на 1.

После выравнивания числа складываются, а затем нормализуются. Нормализация заключается в том, чтобы в самом старшем разряде MSE результирующей мантиссы была 1. Это достигается путем сдвига мантиссы и соответствующего изменения порядка.

Знак результата устанавливается согласно табл. 5

Таблица 5

Определение знака результата при выполнении команд сложения и вычитания

	Знаки исходных операндов		Знаки результата	
	знак OP1	знак OP2	OP1>OP2	OP2>OP1
Сложение	+	+	+	+
	+	-	+	-
	-	+	-	+
	-	-	-	-
Вычитание (OP2-OP1)	+	+	-	+
	+	-	-	-
	-	+	+	+
	-	-	+	-

Если порядки равны, то мантиссы складываются, если задана операция суммирования, или вычитаются, если операция вычитания. Если нужно было произвести операцию вычитания, то после ее выполнения необходимо проверить результат на равенство нулю.

После выполнения сложения и нормализации результат округляется и проверяется на переполнение или потерю порядка.

3.7.1.2. Команда сравнения (CMP)

Команда сравнения (CMP) сравнивает два операнда путем вычитания второго операнда из первого. Результатом выполнения этой команды является загрузка кодов условия следующим образом:

N \leftarrow 1 если OP1 меньше, чем OP2
 Z \leftarrow 1 если OP1 равен OP2
 V \leftarrow 0
 C \leftarrow 0

Алгоритм команды CMP:

1) если знаки не одинаковые, то N \leftarrow знак OP1, и

3.050.001P3

производится запоминание кодов условия (CC);

2) если знаки одинаковые, то производится вычитание порядков $OP1 EXP - OP2 EXP$;

3) если $OP1 EXP > OP2 EXP$, то $N \leftarrow$ знак $OP1$ и производится запоминание CC ;

если $OP1 EXP < OP2 EXP$, то $N \leftarrow$ инверсное значение знака $OP1$ и производится запоминание CC ;

4) если $OP1 EXP = OP2 EXP$, то производится вычитание мантисс;

5) если результирующая мантисса = 0, то в бит Z записывается единица ($Z \leftarrow 1$) и производится запоминание CC .

Если MSB результирующей мантиссы = 0, а сама мантисса не равна 0, то в бит N записывается знак $OP1$ ($N \leftarrow$ знак $OP1$) и производится запоминание CC .

Если MSB результирующей мантиссы = 1, то $N \leftarrow$ инверсное значение знака $OP1$ и производится запоминание CC .

5.7.1.3. Команда вычисления полинома (POLY)

Команда POLY вычисляет полиномиальное выражение вида:

$$A(0) + A(1)X + A(2)X^{**2} + A(3)X^{**3} + \dots$$

у которого наибольший показатель степени у X может быть 31. В команде требуются три спецификатора операндов:

1) ARG - аргумент (т.е. X);

2) DEGREE - наибольший показатель степени, в которую должен быть возведен X;

3) TABLE - адрес таблицы коэффициентов. Первый коэффициент в таблице фактически является последним коэффициентом полинома.

Полиномиальное выражение вычисляется следующим образом:

$$(((C(D)*X + C(D-1))*X + C(D-2))*X \dots \dots + C(1))*X + C(0)$$

где $C(D)$ = коэффициенты.

После умножения выделяется следующее число бит для

сложения:

формат F: 31 разряда;

формат D: 63 разряда;

формат G: 63 разряда;

формат H: 127 разрядов.

Следующий коэффициент затем прибавляется к произведению, число округляется и проверяется на наличие особого случая. Затем запускается следующая итерация.

FPA выполняет команду POLY путем выполнения итераций умножения/сложения и посылки результата назад в процессор. Следующая итерация стартует автоматически. Если команда завершена, процессор должен прекратить операцию в FPA.

Алгоритм команды POLY.

Инициализация:

1) запомнить аргумент в ET8, FT8 (FT9 для формата H);

2) запомнить первый коэффициент в ET2, FT2;

3) выходной знак <-- знак OP1 "исключающее ИЛИ" знак OP2;

4) переход к итерации.

ПРИМЕЧАНИЕ. Знак OP1 отражает знак аргумента.

Процедура итерации:

1) поместить аргумент в ET0, FT0, (FT1);

2) вызвать подпрограмму MUL.ROUTINE;

3) прочитать следующий коэффициент и загрузить в ET2, FT2 (FT3 для формата H);

4) вызвать подпрограмму ADD;

5) округлить и проверить на особый случай;

6) провести усечение и запомнить в ET2, FT2 (FT3);

7) запомнить коды условия и результат;

8) выходной знак <-- выходной знак "исключающее или" знак OP1;

9) переход к итерации.

3.050.001P3

ПРИМЕЧАНИЕ. Если в конце итерации MUL/ADD произойдет потеря порядка, то запоминается частичные результаты, и запоминается ошибочный код. Если бит FU установлен, то центральный процессор прекращает операцию в FPA. FPA автоматически стартует следующую итерацию. При переполнении FPA запоминает ошибочный код и останавливает выполнение операции.

5.7.1.4. Команда деления(DIV)

Деление выполняется командой DIV. Для этой команды выполняется следующая операция: частное $\leftarrow OP2/OP1$.

Алгоритм команды DIV:

1) выходной знак \leftarrow знак $OP1$ "исключающее ИЛИ" знак $OP2$;

2) обнулить FQ;

3) загрузить в EQ счетчик разрядов мантиссы;

4) вычесть мантиссу $OP1$ из мантиссы $OP2$, а затем перейти к петле DIV.

Петля DIV (если следующий результат получился положительным):

сдвинуть FQ влево, вдвинуть единицу;
вычесть $OP1$ из $OP2$;
декрементировать EQ, если его содержимое не равно 0 (NEQ.0), перейти к петле DIV;

Если предыдущий результат получился отрицательным:

сдвинуть FQ влево, вдвинуть ноль;
сложить $OP1$ с $OP2$;
декрементировать EQ, если его содержимое не равно 0 (NEQ.0), перейти к петле DIV.

Выход из петли DIV:

5) нормализация;

6) округление;

7) установка битов кодов условия и запоминание результата.

Команда DIVL используется только для деления длинных слов.

Алгоритм DIVL:

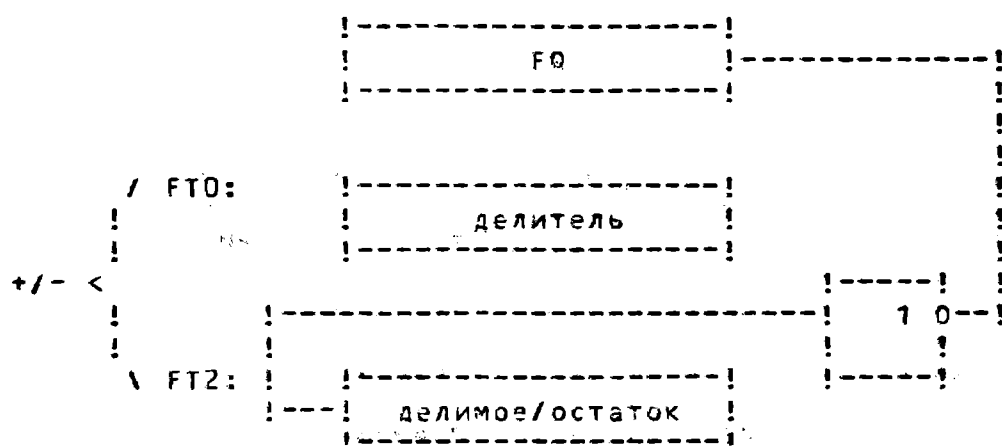
1) так как целые числа представляются в дополнительном коде, необходимо их прозерить, не являются ли

они отрицательными. Если операнд отрицательный, он инвертируется и ET1 инкрементируется (первоначально он был обнулен). Так, если ET2 = 1 после того, как оба операнда проверены и инвертированы, если необходимо, то тогда результат должен быть отрицательным;

2) делимое больше или равно делителю? Если нет, то результат = 0;

3) расположить делимое и делитель так, чтобы MSB был в FPC47. Установить в EQ единицу и инкрементировать EQ для каждого выравнивающего сдвига, которого требует делитель по отношению к делимому. В результате получим счетчик петли для петли деления.

Петля деления:



Вычесть (сложить) делитель из делимого (остатка). Инверсия знакового разряда результата является следующим разрядом частного, он также управляет функцией ALU. После петли деления проверяется ET1. Если ET1 равен 1, результат отрицательный. Переполнение проверяется путем проверки FPC47 для положительных чисел. Если FPC47 для положительных чисел равен единице, то произошло переполнение.

5.7.1.5. Команда умножения (MUL)

Команда MUL выполняет умножение форматов чисел F, D, G и H. Алгоритм ее выполнения следующий:

1) выходной знак <--> знак OP1 "исключающее ИЛИ" знак OP2;

2) поместить OP1 (множитель) в FQ;

3.050.001PЭ

- 3) обнулить FT4 (регистр произведения);
- 4) загрузить EQ счетчиком разрядов мантиссы;
- 5) сдвинуть EQ вправо.

ПРИМЕЧАНИЕ. Если $LSB = 1$, сложить $OP2$ и содержимое $FT4$ и сдвинуть вправо. Если $LSB = 0$, сдвинуть содержимое $FT4$ вправо.

6) декрементировать EQ , если его содержимое не равно 0 ($NEQ.0$), перейти к пункту 3;

7) переместить $FT4$ (произведение) в $FT0$;

8) нормализация.

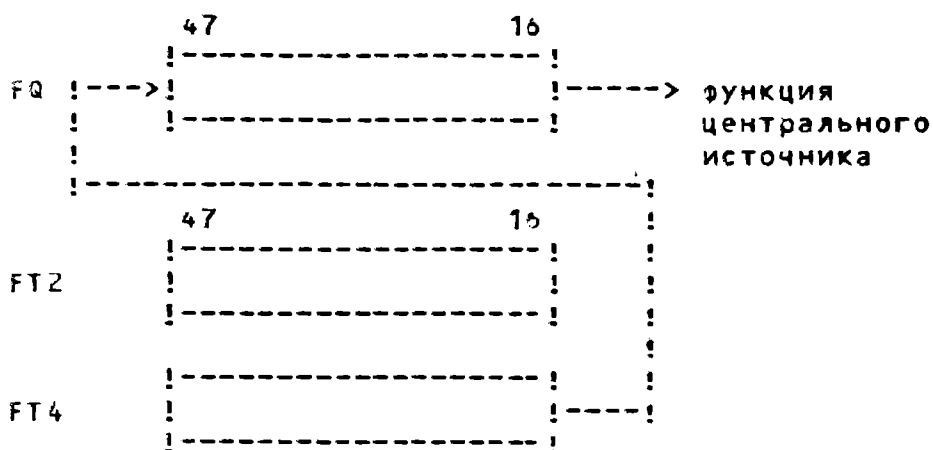
Когда мантисса нормализуется, порядок изменяется следующим образом, для каждого левого сдвига, порядок декрементируется, для каждого правого - инкрементируется.

9) округление.

FPA округляет результат операции над числами с плавающей запятой. Это осуществляется путем прибавления константы округления к результату. Константа округления зависит от типа данных и имеет единицу в том номере разряда, который на единицу меньше, чем LSB . (например для формата F константа округления имеет нули во всех разрядах, кроме разряда 31, в котором будет единица).

10) установка кодов условия (CC) и запоминание результата.

ПРИМЕЧАНИЕ. LSB множителя зависит от типа данных. Микросхема PAL умножения/деления (MUL/DIV) выбирает в зависимости от типа данных LSB (самый младший значащий разряд).



3.050.001PЭ

Команда MULL - команда целочисленного умножения MULL умножает только длинные слова. Умножение целых чисел имеет в основном тот же самый алгоритм, что и умножение чисел с плавающей запятой, кроме того, что эта команда использует пути данных для целых чисел.

Проверка переполнения также отличается: в конце умножения регистр FQ должен содержать расширение знакового разряда (FPA047) FT4. Если это не так, то произошло переполнение.

Умножение повышенной точности и выделение целого числа выполняется командой EMOD.

Основная функция микропрограммы команды EMOD - перемножить расширение множителя (MIER.EXT) на множимое (MAND), запустить петля умножения для остальных разрядов и вызвать подпрограмму CVT.FLT. В функции этой команды входят обработка нулевого операнда, установка кодов условия и обработка особых случаев.

Команда EMOD выполняется следующим образом:

```

      !--> OP1  !--> OP2  !--> OP3
      !         !         !
      !         !         !
TEMP <--- (MIER* (MIER.EXT)* (MAND)
      !
      !
      !--> (вложение)

```

MIER.EXT представляет собой байт для F и D, 11 разрядов для формата G (слева расположенных) и 15 - для формата H (слева расположенных):

После выполнения команды имеется два результата:

- 1) мантисса (того же формата данных, как и в команде);
- 2) целое число (длинное слово).

Аппаратура работает так, что расширение множителя (MIER.EXT) загружается в разряды [31:16] регистра FT4. Микропрограмма может заставить MUL/DIV ПМЛ выбрать Q16 в качестве недостающего LSB множителя. Таким образом, сначала осуществляется умножение на расширение множителя, а затем на OP1. При этом используется та же петля умножения.

Команда EMOD выполняется в следующей последовательности:

- 1) загрузить FT4 в FQ (MIER.EXT --> FQ);
- 2) EQ <-- счетчик циклов (8 = F, D; 11 = G; 15 = H);
- 3) установить Q16;

3.050.001P3

4) выполнить циклы умножения до тех пор, пока $EQ = 0$ здесь вызывается та же петля умножения, что и при обычном умножении;

5) $FQ \leftarrow FT0$; в FQ записывается множитель;

6) $EQ \leftarrow$ счетчик числа разрядов;

7) вызов подпрограммы MUL ;

8) вызов подпрограммы выделения целого числа;

9) вызов подпрограммы обработки целого числа;

10) нормализация мантииссы;

11) округление;

12) проверка на переполнение целого числа;

13) установка кодов условия (CC) и запоминание результата.

5.7.2. Команды преобразования чисел

Любые числа с плавающей запятой в форматы целых чисел преобразуются с помощью двух команд: $CVT(F, D, G, H)$ в (B, W, L) и $CVTR(F, D, G, H)$ в L .

Все команды преобразования похожи друг на друга, главное различие между ними - в числе циклов преобразования в зависимости от типов данных.

Если число с плавающей запятой слишком большое, чтобы быть представленным в формате целого, то устанавливается бит V , а целое число отрезает младшую часть мантииссы.

Команды CVT выполняются в следующей последовательности:

1) вычесть смещение из порядка, полученный результат будет указывать на число разрядов в целом числе;

$EQ \leftarrow ET0 - ET4$,

где $ET0$ = порядку

$ET4$ = смещению порядка

2) если EQ будет иметь отрицательный результат, то исходное число не имеет целой части, в результате будет запоминаться 0;

3) если содержимое EQ не отрицательное, то проверя-

ется нет ли переполнения;

$EQ = ET0 - ET4$
(число разрядов в целом числе)

$ET7 \leftarrow ET6 - E0$

где $ET6$ = числу разрядов для целого в данном формате (для данного слова число разрядов равно 32);

4) если содержимое $ET7$ не равно и не меньше 0, то осуществляется перевод на петлю преобразования.

ПРИМЕЧАНИЕ. $ET7$ = числу разрядов мантиисы (числу разрядов целого).

5) если же оказалось, что для данного числа требуется больше разрядов, чем разрядная сетка данного типа целого числа, то исходное число слишком большое, чтобы быть представленным в данном типе целого числа;

6) если $ET7$ больше либо равно нулю, тогда делается проверка, какие разряды числа будут значащими.

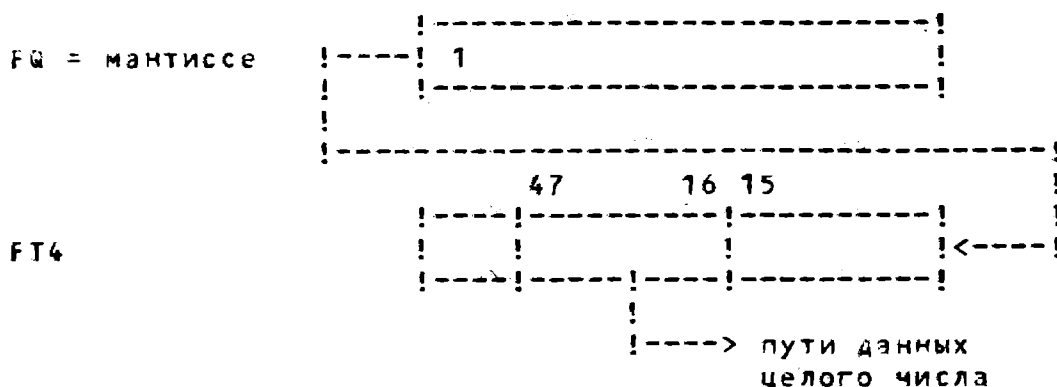
$ET7 \leftarrow ET7 - ET4;$

$ET7$ = числу разрядов целого, в формате которого переводится результат

$ET4$ = числу разрядов в результирующем целом числе;

7) если $ET7$ меньше 0, тогда результат = 0, и должен быть установлен V-бит. Если $ET7$ больше либо равно 0, тогда осуществляется переход к петле преобразования.

Петля преобразования: пересылка $FT0$ в FQ



Затем осуществляется левый сдвиг FQ и $FT4$ на число разрядов, указанное в EQ , которое содержит число разрядов целого числа.

В конце петли преобразования это число должно быть выравнено с путями данных мантиисы путем выполнения 12

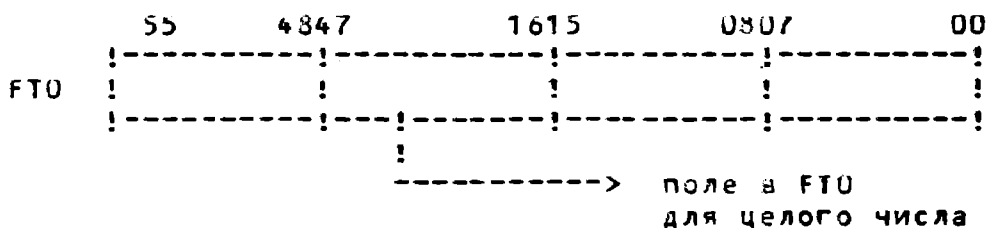
двойных сдвигов.

Команда CVT(B, W, L) (F, D, G, H) преобразует целое число в число с плавающей запятой.

Целое число любого формата может быть преобразовано в число с плавающей запятой также любого формата без переполнения или потери порядка. Поскольку команда преобразования CVTLF может потерять значащие цифры, то она требует округления.

Алгоритм преобразования:

1) целое число загружается в пути данных целого (в поле разрядной сетки целого числа).



2) самый старший разряд целого числа (MSB) выравнивается с FPAСБ5:

для байта MSB = 23;

для слова MSB = 31;

для длинного слова MSB = 47.

Все это требует:

4 двойных левых сдвига для длинного слова;

12 двойных левых сдвигов для слова;

16 двойных левых сдвигов для байта.

3) после выравнивания целого числа проверяется его MSB. Если он равен единице, то от числа берется дополнительный код и устанавливается знаковый разряд.

4) F_ц ← избыток порядка плюс число разрядов в данном формате целого числа.

5) число затем нормализуется (и округляется, если CVTLF), устанавливаются коды условия (CC), и происходит запоминание результата.

Пример: Команда CVTLF, где LW = 4000000.

Алгоритм выполнения команды CVTLF:

3.050.001P3

	55	48	47	
1) загрузить FT0:	-----			
		04000000		

2) выравнять FPAC47	-----			
с FPAC55:	04	000000		00

3) загрузить FQ сдвигом (избытком) порядка плюс числом разрядов целого числа в формате длинного слова, т. е. 32(10-ная с/с): EQ <-- 80 + 20.

4) MBS мантиссы = 0, поэтому знак <-- 0.

5) нормализация мантиссы:

EQ	-----	-----
	A0	04 0-----0
	-----	-----
	-----	-----
	9B	80 -----0
	-----	-----
	-----	-----
	!---- после 5 сдвигов влево	

FPA выполняет четыре команды преобразование одного формата числа с плавающей запятой в другой:

CVTF (D, G, H); CVTD (F, H); CVTG (F, C); CVTH (F, D, G).

Чтобы преобразовать один формат числа с плавающей запятой в другой необходимо:

1) вычесть избыток порядка из порядка (имеется в виду избыток исходного формата);

2) привести к порядку избыток формата, в который производится преобразование;

3) произвести округление, если это необходимо (таким образом, команда CVTFD не требует округления);

4) проверить, не возникло ли переполнение или потеря порядка.

Пример: CVTFG 4080

		55	32 31	00
загрузить FT0	---	-----	-----	-----
		61	10-----0	0-----0
		-----	-----	-----
			FTQ	

3.050.001PЭ

				55	32	31	00
вычесть	!-----!			!-----!	!-----!	!-----!	!-----!
избыток	----- ! 01 !	FTQ	!	10-----0 !	0-----0 !		
порядка	!-----!			!-----!	!-----!	!-----!	!-----!
				55	32	31	00
прибавить	!-----!			!-----!	!-----!	!-----!	!-----!
новый избыток	-- ! 401 !	FTQ	!	10-----0 !	0-----0 !		
порядка	!-----!			!-----!	!-----!	!-----!	!-----!

Никакого переполнения или потери порядка не произошло (они не возможны для данного типа преобразования).

Затем необходимо скомпоновать число в новом формате и запомнить результат: 4010.

5.8. Работа процессора с плавающей запятой

5.8.1. функциональное описание

Блок-схема процессора плавающей запятой дана на рис. 19 в приложении.

Основной частью процессора операций с плавающей запятой является схема путей данных, которая обрабатывает операнды различной длины. Операнды поступают в процессор FPA из процессора порциями по 32 разряда через шину BUS Y. Прежде чем попасть на внутреннюю шину BUS FPA операнды с BUS Y проходят схемы приемопередатчиков с тремя состояниями. Пути данных разделены на поля, которые обрабатывают порядок числа и мантиссу. Кроме того, имеются схемы управления знаком мантиссы и формирования кодов условия. Управление схемой путей данных осуществляется от управляющих сигналов, поступающих от управляющей памяти (памяти микрокоманд).

Команды арифметики с плавающей запятой, которые должны быть обработаны FPA, поступают из центрального процессора по линиям шины BUS I3 D L7:U1 в схему декодирования команд FPA. Код операции команды преобразуется этой схемой в пятиразрядный адрес DECODE ROML4:0JH, который затем используется микросеквенсором для формирования окончательного адреса управляющей памяти. Этот микроадрес (BUS NUAL9:0JH) выбирает из управляющей памяти 48-разрядное микрослово (микрокоманду). Выбранная микрокоманда принимается в регистр микрокоманд, из которого управляющие сигналы поступают на управление логикой путей данных.

Пять младших разрядов 10-разрядного микроадреса мо-

3.050.001PЭ

гут модифицироваться в микросеквенсере в зависимости от сигналов, генерируемых логикой путей данных или схемой декодирования команд, таким образом осуществляя процедуру ветвления. Два младших разряда [1:0] поля микроадреса обычно могут изменяться схемой управления ветвления. Расширенные функции ветвления могут осуществляться путем объединения по "ИЛИ" разрядов [4:2] поля микроадреса с сигналами состояния (сигналами изменения адреса). Таким образом, максимально могут изменяться пять разрядов микроадреса.

Схема контроля по паритету FPA проверяет каждую микрокоманду, считанную из управляющей памяти. Если обнаруживается ошибка по паритету, то эта схема генерирует сигнал FORCE LOW UADDRL, который устанавливает на всех 10 линиях адреса микросеквенсера логические нули. Это будет адрес следующей микрокоманды, вызывающей микроподпрограмму обработки ошибки паритета.

FPA имеет группу приемопередатчиков с тремя состояниями для осуществления чтения/записи адреса микрокоманды. В диагностическом режиме микроадрес BUS NUA [9:0]H может выдвзаться на BUS Y (как BUS Y [9:0]H) для последующей проверки в процессоре. В режиме записи центральный процессор может установить свой микроадрес (который также передается по BUS Y) к управляющей памяти FPA, после установки которого микропрограмма начнет исполняться с этого микроадреса. Указанные операции используются для проверки микросеквенсера, управляющей памяти и логики путей данных.

5.8.1.1.

Прохождение данных

Работа аппаратуры при выполнении команд арифметики с плавающей запятой осуществляется следующим образом. Процессор читает код операции, который затем пересылает FPA по шинам BUS IS. FPA после декодирования команды вызывает соответствующую микропрограмму. При последующих обращениях процессор передает FPA операнды по шине BUS Y. Затем FPA обрабатывает входные данные в зависимости от выполняемой команды. Результат исполнения команды FPA устанавливается на шине BUS Y и передается процессору.

В процессе работы схемы путей данных вырабатывают сигналы состояния (сигналы изменения микроадреса). Эти сигналы, воздействуя на базовый микроадрес из управляющей памяти, могут его изменить перед тем, как он поступит на шины BUS NUA [09:00]. Таким образом осуществляется ветвление.

Если возникает определенные ситуации во время цикла передач данных между FPA и процессором, последний прекращает работу FPA. Об ошибках или особых случаях FPA

сообщает процессору через шину BUS Y.

Когда операнды читаются, логика путей данных FPA настраивается на прием их с BUS Y. Первоначально код операции с BUS Ib поступает на память ROM декодирования команды. 5-разрядное выходное поле с этого ROM используется микросеквенсером KM18U48Y1. В результате своей работы микросеквенсер генерирует адрес микрокоманды (микроадрес) на шинах BUS NUAL9:UJ, который поступает на вход управляющей памяти PROM. По этому адресу считывается 48-разрядное управляющее микрослово (микрокоманда), которое принимается в регистр микрокоманд CSR и осуществляет управление схемами FPA. Эта микрокоманда инициирует следующие действия:

1) настраивает микропроцессоры KM18U48C1, обслуживающие как мантиссу, так и порядок, на установку в ноль рабочих регистров порядка EWR(0) и мантиссы FWR(0) с тем, чтобы быть готовым принять первый операнд (OP1) с линии BUS Y;

2) устанавливает сигнал загрузки, разрешающий прием информации в регистры EWR и FWR. Этот сигнал является результатом определенных значений толеи CLK и MOD, входящих в состав микрокоманды;

ПРИМЕЧАНИЕ. Сигнал загрузки всегда очищается в начале каждой команды.

3) поле адреса следующей микрокоманды из управляющей памяти PROM используется для формирования микроадреса на шинах BUS NUAL9:UJ. В результате вызывается соответствующая микропрограмма. На диаграммах микропрограмм FPA это будет обозначаться следующим образом:

CALL(FET.FLT) (вызов микропрограммы FET.FLT)
или
CALL(INT.FLT)

После вызова этой микропрограммы производится постоянная загрузка (то есть повторение цикла загрузки) рабочих регистров (WR) логики путей данных до тех пор, пока центральный процессор не установит сигнал CPU DATA AVAIL L.

На рис. 20 и 21 в приложении показано, как производится загрузка операндов в схему путей данных. В том случае, если операнды имеют длину более, чем одно длинное слово (для форматов D, G или H), FPA синхронизирует свою работу с процессором только для первого слова, остальные длинные слова принимаются FPA без указанной выше синхронизации.

После того, как FPA прочел все операнды из процессора, длительность микрокоманды уменьшается с 270 до 180 ns. Это уменьшение производится в самом начале микропрограммы исполнения команд.

3.050.001PЭ

Поскольку порядок в форматах данных G и H не спозиционирован под схему путей данных порядка, часть его должна загружаться в схему путей данных мантиссы. Затем эта часть порядка должна быть передвинута в схему путей данных порядка. Для обоих операндов это производится одновременно в микропрограмме обработки расширенных порядков. Для этого OP2 помещается в Q-регистр порядка (EQ) и в Q-регистр мантиссы (FQ), далее EQ и FQ одновременно сдвигаются с рабочими регистрами EWR(O) и FWR(O), содержащими OP1. Затем схема управления сдвигом мантиссы направит MSB регистров FQ и FWR(O) соответственно на входы левого сдвига EQ и EWR(O).

Процедура, указанная выше, выполняется для форматов G. Для форматов данных H в данный момент времени может обрабатываться только один операнд так как и рабочий регистр мантиссы (FWR) и Q-регистр мантиссы (FQ) используются для сдвига только мантиссы операнда H. Младшая половина мантиссы H первоначально загружается в FQ, а старшая во временную ячейку FWR. Затем выполняется левый сдвиг, а MSB регистра FQ поступает на вход левого сдвига временной ячейки FWR. После этого MSB ячейки FWR поступает в EWR. Таким образом, для обработки операнда H требуется семь сдвигов.

После обработки операнда G или H в ПМЛ ветвления (BRANCH 3 PAL) устанавливаются сигналы OP1 EQ 0 и OP2 EQ 0. Для операндов F и D это делается автоматически, когда принимаются разряды знака. Однако, для операндов G и H это невозможно, поскольку часть поля порядка для этих типов данных первоначально загружается в схему путей данных мантиссы.

После того, как процессор передал FPA операнды и попытался вычислить адрес приемника, он готов принять коды условия (путем установки сигнала READ PORT L), и затем переходит в ожидание до тех пор, пока FPA не установит сигнал ACC SYNC H или не произойдет прерывание. Если произошло прерывание, процессор обычно прекращает работу FPA и переходит на обслуживание прерывания.

Аналогичную работу выполняет при запоминании результата FPA. Он формирует коды условия и выполняет переход на цикл ожидания, пока процессор не установит сигнал READ PORT L. В этот момент он также устанавливает сигнал ACC SYNC H.

Перед запоминанием результата FPA должен его подготовить. Это означает выдвигание скрытого бита и выполнение необходимого числа сдвигов для перемещения порядка в схему путей данных мантиссы. FPA проверяет также установлен ли сигнал загрузки схемы путей данных или нет.

Процессор прекращает работу FPA в следующих случаях:

- 1) при возникновении прерывания;

3.050.001PЭ

- 2) при ошибках диспетчера памяти;
- 3) при получении неправильного режима адресации;
- 4) в конце исполнения команды POLY.

Чтобы прервать работу FPA процессор подставляет в качестве очередного адреса микрокоманды FPA микроадрес 7. Этот адрес инициирует микроподпрограмму, которая приводит в исходное состояние некоторые регистры FPA и переводит его в петлю ожидания.

Имеется ряд ситуаций, при которых FPA должен указать об этом в процессор:

- 1) переполнение (особый случай);
- 2) потеря порядка (особый случай);
- 3) неправильный операнд;
- 4) деление на ноль;
- 5) ошибка паритета.

Если любая из ошибок имела место, FPA устанавливает бит "С" в коде условия, который при передаче процессору является LS3 на шине BUS Y. Поскольку процессор проверяет этот бит первым во время операции запоминания результата, наличие единицы в нем немедленно вызовет микропрограмму обработки ошибок. Далее процессор получает из FPA код ошибки (в формате длинного слова). Этот код ошибки вместе с кодами условия используется процессором для определения ситуации, которая возникла в FPA. Код ошибки формируется микропрограммой FPA и посылается в процессор. Значения кодов ошибки приведены в табл.6.

Таблица 6

Коды ошибок	
Код	Ошибка
0	! Переполнение, если V-бит=1
0	! Потеря порядка, если V-бит=0
7F80	! Неправильный операнд
FF80	! Деление на ноль
X-XI (LSB=1)	! Ошибка паритета

После того, как FPA передал код ошибки по шинам BUS Y, он подготавливается для выполнения следующей команды и переходит в петлю ожидания. Однако, если произойдет ошибка паритета, FPA остается в микрокоманде 1, и чтобы запустить его снова, процессор должен установить FPA микроадрес.

5.8.1.2. Синхронизация процессора плавающей запятой

FPA работает с длительностями микрокоманд 180 и 270 ns. Короткая длительность микрокоманды 180 ns является нормальной длительностью FPA в процессе его работы по выполнению команды. Более длинный цикл микрокоманды 270 ns используется FPA при получении им операндов или команды из процессора, а также при выполнении запоминания результата исполнения команды. Логика формирования синхросигналов (рис.22 в приложении) состоит из синхрогенератора (реализованного на микросхеме ПМЛ) и схем "НЕ-И".

Логика синхронизации генерирует сигналы DPO CLK L, DP1 CLK L и REG CLK L, которые используются для работы управляющей памяти, логики путей данных, логики ветвления и схем управления. Хотя эти сигналы выдаются тремя отдельными схемами "НЕ-И" (для обеспечения нагрузочной способности), они генерируются идентичным способом. Логика синхронизации генерирует также сигналы IR CLK L и IR CLK H, которые используются для работы схем декодирования команд. Импульс 45 ns TRISTATE DISA H, выдаваемый в начале каждой микрокоманды, отключает приемопередатчики FPA для того, чтобы исключить их одновременное включение.

Синхрогенератор ПМЛ генерирует сигналы или SLOW PATH ENAB H, или FAST PATH ENAB H, а также FP PH1 и CPU PHU H (рис.23 и 24 в приложении). Эти сигналы используются вентелями, предназначенными для формирования цикла времени 270/180 ns. На вход указанного ПМЛ поступают ENB CLK(1)H и BASIC CLOCK H (от сигнала PORT CLOCK L контроллера памяти). Эти входы используются для генерации сигналов DP1 CLK L, DPO CLK L и REG CLK L. BASIC CLOCK H используется также для генерации IR CLK H и IR CLK L.

Когда нет сигнала FAST CYCLE L, работает медленный цикл. Во время операций с медленным циклом синхрогенератор ПМЛ генерирует сигнал SLOW PATH ENB, который совместно с CPU P2 H (рис.25 в приложении) выдает сигналы синхронизации.

На рис.26 в приложении показано формирование

5.050.001P3

быстрых и медленных циклов.

Если процессор установил сигнал TRAP ACC L или READ ACC UPC L и работает в PH1, то с указанного ПМЛ берутся сигналы FP PH1 H и FAST PATH ENB H для генерации CLK ENB H.

Во время нормального быстрого цикла (рис.27 в приложении), когда сигналы TRAP ACC или READ ACC UPC не установлены процессором, для генерации CLK ENB H используются сигналы FP PH1 и FAST PATH ENB H.

Когда процессор устанавливает сигнал READ ACC UPC L, генератор ПМЛ производит сигнал CLOCK OFF, который отключает работу быстрого и медленного циклов, таким образом запрещая прием в регистры FPA.

Режим короткого цикла микрокоманды снимается, когда процессор устанавливает сигнал TRAP ACC L. С этого момента FPA будет синхронизировать свою работу с процессором.

Входной сигнал схемы синхронизации READ ACC UPC L заставит выдать FPA процессору состояние BUS NUA из микросеквенсера, когда будет установлен сигнал CPU RCV DATA L.

Когда процессор устанавливает сигнал FORCE UADDR L, сигнал FAST CYCLE (внутри генератора ПМЛ) снимается, и короткий цикл FPA растягивается (насколько требуется) так, чтобы в конце текущего цикла FPA смог бы засинхронизироваться с процессором.

На рис.28 и 29 в приложении показано замедление циклов FPA. Это может произойти, когда FPA меняет длительность цикла (по соответствующей микрокоманде) или когда процессор устанавливает сигналы TRAP ACC L или READ ACC UPC L. Любой из этих сигналов замедлит работу FPA до тех пор, пока он не засинхронизируется с процессором.

5.8.1.3. Логика декодирования команд

Логика декодирования команды FPA (рис. 30 в приложении) дешифрует команду арифметики с плавающей запятой, полученную по шине BUS IR. В результате декодирования формируются:

1) 5-разрядное смещение для начального микроадреса, вырабатываемого схемой микросеквенсера;

2) 2-разрядный код длины данных (SIZE1:01H).

Код длины данных указывает схемам управления тип

3.050.001P3

данных (F, D, G или H), который поступит из процессора, и настроит FPA на обработку операндов соответствующего типа. Декодирование команды производится соответствующей схемой ROM, схемой управления расширенными функциями и мультиплексором.

Начальная часть микропрограммы FPA зависит от поступившего кода операции команды (BUS IB D[7:0]H), который является адресом к памяти ROM с организацией 512x8. Выходом этой памяти ROM является DECODE ROM [7:0], часть из которого поступает на микросеквенсор для участия в формировании микроадреса для управляющей памяти на шине BUS NUA[9:0]H. Этот адрес микрокоманды будет начальным в микропрограмме FPA (рис. 31 в приложении). Выход ROM DECODE ROM[6:0] поступает на вход мультиплексор-летчей, которые в свою очередь управляются IRD STATE L и IR CLK H. Выходами этих летчей являются INSTR ENCC[4:0]H и SIZE[1:0]H.

На выходах летчей SIZE[1:0] указывается тип данных (F, D, G или H), который будет получен в качестве операндов от процессора по шинам BUS Y. В табл.7 дается соответствие между получаемыми типами данных и их кодами в поле SIZE[1:0]H.

На рис.32 в приложении показаны входные сигналы мультиплексора при нормальной операции и в диагностическом режиме. В диагностическом режиме процессор устанавливает входные сигналы на BUS FPA D[17:10] для проверки работы указанного мультиплексора и других схем декодирования команды.

Таблица 7

Кодирование поля SIZE[1:0]

Значение поля SIZE[1:0]H	Используемый тип данных
0	F (формат данных одинарной точности)
1	D (формат данных двойной точности)
2	G (формат данных с расширенным порядком)
3	H (формат данных учетверенной точности с расширенным порядком)

Прием осуществляется при установке BUS FPA D[18 H и TRAP ACC L, которые заставляют внутри синхрогенератора FPA сигналу IRD+FORCE H объединится по "И" с сигналом CPU P2 H, чтобы произвести IR CLK H. Если IRD STATE L не установлен, то тогда выбираются BUS FPA D[17:10], которые загружаются далее в регистр команд. Таким образом, BUS FPA D[17:10] после принятия в регистр команд формируют INSTR ENCC[4:0]H и SIZE[1:0].

3.050.001P3

Когда по шинам BUS IB поступает расширенный код операции, на выходе схемы управления расширенных функций устанавливается сигнал EXTENDED FUNC (1) H.

5.8.1.4. Формирование микроадреса

Логика микросеквенсера FPA (рис. 33 в приложении) генерирует последовательность 10-разрядных микроадресов (на шину BUS NUA[9:0]H), которая затем поступает на управляющую память. По этим микроадресам из памяти считываются микрокоманды, которые генерируют управляющие сигналы для различных схем FPA.

Логика микросеквенсера содержит три 4-разрядных микросеквенсера K1804BY1 плюс схемы управления. Хотя три микросхемы K1804BY1 могут генерировать 12-разрядный микроадрес, они сконмутированы так, чтобы генерировать только 10-разрядный адрес микрокоманды. Этого достаточно для того объема управляющей памяти, который имеет FPA.

Микросеквенсер имеет две группы входов. Одна из этих групп принимается в начале работы FPA с логики декодирования команд в виде сигналов DECODE ROM[4:0]H. Другая группа входов представляет собой регистровый вход, который управляется 10-разрядным полем микроадреса CSL[9:0] из управляющей памяти. Это поле может быть модифицировано и тем самым осуществляется ветвление.

Три микропрограммных секвенсера KM1804BY1 (рис. 34 в приложении) содержат четырехходовый мультиплексор, который используется, чтобы выбрать:

- 1) адресный регистр (AR);
- 2) прямые входы (D);
- 3) микропрограммный счетчик (MPS);
- 4) содержимое стека (F).

Один из перечисленных входов может выбираться в качестве базового адреса следующей микрокоманды. Выбор осуществляется в зависимости от кода на двух выходных линиях логики выбора адреса. Этот код управляется U[CTRL][4:2](1)H от логики ветвления FPA.

Адресный регистр KM1804BY1 состоит из четырех фронтových D-триггеров, принимающих информацию по импульсу DPU CLK L, поступающему из схемы синхронизации FPA. Новые данные вводятся в регистр по переходу с низкого уровня на высокий импульс DPU CLK. Выходы адресного регистра поступают в микросхеме KM1804BY1 на мультиплексор, как источник для следующего адреса микрокоманды (микроадрес NUA[y:0]H).

3.030.001PЭ

На прямой вход мультиплектора (D) поступают DECODE ROM[4:0]H логики декодирования команд. Этот вход используется для следующего микроадреса в состоянии IRD.

Вход С микросхемы KM18048Y1 заставляет адрес микрокоманд в регистре KM18048Y1 последовательно увеличиваться с каждым следующим импульсом DP0 CLK на единицу (то есть текущий BUS NUA[9:0]H плюс 1).

Содержимое стека (файла) также может использоваться, как источник следующего адреса микрокоманд. Стек используется для возврата к адресу микрокоманды, с которого ранее был произведен уход на микроподпрограмму. Стек содержит встроенный в него указатель стека (SP), который всегда указывает на последнюю записанную в стек ячейку. Это позволяет обращаться к стеку без процедур "заталкивания" и "выталкивания".

SP работает как реверсивный счетчик с использованием отдельных входов POP L и FILE ENB. Когда вход FILE ENB микросеквенсера KM18048Y1 имеет низкий уровень, а POP L - высокий, то осуществляется операция записи в стек. При этом указатель стека увеличивается, а в стек последовательно записывается содержимое счетчика микрокоманд, который содержит адрес текущей микрокоманды плюс 1.

Если оба входа FILE ENB и PUSH микросеквенсера KM18048Y1 имеют низкие уровни, то осуществляется операция считывания из стека. С помощью этой операции производится возврат из микропрограмм. Адрес возврата равен адресу вызова плюс 1. Следующий переход с низкого уровня на высокий сигнала DP0 CLK заставит SP декрементироваться. Если FILE ENB имеет высокий уровень, то действия, выполняемые SP не зависят от других входов.

Связь через указатель стека такова, что может быть достигнута любая комбинация записи, считывания и обращения к стеку. Так как глубина стека - четыре слова, то могут быть задействованы до четырех микроподпрограмм.

Входной сигнал FORCE ZERO микросеквенсера KM18048Y1 предназначен для того, чтобы установить в нулевое положение все 10 микроадресных линий BUS NUA[9:0]H. Когда установлен сигнал FORCE LOW UADDR L соответствующей логикой управления, все 10 адресных выходов имеют нулевое значение независимо от других входов (кроме OUTPUT ENABLE).

Формирование следующего микроадреса при ветвлениях осуществляется в зависимости от сигналов изменения микроадреса (сигналов состояния), которые в свою очередь вырабатываются логикой путей данных, а также зависят от выполняемой команды. Эти сигналы могут воздействовать на BUS NUA[1:0]H или BUS NUA[4:0]H. Логика ветвления состоит из регистра состояния и пяти микросхем ПМЛ. Четыре из них используются при обычных ветвлениях при формиро-

5.030.001P3

вании двух самых младших разряда BUS NUA, и все указанные микросхемы ПМЛ участвуют при выполнении расширенных ветвлений.

Сигналы состояния от логики путей данных записываются в регистр состояния. Они принимаются в регистр по импульсу CPU CLK L и затем поступают на входы микросхем ПМЛ, формирующих сигналы ветвления. Эти ПМЛ управляются полем UBCTL[4:0](0)H, поступающим из управляющей памяти. Это поле определяет, какой разряд состояния (сигнал изменения микроадреса) или комбинация этих разрядов поступят на выходные линии ПМЛ BRANCH[1:0]H. В табл. 8 приводится перечень сигналов, которые выбираются полем ветвления для модификации адресов микрокоманд.

Таблица 8

Кодирование выходных линий BRANCH[1:0]

UBCTL [4:0](1) H (16 с/с)	Выходные линии ПМЛ ветвления		
0	BRAN1 EXP COUT	BRAND GRAND	Специальные условия
1	SIGN OUT	HUGE	
2	CPU DATA AVAIL	SINGLE	Установка сигнала SYNC
3	CPU DATA AVAIL	ADD+SUB	Установка сигнала SYNC
4	FRAC COUT	EXT FUNC	
5	OP1 SIGN	EMOD	
6	FRAC55 F3	SINGLE	
7	OP2 SIGN	ADD+SUB	
8	EXP COUT	EXP15 F3	
9	SIGN OUT	OP2=0	
A	CPU DATA AVAIL	ZERO	
B	CPU DATA AVAIL	ZERO	
C	OP2 SIGN	(OP1+OP2)/=0	
D	OP1 SIGN	(OP1+OP2)/=0	
E	FRAC55 F3	0	
F	FRAC COUT	EXP15 F3	

Продолжение табл. 8

UBCTL			
L4:01(1)			
H			
(16 с/с)			
Выходные линии ПМЛ ветвления			
10	MUL 1I	FRAC55 Q3	
11	F47 F3	EXT00 Q0	
12	FRAC[55:00]=0	DIV 13	
13	FRAC[47:16]=0	ZERO	
14	FRAC[55:00]=0	CPU RCV DATA	
15	ZERO	ZERO	Нулевое ветвление
16	ZERO	CPU RCV DATA	Установка сигнала SYNC
17	FRAC[55:7]=0	ZERO	
18	EXPONENT=0	EXP15 F3	
19	OP1=0	OP2=0	
1A	ZERO	ZERO	Вызов подпрограммы
1B	SUMPATH	ZERO	
1C	ZERO	(OP1+OP2)/=0	Возврат из подпрограммы
1D	ZERO	(OP1+OP2)/=0	Возврат из подпрограммы
1E	ZERO	ZERO	Возврат из подпрограммы
1F	ZERO	EXP15 F3	Возврат из подпрограммы

Расширенное ветвление воздействует на выходы микро-секвенсера BUS NUA[4:2]. Это ветвление иногда используется для осуществления переходов по многим направлениям и при этом используются поля управляющей памяти CLK CTL и MOD. Старшие два разряда [4:3] поля UBCTL определяют, какой тип расширенного ветвления должен использоваться. В табл. 9 приводятся варианты расширенного ветвления.

Таблица 9

Расширенное ветвление			
Значение !	Разряды расширенного ветвления		
U3CTL !	[4:3](1)H!		
	BRAN4	BRAN3	BRAN2
0	DOUB OPER	ADD+SUB	FRAC31-EXT00=0
1	SIZE1	SIZE0	FRAC31:0J=0
2	DOUB OPER	ADD+SUB	ZERO
3	INSTR ENC2	INSTR ENC1	INSTR ENC0

5.8.1.5. Управляющая память

Выполнение операций арифметики с плавающей запятой осуществляется по микропрограмме, представляющей собой последовательность микрокоманд. Микрокоманды считываются из управляющей памяти (рис.35 в приложении) и генерируют сигналы управления данными, которые и используются логикой путей данных. После того, как операнды с шины BUS Y загружены в схему путей данных, они начинают там обрабатываться в соответствии с сигналами, поступающими из микрокоманд.

Схема управляющей памяти FPA состоит из памяти PROM и нескольких регистров. Управляющая память PROM содержит 1К 48-разрядных микрослов. Каждое микрослово (микрокоманда) содержит 2-разрядное поле паритета. 48-разрядное микрослово считывается из памяти PROM в регистр микрокоманд в зависимости от адреса на шинах BUS NVALY:0J H, поступающего из микросеквенсера. Из регистра микрокоманд управляющие сигналы поступают на логику путей данных, а поле микроадреса - на микросеквенсер. На рис.36 в приложении представлен формат микрокоманды, а в табл.10 дается описание и назначение всех функциональных полей, входящих в состав микрокоманды.

Таблица 10

Поля управляющей памяти		
CS !	Функция !	Описание
47	ACC SYNC	Сигнал синхронизации (идентификации) FPA
46	бит паритета P1	Бит паритета для проверки разрядов: CSL44:39J, CSL36:22J и CSL14:09J
45	бит паритета P0	Бит паритета для проверки разрядов: CSL47J, CSL38:37J, CSL21:15J и

Продолжение табл. 10

CS	Функция	Описание																																												
		CS[08:00]																																												
44:43	Управляющее поле приемника для порядка (EXP DST)	<p>Это поле указывает, куда перелается порядок с выходов ALU. Нормально информация с выходов ALU может приниматься или в рабочий регистр (WR) или Q-регистр.</p> <p>EXT DST[1:0] Куда принимается</p> <p>00 Q-регистр</p> <p>01 Рабочий регистр (WR)</p> <p>10 Правый сдвиг и запись в WR</p> <p>11 Левый сдвиг и запись в WR</p>																																												
42:39	Управление путями данных порядка (EXP CTL)	<p>Это поле задает код операции в ALU микропроцессора КМ1804BC1 для операций приемника и источника. Результат этой операции может приниматься в рабочий регистр (WR) или Q-регистр в зависимости от кода в поле приемника для порядка. Функции, помеченные звездочкой (*), дают результат, который может приниматься только в рабочий регистр (WR).</p> <table border="1"> <thead> <tr> <th>EXP</th> <th>Функция</th> <th>EXP</th> <th>Функция</th> </tr> </thead> <tbody> <tr> <td>CTL[3:0]</td> <td></td> <td>CTL[3:0]</td> <td></td> </tr> <tr> <td>0000</td> <td>0 или 0</td> <td>1000</td> <td>Q - 1</td> </tr> <tr> <td>0001</td> <td>B - A</td> <td>1001</td> <td>Q + 1</td> </tr> <tr> <td>0010</td> <td>A - B</td> <td>1010</td> <td>A</td> </tr> <tr> <td>0011</td> <td>B + A</td> <td>1011</td> <td>Q</td> </tr> <tr> <td>0100</td> <td>A или B</td> <td>1100</td> <td>0</td> </tr> <tr> <td>0101</td> <td>A и B</td> <td>1101</td> <td>B</td> </tr> <tr> <td>0110</td> <td>A - Q</td> <td>1110</td> <td>Q - A</td> </tr> <tr> <td>0111</td> <td>A + B +</td> <td>1111</td> <td>отсутствие операции</td> </tr> <tr> <td></td> <td>FRAC COUT</td> <td></td> <td></td> </tr> </tbody> </table>	EXP	Функция	EXP	Функция	CTL[3:0]		CTL[3:0]		0000	0 или 0	1000	Q - 1	0001	B - A	1001	Q + 1	0010	A - B	1010	A	0011	B + A	1011	Q	0100	A или B	1100	0	0101	A и B	1101	B	0110	A - Q	1110	Q - A	0111	A + B +	1111	отсутствие операции		FRAC COUT		
EXP	Функция	EXP	Функция																																											
CTL[3:0]		CTL[3:0]																																												
0000	0 или 0	1000	Q - 1																																											
0001	B - A	1001	Q + 1																																											
0010	A - B	1010	A																																											
0011	B + A	1011	Q																																											
0100	A или B	1100	0																																											
0101	A и B	1101	B																																											
0110	A - Q	1110	Q - A																																											
0111	A + B +	1111	отсутствие операции																																											
	FRAC COUT																																													
38:30	Управление путями данных мантиссы (FRAC CTL)	<p>Это поле полностью соответствует сигналам микропроцессора КМ1804BC1</p>																																												
29:26	Поле адреса "A" (A ADDR)	<p>Это поле адресует порт "A" рабочих регистров (WR) микропроцессора КМ1804BC1 для путей данных как порядка, так и мантиссы. Если поле синхронизации равно значению, которое разрешает принимать знак результата (SYGN OUT), то младшие три разряда поля адреса "A" управляют функцией для получения знака результата.</p>																																												

Продолжение табл. 10

CS	Функция	Описание
		A ADDR[2:0] Значение SIGN OUT 000 OP1 SIGN (знак OP1) 001 ZERO 010 Знак OP1 исключительное или знак OP2 011 Знак OP1 исключительное или знак результата 100 Знак OP2 101 Единица 110 Знак OP2 111 Единица
25:22	Поле адреса "B" (B ADDR)	Это поле адресует порт "B" рабочих регистров (WR) микропроцессоров KM18U4BC1 как для путей данных порядка, так и для путей данных мантиссы. Этот адрес является адресом записи.
21:20	Поле модификации (MOD)	Это поле используется, во-первых, для расширения других полей и, во-вторых, для выполнения специальных функций: 1) MOD[1:0]=00 Отсутствие операции 2) MOD[1:0]=01 Расширяет поле синхронизации 3) MOD[1:0]=10 включает MUL/DIV 4) MOD[1:0]=11 включает загрузку или запоминание При расширении поля синхронизации число функций, которое может быть выполнено этим полем удваивается. Включение MUL/DIV обеспечивает подключение некоторой условной логики для умножения и деления. Какая логика должна подключиться, определяет схема управления кодом операции. Включение загрузки или запоминания дает возможность загрузить или запомнить мантиссу и порядок. Что будет выполняться, загрузка или запоминание, определяется сигналом загрузки, который устанавливается кодом синхронизации. Кто должен загружаться или запоминаться, определяется полем сдвига.
19:18	Поле сдвига (SHF)	Это поле имеет много различных функций в зависимости от выполняемой операции. Загрузка: 1) SHF=00 Операнд одинарной точности с плавающей запятой. Загружаются SIGN EXP[7:0] FRACT[55:32] 2) SHF=01 Загрузка MOD. EXT[7:0] 3) SHF=10 Загрузка двойной точности

Продолжение табл. 10

СЗ	Функция	Описание
		операнда с плавающей запятой или загрузка целого.
		загружаются
		FRAC[31:16] или FRAC[55:00]
		В зависимости от того загружается целое или нет. Если загружается целое, младшие 16 разрядов должны маскироваться микрокомандой.
		4) SHF=11 Загрузка формата H: EXT[7:0]#FRAC[55:32]
		Запоминание:
		1) SHF=00 Запоминание первого слова SIGN#EXT[7:0] FRAC[55:32]
		2) SHF=01 Запоминание кода условия
		3) SHF=10 Запоминание второго слова FRAC[31:00]
		4) SHF=11 Запоминание операнда "H" EXT[7:0] FRAC[55:32]
		Сдвиги - поле сдвига также определяет, какие значения сдвигаются в различные регистры путей данных.
		Правый сдвиг - поле сдвига определяет, что сдвигается в MSB путей данных мантиссы.
		SHF[1:0] FRAC55 Q3 FRAC55 R3
		00 Порядок Q0 Порядок R0
		01 Расширение R0 FRAC COUT
		10 Ноль EXT00 R0 SAVE
		11 Расширение R0 ноль
		Когда поле синхронизации равно значению 011 (изменения при сдвиге мантиссы), поле сдвига работает следующим образом:
		SHF[1:0] FRAC55 Q3 FRAC55 R3
		00 Расширение R0 Порядок R0
		01 Единица Единица
		10 Ноль EXT00 R0 SAVE
		11 Ноль Ноль
		Левый сдвиг - при выполнении левого сдвига поле сдвига определяет, что сдвигается в секции мантиссы и порядка.
		SHF Порядок Мантисса
		[1:0] Q0 R0 Q0 R0
		00 FRAC55 Q3 FRAC55 Q3 Ноль Ноль
		01 Ноль Ноль Ноль FRAC55 R3 SV
		10 Единица Единица Единица Единица
		11 FRAC55 Q3 FRAC55 R3 QIN FRAC55 Q3
		Последняя комбинация используется для сдвига операнда в формате "H", при этом старшая часть его в QR, а млад-

Продолжение табл. 10

CS	Функция	Описание
		! шая часть - в FQ. Это дает возмож- ! ность сдвинуть весь операнд за один ! раз. После завершения операнд "Н" бу- ! дет находиться в FWR 55 - EXT 0 и ! FQL55:7]. Заметим, что QIN поступает ! в младший бит расширения в Q-регистре; ! он всегда равен нулю, если выполняет- ! ся команда не деления.
17:15	! Поле уп- ! равления ! синхрони- ! зацией ! (CLK CTL)	! Это поле может выполнить до 16 функ- ! ций при использовании совместно с ним ! поля модификации. Поле MOD не равно ! расширению поля синхронизации: ! 1) CLK CTL=000. Включает прием для ! OP1=0 и OP2=0. Разрешает установку ! триггеров OP1 EQ 0 и OP2 EQ 0. ! Это значение включает прием в два ! триггера (находящихся внутри ПМЛ), ! которые указывают, какой из опе- ! рандов равен нулю. Триггер OP2=0 ! загружается сигналом EXP=0, а ! триггер OP1=0 - сигналом OP2=0. ! 2) CLK CTL=001. Принимает R5 SAVE для ! формата "Н". ! Это значение поля сохраняет FRAC55 ! R5 в ПМЛ D43 до следующего появле- ! ния этого кода. Это необходимо для ! сохранения R5 при делении операн- ! дов в формате "Н". ! 3) CLK CTL=010. Ничего не делается. ! 4) CLK CTL=011. Включает изменения ! при сдвиге мантиссы. ! Это значение поля совместно с по- ! лем сдвига дает возможность ! сдвинуть единицу или ноль в MSB ! (самый старший разряд) мантиссы. ! 5) CLK CTL=100. Принимает знак ре- ! зультата. Это значение поля разре- ! шает прием в триггер знака. Какое ! значение знака будет принято при ! этом, определяется младшими тремя ! разрядами поля адреса "А". ! 6) CLK CTL=101. Принимает знак OP2. ! Это значение поля разрешает при- ! ем знака второго операнда. ! 7) CLK CTL=110. По этому коду поля ! осуществляется прием кодов усло- ! вия. Сдвигаемые разряды будут ус- ! танавливать биты V и C. Это дела- ! ется для ошибочных ситуаций. Норм- ! ально оба бита ожидаются. ! 8) CLK CTL=111. Принимает знак OP1. ! Это значение поля разрешает прием ! знака первого операнда.

3.050.001PЭ

Продолжение табл. 10

СЗ	Функция	Описание
		Значение поля MOD = расширению функций поля синхронизации, то есть MOD=01:
		1) CLK CTL=000. Изменение функции запоминания. Это значение поля инвертирует нормальное запоминание от запоминания плавающего формата к целому и наоборот. Это используется при выполнении команды EMOD.
		2) CLK CTL=001. Запускает быстрый цикл. Это значение поля осуществляет срабатывание триггера короткого цикла. Когда он установлен, время исполнения одной микрокоманды устанавливается равным 180 нс, когда он в нулевом состоянии, это время равно 270 нс и при этом FPA работает синхронно CPU.
		3) CLK CTL=010. Включить литерал. Это значение поля подключает 8-разрядный литерал на FPA BUS D [14:07]. Он может быть загружен в пути данных порядка и мантисы. При загрузке константы в пути данных мантисы константа загружается одновременно в EXT[6:0] и FRAC[30:23]. В большинстве случаев желательно загружать расширение константой, при этом другие части должны быть замаскированы.
		4) CLK CTL=011. Устанавливает триггер загрузки. Это значение поля устанавливает триггер загрузки. Когда поле MOD равно значению загрузки или запоминания, аппаратура прерывает их при загрузке. Этот сигнал очищается, когда в следующий раз будет установлен этот код. Сигнал загрузки устанавливается в ноль сигналом FORCE UADDR.
		5) CLK CTL=100. Принимает знак результата. Этот код поля разрешает прием в триггер знака результата.
		6) CLK CTL=101. Изменение входного переноса CIN. Этот код поля устанавливает следующий перенос мантисы (входной) равным значению выходного переноса мантисы в текущем состоянии. Это используется при сложении операндов.

Продолжение табл. 10

CS	Функция	Описание
		дво в формате "H".
		7) CLK CTL=110.
		Это значение поля устанавливает бит, который указывает логике умножения выбрать FRACT6 Q0 в качестве самого младшего разряда (LSB) множителя. Это используется при расширенном умножении. Этот сигнал устанавливается в ноль сигналом FORCE UADDR.
		8) CLK CTL=111. Расширенное ветвление.
		Этот код расширяет поле ветвления с двух до пяти разрядов (см. раздел, посвященный организации ветвлений в секвенсере).
14:10	Поле управления ветвления (VCTL)	Это поле определяет, какие сигналы состояния (сигналы изменения микроадреса) будут выбраны для модификации адреса микрокоманды. Эти сигналы, объединяясь по "ИЛИ" с соответствующими разрядами поля микроадреса UPF, совместно формируют адрес следующей микрокоманды (NUA). См. раздел 5.6.1.4, описывающий микросеквенсер.
9:0	Поле микроадреса (UPF)	Это поле определяет следующий микроадрес. Микроадрес, определяемый этим полем, может модифицироваться при использовании поля ветвления. Младшие восемь разрядов поля UPF используются также как поле литерала, когда последнее поле используется, адрес управляющей памяти берется из UPC (счетчик команд).

5.8.2. Обработка данных

Числа с плавающей запятой, которые процессор посылает FPA, обрабатываются логикой путей данных. Эти числа обрабатываются в соответствии с управляющими сигналами, поступающими из управляющей памяти. После получения результата обработки последний направляется в процессор. Как видно из рис. 37, 38 в приложении, логика путей данных состоит из секций порядка и мантиссы. Обе эти секции построены на 4-разрядных микропроцессорных секциях KM1804BC1.

5.8.3.

4-разрядная микропроцессорная секция KM1804BC1

4-разрядная микропроцессорная секция KM1804BC1 состоит из рабочих регистров (память RAM), Q-регистра, арифметико-логического устройства (ALU) и схем управления (рис.39 в приложении).

Рабочие регистры (WR) представляют собой местную память микропроцессора, где могут запоминаться как промежуточные, так и окончательные результаты арифметических и логических операций.

Арифметико-логическое устройство (ALU) является той частью путей данных, которая выполняет в процессоре FPA логические и арифметические операции в соответствии с управляющими сигналами, поступающими из управляющей памяти. На входы к ALU подключаются выходы с трехходового мультиплексора. На входы этого мультиплексора подсоединяются внешние входы данных, выходы порта "А" памяти RAM и входы установки нуля. На входы S ALU через мультиплексор можно подать выходы портов "А" и "В" памяти RAM, Q-регистр и установку нуля.

Выходные данные ALU (F) могут быть направлены в Q-регистр или память RAM, а также через двухходовой мультиплексор, на другой вход которого поступает выход порта "А" памяти RAM, - на выходные передатчики и затем на шину FPA. Дешифратор функции ALU определяет, какая логическая или арифметическая функция должна быть выполнена, а дешифратор приемника результата - куда должен быть направлен результат внутри микропроцессора или за его пределы на внешнюю шину.

Q-регистр загружается с выходов ALU и используется для формирования частного во время операции деления. Он также используется как регистр для временного запоминания данных. Выход Q-регистра может поступать на вход самого себя, а также сдвигаться вправо или влево.

5.8.4. Пути данных для обработки порядка

Пути данных порядка (рис.40 в приложении) используются для операций над порядком, подсчета циклов и проверки переполнения и потери порядка. Пути данных порядка состоят из 4-разрядных микропроцессоров, каждый из которых содержит 16 рабочих регистров (WR). Все 16 рабочих регистров адресуются через EXP A/B ADDR[3:0] от управля-

5.050.001PЭ

ющей памяти. Некоторые из рабочих регистров содержат константы, которые перечислены в табл.11.

Источник путей данных порядка, функция ALU и бит I(6) поля приемника порядка (I[6:8]) управляются путем декодирования EXP CODE[3:0](1)H от управляющей памяти. В силу этого не все функции (табл.12) КМ1804BC1 используются.

Таблица 11

Константы рабочих регистров порядка (в памяти RAM)		
Адрес (WR)	Константа	Использование
F	7FFF	Максимальный порядок в формате "H"
E	0400	Избыток порядка для формата "G"
D	07FF	Максимальный порядок в формате "G"
C	00FF	Максимальный порядок для форматов "F" и "D" (для чисел одинарной и двойной точности)
B	4000	Избыток порядка для формата "H"
A	0000	Нулевая константа
9	0001	Константа "единица"
5	18	Счетчик разрядов мантиссы

Таблица 12

Выбор функции, выполняемой над порядками	
EXP CODE [3:0](1)H	Выбираемая функция
0000	D ИЛИ U
0001	B - A
0010	A - B
0011	B + A
0100	A ИЛИ B
0101	A И B
0110	A - Q
0111	A + B + FRAC COUNT
1000	Q - 1

Продолжение табл. 12

EXP CODE !	Выбираемая функция
L3:01(1)H!	
1001 !	$Q + 1$
1010 !	A
1011 !	Q
1100 !	0
1101 !	Q
1110 !	$Q - A$
1111 !	Отсутствие операции

5.3.5. Пути данных для обработки мантиссы

Пути данных мантиссы содержат 16 микропроцессоров KM16043C1 и таким образом имеют длину в 64 разряда. Такая длина ориентирована на загрузку операндов в формате "H". Пути данных мантиссы состоят (рис.36,37 в приложении) из старшей части мантиссы [55:32], средней части [31:00] и части целого [47:16] плюс расширенной части EXP[7:0].

Пути данных мантиссы управляются I[8:0] и "A", "B" ADDR[3:0] и, поступающей из управляющей памяти. Биты I[8:0] выбирают функцию, которая должна быть выполнена с мантиссой, а "A", "B" ADDR[3:0]H управляет рабочими регистрами (памятью RAM). Младшая и средняя части мантиссы загружаются прямо с шины данных FPA. Часть старшей секции [55:48] загружается с данными, которые проходят через ИМЛ скрытого бита.

В семи из 16-ти 64-разрядных рабочих регистров (памяти RAM) путей данных мантиссы содержатся константы, перечисленные в табл.13.

Таблица 13

Константы рабочих регистров мантиссы

Адрес !	Константа	Использование
WR !		
E !	000000000000004000 !	Для округления числа в формате H
F !	00000000000000080 !	Для округления числа в формате D

Продолжение табл. 13

Адрес	Константа	Использование
G	00000000000000400	Для округления числа в формате G
C	00000800000000000	Для округления числа в формате F
B	000000000000000FF	Маска расширения
A	00000001FFFFFFFFFF	Маска расширения и средней части мантиссы
9	0000000000FFFF	Маска целого числа

Внутренняя 32-разрядная шина FPA (BUS FPA D[31:00]) не может за один раз загрузить мантиссу длиной в 64 разряда. Поэтому рабочие регистры путей данных мантиссы загружаются по частям. Когда рабочие регистры загружаются, управляющие поля настраиваются на выполнение функции.

$$WR(X) \leftarrow \text{D или 0}$$

Может быть установлена операция NOOP (отсутствие операции) путем установки 1(7) для младшей части мантиссы. Это преобразует функцию записи в рабочий регистр в NOOP. Микрокоманда из управляющей памяти определяет, какие части записываются с использованием полей модификации и сдвига (MOD и SHF).

5.8.6. Логика формирования знака

Каков знак результата, FPA сообщает процессору через BUS FPA D15 н. Логика определения знака состоит из микросхемы ПМЛ, которая синхронизируется данными из управляющей логики FPA.

Знаковая ПМЛ (рис.41 в приложении) принимает знаки первого и второго операнда, результирующий знак (SIGN OUT) и сигнал SUMPATH, который указывает, какая операция (суммирование или вычитание) должна быть выполнена при реализации команды ADD или SUB. Знаковая ПМЛ содержит триггер SIGN.OUT (результатирующий знак), который может загружаться:

- 1) знаком первого операнда (OP1);
- 2) знаком второго операнда (OP2);
- 3) результатом операции "исключающее ИЛИ" над знаками первого и второго операндов;

4) результатом операции "исключающее ИЛИ" (XOR) над знаком первого операнда и SIGN OUT;

5) единицей;

6) нулем.

Для большинства команд, выполняемых FPA, знаковые разряды первого и второго операндов загружаются в триггеры OP1 и OP2 (находящихся в микросхеме ПМЛ) во время процедуры загрузки операндов. В триггер SIGN OUT принимается выходной знак результата.

Когда FPA выполняет команду POLY, триггер OP1 в ПМЛ загружается знаком аргумента. После загрузки он остается без изменений во время выполнения команды. Триггер OP2 в ПМЛ загружается каждый раз знаком коэффициента. Триггер ПМЛ SIGN OUT получает текущий результирующий знак. На знаковую ПМЛ поступают входы POLY N и EXP A ADDR[2:0]H. Она генерирует выходы BUS FPA 015 H, SUMPATH(1)H, OP1,2 SIGN OUT(1)H. Сигнал POLY N поступает из логики ветвления FPA, а EXP A ADDR[2:0]H генерируется управляющей памятью. BUS FPA 015 H поступает в процессор, а другие выходы (SUMPATH(1)H, OP1,2 SIGN OUT(1)H) используются логикой ветвления FPA. функция знаковой ПМЛ SIGN OUT управляется выходом из управляющей памяти EXP A ADDR[2:0]H. Эти функции в зависимости от кодирования этого поля приведены в табл.14.

5.9. Средства наладки FPA

Процессор операций над числами с плавающей запятой содержит схемы, которые позволяют записать со стороны процессора любой адрес микрокоманды. Это делается с помощью сигналов TRAP ACC L или READ ACC UPC L, логики записи/чтения микроадреса, которая состоит из управления записью/чтением, приемопередатчиков на шину и их схем включения.

Таблица 14

Кодирование работы ПМЛ формирования знака в зависимости от управляющего им кода

EXP A ADDR[2:0]H (3-ная с/с)	! Сигнал на выходе PAL SIGN OUT EXP
0	! OP1 SIGN
1	! ZERO
2	! OP1 SIGN "исключающее ИЛИ" OP2 SIGN
3	! OP1 SIGN "исключающее ИЛИ" SIGN OUT

Продолжение табл. 14

EXP A ADDR12:0JH ! Сигнал на выходе PAL SIGN OUT EXP
(8-ная с/с) !

4	! OP2 SIGN
	!
5	! Единица
	!
6	! OP2 SIGN
	!
7	! Единица

5.9.1. Установка микроадреса

Когда процессор выдает сигнал TRAP ACC L, логика записи/чтения микроадреса (рис.42 в приложении) генерирует FORCE UADDR (1)H. Этот сигнал используется для запрещения выхода микросеквенсера. Микроадрес, который хочет записать процессор, устанавливается на BUS Y D019:0JH. Затем выход логики записи/чтения микроадреса FPA BUS NUA19:0JH используется в качестве адреса следующей микрокоманды вместо запрещенного выхода микросеквенсера BUS NUA19:0JH.

5.9.2. Чтение микроадреса

В процессе микродиагностики логика чтения микроадреса используется для чтения выхода микросеквенсера BUS NUA19:0JH на BUS Y для последующей передачи его в процессор. Во время операции чтения микроадреса процессор устанавливает сигнал READ ACC UPC L. Он запрещает выдачу синхроимпульсов FPA. Он также коммутирует выход микросеквенсера BUS NUA19:0JH на шину FPA через шинные передатчики логики записи/чтения микроадреса. В следующий момент процессор генерирует RCV DATA L, и выход BUS NUA19:0JH поступает на BUS Y D019:0JH. Сигнал RCV DATA L также вновь разрешает выдачу синхроимпульсов FPA.

5.9.3. Логика формирования паритета

на паритет проверяется каждое 43-разрядное микрослово (микрокоманда), которое считывается из управляющей памяти. Микрослово содержит только два бита паритета, каждое из которых предназначено для определенных разрядов микрокоманды. Рис.43 в приложении показывает какие

поля проверяются (контролируются) разрядами паритета. Логика паритета состоит из трех схем контроля по паритету, памяти PRDM и ПМЛ управления паритетом. Сумма паритетного бита и разрядов полей, которые данный бит контролирует, должна быть четной.

Когда обнаруживается ошибка паритета, логика паритета генерирует сигнал FORCE LOW UADDR L, который устанавливает на выходах микросеквенсора NUAL9:0J логические нули. В результате запускается процедура обработки ошибки паритета, которая просто циклитса в микрокоманде с нулевым адресом, постоянно запоминая ошибку паритета. Процессор первоначально интерпретирует это, как особый случай и запрашивает код ошибки. FPA затем передает код ошибки. FPA снова передает ошибку паритета, которую процессор интерпретирует как ошибку паритет. Из подпрограммы обработки ошибки FPA выходит с помощью процессора.

Выходами ПМЛ управления паритетом являются BUS FPA DL3:0J H и FORCE LOW UADDR L. Первый из них представляет собой 4-разрядное поле, и при ошибке паритета в разряде поля BUS FPA 000 устанавливается логическая единица. Она информирует процессор, что произошла ошибка паритета.

Разряды ошибки, которые были установлены ПМЛ управления паритетом, будут оставаться на выходных линиях BUS FPA DL3:0J H до тех пор, пока они не очистятся сигналом FORCE UADDR(1)H. Они передаются на шину BUS FPA при сигнале READ UADDR(1)H.

6. ПОДГОТОВКА К РАБОТЕ

6.1. Порядок установки

Помещение, в котором устанавливается и будет эксплуатироваться процессор, должно быть отапливаемым в зимнее время. Перед установкой ФРА необходимо его распаковать и провести расконсервацию. Распаковка в зимнее время должна проводиться в отапливаемом помещении, с предварительной выдержкой в нераспакованном виде в нормальных климатических условиях (температура воздуха от 15 до 25 градусов С, относительная влажность воздуха от 30 до 80%) в течении 12 ч. Должна быть проведена проверка комплектности эксплуатационных документов и микродиагностических тестов процессора. Следует провести внешний осмотр, проконтролировать на отсутствие механических повреждений и дефектов, коррозии, царапин, отсутствие посторонних предметов и повреждений на плате, удалить пыль.

6.2. Первоначальное включение

Подача напряжения 5V осуществляется при включении комплекса ВК СМ 1700. Проверка уровня электропитания осуществляется в комплексе ВК СМ 1700.

7. ИЗМЕРЕНИЕ ПАРАМЕТРОВ, РЕГУЛИРОВАНИЕ И НАСТРОЙКА

Перед установкой платы FPA необходимо убедиться в отсутствии короткого замыкания между шинами питания +5V и 0V. Измерение рекомендуется производить омметром Ц 4353 (класс точности 1,5), установленным в режим измерения сопротивления на диапазоне - "Омы". Возможно применение других измерительных приборов аналогичного класса. При отсутствии короткого замыкания между шинами питания, установить плату FPA.

Дальнейшую проверку процессора следует производить согласно разделу 9 "Характерные неисправности, методы их обнаружения".

8. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Техническое обслуживание процессора плавающей запятой осуществляется персоналом, имеющим право на техническое обслуживание ЭВМ, в которую входит процессор.

Техническое обслуживание ФРА должно производиться в помещениях, гарантирующих защиту от воздействия пыли, агрессивной среды и электромагнитных полей.

Для обеспечения бесперебойной работы ФРА необходимо проводить ежеквартальные контрольно-профилактические работы. При проведении этих работ ФРА исключается из состава ЭВМ, в которой он находится.

Продолжительность проведения профилактического обслуживания зависит от состояния ФРА, но не превышает 8 ч.

При проведении ежеквартальных работ и после устранения неисправностей необходимо провести проверку работоспособности ФРА с помощью однократного прогона микродиагностических тестов.

В ежеквартальные контрольно-профилактические работы входят:

внешний осмотр;

удаление пыли и грязи;

чистка контактов блоков элементов;

проверка монтажных соединений.

Для проведения различных видов технического обслуживания требуется материалы:

отходы хлопчатобумажных материалов по ГОСТ 4644-75;

спирт этиловый ректификованный технический высшего сорта по ГОСТ 18300-72;

марля медицинская по ГОСТ 9412-77.

На поверхности ФРА не должно быть царапин, вмятин и других повреждений.

Удаление пыли и грязи выполняется чистой сухой хлопчатобумажной тканью.

Чистка контактов производится медицинской марлей, смоченной спиртом.

Проверка монтажных соединений производится визуально. Проверяется состояние монтажных соединений и паяк, при этом не должно быть нарушений изоляции, пайки должны быть покрыты лаком.

9. ХАРАКТЕРНЫЕ НЕИСПРАВНОСТИ, МЕТОДЫ ИХ ОБНАРУЖЕНИЯ

Возможными неисправностями FPA являются:

выход из строя интегральных микросхем;

обрыв линии связи.

Методом обнаружения характерных неисправностей является прогон микродиагностических тестов FPA, написанных в микрокодах центрального процессора. Эти микродиагностические тесты запускаются под управлением микромонитора, который во время выполнения теста находится в оперативной памяти консольного процессора.

Загрузка микродиагностических тестов FPA выполняется следующим образом:

поставить кассету с системой микродиагностических тестов в механизм консольного ввода;

повернуть ключ включения машины в положение "МЕСТН" и ждать появления идентификатора системы микродиагностического обеспечения MIC> на консольном терминале. Появление MIC> говорит о том, что программа микромонитора находится в оперативной памяти консольного процессора;

Набрать на консольном терминале следующие командные строки:

MIC>SE TR <ВК> - установка флага трассы для наблюдения за выполнением микротестов;

MIC>DI BO FPA <ВК> - запуск микротестов, проверяющих процессор.

При прогоне микротестов FPA, в случае обнаружения ошибки на консольном терминале появится сообщение:

SECT TST ERR EXP REC OTHER MASK MODULE

SECT - имя микротеста;
TST - номер микротеста, где ошибка;
ERR - номер ошибки;
EXP - ожидаемые данные;
REC - полученные данные;
OTHER - некоторая дополнительная информация;
MASK - маска ошибок. Биты, установленные в 1 в маске, соответствуют битам результата, которые не проверяются;
MODULE - предполагаемый источник ошибки.

Подробно предполагаемые источники ошибок приведены в 00076-01-13-01 "Система микродиагностического обеспечения ВК СМ 1700. Описание программы".

10. ТРАНСПОРТИРОВАНИЕ И ХРАНЕНИЕ

Транспортирование ФРА производить в упакованном виде в транспортной таре всеми видами транспорта на любые расстояния при условии воздействия следующих климатических факторов:

температура окружающего воздуха от минус 50 до +50 градусов С;

относительная влажность воздуха до 95% при температуре 30 градусов С;

атмосферное давление от 84 до 107 кПа;

транспортная тряска с ускорением 3 Г при частоте ударов от 80 до 120 в минуту.

ФРА должен храниться в складских помещениях в упаковке при температуре от 15 до 25 градусов С и относительной влажности воздуха не более 85% на стеллажах.

