

## **ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2**

### **2.1 ОБЗОР**

В этой главе описываются архитектура и функции трех вычислительных устройств: арифметико-логического устройства (АЛУ), умножителя-накопителя и устройства циклического сдвига.

Каждый процессор семейства ADSP-2100 является 16-разрядным процессором с фиксированной точкой. Большинство операций предполагает представление чисел в дополнительном коде, в остальных случаях предполагается работа с беззнаковыми числами или с простыми последовательностями двоичных символов. Предусмотрена поддержка арифметических операций над несколькими словами и операции с блочной плавающей точкой. Более подробная информация о форматах представления чисел, поддерживаемых процессорами семейства ADSP-2100, дается в Приложении С.

Знаковые числа в процессорах семейства ADSP-2100 всегда представлены в дополнительном коде. Двоично-десятичный формат, числа со знаком, обратный дополнительный код и коды с избытком  $n$  не используются.

#### **2.1.1 Последовательности двоичных символов**

Это самое простое двоичное представление; 16 бит здесь представлены как набор двоичных разрядов. Примерами вычислений с использованием этого формата являются логические операции: NOT, AND, OR, XOR. В этих операциях операнды используются как последовательности двоичных символов, а знаковый бит и положение двоичной точки не предусмотрены.

#### **2.1.2 Беззнаковый формат**

Беззнаковые двоичные числа могут быть представлены как положительные числа, в два раза превышающие по модулю знаковые числа той же длины. Наименее значимые слова чисел с повышенной точностью считаются беззнаковыми числами.

#### **2.1.3 Знаковые числа в дополнительном коде**

Когда мы говорим о семействе ADSP-2100, «знаковый» относится к представлению в двоичном коде. Большинство операций процессоров семейства поддерживают или предполагают арифметические операции над числами, представленными в дополнительном коде. В семействе ADSP-2100 не используется

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

двоично-десятичный формат, числа со знаком, обратный дополнительный код и коды с избытком  $n$ .

### 2.1.4 Представление дробных чисел в формате 1.15

Арифметические операции процессоров семейства ADSP-2100 оптимизированы для операций над числовыми значениями в дробном двоичном формате, обозначенном как 1.15 (одна точка пятнадцать). В этом формате имеется один знаковый бит (самый старший бит) и пятнадцать бит, отведенных под дробную часть и представляющих значения от  $-1$  до  $+1$  минус один младший бит.

На рис. 2.1 показаны веса битов для чисел в формате 1.15. Ниже даются примеры чисел в формате 1.15 и их десятичные эквиваленты.

Число в формате 1.15	Его десятичный эквивалент
0x0001	0.000031
0x7FFF	0.999969
0xFFFF	-0.000031
0x8000	-1.000000

$2^{-0}$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$	$2^{-12}$	$2^{-13}$	$2^{-14}$	$2^{-15}$
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------

Рис. 2.1 Вес бит для чисел в формате 1.15

### 2.1.5 Арифметические операции АЛУ

Во всех операциях АЛУ, за исключением примитивов знакового деления (DIVS), операнды и результаты представляются как простые двоичные последовательности из 16 бит. Различные биты (разряды) состояния представляют результаты в знаковом формате: код переполнения (AV) и флаг отрицание (AN).

Логика бита переполнения (AV) базируется на арифметических операциях с представлением чисел в дополнительном коде. Он устанавливается в тех случаях, когда самый старший бит изменяется непредусмотренным знаком операндов и сущностью операции образом. Например, при сложении двух положительных чисел должен получиться положительный результат; изменение знакового бита означает переполнение, и устанавливается AV. Сложение отрицательного и положительного числа может иметь и отрицательный, и положительный результат, но не может привести к переполнению.

Логика бита переноса (AC) базируется на арифметических операциях с беззнаковыми числами. Он устанавливается в тех случаях, когда перенос генерируется битом 16 (самым старшим битом). Бит AC более полезен для младших бит слова при операции с несколькими словами.

### 2.1.6 Арифметические операции умножителя

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

Результаты, полученные в умножителе, представляют собой последовательности двоичных символов. Входные значения интерпретируются согласно информации, представленной в самой команде (знаковое число - во сколько раз - знаковое, беззнаковое - во сколько раз - беззнаковое, смешанное умножение, округление). Предполагается, что 32-разрядный результат умножителя представлен как знаковый, так как он далее расширяется по знаку в наборе регистров MR разрядностью 40 бит.

Процессоры семейства ADSP-2100 поддерживают два режима преобразования формата: дробный режим для дробных операндов, формат 1.15 (1 знаковый бит и 15 бит для представления дробной части), и целочисленный режим для целочисленных операндов, формат 16.0.

При перемножении процессором двух операндов в формате 1.15 получается число в формате 2.30 (2 знаковых бита и 30 бит для представления дробной части). В режиме умножения дробных чисел умножитель автоматически сдвигает полученный им результат P влево перед передачей его в регистр результатов умножителя (MR). Благодаря этому сдвигу результат умножителя получается в формате 1.31, который можно округлить до формата 1.15. Эта операция показана на рис. 2.7 в разделе, посвященном умножителю.

В режиме целочисленного умножения левый сдвиг не происходит. Например, если операнды представлены в формате 16.0, 32-разрядный результат умножителя должен быть в формате 32.0. Нет никакой необходимости в сдвиге влево; он только изменит числовое представление. Данное положение иллюстрируется на рис. 2.8 в разделе, посвященном умножителю.

### 2.1.7 Арифметические операции устройства сдвига

Многие операции в устройстве сдвига явным образом расписаны для знаковых (представленных в дополнительном коде) или беззнаковых величин: логический сдвиг предполагает представление величин без знака или значений последовательностей двоичных символов, а арифметический сдвиг предполагает представление чисел в дополнительном коде.

Логика порядка числа допускает использование чисел, представленных в дополнительном коде. Эта логика поддерживает операции с блочной плавающей точкой, которые также базируются на представлении дробной части в дополнительном коде.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

### 2.1.8 Заключение

В таблице 2.1. представлены некоторые характеристики арифметических операций процессоров семейства ADSP-2100. Кроме типов представления чисел, описываемых в этом разделе, компилятор C процессоров семейства поддерживает 32-разрядный формат для чисел с плавающей точкой, при котором одно 16-разрядное слово является порядком числа, а другое - мантиссой (обратитесь к руководству «ADSP-2100 Family C Tools Manual»).

Таблица 2.1

Форматы чисел для арифметических операций		
ОПЕРАЦИЯ	ФОРМАТЫ	
<i>АЛУ</i>	<i>Операнды</i>	<i>Результат</i>
сложение	знаковые или беззнаковые	выдаются флаги
вычитание	знаковые или беззнаковые	выдаются флаги
логические операции	цепочка двоичных символов	в том же формате, что и операнды
деление	явно знаковые/беззнаковые	в том же формате, что и операнды
переполнение АЛУ	знаковые	в том же формате, что и операнды
бит переноса АЛУ	16-разрядные беззнаковые	в том же формате, что и операнды
насыщение АЛУ	знаковые	в том же формате, что и операнды
<i>Умножитель, дробный режим</i>		
умножение (P)	1.15 явно знаковые/беззнаковые	32 бита (2.30)
умножение (MR)	1.15 явно знаковые/беззнаковые	2.30 после сдвига 1.31
умножение/сложение	1.15 явно знаковые/беззнаковые	2.30, после сдвига 1.31
умножение/вычитание	1.15 явно знаковые/беззнаковые	2.30, после сдвига 1.31
насыщение умножителя	знаковые	в том же формате, что и операнды

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

табл. 2.1 (продолжение)

*Умножитель, целочисленный режим*

умножение (P)	1.15 явно знаковые/беззнаковые	32 бита (2.30)
умножение (MR)	16.0 явно знаковые/беззнаковые	32.0 нет сдвига
умножение/сложение	16.0 явно знаковые/беззнаковые	32.0 нет сдвига
умножение/вычитание	16.0 явно знаковые/беззнаковые	32.0 нет сдвига
насыщение умножителя	Знаковые	в том же формате, что и операнды

*Устройство сдвига*

логический сдвиг	беззнаковые/цепочка двоичных символов	в том же формате, что и операнды
арифметический сдвиг	Знаковые	в том же формате, что и операнды
нахождение поряд-ка числа	Знаковые	в том же формате, что и операнды

### 2.2 АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ)

Арифметико-логическое устройство (АЛУ) обеспечивает выполнение стандартного набора арифметических и логических операций. К первым относятся сложение, вычитание, инвертирование (отрицание), инкремент, декремент и нахождение абсолютного значения. Эти операции дополнены двумя примитивами деления, при помощи которых операция деления может выполняться за несколько циклов. К логическим операциям относятся логическое И (AND), ИЛИ (OR), исключающее ИЛИ (XOR) и НЕ (NOT).

#### 2.2.1 Блок-схема АЛУ

Блок-схема АЛУ представлена на рис. 2.2.

АЛУ разрядностью 16 бит имеет два 16-битовых порта ввода - X и Y, и один порт вывода - R. АЛУ принимает сигнал переноса (CI), который является битом переноса из арифметического регистра состояния (ASTAT). В АЛУ генерируется 6 сигналов состояния: состояние-ноль (AZ), отрицания (AN), переноса (AC), переполнения (AV), ввода знака через порт X (AS) и состояние частного (AQ). Все сигналы арифметического состояния фиксируются в арифметическом регистре состояний (ASTAT) в конце цикла. За информацией о том, как каждая команда влияет на флаги АЛУ, обращайтесь к главе «Набор команд» руководства пользователя «ADSP-2100 Assembler Manual».

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

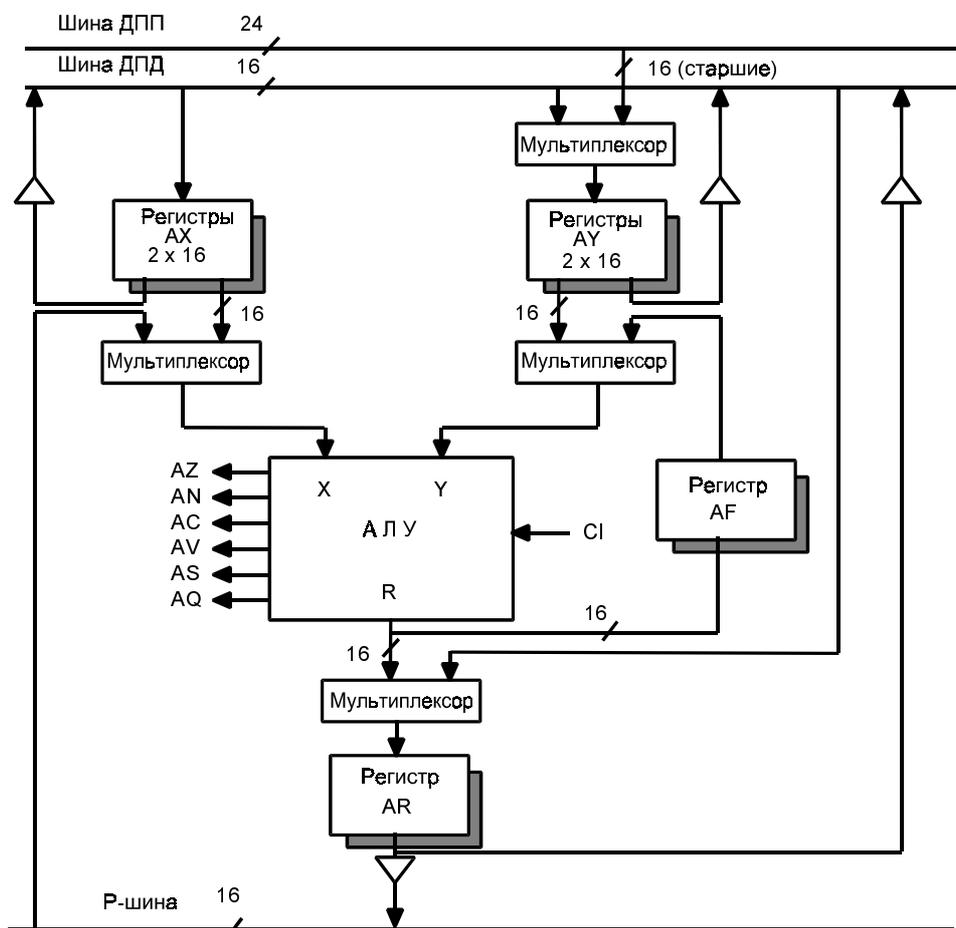
Порт ввода X АЛУ может принимать данные из двух источников: регистров AX или шины результатов (R-шина). R-шина соединяет регистры вывода всех вычислительных устройств, позволяя использовать их как операнды ввода. Имеется два регистра AX: AX0 и AX1. Считывание из этих регистров и запись в них может производиться с шины ДПД. В наборе команд предусматривается также считывание этих регистров на шину ДПП, но в данном случае нет прямого соединения; в этой операции используется устройство обмена данными между шинами ДПП и ДПД. Два выхода регистров AX позволяют осуществлять ввод с одного регистра в АЛУ, в то время как другой передает данные на шину ДПД.

Порт ввода Y АЛУ также может принимать данные из двух источников: регистров AY и регистра обратной связи AF АЛУ. Имеется два регистра AY: AY0 и AY1. Считывание из этих регистров и запись в них производится с шины ДПД, запись может также производиться из шины ДПП. В наборе команд предусматривается считывание из этих регистров на шину ДПП, но в данном случае нет прямого соединения; в этой операции используется устройство обмена данными между шинами ДПП и ДПД. Два выхода регистров AY могут обеспечить ввод данных в АЛУ одновременно с передачей данных на шину ДПД.

Выходные данные АЛУ загружаются либо в регистр обратной связи АЛУ (AF), либо в регистр результатов АЛУ (AR). Регистр AF является внутренним регистром АЛУ и позволяет использовать результат АЛУ прямо в качестве входного значения для порта Y АЛУ. Регистр AR может передавать данные как на шину ДПД, так и на шину результатов. Этот регистр может прямо загружаться с шины ДПД. В наборе команд предусматривается также считывание с AR на шину ДПП, но в данном случае нет прямого соединения; для этой операции используется устройство обмена данными между шинами ДПП и ДПД.

В любой из регистров, связанных с АЛУ, может производиться запись/ считывание в течении одного и того же цикла. Считывание с регистров производится в начале цикла, а запись в регистры - в конце цикла. Следовательно, при считывании из регистра считывается значение, загруженное в этот регистр в конце предшествовавшего цикла. Новое значение, записанное в регистр, не может быть считано до начала последующего цикла. Это позволяет регистрам ввода АЛУ обеспечивать АЛУ операндом в начале цикла и обновляться следующим операндом из памяти в конце того же цикла. Это также позволяет сохранять содержимое регистра результатов в памяти и обновлять этот регистр новым значением результата в течении одного и того же цикла. Пример работы считывания/записи в течение одного и того же цикла можно найти в главе 15 «Набор команд» в разделе «Многофункциональные команды».

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2



**Рис. 2.2** Блок-схема АЛУ

В АЛУ содержится двойной ряд регистров, которые показаны на рис. 2.2 за основными регистрами. В действительности имеется два набора регистров AR, AF, AX и AY. В любой определенный момент времени доступным является только один ряд регистров. Дополнительный ряд регистров может быть активизирован для чрезвычайно быстрого контекстного переключения (например, во время подпрограммы обслуживания прерывания). В таком случае новая задача (т.е. обслуживание прерывания в примере) может быть выполнена без передачи текущих состояний в ЗУ.

Выбор основного или альтернативного ряда регистров управляется битом 0 в регистре состояния режима процессора (MSTAT). Если этот бит принимает

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

значение 0, выбирается основной ряд; если он равен 1, выбирается теневой ряд регистров.

### 2.2.2 Стандартные функции

Список стандартных функций АЛУ

$R = X + Y$	Сложить операнды X и Y
$R = X + Y + CI$	Сложить операнды X и Y и бит переноса
$R = X - Y$	Вычесть операнд Y из X
$R = X - Y + CI - 1$	Вычесть операнд Y из X с «заемом» (единицы старшего разряда при вычитании)
$R = Y - X$	Вычесть X из операнда Y
$R = Y - X + CI - 1$	Вычесть X из операнда Y с «заемом»
$R = -X$	Инвертирование операнда X (в дополнительном коде)
$R = -Y$	Инвертирование операнда Y (в дополнительном коде)
$R = Y + 1$	Инкрементирование операнда Y
$R = Y - 1$	Декрементирование операнда Y
$R = PASS X$	Пропустить операнд X для сохранения результата неизменным
$R = PASS Y$	Пропустить операнд Y для сохранения результата неизменным
$R = 0 (PASS 0)$	Обнуление результата
$R = ABS X$	Найти абсолютное значение операнда X
$R = X AND Y$	Логическое И над операндами X и Y.
$R = X OR Y$	Операция логического ИЛИ над операндами X и Y
$R = X XOR Y$	Операция исключающего ИЛИ над операндами X и Y
$R = NOT X$	Операция логического НЕ над операндом X (в дополнительном коде)
$R = NOT Y$	Операция логического НЕ над операндом Y (в дополнительном коде)

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

### 2.2.3 Регистры ввода/вывода АЛУ

Ниже показаны источники ввода данных в регистры ввода и вывода АЛУ.

<i>Источники ввода данных в порт X</i>	<i>Источники ввода данных в порт Y</i>	<i>Направление вывода дан- ных через порт R</i>
AX0, AX1	AY0, AY1	AR
AR	AF	AF
MR0, MR1, MR2		
SR0, SR1		

MR0, MR1 и MR2 являются регистрами результатов умножителя/ накопителя; SR0 и SR1 - регистры результатов устройства сдвига.

### 2.2.4 Возможность операций с повышенной точностью

Операции с многократно увеличенной точностью поддерживаются в АЛУ сигналом переноса и битом состояния переноса АЛУ (AC). Сигнал переноса является битом состояния AC, который был получен при выполнении предыдущей операции АЛУ. Операция «сложение с переносом» (+C) предназначена для сложения старших разрядов чисел с повышенной точностью. Операция «Вычитание с «заемом» (C - 1 - является «заемом») предназначена для вычитания старших разрядов чисел с многократно увеличенной точностью.

### 2.2.5 Режим насыщения АЛУ

В регистре AR имеется режим насыщения для операций в дополнительном коде, который автоматически устанавливает значение этого регистра максимально отрицательным или положительным, в тех случаях когда в АЛУ получается результат переполнения или потеря значащих разрядов. Этот режим работы АЛУ разрешается установкой бита 3 регистра состояния режима (MSTAT). В случае такого разрешения значение, загружаемое в регистр AR во время операции АЛУ, зависит от состояния переполнения и переноса, генерируемых АЛУ в этом цикле. В таблице на следующей странице представлена информация о загрузке AR при работе в режиме насыщения.

Режим насыщения в умножителе-накопителе, в отличие от операций в режиме насыщения АЛУ, действителен только для одной команды. Разрешение режима насыщения АЛУ, напротив, означает, что все последующие операции выполняются этим способом.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Таблица 2.2

Режим насыщения		
Переполнение (AV)	Перенос (AC)	Содержимое AR
0	0	Выходные данные АЛУ
0	1	Выходные данные АЛУ
1	0	0111111111111111 <i>полномасштабная положительная величина</i>
1	1	1000000000000000 <i>полномасштабная отрицательная величина</i>

В режиме насыщения АЛУ преобразования происходят только в регистре AR; если данные выводятся через регистр AF, происходит возврат по циклу, а насыщение результата отражается флагами состояний.

### 2.2.6 Режим фиксации переполнения АЛУ

Режим фиксации переполнения АЛУ разрешается установкой бита 2 в регистре состояния режима (MSTAT) и приводит к фиксации бита AV, после того как он был установлен. При переполнении АЛУ в этом режиме бит AV устанавливается и сохраняется даже тогда, когда в последующих операциях АЛУ переполнение не генерируется. В этом режиме сброс AV возможен только при записи его нулем прямо с шины ДПД.

### 2.2.7 Деление

АЛУ поддерживает операцию деления. Эта функция достигается за счет добавления цепи сдвига, не показанной на блок-схеме рис. 2.2. Деление осуществляется с помощью двух специальных примитивов деления, которые используются для выполнения невозстанавливаемого условного алгоритма деления сложением-вычитанием. Деление может быть либо знаковым, либо беззнаковым; однако, делимое и делитель должны быть при этом в одном и том же виде. В Приложении В детально описываются различные исключения из нормальной операции деления, описываемой в данном разделе.

Деление с одинарной точностью с 32-разрядным делимым (числителем) и 16-разрядным делителем (знаменателем), в результате которого получается 16-разрядное частное, выполняется в течении 16 циклов. При этом также могут быть подсчитаны частные с более высокой и низкой точностью. Делитель может

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

храниться в регистрах AX0, AX1 или любом из регистров R. Старшие разряды знакового делимого могут начинаться либо в AY1, либо AF. Старшие разряды беззнакового делимого должны находиться в AF. Младшие разряды делимого, независимо от его вида, должны находиться в регистре AY0. В конце операции деления частное будет находиться в AY0.

Первый из двух примитивов деления «деление-знак» (DIVS) выполняется в начале деления при делении знаковых чисел. В этой команде в ходе выполнения операции логического ИЛИ над знаковыми битами делимого и делителя вычисляется знаковый бит частного. Содержимое регистра AY0 сдвигается на одну позицию, таким образом, что полученный знаковый бит становится самым младшим битом. Вычисленный знаковый бит также загружается в бит AQ арифметического регистра состояния. Самый старший бит регистра AY0 сдвигается в положение самого младшего бита регистра AF, а старшие 15 бит AF загружаются младшими 15 битами R из АЛУ, которые не поступая на вход Y идут прямо на выход R. В результате достигается левый сдвиг пары регистров AF - AY0 и перемещение знакового бита частного в положение самого младшего бита. Операция DIVS показана на рис. 2.3.

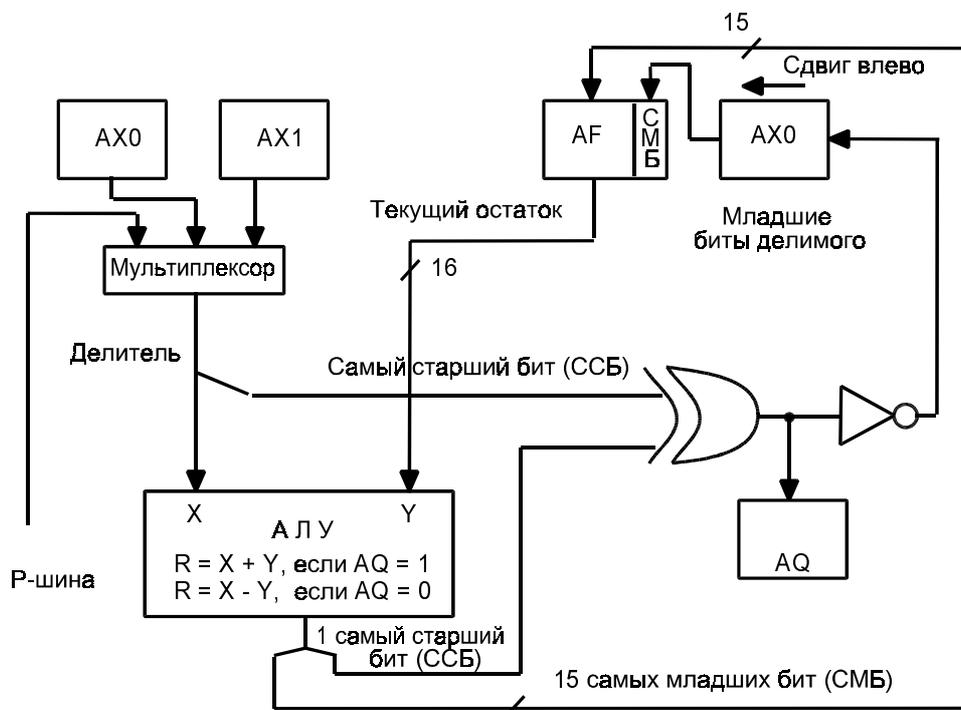
При делении беззнаковых чисел операция DIVS не используется. В данном случае бит AQ в арифметическом регистре состояния (ASTAT) должен быть установлен в 0 путем сброса, осуществляемого вручную. Бит AQ указывает, что для последующих операций предполагается положительное частное.

Вторым примитивом деления является команда «деление-частное» (DIVQ), которая генерирует один бит частного и выполняется повторно для вычисления остальных бит частного. Для беззнакового деления с одинарной точностью команда DIVQ выполняется 16 раз для получения 16-разрядного частного.

Для знакового деления с одинарной точностью эта команда выполняется 15 раз, после того как знаковый бит был вычислен в ходе операции DIVS. Команда DIVQ сдвигает содержимое регистра AY0 на один бит влево, таким образом что новый бит частного может быть помещен в положение самого младшего бита. Состояние бита AQ, сгенерированного в ходе выполнения предыдущей операции, определяет операцию АЛУ по вычислению текущего остатка. Если AQ = 1, АЛУ прибавляет делитель к текущему остатку в AF. Если AQ=0, АЛУ вычитает делитель из текущего остатка в AF. Результат с выхода R загружается в AF, так же как и в предыдущей операции DIVS. Бит AQ вычисляется при помощи операции исключающего ИЛИ над самым старшим битом делимого и самым старшим битом выходного значения АЛУ, а бит частного получается в результате инвертирования этого значения. Бит частного загружается в самый младший бит



## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2



**Рис. 2.4** Операции DIVQ

замечания и примеры можно найти в первом томе издания «*Digital Signal Processing Applications Using the ADSP-2100 Family.*»

Делимое           BBBBB.BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB  
                       NL бит           NR бит

Делитель            BB.BBBBBBBBBBBBBB  
                       DL бит           DR бит

Частное             BBBB.BBBBBBBBBBBB  
                       (NL - DL + 1) бит       (NR - DR - 1) бит

**Рис. 2.5** Формат частного

Переполнение происходит, когда результат не может быть представлен в формате, вычисленном для частного согласно приведенной выше схеме, или когда делитель равен нулю или меньше делимого по абсолютному значению.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

### 2.2.8 Состояния АЛУ

Ниже описываются биты состояния АЛУ в регистре АСТАТ. Полная информация о регистре АСТАТ, мнемонике специальных битов и их позициях дается в главе «Управление программой».

<i>Флаг</i>	<i>Имя</i>	<i>Описание</i>
AZ	Ноль	Логическое ИЛИ-НЕ всех бит в регистре результатов АЛУ. Истинно, если выходное значение АЛУ равно нулю.
AN	Отрицание	Знаковый бит результата АЛУ. Истинно, если выходное значение АЛУ отрицательно.
AV	Переполнение	Исключающее ИЛИ выходных значений переноса двух самых старших слагаемых. Истинно, если АЛУ переполнено.
AC	Перенос	Выходное значение переноса от самого старшего слагаемого.
AS	Знак	Знаковый бит порта ввода X АЛУ. Имеет значение только для команды ABS.
AQ	Частное	Бит частного генерируется только командами DIVS и DIVQ.

### 2.3 УМНОЖИТЕЛЬ-НАКОПИТЕЛЬ (УМНОЖИТЕЛЬ)

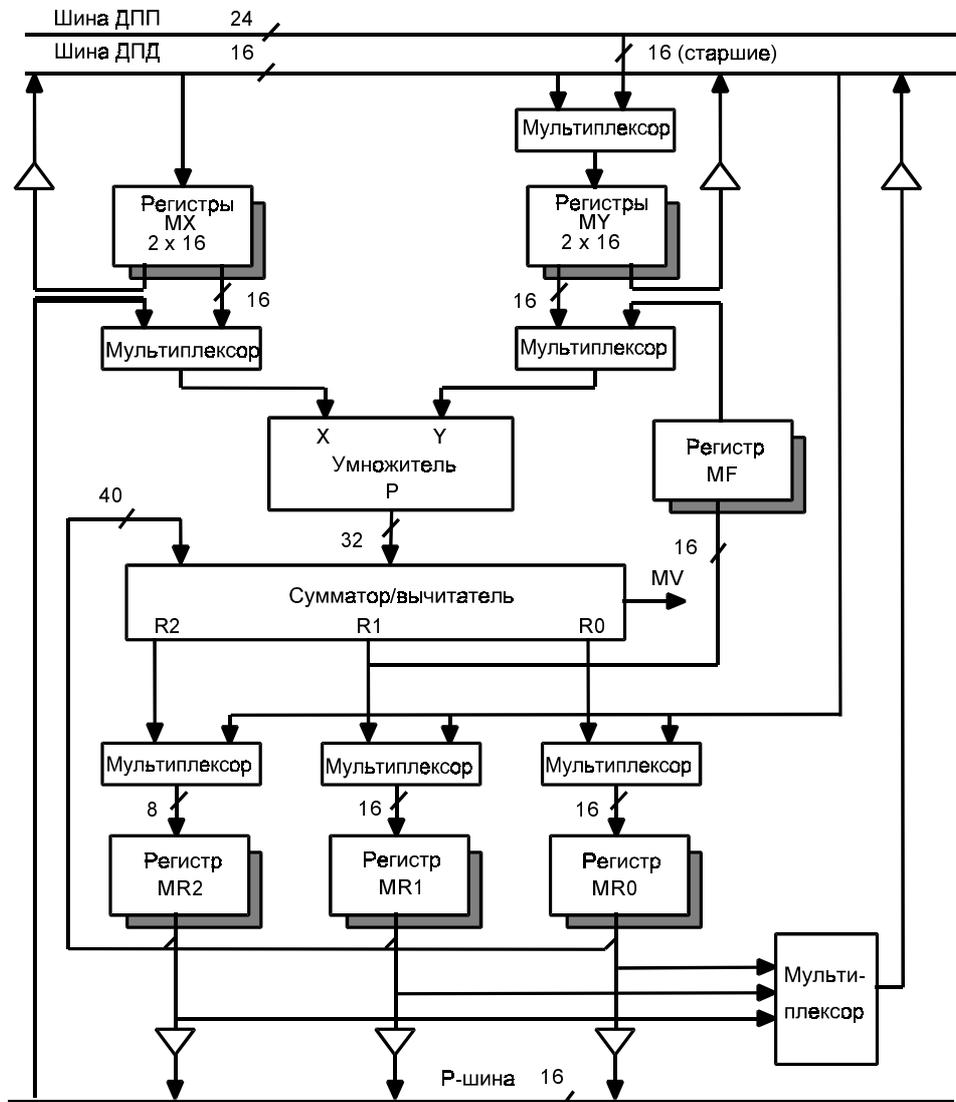
Умножитель обеспечивает высокую скорость умножения, умножение с накопленным сложением и вычитанием, функции насыщения и сброса в ноль. Функция обратной связи позволяет использовать часть выходных значений сумматора в качестве одного из сомножителей в следующем цикле.

#### 2.3.1 Блок-схема умножителя-накопителя

Блок-схема умножителя-накопителя приведена на рис. 2.6.

В умножителе имеется два входных порта X и Y разрядностью 16 бит и порт вывода результата R разрядностью 32 бита. Результат разрядностью 32 бита передается в 40-разрядный сумматор/вычитатель, который прибавляет или вычитает новый результат из содержимого регистра умножителя MR или передает новый результат прямо в MR. Регистр MR имеет разрядность 40 бит. В данном руководстве весь регистр обозначен как MR. В действительности он состоит из трех регистров меньших размеров: MR0 и MR1, которые имеют разрядность 16 бит, и MR2 разрядностью 8 бит.

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2



**Рис. 2.6** Блок-схема умножителя-накопителя

Сумматор / вычитатель имеет разрядность, большую чем 32, чтобы предотвратить переполнение при выполнении серии операций умножителя-накопителя. Бит состояния переполнения умножения (MV) устанавливается, когда сумматор переполняется свыше границы в 32 бита, то есть, когда в верхних

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

девяти битах регистра MR имеются старшие (беззнаковые) биты (при операциях над числами в дополнительном коде).

Регистры ввода/вывода множителя аналогичны регистрам ввода/вывода АЛУ.

Порт ввода X может принимать данные либо из регистров MX, либо из любого регистра по R-шине (результатов). R-шина соединяет регистры вывода всех вычислительных устройств, что позволяет использовать их сразу в качестве операндов ввода. Имеется два регистра MX: MX0 и MX1. Запись в эти регистры и считывание из них может производиться с шины ДПД. Два выхода регистров MX позволяют осуществлять ввод данных в множитель одновременно с передачей данных на шину ДПД.

Порт ввода Y может принимать данные либо из регистров MY, либо из регистра MF. Имеется два регистра MY: MY0 и MY1. Запись в эти регистры и считывание с них может производиться с шины ДПД, а запись также может быть произведена и с шины ДПП. В наборе команд предусматривается также считывание из этих регистров на шину ДПП, но в данном случае нет прямого соединения; в этой операции используется устройство обмена данными между шинами ДПД и ДПП. Один из регистров MY может обеспечивать ввод данных в множитель, в то время как другой одновременно передает данные на шину ДПД.

Выходное значение множителя-аккумулятора поступает либо в регистр MF, либо в регистр MR. MF является регистром обратной связи и позволяет использовать биты результата 16-31 в качестве вводимых через порт Y данных в последующем цикле. Регистр сумматора/вычитателя MR разрядностью 40 бит разделен на три составные части: MR0, MR1 и MR2. Каждый из этих регистров может загружаться непосредственно с шины ДПД и передавать данные либо на шину ДПД, либо на R-шину.

Запись и считывание из любого регистра, связанного с множителем-накопителем, может осуществляться в одном и том же цикле. Считывание из регистров происходит в начале цикла, а запись в регистры - в конце цикла. Следовательно, при считывании из регистра считывается значение, загруженное в этот регистр в конце предыдущего цикла. Новое значение, записанное в регистр, невозможно считать до следующего цикла. Это позволяет регистру ввода обеспечивать множитель операндом в начале цикла и обновлять свое содержимое следующим операндом из памяти в конце этого же цикла. Это также позволяет хранить содержимое регистра результатов в памяти и обновлять его новым результатом в течение этого же цикла. Пример считывания и записи в течение одного цикла можно найти в главе 15 «Набор команд» в разделе «Многофункциональные команды».

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

В умножителе имеется двойной ряд регистров, которые можно видеть на рис. 2.6 за основными регистрами. В действительности существует два набора регистров MR, MF, MX, MY. В любой определенный момент времени доступен только один ряд. Дополнительный ряд регистров может быть активизирован для чрезвычайно быстрого контекстного переключения. Таким образом, новая задача, например программа обслуживания прерываний, может выполняться без необходимости передачи текущих состояний в ЗУ.

Выбор основного или альтернативного ряда регистров определяется битом 0 в регистре состояния режима процессора (MSTAT). Если этот бит равен 0, то выбран основной ряд, если он равен 1 - альтернативный ряд регистров.

### 2.3.2 Операции умножителя-накопителя

В данном разделе описываются функции умножителя, форматы ввода данных и поведение умножителя при насыщении и переполнении.

#### 2.3.2.1 Стандартные функции

Функциями умножителя-накопителя являются:

$X*Y$	Умножение операндов X и Y
$MR + X*Y$	Умножение операндов X и Y и сложение результата с содержимым регистра MR
$MR - X*Y$	Умножение операндов X и Y и вычитание полученного результата из содержимого регистра MR.
0	Обнуление результата регистра MR.

Процессоры семейства ADSP-2100 имеют два режима для стандартной операции умножения с накоплением: режим умножения дробных чисел (в формате 1.15) и режим умножения целых чисел (в формате 16.0).

В режиме умножения дробных чисел результат P разрядностью 32 бита регулируется по формату, то есть дополняется по знаку и сдвигается на один бит влево, перед добавлением к содержимому регистра MR. Например, бит 31 результата P выравнивается до бита 32 регистра MR (то есть бита 0 регистра MR0). Самый младший бит содержит 0. Формат результата умножения в режиме дробных чисел показан на рис. 2.7.

В режиме умножения целых чисел содержимое регистра результата P разрядностью 32 бита не сдвигается перед добавлением его к содержимому регистра MR. На рис.2.8 показано расположение результата умножения целых чисел.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Режим умножения задается битом 4 регистра состояния режима (MSTAT). Если указанный бит равен 1, то выбран режим умножения целых чисел. В противном случае выбран режим умножения дробных чисел. В обоих режимах результат умножения P помещается в 40-разрядный сумматор/вычитатель, в котором новое значение результата добавляется к текущему значению регистра MR, и в итоге получается конечный результат P разрядностью 40 бит.



Рис. 2.7 Формат результата умножения дробных чисел

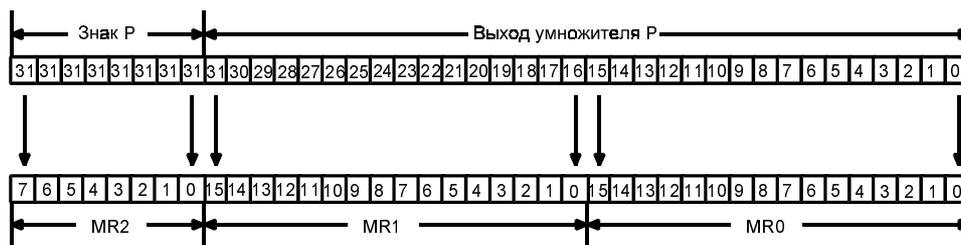


Рис. 2.8 Формат результата умножения целых чисел

### 2.3.2.2 Форматы ввода данных

Чтобы обеспечить умножение с повышенной точностью, при вводе операндов X и Y в умножитель возможны любые комбинации чисел в дополнительном коде и в беззнаковом формате.

Операнд X		Операнд Y
знаковый	*	знаковый
беззнаковый	*	знаковый
знаковый	*	беззнаковый
беззнаковый	*	беззнаковый

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

Форматы ввода данных задаются как часть команды. Они динамически выбираются каждый раз, когда используется умножитель.

Режим «знаковый\*знаковый» используется при умножении двух знаковых чисел с одинарной точностью или старших слов двух знаковых чисел с повышенной точностью.

Режимы «беззнаковый\*знаковый» и «знаковый\*беззнаковый» используются при умножении старших бит одного знакового числа с повышенной точностью на младшие биты другого, или при умножении знакового числа с одинарной точностью на беззнаковое число с одинарной точностью.

Режим «беззнаковый\*беззнаковый» используется при умножении беззнаковых чисел с одинарной точностью или всех бит, кроме старших, двух знаковых чисел с повышенной точностью.

### 2.3.2.3 Регистры ввода/вывода умножителя-накопителя

Источники ввода данных и направление вывода данных умножителя приведены ниже.

<i>Источники ввода данных в порт X</i>	<i>Источники ввода данных в порт Y</i>	<i>Направление вывода данных через порт R</i>
MX0, MX1	MY0, MY1	MR (MR2, MR1, MR0)
AR	MF	MF
MR0, MR1, MR2		
SR0, SR1		

### 2.3.2.4 Операции регистра MR

Как уже было показано на блок-схеме, регистр MR делится на три части: MR0 (биты 0-15), MR1 (биты 16-31) и MR2 (биты 32-39). Каждый из этих регистров может быть загружен с шины ДПД и может передавать данные на Р-шину или шину ДПД.

8-разрядный регистр MR2 привязан к младшим 8 битам этих шин. Когда MR2 передает данные на шину ДПД или Р-шину, происходит добавление знаковых бит для формирования 16-разрядной величины. Регистр MR1 также имеет возможность автоматического добавления знаковых бит. Когда MR1 загружается с шины ДПД, каждый бит регистра MR2 устанавливается как знаковый бит (самый старший бит) MR1, таким образом, MR2 представляется как добавление к MR1. Чтобы загрузить в MR2 значение, отличное от этих дополнительных бит знака MR1, следует загружать MR2, после того как MR1 был уже загружен.

Загрузка MR0 не влияет на содержимое MR2 и MR1; при загрузке MR0 добавления знаковых бит не происходит.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

### 2.3.2.5 Переполнение и насыщение умножителя

Сумматор/вычитатель генерирует сигнал переполнения (MV), который загружается в арифметический регистр состояний (ASTAT) каждый раз, когда выполняется операция умножителя. Бит MV устанавливается тогда, когда результат накопителя, интерпретируемый как число в дополнительном коде, выходит за пределы 32-разрядной (MR1/MR2) границы. Таким образом, MV устанавливается если старшие девять бит MR не являются все нулями или единицами.

В регистре MR предусмотрен режим насыщения, благодаря которому MR устанавливается на максимальное положительное или отрицательное значение при переполнении или потере значащих разрядов. Операция насыщения зависит от бита состояния переполнения (MV) в арифметическом регистре состояний (ASTAT) и от самого старшего бита регистра MR2. В следующей таблице представлена операция насыщения регистра MR.

Таблица 2.3

		<b>Результат команды насыщения умножителя</b>	
<i>MV</i>	<i>Самый старший бит MR2</i>	<i>Содержимое регистра MR после насыщения</i>	
0	0 или 1	нет изменений	
1	0	00000000 0111111111111111 1111111111111111	<i>полномасштабная положительная величина</i>
1	1	11111111 1000000000000000 0000000000000000	<i>полномасштабная отрицательная величина</i>

Насыщение в умножителе, в отличие от АЛУ, лучше понимать как команду, нежели режим. Команда насыщения предназначена для использования при завершении последовательности умножений с накоплением, таким образом, чтобы промежуточное переполнение не приводило к насыщению сумматора.

Переполнение за самый старший бит MR2 не допускается. Иначе знаковый бит результата безвозвратно теряется, и при насыщении может получиться неверное значение. Но для достижения такого состояния требуется более 255 переполнений (типа MV).

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

### 2.3.2.6 Режим округления

В сумматоре имеется возможность округления 40-разрядного результата R на границе между 16 и 15 битами. Округление может быть задано как часть кода команды. Округленное выходное значение направляется либо в регистр MR, либо в регистр MF. При округлении с выводом результата в регистр MF содержимое этого регистра представляет собой округленный 16-разрядный результат. Аналогичным образом, при выборе MR в качестве регистра вывода округленный 16-разрядный результат содержится в MR1; округленное значение в MR1 отражается и на содержимом MR2, а MR1 и MR2 содержат, таким образом, округленный 24-разрядный результат.

В сумматоре используется округление по схеме без смещения. Условный метод округления со смещением заключается в добавлении 1 к 15-тому биту в цепочке сумматора. Округление по данному методу приводит к чисто положительному смещению, так как срединное значение (когда MR0 = 0x8000) всегда округляется в сторону увеличения. Это смещение устраняется в сумматоре за счет принудительной установки 16-го бита выводимого результата в 0 при обнаружении такого половинного значения. В результате этой установки нечетные значения MR1 округляются в сторону увеличения, а четные значения MR1 - в сторону уменьшения, что приводит в конечном итоге к нулевому смещению при большой выборке в случае равномерно распределенных значений.

Ниже приводятся два примера округления, в которых значок «x» использован для представления любого шаблона бит (когда все биты не равны 0). В первом примере показана типичная операция округления.

<i>Пример 1</i>	<i>MR2</i>	<i>MR1</i>	<i>MR0</i>
Неокругленное значение:	xxxxxxx	xxxxxxx00100101	1xxxxxxxxxxxxxx

Бит 15 = 1

Добавить 1 к 15-ому биту с последующим переносом		1	
Округленное значение:	xxxxxxx	xxxxxxx00100110	0xxxxxxxxxxxxxx

Компенсирование для избежания смещения становится видимым, когда младшие 15 бит равны 0, а 15-ый бит равен 1, т.е. при половинном значении.

<i>Пример 2</i>	<i>MR2</i>	<i>MR1</i>	<i>MR0</i>
Неокругленное значение:	xxxxxxx	xxxxxxx01100110	10000000000000

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Бит 15 = 1, а биты 0-14 = 0

Добавить 1 к 15-ому биту с последующим переносом 1

xxxxxxx    xxxxxxxx01100111    0000000000000000

Так как 16-ый бит равен 1, принудительная установка его на 0

Округленное значение:    xxxxxxxx    xxxxxxxx01100110    0000000000000000

В последнем случае 16-ый бит принудительно устанавливается в 0. Этот алгоритм используется в каждой операции округления, но становится явным, только при таком наборе младших 16 бит, который показан в последнем примере.

### 2.3.2.7 Округление со смещением (ADSP-217х, ADSP-218х, ADSP-21msp5х)

В процессорах ADSP-217х, ADSP-218х и ADSP-21msp58/59 имеется режим, позволяющий производить, помимо нормального округления, операцию округления со смещением. Этот режим устанавливается битом BIASRND (бит 12 в регистре управления автобуферизацией последовательного порта SPORT0). Когда бит BIASRND равен 0, производится нормальное округление без смещения. Когда бит BIASRND равен 1, вместо нормального округления производится округление со смещением. При работе в режиме округления со смещением, в большую сторону при значении MR0, равном 0x8000, округляются все, а не только нечетные значения MR1, например:

<i>Значение MR0 до округления</i>	<i>Результат округления со смещением</i>	<i>Результат округления без смещения</i>
00-0000-8000	00-0001-8000	00-0000-8000
00-0001-8000	00-0002-8000	00-0002-8000
00-0000-8001	00-0001-8001	00-0001-8001
00-0001-8001	00-0002-8001	00-0002-8001
00-0000-7FFF	00-0000-7FFF	00-0000-7FFF
00-0001-7FFF	00-0001-7FFF	00-0001-7FFF

Данный режим действует только, когда в регистре MR0 содержится значение 0x8000; во всех остальных случаях производятся операции округления без смещения. Такой режим позволяет более эффективно осуществлять алгоритмы, в которых используется округление со смещением, например, подпрограммы сжатия речи в системах GSM. В остальных алгоритмах предпочтительнее округление без смещения.

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

### 2.4 УСТРОЙСТВО ЦИКЛИЧЕСКОГО СДВИГА

Устройство сдвига обеспечивает полный набор функций сдвига для 16-разрядных входных значений, в результате чего на выходе получается 32-разрядное значение. Сюда относятся арифметический и логический сдвиги и операция нормализации. Устройство сдвига также выполняет операции нахождения порядка и нахождения общего порядка для целого блока чисел. Эти базовые функции могут комбинироваться для эффективного управления преобразованием форматов чисел, включая представление с плавающей точкой.

#### 2.4.1 Блок-схема устройства сдвига

Блок-схема устройства сдвига приведена на рисунке 2.9. В устройстве сдвига можно выделить следующие компоненты: массив устройства сдвига, логическое устройство ИЛИ/ПЕРЕДАЧА (OR/PASS), определитель порядка и логическое устройство сравнения порядка.

Массив устройства сдвига представляет собой регистр циклического сдвига размером 16 на 32 бита. В него вводится 16-разрядное значение, которое может быть помещено в любое место 32-разрядной области результата, от крайне правой до крайне левой, за один цикл. Таким образом, возможно 49 различных расположений в 32-разрядной области результата. Расположение входных 16 бит определяется управляющим кодом (C) и опорным сигналом HI/LO.

Массив устройства сдвига с обслуживающими его логическими устройствами окружены набором регистров. Регистр ввода устройства сдвига (SI), обеспечивает ввод данных в массив устройства сдвига и в определитель порядка.

Считывание из регистра SI разрядностью 16 бит и запись в него может производиться с шины ДПД. Ввод данных в массив устройства сдвига и определитель порядка также производится из регистров AR, SR или MR по P-шине. Регистр результата устройства сдвига (SR) имеет разрядность 32 бита и разделяется на два 16-разрядных регистра, SR0 и SR1. Загрузка в регистры SR0 и SR1 может производиться с шины ДПД, а вывод результата - либо на шину ДПД либо на P-шину. Регистр SR также имеет обратную связь с логическим устройством ИЛИ/ПЕРЕДАЧА для обеспечения операций сдвига с двойной точностью.

Регистр SE («порядок устройства сдвига») разрядностью 8 бит содержит значение порядка при проведении операций нормализации и денормализации. Регистр SE может считываться и загружаться с 8 младших битов шины ДПД. Это значение 8.0 в дополнительном коде.

Регистр SB («блок устройства сдвига») используется для операций с блочной плавающей точкой, во время которых в нем содержится значение блочного

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

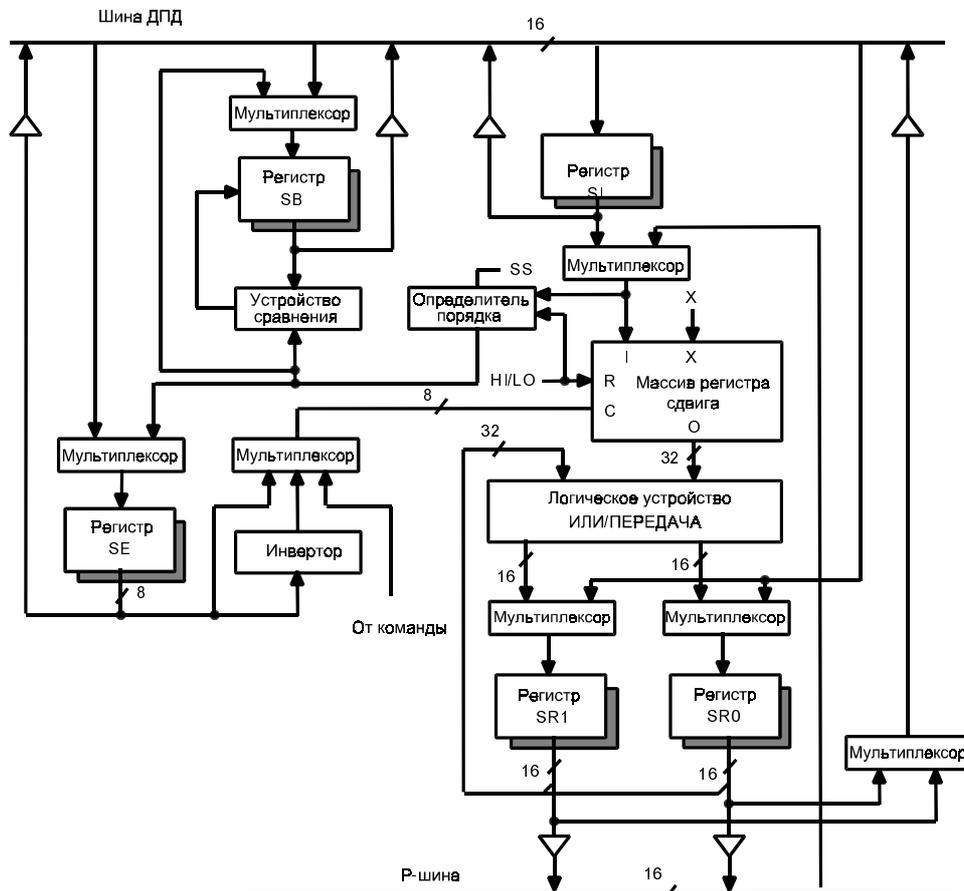


Рис. 2.9 Блок-схема устройства сдвига

порядка, то есть значение, на которое должен произойти сдвиг для нормализации самого большого числа в блоке чисел. 5-разрядный регистр SB считывается и загружается с 5 младших битов шины ДПД. Это значение 5.0 в дополнительном коде.

Всегда, когда значения регистров SE или SB передаются на шину ДПД, в них добавляются знаковые биты для формирования 16-разрядного значения.

Запись в любой из регистров SE, SI или SB и считывание из них могут производиться за один цикл. Считывание из регистров производится в начале цикла, а запись в них - в конце цикла. Следовательно, при считывании со всех регистров производится считывание значений, загруженных в них в конце предыдущего цикла. Новое значение, записываемое в регистр, не может быть

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

считано до начала следующего цикла. Таким образом, регистр ввода обеспечивает операнд для устройства сдвига в начале цикла и обновляется значением следующего операнда в конце того же цикла. Это также позволяет сохранять содержимое регистра результата в памяти и обновлять его новым результатом в течение того же цикла. Пример такого считывания и записи в течение одного цикла приводится в разделе «Многофункциональные команды» в главе 15 «Набор команд».

В устройстве сдвига имеется еще один ряд регистров, который показан на рис. 2.9 за основными регистрами. В действительности существует два набора регистров SE, SB, SI, SR1 и SR0. В любой определенный момент времени доступным является лишь один ряд регистров. Дополнительный ряд регистров может быть активизирован для быстрого контекстного переключения. В этом случае новая задача, например, подпрограмма обслуживания прерываний, может быть выполнена без передачи текущих состояний в ЗУ.

Выбор основного или альтернативного ряда регистров определяется битом 0 в регистре состояния режима процессора (MSTAT). Когда этот бит равен 0, то выбирается основной ряд регистров; когда он равен 1, выбирается альтернативный ряд регистров.

Сдвиг входного значения зависит от управляющего кода (C) и опорного сигнала HI/LO. Управляющий код представляет собой 8-разрядную знаковую величину, которая показывает направление и число позиций, на которое сдвигается входное значение. Положительный код означает сдвиг влево (вверх), а отрицательный код - вправо (вниз). Управляющий код может иметь три источника: содержимое регистра порядка устройства сдвига (SE), отрицательное содержимое регистра SE или непосредственно значение из команды.

Опорный сигнал HI/LO определяет опорную точку сдвига. В состоянии HI, все сдвиги происходят относительно SR1 (старших бит результата), а в состоянии LO - относительно SR0 (младших бит). Модификатор HI/LO полезен при сдвиге 32-разрядных значений, так как позволяет осуществлять сдвиг и старших и младших бит числа при одном и том же управляющем коде. Возможность выбора вида модификатора HI/LO появляется каждый раз при использовании устройства сдвига.

Все биты справа от входного значения в области результата заполняются нулями, а биты слева от него - дополнительным битом (X). Дополнительный бит может устанавливаться тремя возможными источниками в зависимости от выполняемой команды. Ими являются самый старший бит входного значения, бит AC арифметического регистра состояния (ASTAT) или ноль.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

В табл. 2.4 выходное значение устройства сдвига показано как функция управляющего кода и модификатора HI/LO.

Логическое устройство ИЛИ/ПЕРЕДАЧА позволяет соединять части числа с повышенной точностью после сдвига в одно число. Когда выбирается режим PASS (ПЕРЕДАЧА), массив данных устройства сдвига просто передается в регистр результата устройства сдвига (SR) без изменений. Когда выбирается режим OR (ИЛИ), массив данных устройства сдвига перед загрузкой в SR поразрядно подвергается логической операции ИЛИ с текущим содержимым регистра SR.

Определитель порядка находит порядок входного значения устройства сдвига. Имеется три возможных способа интерпретации входного значения. В состоянии HI входное значение интерпретируется как число с одинарной точностью или как старшие биты числа с двойной точностью. Определитель порядка определяет число ведущих знаковых бит и формирует код, который указывает, на сколько позиций должно быть сдвинуто входное значение, чтобы устранить все биты, кроме знаковых. Этот код является отрицательным и, следовательно, может быть эффективным порядком мантииссы, образованной после удаления лишних знаковых бит.

В расширенном состоянии HI (HIX) входное значение интерпретируется как результат сложения или вычитания, выполняемого в АЛУ, которое может быть переполнено. Тогда определитель порядка учитывает состояние переполнения (AV). Если установлен бит AV, то выводится порядок +1. Чтобы указать на необходимость дополнительного бита в нормализованной мантииссе (бит переноса АЛУ); если AV не установлен, то состояние HIX функционирует также как HI. При выполнении функции извлечения порядка в режимах HI и HIX определитель порядка также выводит знаковый бит устройства сдвига (SS), который загружается в арифметический регистр состояния (ASTAT). Знаковый бит равен самому старшему биту входного значения устройства сдвига, за исключением тех случаев, когда установлен бит AV; если AV установлен в расширенном состоянии HI (HIX), то для восстановления знакового бита переполненного значения инвертируется самый старший бит.

В состоянии LO входное значение интерпретируется как младшие биты числа с двойной точностью. В этом состоянии определитель порядка определяет бит SS арифметического регистра состояния (ASTAT) как знаковый бит этого числа. Выходное значение определителя порядка загружается в регистр SE только, когда SE содержит -15. Это происходит только в том случае, когда старшие биты, которые должны обрабатываться в первую очередь, содержат все знаковые биты. Выходное значение определителя порядка сдвигается на -16, чем и объясняется

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

Таблица 2.4

Характеристика массива устройства сдвига						
НИ	управляющий код	ЛО	выходной массив устройства сдвига			
+16 до +127		+32 до +127	00000000	00000000	00000000	00000000
+15		+31	R0000000	00000000	00000000	00000000
+14		+30	PR000000	00000000	00000000	00000000
+13		+29	NPR00000	00000000	00000000	00000000
+12		+28	MNPR0000	00000000	00000000	00000000
+11		+27	LMNPR000	00000000	00000000	00000000
+10		+26	KLMNPR00	00000000	00000000	00000000
+9		+25	JKLMNPR0	00000000	00000000	00000000
+8		+24	IJKLMNPR	00000000	00000000	00000000
+7		+23	HIJKLMNP	R0000000	00000000	00000000
+6		+22	GHIJKLMN	PR000000	00000000	00000000
+5		+21	FGHIJKLM	NPR00000	00000000	00000000
+4		+20	EFGHIJKL	MNPR0000	00000000	00000000
+3		+19	DEFGHIJK	LMNPR000	00000000	00000000
+2		+18	CDEFGHIJ	KLMNPR00	00000000	00000000
+1		+17	BCDEFGHI	GKLMNPR0	00000000	00000000
0		+16	ABCDEF GH	IJKLMNPR	00000000	00000000
-1		+15	XABCDEFG	HIJKLMNP	R0000000	00000000
-2		+14	XXABCDEF	GHIJKLMN	PR000000	00000000
-3		+13	XXXABCDE	FGHIJKLM	NPR00000	00000000
-4		+12	XXXXABCD	EFGHIJKL	MNPR0000	00000000
-5		+11	XXXXXABC	DEFGHIJK	LMNPR000	00000000
-6		+10	XXXXXXAB	CDEFGHIJ	KLMNPR00	00000000
-7		+9	XXXXXXXA	BCDEFGHI	JKLMNPR0	00000000
-8		+8	XXXXXXXX	ABCDEF GH	IJKLMNPR	00000000
-9		+7	XXXXXXXX	XABCDEFG	HIJKLMNP	R0000000
-10		+6	XXXXXXXX	XXABCDEF	GHIJKLMN	PR000000
-11		+5	XXXXXXXX	XXXABCDE	FGHIJKLM	NPR00000
-12		+4	XXXXXXXX	XXXXABCD	EFGHIJKL	MNPR0000
-13		+3	XXXXXXXX	XXXXXABC	DEFGHIJK	LMNPR000
-14		+2	XXXXXXXX	XXXXXXAB	CDEFGHIJ	KLMNPR00
-15		+1	XXXXXXXX	XXXXXXXA	BCDEFGHI	JKLMNPR0
-16		0	XXXXXXXX	XXXXXXXX	ABCDEF GH	IJKLMNPR
-17		-1	XXXXXXXX	XXXXXXXX	XABCDEFG	HIJKLMNP
-18		-2	XXXXXXXX	XXXXXXXX	XXABCDEF	GHIJKLMN
-19		-3	XXXXXXXX	XXXXXXXX	XXXABCDE	FGHIJKLM
-20		-4	XXXXXXXX	XXXXXXXX	XXXXABCD	EFGHIJKL
-21		-5	XXXXXXXX	XXXXXXXX	XXXXXABC	DEFGHIJK
-22		-6	XXXXXXXX	XXXXXXXX	XXXXXXAB	CDEFGHIJ
-23		-7	XXXXXXXX	XXXXXXXX	XXXXXXXA	BCDEFGHI
-24		-8	XXXXXXXX	XXXXXXXX	XXXXXXXX	ABCDEF GH
-25		-9	XXXXXXXX	XXXXXXXX	XXXXXXXX	XABCDEFG
-26		-10	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXABCDEF
-27		-11	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXABCDE
-28		-12	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXABCD
-29		-13	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXABC
-30		-14	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXAB
-31		-15	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXA
-32 до -128		-16 до -128	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

ABCDEF GHIJKLMNPR представляет вводимую комбинацию разрядностью в 16 бит,  
X - бит расширения

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

то, что, фактически, входное значение представляет собой младшие 16 бит 32-разрядного числа. В таблице 2.5. приведены характеристики определителя порядка для всех трех режимов.

Логическое устройство сравнения порядка используется для нахождения самого большого значения порядка в массиве входных значений устройства сдвига. В соединении с определителем порядка, логическое устройство сравнения порядка находит порядок блока. Компаратор сравнивает значение порядка, извлеченное определителем порядка, со значением, хранящемся в регистре блочного порядка устройства сдвига SB и обновляет содержимое SB только, когда извлеченное значение порядка превышает значение в регистре SB. См. примеры, данные ниже.

### 2.4.2 Операции устройства сдвига

Устройство сдвига выполняет следующие функции (мнемоника команд показана в скобках):

- Арифметический сдвиг (ASHIFT)
- Логический сдвиг (LSHIFT)
- Нормализация (NORM)
- Определение порядка (EXP)
- Нахождение (корректировка) блочного порядка (EXPANDJ)

Эти основные команды устройства сдвига применяются при различных арифметических операциях. В следующем разделе даются примеры выполнения этих трех функций с одинарной и многократной точностью.

- Извлечение блочного порядка
- Непосредственный сдвиг
- Нормализация
- Денормализация

Функции устройства сдвига (арифметический сдвиг, логический сдвиг, нормализация и денормализация) могут дополнительно определяться режимами [SR OR] и HI/LO, чтобы облегчить выполнение операций с повышенной точностью. В режиме [SR OR] над результатом устройства сдвига и текущим содержимым SR выполняется операция логического ИЛИ. Это используется для соединения 16-разрядных частей в 32-разрядное значение в регистре SR. HI и LO модифицируют опорную точку сдвига: по старшим или младшим битам 32-разрядного регистра SR. Входные значения для этих операций устройства сдвига берутся или из регистра SI, или из любого другого регистра результатов, а 32-разрядный результат сдвига загружается в регистр SR.

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

Таблица 2.5

### Характеристики определителя порядка

<b>Режим НI</b>		<b>Режим НIХ</b>		
Входной массив устройства	Результат	AV	Входной массив устройства	Результат
		1	сдвига	
		0	DDDDDDDD DDDDDDDD	+1
SNDDDDDD DDDDDDDD	0	0	SNDDDDDD DDDDDDDD	0
SSNDDDDD DDDDDDDD	-1	0	SSNDDDDD DDDDDDDD	-1
SSSNDDDD DDDDDDDD	-2	0	SSSNDDDD DDDDDDDD	-2
SSSSNDDD DDDDDDDD	-3	0	SSSSNDDD DDDDDDDD	-3
SSSSSNDD DDDDDDDD	-4	0	SSSSSNDD DDDDDDDD	-4
SSSSSSND DDDDDDDD	-5	0	SSSSSSND DDDDDDDD	-5
SSSSSSSN DDDDDDDD	-6	0	SSSSSSSN DDDDDDDD	-6
SSSSSSSS NDDDDDDD	-7	0	SSSSSSSS NDDDDDDD	-7
SSSSSSSS SNDDDDDD	-8	0	SSSSSSSS SNDDDDDD	-8
SSSSSSSS SSNDDDDD	-9	0	SSSSSSSS SSNDDDDD	-9
SSSSSSSS SSSNDDDD	-10	0	SSSSSSSS SSSNDDDD	-10
SSSSSSSS SSSSNDDD	-11	0	SSSSSSSS SSSSNDDD	-11
SSSSSSSS SSSSSNDD	-12	0	SSSSSSSS SSSSSNDD	-12
SSSSSSSS SSSSSSND	-13	0	SSSSSSSS SSSSSSND	-13
SSSSSSSS SSSSSSSN	-14	0	SSSSSSSS SSSSSSSN	-14
SSSSSSSS SSSSSSSS	-15	0	SSSSSSSS SSSSSSSS	-15

### Режим LO

SS	Входной массив устройства	Результат	
	сдвига		
S	NDDDDDDD DDDDDDDD	-15	
S	SNDDDDDD DDDDDDDD	-16	
S	SSNDDDDD DDDDDDDD	-17	
S	SSSNDDDD DDDDDDDD	-18	
S	SSSSNDDD DDDDDDDD	-19	
S	SSSSSNDD DDDDDDDD	-20	
S	SSSSSSND DDDDDDDD	-21	
S	SSSSSSSN DDDDDDDD	-22	
S	SSSSSSSS NDDDDDDD	-23	
S	SSSSSSSS SNDDDDDD	-24	
S	SSSSSSSS SSNDDDDD	-25	
S	SSSSSSSS SSSNDDDD	-26	
S	SSSSSSSS SSSSNDDD	-27	
S	SSSSSSSS SSSSSNDD	-28	
S	SSSSSSSS SSSSSSND	-29	
S	SSSSSSSS SSSSSSSN	-30	
S	SSSSSSSS SSSSSSSS	-31	

**S** = знаковый бит  
**N** = беззнаковый бит  
**D** = незначащий бит

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

### 2.4.2.1 Регистры ввода/вывода

Источники входных значений устройства сдвига и направление вывода результатов даны ниже:

<i>Источники ввода данных</i>	<i>Направление вывода результатов</i>
SI	SR (SR0, SR1)

AR

MR0, MR1, MR2

SR0, SR1

### 2.4.2.2 Нахождение блочного порядка

Эта функция определяет порядок самого большого по абсолютной величине числа в массиве чисел. Для ее реализации используется команда EXPANDJ. Ниже дается последовательность шагов при типичном случае.

*A. Загрузка значения -16 в регистр SB*

Регистр SB используется для хранения порядка всего блока чисел. После выполнения серии операций EXPANDJ в регистре SB возможны значения в диапазоне от -15 до 0. Если новое значение больше текущего, то логическое устройство сравнения порядка обновляет первоначальное содержимое регистра SB. При загрузке этого регистра значением -16 его содержимое всегда будет меньше, чем любые найденные значения порядка.

*B. Обработка первого элемента массива:*

Массив(1) = 11110101 10110001

Порядок = -3

- $3 > SB(-16)$

SB получает новое значение -3

*C. Обработка следующего элемента массива:*

Массив(2) = 00000001 01110110

Порядок = -6

- $6 < -3$

SB сохраняет значение -3

*D. Продолжение обработки элементов массива*

Если находится такой элемент массива, порядок которого больше содержащегося в SB, то это новое значение порядка загружается в регистр SB. После такой обработки всех элементов массива в регистре SB оказывается порядок наибольшего числа во всем блоке. Операция нормализации в данном случае не выполняется. Операция EXPANDJ является чисто проверочной. Значение, содержащееся в регистре SB, может передаваться в регистр SE и использоваться для нормализации блока при следующем прогоне через устройство сдвига. Или оно может быть просто связанным с обрабатываемыми данными для последующей интерпретации.

### 2.4.2.3 Непосредственный сдвиг

## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

При непосредственном сдвиге входная последовательность бит просто сдвигается вправо (сдвиг вниз) или влево (сдвиг вверх) на заданное число бит. Величина сдвига и его направление задаются непосредственно в команде (примеры этой команды можно найти в главе 15 «Набор команд»). Значение, управляющее сдвигом, является 8-разрядным знаковым числом. При непосредственном сдвиге регистр SE не используется и не изменяется.

Следующий пример показывает сдвиг вправо относительно старших бит SR (SR1). Это сдвиг в режиме HI.

SI = 0xB6A3;

SR = LSHIFT SI BY -5 (HI);

Входное значение 10110110 10100011

Величина сдвига -5

SR            00000**101 10110101 00011** 000 000000

А теперь то же самое значение сдвигается в другом направлении, относительно младших бит регистра SR (LO).

SI = 0xB6A3;

SR = LSHIFT SI BY 5 (LO);

Входное значение 10110110 10100011

Величина сдвига +5

SR            00000000 000**10110 11010100 011** 00000

Кроме направления операции сдвига, сдвиг может быть либо арифметическим (ASHIFT), либо логическим (LSHIFT). Например, ниже приведен логический сдвиг относительно старших бит SR (HI).

SI = 0xB6A3;

SR = LSHIFT SI BY -5 (HI);

Входное значение 10110110 10100011

Величина сдвига -5

SR            00000**101 10110101 00011** 000 00000000

Следующий пример показывает арифметический сдвиг для того же входного значения и величины сдвига.

SI = 0xB6A3;

SR = ASHIFT SI BY -5 (HI);

Входное значение 10110110 10100011

Величина сдвига -5

SR            11111**101 10110101 00011** 000 00000000



## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

<i>Модификатор = LO, нет [SR OR]</i>	<i>Операция сдвига = логический, SE = -3</i>
Первая половина входного значения	01110110 01011101 (младшие биты требуемого результата)
SR	00000000 00000000 000 <b>01110 11001011</b>
<i>Модификаторы = HI, [SR OR]</i>	<i>Операция сдвига = арифметический, SE = -3</i>
Вторая половина входного значения	10110110 10100011 (старшие биты требуемого результата)
SR	<b>11110110 11010100 011</b> 01110 11001011

### 2.4.2.5 Нормализация

Числа с избыточными знаковыми битами требуют нормализации. Нормализация представляет собой процесс сдвига числа в дополнительном коде внутри некоторой области таким образом, что самый правый знаковый бит выравнивается по позиции самого старшего бита этой области, при этом записывается количество позиций, на которое было сдвинуто число. Эту операцию можно представить как преобразование числа с фиксированной точкой в число с плавающей точкой, при котором генерируются мантисса и порядок.

Нормализация осуществляется в два этапа. На первом этапе определяется порядок. На втором - непосредственно осуществляется сдвиг. На первом этапе используется команда EXP, которая определяет значение порядка и загружает его в регистр SE. Команда EXP опознает модификаторы HI и LO. На втором этапе используется команда NORM. Эта команда опознает не только HI и LO, но и модификатор [SR OR]. Команда NORM использует инверсное значение регистра SE в качестве управляющего кода сдвига, чтобы сдвиг выполнялся в правильном направлении.

Ниже приведен пример нормализации для входного значения с одинарной точностью.

SE = EXP AR                      (HI)

*Определяет порядок с модификатором = HI*

Входное значение            11110110 11010100

SE устанавливается на    -3

*Нормализация, с модификатором = HI Сдвиг на величину, содержащуюся в SE*

Входное значение            11110110 11010100

SR                                **10110110 10100000 00000000 00000000**

При операции нормализации над входным значением с одинарной точностью может использоваться либо модификатор HI, либо - LO в зависимости от того, в какой регистр, SR1 или SR0, соответственно, желательно поместить результат.

Значения с двойной точностью обрабатываются по той же схеме. На первом этапе определяется порядок, а на втором осуществляется нормализация двух частей входного значения. Однако, для чисел с двойной точностью каждый этап подразделяется на две операции.



## ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА 2

2. *Определение порядка, модификатор = LO*

Вторая половина входного значения  
11110110 11010100  
SE установлен на -19

3. *Нормализация, модификаторы = HI, нет [SR OR], SE = -19 (инвертируется)*

Первая половина входного значения  
11111111 11111111  
SR 00000000 00000000 00000000 00000000

Все значения SE меньше -15 (в результате сдвига на +16 или больше) сдвигают входное значение влево полностью за пределы шкалы.

4. *Нормализация, модификаторы = LO, [SR OR], SE = -19 (инвертируется)*

Вторая половина входного значения  
11110110 11010100  
SR **10110110 10100** 000 0000000 00000000

При нормализации возможна ситуация, требующая расширенного состояния НХ. Это происходит при нормализации регистра результатов АЛУ (AR), который может быть переполнен. При этой операции бит переполнения (AV) арифметического регистра состояний (ASTAT) и бит переноса (AC) считываются в конъюнкции со значением в регистр AR. Если происходит переполнение, то флаг AV устанавливается равным 1, а флаг AC содержит истинный знак значения в дополнительном коде.

Например, заданы следующие условия:

AR = 11111010 00110010  
AV = 1, указывает на переполнение  
AC = 0, истинный знаковый бит этого значения.

1. *Определение порядка, модификатор = HIX*

SE получается установленным на +1

2. *Нормализация, модификатор = HI, SE = 1*

AR = 11111010 00110010  
SR = **0**1111101 00011001

Бит AC представляется как знаковый бит, что показано выше выделением этого бита жирным шрифтом.

## 2 ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Операция НІХ выполняется соответствующим образом, независимо от того, имело ли место в действительности переполнение. Это доказывает следующий пример.

AR = 11100011 01011011

AV = 0, указывает на отсутствие переполнения

AC = 0, не значащий бит, если AV = 0

1. *Определитель порядка, модификатор = НІХ*

SE устанавливается на -2

2. *Нормализация, модификатор = НІ, SE = -2*

AR = 11100011 01011011

SR = **10001101 01101** 000 0000000 00000000

Бит AC не используется в качестве знакового. Уже одного взгляда на таблицу 2.4 достаточно, чтобы заметить, что, когда AV не указывает на переполнение, режим НІХ идентичен режиму НІ. При осуществлении операции NORM в режиме LO дополнительный бит расширения равен 0; при осуществлении операции NORM в режиме НІ бит расширения равен AC.