

## **ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С**

### **С.1 ОБЗОР**

Аппаратное обеспечение процессоров семейства ADSP-21XX поддерживает 16-разрядные данные с фиксированной точкой. Вычислительные устройства этих процессоров обладают особыми характеристиками, которые позволяют работать с другими форматами за счет соответствующего использования программных средств. В данном приложении разъясняются различные аспекты форматов данных разрядностью 16 бит, а также рассказывается, с помощью каких программных средств можно реализовать операции с блоками чисел в формате с плавающей точкой.

### **С.2 БЕЗЗНАКОВЫЙ ИЛИ ЗНАКОВЫЙ ФОРМАТ : ДОПОЛНИТЕЛЬНЫЙ КОД**

Беззнаковые двоичные числа можно представить как положительные значения с абсолютной величиной, почти в два раза превышающей абсолютное значение знакового числа той же длины. Самые младшие слова чисел с многократной точностью также представляются в беззнаковом формате.

Процессоры семейства ADSP-21XX поддерживают знаковые числа, представленные в формате с дополнительным кодом. Двоично-десятичный формат, формат с избыточными битами и величины со знаком не поддерживаются этими процессорами.

### **С.3 ЦЕЛЫЕ И ДРОБНЫЕ ЧИСЛА**

Процессоры семейства ADSP-21XX, за исключением процессора ADSP-2100, который не выполняет умножения целых чисел, поддерживают форматы как целочисленных, так и дробных данных. В целых числах двоичная точка расположена справа от младшего бита, поэтому все биты их абсолютного значения имеют вес равный или превышающий 1. Этот формат показан на рис.С.1. Обратите внимание, что в дополнительном формате знаковый бит имеет отрицательный вес.

В дробном формате двоичная точка лежит внутри числа, поэтому некоторые или все биты его абсолютного значения имеют вес меньше 1. На рис.С.2. показан формат числа, двоичная точка которого расположена слева от трех самых младших бит, и указан их вес.

## ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С



Рис.С.1 Целочисленный формат

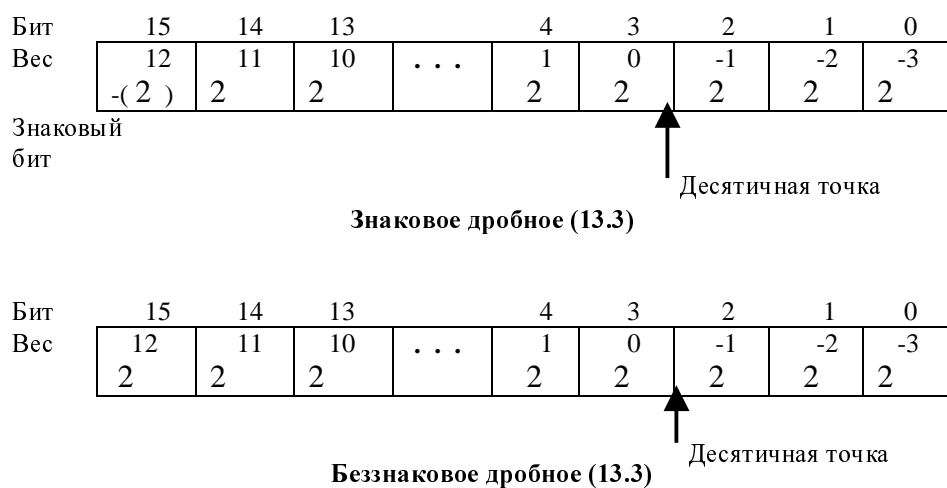


Рис.С.2 Пример дробного формата

Для обозначения формата используются две цифры, разделенные точкой (.); первая из них обозначает количество бит слева от двоичной точки, вторая - количество бит справа от двоичной точки. Например, формат 16.0 является

## ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С

целочисленным, так как все биты числа в таком формате расположены слева от двоичной точки. Число, показанное на рис. С.3. имеет формат 13.3

В таблице С.1 приведены числа разрядностью 16 бит, которые могут быть представлены в дробных форматах.

Таблица С.1

### Дробные форматы и их диапазоны

Формат	Число целых бит	Число дробных бит	Наибольшее положительное значение (0x7FFF) в десятичной системе счисления	Наименьшее отрицательное значение (0x8000) в десятичной системе счисления	Значение наименьшего значущего бита (0x0001) в десятичной системе счисления
1.15	1	15	0.999969482421875	-1.0	0.000030517578125
2.14	2	14	1.999938964843750	-2.0	0.000061035156250
3.13	3	13	3.999877929687500	-4.0	0.000122070312500
4.12	4	12	7.999755859375000	-8.0	0.000244140625000
5.11	5	11	15.999511718750000	-16.0	0.000488281250000
6.10	6	10	31.999023437500000	-32.0	0.000976562500000
7.9	7	9	63.998046875000000	-64.0	0.001953125000000
8.8	8	8	127.996093750000000	-128.0	0.003906250000000
9.7	9	7	255.992187500000000	-256.0	0.007812500000000
10.6	10	6	511.984375000000000	-512.0	0.015625000000000
11.5	11	5	1023.968750000000000	-1024.0	0.031250000000000
12.4	12	4	2047.937500000000000	-2048.0	0.062500000000000
13.3	13	3	4095.875000000000000	-4096.0	0.125000000000000
14.2	14	2	8191.750000000000000	-8192.0	0.250000000000000
15.1	15	1	16383.500000000000000	-16384.0	0.500000000000000
16.0	16	0	32767.000000000000000	-32768.0	1.000000000000000

## С.4 УМНОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ

При выполнении сложения и вычитания оба операнда должны быть представлены в одном и том же формате (знаковом или беззнаковом с одинаковым расположением двоичной точки), который совпадает с форматом результата. Сложение и вычитание знаковых и беззнаковых входных чисел выполняется одинаковым образом.

При умножении, однако, входные значения могут иметь различные форматы, от которых будет зависеть результат. Язык Ассемблер процессоров семейства ADSP-2100 позволяет описать, являются ли оба входных значения знаковыми, беззнаковыми или и тем и другим (смешанный режим). Расположение

## ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С

двоичной точки в результате определяется из ее расположения в каждом из входных значений, что показано на рис.С.3. Произведением двух 16-разрядных чисел будет 32-разрядное число. Произведение двух значений с форматами M.N и P.Q будет иметь формат (M+P).(N+Q). Например, произведение двух чисел в формате 13.3 будет иметь формат 26.6, а произведением двух чисел в формате 1.15-2.30.

Главное правило:		Пример для 4-х бит :		Пример для 16-ти бит :	
X	M.N P.Q	X	1.111 формат 1.3 11.11 формат 2.2	X	5.3 5.3
<hr/> (M+P) . (N+Q)		<hr/> 1111 1111 1111 1111 <hr/>		<hr/> 10.6	
					1.15 1.15 <hr/> 2.30
			111.00001 формат 3.5=(1+2).(2+3)		

**Рис. С.3** Формат результата умножения

Произведением двух чисел в формате с дополнительным кодом будет число с двумя знаковыми битами. Так как один из этих битов является избыточным, все биты результата можно сдвинуть на 1 бит влево. Кроме того, если одно входное значение было в формате 1.15, то после сдвига влево формат результата окажется равным формату другого входного значения (с 16 битами дополнительной точности). Например, при умножении числа в формате 1.15 на число в формате 5.11 получается число в формате 6.26. После сдвига влево на 1 бит результат имеет формат 5.27, т.е. представляет собой число в формате 5.11 плюс 16 самых младших бит.

В процессорах семейства ADSP-2100 имеется особый режим (называемый дробным), при котором результат умножения всегда подвергается сдвигу на 1 бит влево перед записью его в регистр результата. (Этот режим всегда активен в процессоре ADSP-2100, в других процессорах он может опускаться). Сдвиг влево позволяет удалить один излишний знаковый бит, когда оба операнда были знаковыми числами, и получить в итоге результат в правильном формате.

Когда оба операнда имеют формат 1.15, результат получается в формате 2.30 (30 бит в дробной части). После сдвига влево формат результата становится равным 1.31 и может округляться до 1.15. Таким образом, при использовании дробного формата наиболее удобно работать с форматом 1.15.

В целочисленном режиме сдвига влево не происходит. Этот режим используется, когда оба операнда являются целыми числами (в формате 16.0). 32-разрядный результат имеет формат 32.0 и также является целым числом.

## ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С

Целочисленный режим не доступен только в процессоре ADSP-2100, в котором результат в формате 32.0 сдвигается до формата 31.1. Так как самый старший бит при этом остается доступным в 40-разрядном накопителе, результат может быть исправлен путем сдвига вправо.

Во всех процессорах семейства, кроме процессора ADSP-2100, целочисленный и дробный режимы управляются битом в регистре MSTAT. При перезапуске по умолчанию для совместимости с ADSP-2100 устанавливается дробный режим.

### С.5 ФОРМАТ С БЛОЧНОЙ ПЛАВАЮЩЕЙ ТОЧКОЙ

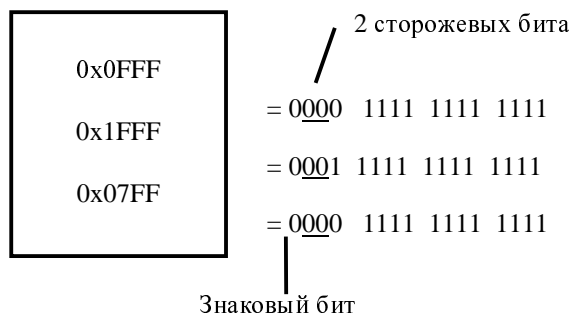
Формат с блочной плавающей точкой дает процессору, работающему с числами с фиксированной точкой, выигрыш в увеличении динамического диапазона, свойственного формату с плавающей точкой, без каких-либо дополнительных издержек, необходимых для арифметических вычислений с плавающей точкой. Однако, для работы в формате с блочной плавающей точкой, необходимо дополнительное программирование процессора.

Число с плавающей точкой имеет порядок, который указывает на расположение двоичной точки в реальном значении. В формате с блочной плавающей точкой набор (блок) значений данных имеет один и тот же общий порядок. Чтобы преобразовать значения из формата с блочной фиксированной точкой в формат с блочной плавающей точкой, каждое из них должно быть сдвинуто влево на одну и ту же величину, эта величина сдвига и будет сохранена в качестве блочного порядка.

Как правило, формат с блочной плавающей точкой позволяет сдвигать незначимые старшие биты и увеличивать тем самым точность, возможную для данного значения. Формат с блочной плавающей точкой может также использоваться, чтобы исключить возможность переполнения значения данных, такой пример приводится на рис.С.4. В каждом из трех данных имеется, по крайней мере, 2 незначимых избыточных знаковых бита. Каждое значение может увеличиться на эти два бита (две степени абсолютного значения) до того, как произойдет переполнение; поэтому эти биты называются *сторожевыми*. Если известно, что при вычислении ни одно из значений не увеличится более, чем на эти два бита, то операция будет осуществлена без потери данных. Однако, после ее завершения следует согласовать блок данных таким образом, чтобы восстановить сторожевые биты перед следующей операцией.

На рис. С.5 показаны такие обработанные, но еще несогласованные данные. Выравнивание блока данных с плавающей точкой выполняется следующим образом. В начале регистр SB содержит значение -2, что соответствует

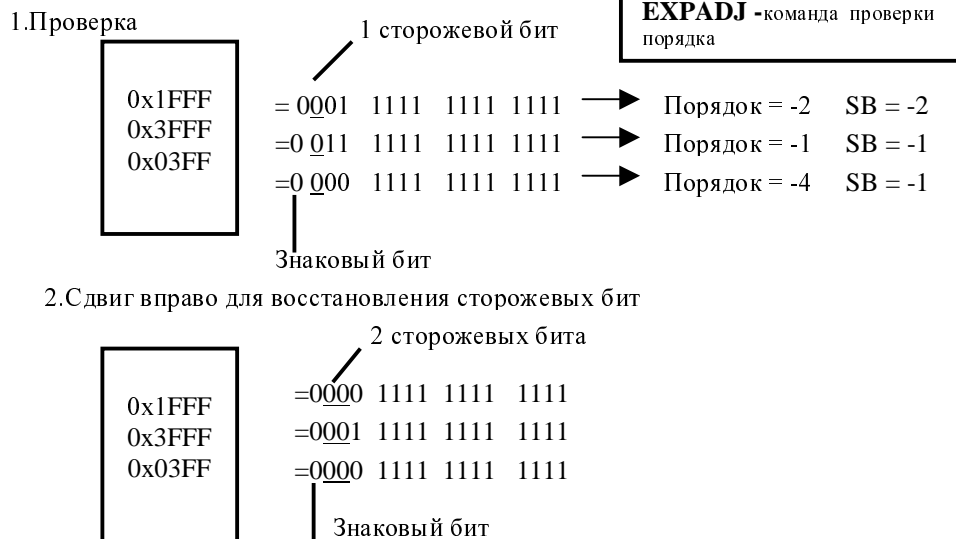
## ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ Приложение С



Для обнаружения в 2 бита установите SB = -2

**Рис. С.4 Данные со сторожевymi битами**

двум сторожевым битами. Во время обработки каждое получаемое значение данных проверяется командой EXPADJ, которая подсчитывает количество избыточных знаковых бит и корректирует значение в регистре SB, если этих избыточных бит оказывается меньше 2. В данном примере, после обработки SB=-1, т.е. указывает на то, что блок данных должен подвергнуться сдвигу на 1 бит вправо для сохранения двух сторожевых бит. Если бы после обработки SB было равно 0, то нужно было бы сдвинуть блок на 2 бита вправо. В обоих случаях, значение блочного порядка, обновляется в соответствии с требуемым сдвигом.



**Рис.С.5 Выравнивание блока данных с плавающей точкой**