Модули за числа с плаваща запетая

TECHNICAL UNIVERSITY SOFIA

Автор: гл. ас. д-р инж. Любомир Богданов

Европейски съюз

ПРОЕКТ BG051PO001--4.3.04-0042

„Организационна и технологична инфраструктура за учене през целия живот и развитие на компетенции"

Проектът се осъществява с финансовата подкрепа на Оперативна програма „Развитие на човешките ресурси", съфинансирана от Европейския социален фонд на Европейския съюз

Инвестира във вашето бъдеще!

ЕСФ България

Европейски социален фонд

Съдържание

1. Представяне на числа в двоичен вид
2. Модули и копроцесори за работа с числа с плаваща запетая (FPU)
3. Умножители (MPY)
4. Модул за защита на паметта (MPU)
5. Модул за организация на паметта (MMU)
6. Кеш памети

Представяне на числа в двоичен вид

Цифровите схеми могат да обработват само числа, представени с битове, които може да са 0 или 1.

Проблем:
*с 1 бит може да се представят само две числа, но за да върши нещо полезно, μPU трябва да може да обработва и по-големи числа, например 0 ÷ 10, 100 ÷ 200, 1000000 ÷ 3500000, и т.н.

Решение:
*да се използва група от битове, която представя числата в двоична бройна система. Колкото повече бита включва една група, толкова по-голямо десетично число може да представи тя.

Представяне на числа в двоичен вид

Прав код без знак (straight binary) – групата от битове се съпоставя директно на положителни числа от десетичната бройна система [1].

Decimal number	Binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Представяне на числа в двоичен вид

Проблем:

*Как може да се представят целочислени стойности със знак?

Решение:

*прав код (signed magnitude)

*обратен код (one’s complement)

*допълнителен код (two’s complement)

Проблем:

*Как може да се представят дробни числа (числа със запетая)?

Решение:

*Числа с фиксирана запетая

*Числа с плаваща запетая (IEEE 754-1985)

Представяне на числа в двоичен вид

Прав код (signed magnitude) – най-старшият бит (т.е. битът най-вляво) от двоичното число се заделя, за да се указва знак. Когато MSb = 1, значи че знакът е минус, а оставащите младши битове указват отрицателно число. Когато MSb = 0, знакът е плюс и числото е положително.

Проблем:
при такова представяне, в обхвата от всички числа има **две нули.**

1000.0000 (0-)
0000.0000 (0+)

Представяне на числа в двоичен вид

Работата с такива числа ще доведе до усложняване на хардуера, за да се преодолее този проблем.

Decimal number	8-bit binary signed magnitude number
-127	1111 1111
...	...
-3	1000 0011
-2	1000 0010
-1	1000 0001
-0	1000 0000
+0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
...	...
127	0111 1111

Представяне на числа в двоичен вид

Обратен код (one’s complement) – отрицателните числа се представят с инвертирани битове на техния положителен еквивалент.

При събиране на две такива числа трябва да се следи за пренос (Carry). Ако има, този бит трябва да бъде добавен обратно към резултата, иначе числото ще е грешно.

Проблем:
при такова представяне, също има **две нули.**

1111.1111 (0-)

0000.0000 (0+)

Представяне на числа в двоичен вид

Decimal number	8-bit binary one's complement number
-127	1000 0000
…	…
-3	1111 1100
-2	1111 1101
-1	1111 1110
-0	1111 1111
+0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
…	…
127	0111 1111

Представяне на числа в двоичен вид

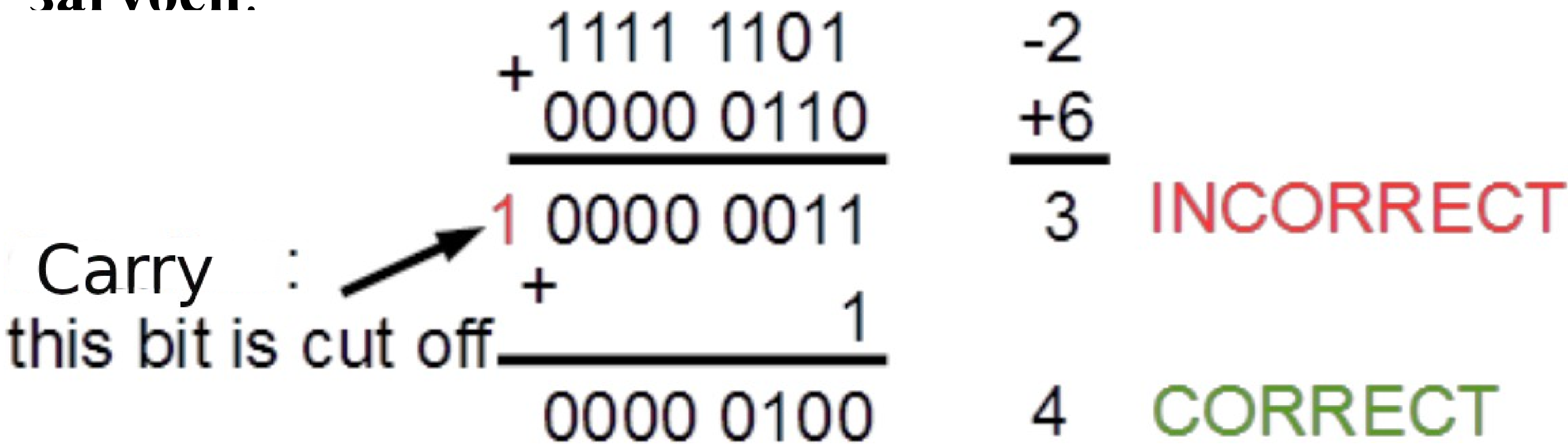
Пример – да се съберат числата -5 и +3. Нека те се представят с обратен код.

$$\begin{array}{rrrl} & 1111\ 1010 & -5 & \\ + & 0000\ 0011 & +3 & \\ \hline & 1111\ 1101 & -2 & \text{CORRECT} \end{array}$$

Представяне на числа в двоичен вид

Пример – да се съберат числата -2 и +6. Нека те се представят с обратен код.

ВНИМАНИЕ! Дължината на целочислените стойности е фиксирана (n = 4, 8, 16, 24, 32 бита). Всеки бит, който се пренесе на позиция n + 1 **бива загубен**.



Представяне на числа в двоичен вид

Допълнителен код (two’s complement) – отрицателните числа се представят с инвертирани битове на техния положителен еквивалент и след това добавяне на числото 1.

При такова представяне **има само една нула** (0000.0000).

Аритметиката се извършва **със същия хардуер**, с който се извършва аритметиката на числа, представени с **прав код без знак**.

Компилаторът е отговорен за преобразуването на числата в допълнителен код, преди да се заредят в паметта.

Представяне на числа в двоичен вид

С 8-битов допълнителен код могат да се представят числата -128 ÷ +127 (докато с обратния код имаме неефективно ползване на битовете: -127 ÷ +127).

Допълнителният код е най-често използвания в цифровите системи.

Представяне на числа в двоичен вид

Decimal number	8-bit binary one's complement number
-128	1000 0000
…	…
-3	1111 1101
-2	1111 1110
-1	1111 1111
0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
…	…
127	0111 1111

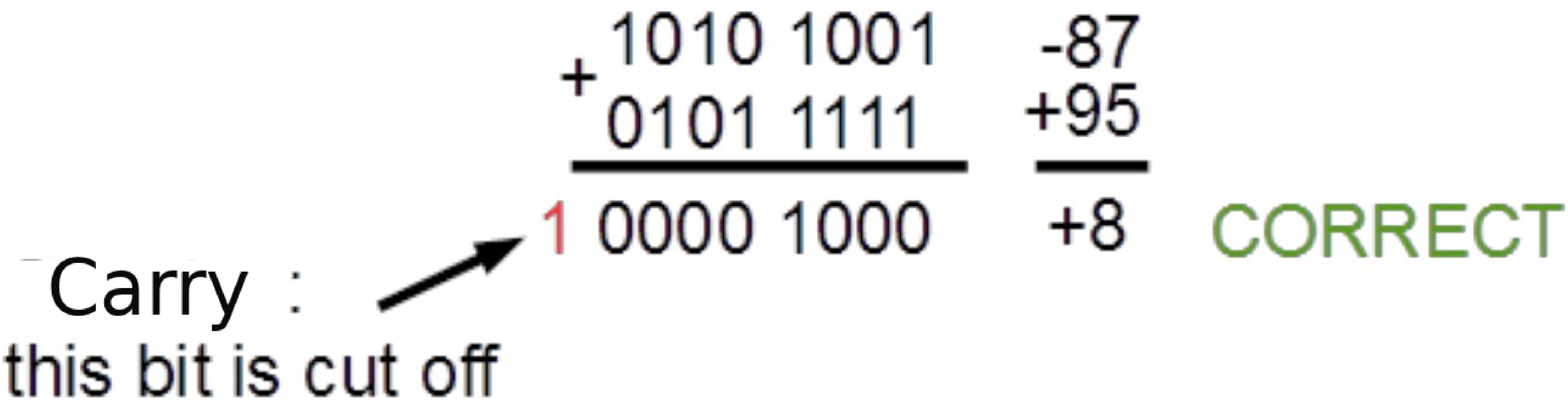
Представяне на числа в двоичен вид

Пример – да се изчисли допълнителния код на числото -87.

```
(bitwise invert)~0101 0111      +87
                 1010 1000
                +        1
                 1010 1001      -87
```

Представяне на числа в двоичен вид

Пример – да се изчисли сумата на числата в допълнителен код (-87) + (+95).



Забележка: всъщност Carry се съхранява в STATUS регистъра на μPU ядро и може да се използва за аритметика с числа с произволна разредност (софтуерно подсилване на μPU).

Представяне на числа в двоичен вид

Числа с фиксирана запетая.

Представяне на числа в двоичен вид

Числа с плаваща запетая (floating point numbers) – двоични числа, които се съпоставят на дробни числа в десетична бройна система. Съществуват различни стандарти, но най-често използвания е **IEEE754-1985,** който гласи:

Едно 32-битово дробно число се представя със следните битови полета:
*1 бит за **знак**
*8 бита за **експонента**
*23 бита за **мантиса** (mantissa, significand)

Представяне на числа в двоичен вид

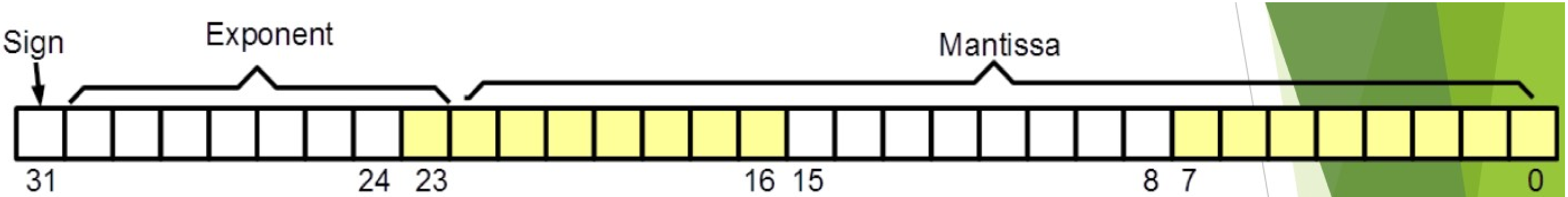
Числото +4.567 може да се представи изцяло с целочислени стойности: $+4567 \times 10^{-3}$

$+ \quad \rightarrow$ знак

$4567 \rightarrow$ мантиса

$10^{-3} \rightarrow$ експонента -3 с основа 10

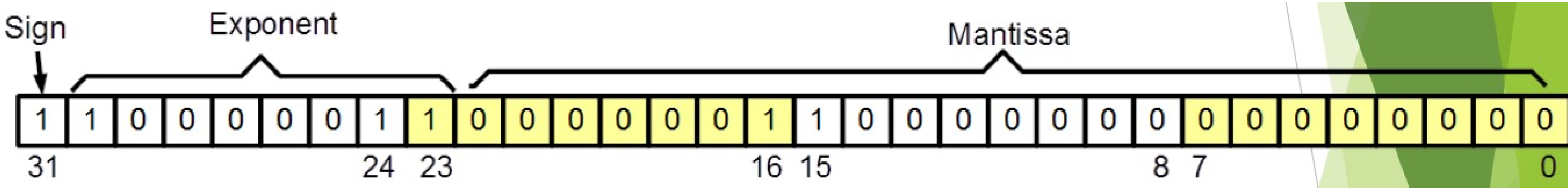
Представяне на числа в двоичен вид



$$float = (-1)^{sign} \times 2^{\exp onent-127} \times \left(1 + \sum_{n=1}^{23} bit_{23-n} \times 2^{-n}\right)$$

Представяне на числа в двоичен вид

Пример – намерете десетичния еквивалент на числото, представено с IEEE754-1985 дробно число:



*Знак: $(-1)^1 = -1$

*Експонента: $10000011_{(2)}=131_{(10)} \rightarrow 2^{131-127}=2^4$

*Мантиса: $1 + 0x2^{-1} + 0x2^{-2} + 0x2^{-3} + 0x2^{-4} + 0x2^{-5} + 0x2^{-6} + 1x2^{-7} + 1x2^{-8} + \ldots = 1 + 1/2^7 + 1/2^8 = 1.01171875$

*Резултат: $-1 \text{ x } 2^4 \text{ x } 1.01171875 = -16.1875$

*Или използвайте онлайн конвертора [2] :-)

Литература

[1] Jason Albanus, “Coding Schemes Used with Data Converters”, SBAA042A, Texas Instruments, 2015.
[2] https://www.h-schmidt.net/FloatConverter/IEEE754.html