Logical Design

CS 221

Prof.Dr. Mohamed Osama Khozium

3. Number Systems

Location in course textbook

Chapt. 1

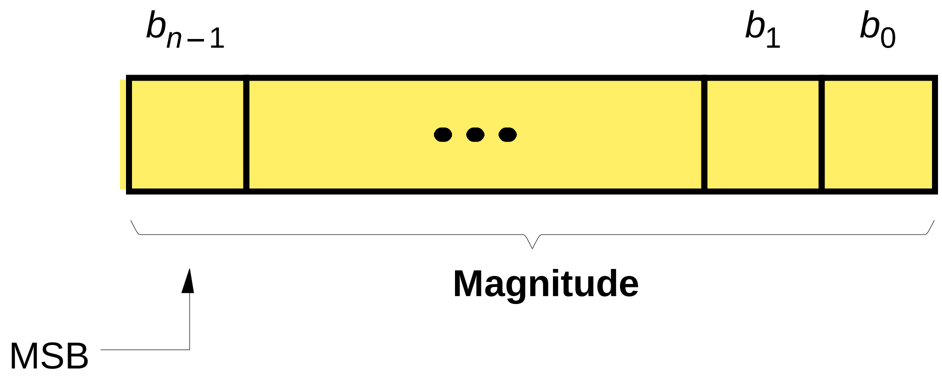
Representation of Signed Numbers

- Positive number representation same in most systems
- Major differences are in how negative numbers are represented
- Three major schemes:
  - sign and magnitude
  - ones complement
  - twos complement

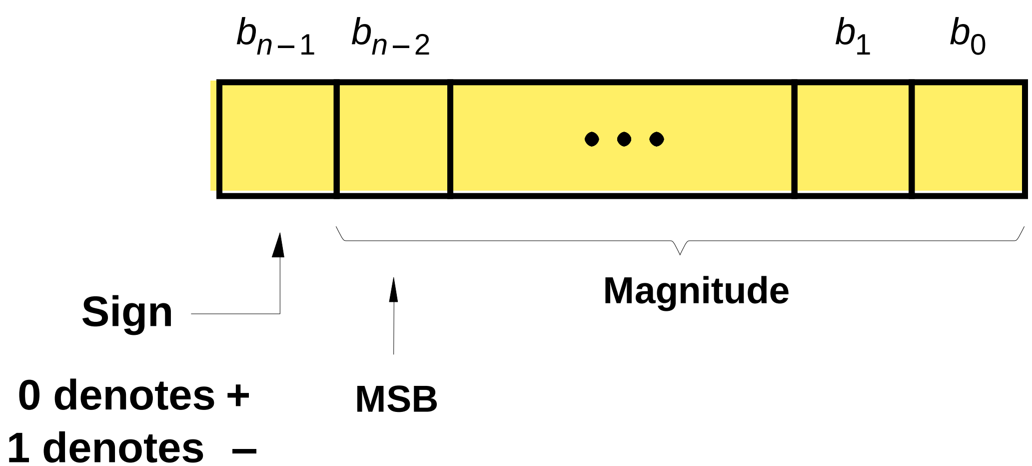
Negative Number Representation

- Assumptions:
  - we'll assume a 4 bit machine word
  - 16 different values can be represented
  - roughly half are positive, half are negative
  - sign bit is the MSB; 0 = plus, 1 = minus

Representation of Negative Numbers

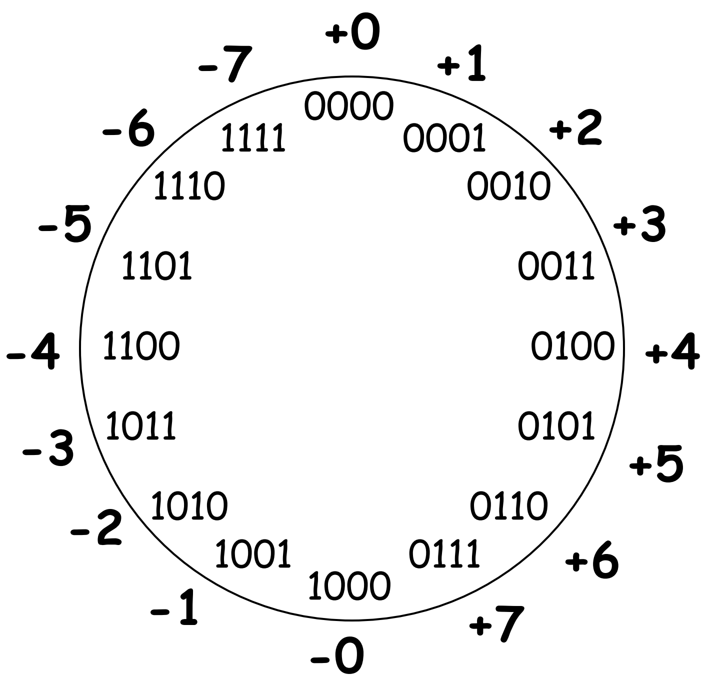


(a) Unsigned number



(b) Signed number

Sign-Magnitude Representation



High order bit is sign: 0 = positive (or zero), 1 = negative

Three low order bits is the magnitude: 0 (000) thru 7 (111)

Number range for n bits = $\pm 2^{n-1} - 1$

Two representations for 0

0 100 = +4
1 100 = - 4

The major disadvantage is that we need separate circuits to both add and subtract

Number magnitudes need to be compared to get the right result

Subtraction of Numbers using Ones Complement

Example :

Subtract 1 from 1000 (binary sys.)

$$\begin{array}{r} 1000\ (8) \\ -\quad 1\ (1) \\ \hline \end{array}$$

1- Complete the subtrahend by zeros to be the same digits like the minuend

1 → 0001

2 – Get the ones complement for the subtrahend

0001 → 1110

3 – Add the ones complement to minuend

$$\begin{array}{r} 1000 \\ +\ 1110 \\ \hline 10110 \end{array}$$

If there is a carry then add 1 and neglect the carry …then 10110 → 0111(7)

If there is no carry then get the 1's complement and add (-) it will be negative

Subtraction of Numbers using Ones Complement

Another Example :

1010100 (84)
-
1000011 (67)

____

- Get the ones complement for the subtrahend
  1000011 → 0111100

- Add the ones complement to minuend

$$
\begin{array}{r}
1\,0\,1\,0\,1\,0\,0 \\
+\ \ 0\,1\,1\,1\,1\,0\,0 \\
\hline
1\ \ 0\,0\,1\,0\,0\,0\,0
\end{array}
$$

- If there is a carry then add 1 and neglect the carry ….
- then 001000 → 001001 (17)

Subtraction of Numbers using Ones Complement

Another Example :

010011 (19)
-
010101 (21)

– Get the ones complement for the subtrahend

010101 → 101010

– Add the ones complement to minuend

$$\begin{array}{r} 010011 \\ + \\ 101010 \\ \hline 111101 \end{array}$$

- No carry … then get the 1's complement and put (-)

111101 → 000010 → - 000010 (- 2)

Subtraction of Numbers using Ones Complement

Another Example :

  1000011 (67)
-
  1010100 (84)
  ____

- Get the ones complement for the subtrahend
  1010100 → 0101011

- Add the ones complement to minuend

    1000011
  +
    0101011
    ____
    1101110

- No carry … then get the 1's complement and put (-)
  1101110 → 0010001 → - 0010001 (- 17)

Twos Complement

- Most common scheme of representing negative numbers in computers
- Affords natural arithmetic (no special rules!)
- To represent a negative number in 2's complement notation…
  1. Decide upon the number of bits (*n*)
  2. Find the binary representation of the +ve value in *n*-bits
  3. Flip all the bits (change 1's to 0's and vice versa)
  4. Add 1

Shortcut …start from right put each zero the same and the first one then change each zero by one and each one by zero

0110100 → 1001100

Twos Complement Example

- Represent -5 in binary using 2's complement notation
  1. Decide on the number of bits

     6 (for example)

  2. Find the binary representation of the +ve value in 6 bits

     000101

     +5

  3. Flip all the bits

     111010

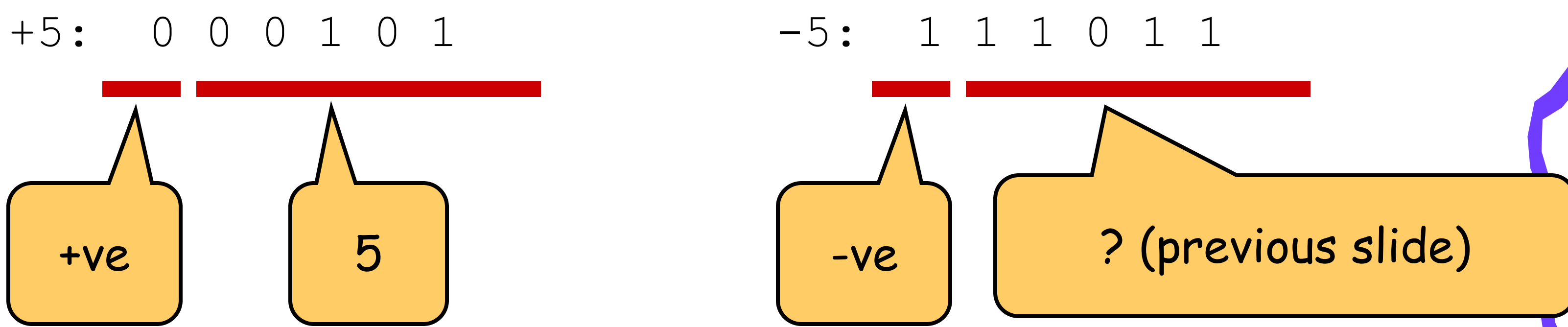
  4. Add 1

```
  111010
+      1
--------
  111011
```

-5

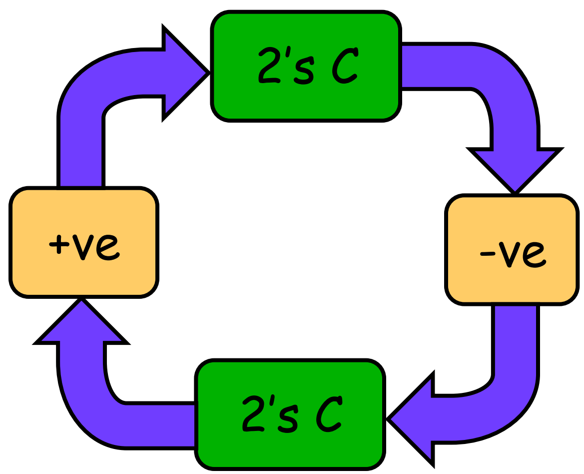
Sign Bit

- In 2's complement notation, the MSB is the sign bit (as with sign-magnitude notation)
  - 0 = positive value
  - 1 = negative value

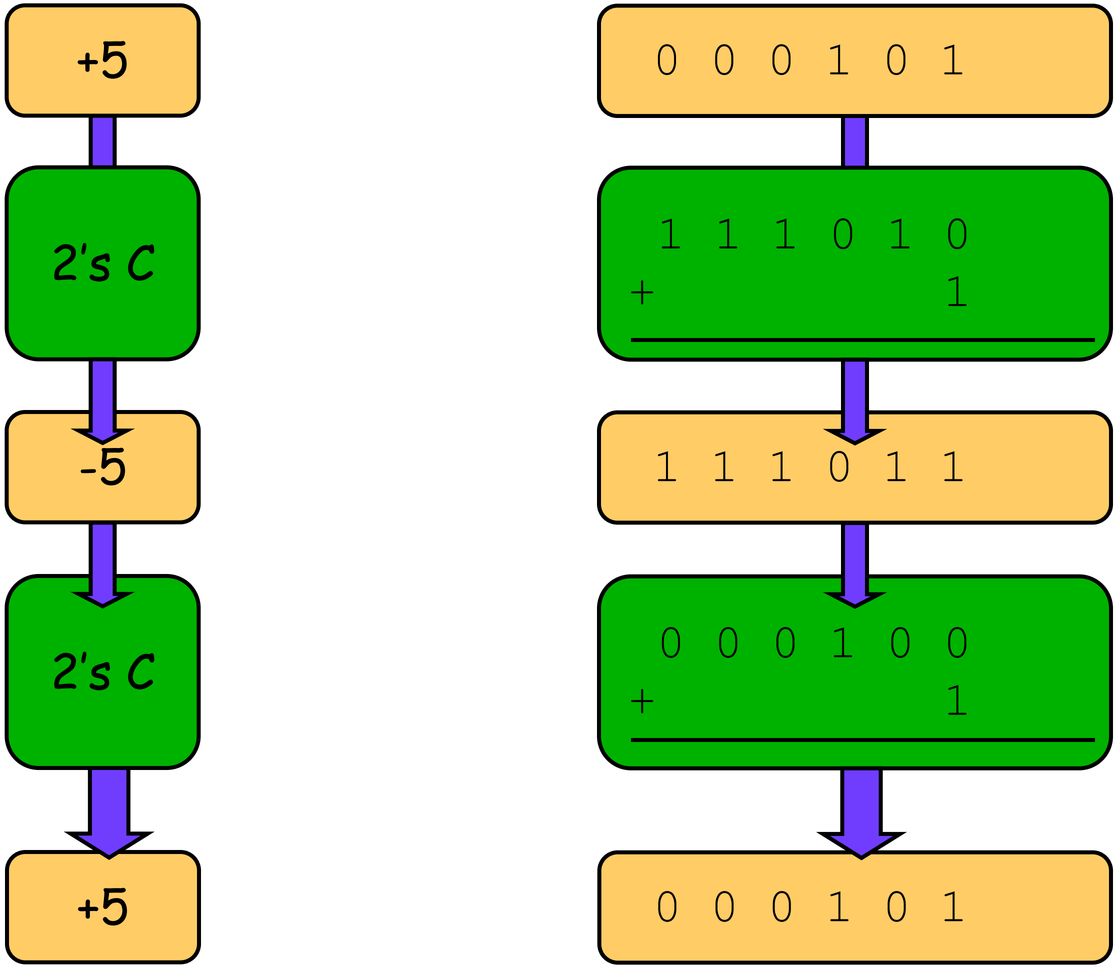


"Complementary" Notation

- Conversions between positive and negative numbers are easy
- For binary (base 2)…



Example



Exercise – 2's C conversions

- What is -20 expressed as an 8-bit binary number in 2's complement notation?
  - Answer:________
- 1100011 is a 7-bit binary number in 2's complement notation. What is the decimal value?
  - Answer:________

Skip answer

Answer

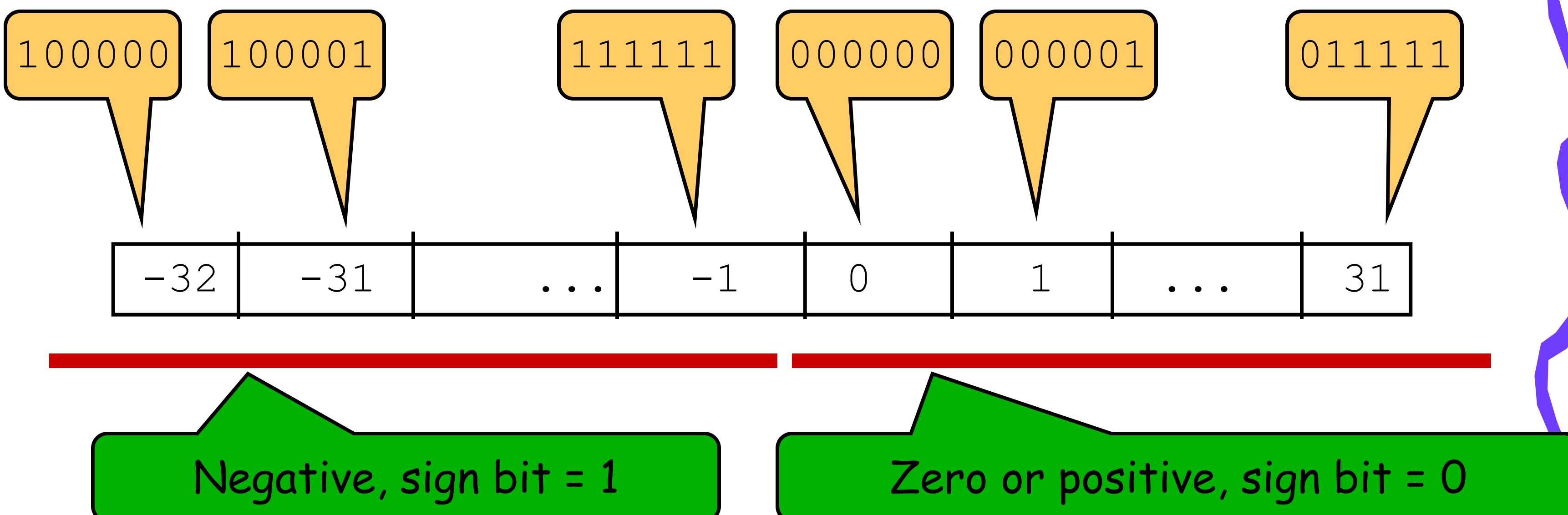
Exercise – 2's C conversions

Answer

- What is -20 expressed as an 8-bit binary number in 2's complement notation?
  - Answer: 11101100
- 1100011 is a 7-bit binary number in 2's complement notation. What is the decimal value?
  - Answer: -29

Range for 2's Complement

- For example, 6-bit 2's complement notation



One’s Complement subtraction

01010 (10)
-
00111 (7)
One’s complement for subtrahend then add
 01010
 +
 11000
____
100010
Neglect carry and add 1
 00011 (+3)

Two’s Complement subtraction
01010 (10)
-
00111 (7)
Two’s complement for subtrahend then add
 01010
 +
 11001
____
100011
Neglect carry
 00011 (+3)

One’s Complement subtraction

00111 (7)

-

01010 (10)

One’s complement for subtrahend then add

00111

+

10101

____

11100

N0 carry and get one’s complement again…put -

-00011 (-3)

Two’s Complement subtraction

00111 (7)

-

01010 (10)

Two’s complement for subtrahend then add

00111

+

10110

____

11101

No carry Two’s complement again

- 00011 (-3)

Ranges (revisited)

No. of bits	Binary					
	Unsigned		Sign-magnitude		2's complement	
	Min	Max	Min	Max	Min	Max
1	0	1				
2	0	3	-1	1	-2	1
3	0	7	-3	3	-4	3
4	0	15	-7	7	-8	7
5	0	31	-15	15	-16	15
6	0	63	-31	31	-32	31
Etc.						

In General (revisited)

No. of bits	Binary					
	Unsigned		Sign-magnitude		2's complement	
	Min	Max	Min	Max	Min	Max
n	0	$2^n - 1$	$-(2^{n-1} - 1)$	$2^{n-1}-1$	-2^{n-1}	$2^{n-1} - 1$

2's Complement Addition

- Easy
- No special rules
- Just add

What is -5 plus +5?

- Zero, of course, but let's see

Sign-magnitude

```
-5:    10000101
+5:   +00000101
      ---------
       10001010
```

Twos-complement

```
        1 1 1 1 1 1 1 1
-5:    11111011
+5:   +00000101
      ---------
       00000000
```

2's Complement Subtraction

- Easy
- No special rules
- Just subtract, well … actually … just add!

$$A - B = A + (-B)$$

add

2's complement of B

What is 10 subtract 3?

- 7, of course, but…
- Let's do it (we'll use 6-bit values)

10 − 3 = 10 + (−3) = 7

```
  +3: 000011

1s C: 111100
  +1:      1
  -3: 111101
```

```
 001010
+111101
 000111
```

What is 10 subtract -3?

(-(-3)) = 3

- 13, of course, but…
- Let's do it (we'll use 6-bit values)

```
10 - (-3) = 10 + (-(-3)) = 13
```

```
  -3: 111101

1s C: 000010
  +1:      1
  +3: 000011
```

```
 001010
+000011
 001101
```

Thank You