Computer Organization & Architecture

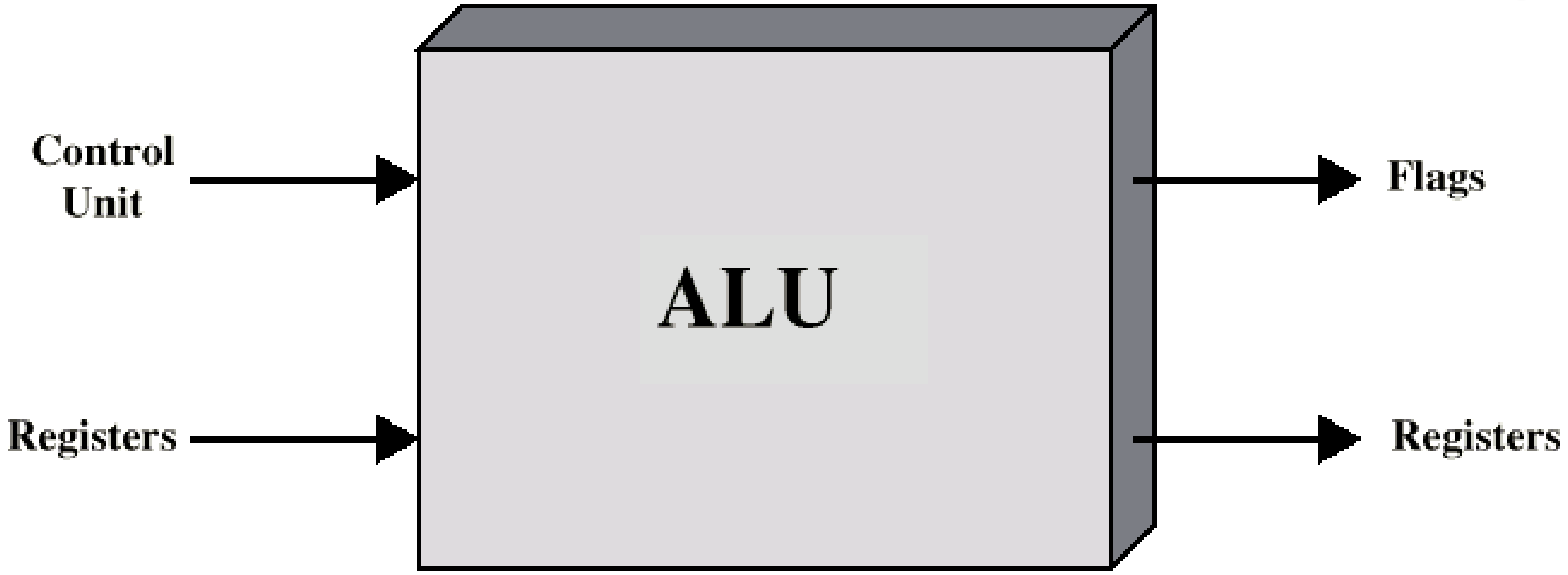
Agenda

- Number Systems
- Conversion Among Number Systems
- Signed Number Representation

Arithmetic & Logic Unit

- Does the calculations
- Everything else in the computer is there to service this unit
- Handles integers
- May handle floating point (real) numbers
- May be separate (math co-processor)

ALU Inputs and Outputs



Number Systems

- Decimal
- Binary
- Octal
- Hexadecimal

Conversion Among Number Systems

- Binary/Octal/Hexadecimal → Decimal

Conversion Among Number Systems…cont.

- Decimal/Octal/Hexadecimal → Binary

Conversion Among Number Systems…cont.

- Binary → Octal/Hexadecimal

Conversion Among Number Systems…cont.

- Octal/Hexadecimal → Hexadecimal/Octal

Number Representation

- **Numbers**
  - Integer (Fixed Point) Numbers
    - Unsigned
    - Signed
  - Real (Floating Point) Numbers
    - Unsigned
    - Signed

Unsigned Number Representation

- If the word size is n bits, the range of numbers that can be represented is from 0 to $(Base)^n - 1$
- For binary number system it form is from 0 to $(2)^n - 1$

Word size	Range for numbers
4	0, 1, 2, 3
8	0, 1, 2, 3,…,7
16	0, 1, 2, 3,…,15
32	0, 1, 2, 3,…,31

Signed Number Representation

- Negative numbers are represented as
  - One’s complement
  - Two’s complement
  - Sign-Magnitude

Sign-Magnitude

- Left most bit is sign bit
- 0 means positive
- 1 means negative
- +18 = 00010010
- -18 = 10010010

Problems:

- Need to consider both sign and magnitude in arithmetic
- Two representations of zero (+0 and -0)

Two’s Complement

- +3 = 00000011
- +2 = 00000010
- +1 = 00000001
- +0 = 00000000

-3 = 11111101
-2 = 11111110
-1 = 11111111
-0 = 00000000

Range of Two’s Complement Numbers

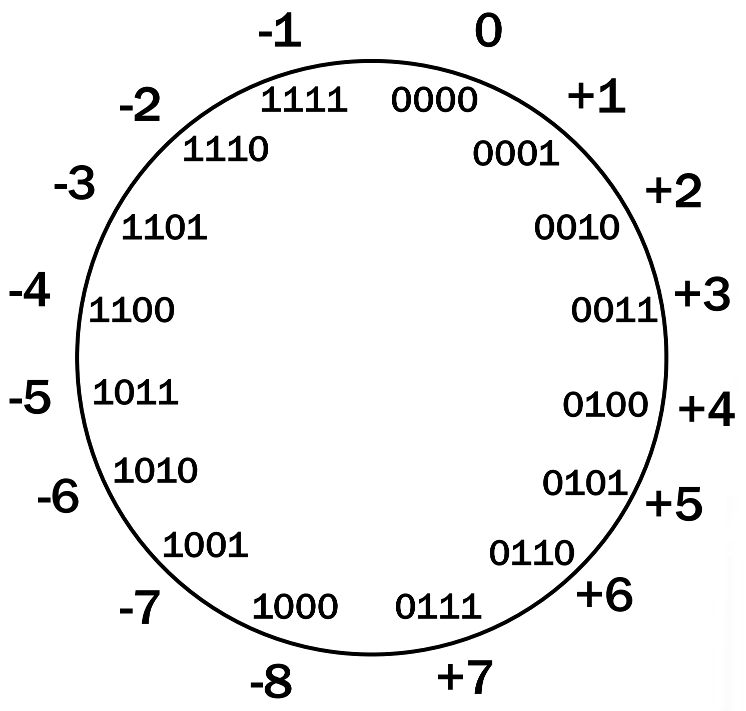
- If the word size is n bits, the range of numbers that can be represented is from $-(2^{n-1})$ to $+(2^{n-1}-1)$. A table of word size and the range of 2's complement numbers that can be represented is shown here:

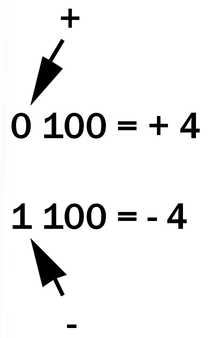
Word size	Range for 2’s complement numbers
4	-8 to +7
8	-128 to +127
16	-32768 to +32767
32	-2147483648 to +2147483647±2 × 10^{+9} (approx.)

2's complement

Only one representation for 0

One more negative number than positive numbers





Range of Numbers

- 8 bit 2's complement
  - +127 = 01111111 = 2^7 -1
  - -128 = 10000000 = -2^7
- 16 bit 2's complement
  - +32767 = 011111111 11111111 = 2^{15} - 1
  - -32768 = 100000000 00000000 = -2^{15}

Two's Complement

If a number is represented in n = 8-bits

Value in Binary:

a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Value in Decimal:

$$2^7.a_7 + 2^6.a_6 + 2^5.a_5 + 2^4.a_4 + 2^3.a_3 + 2^2.a_2 + 2^1.a_1 + 2^0.a_0$$

- +3 = 00000011
- +2 = 00000010
- +1 = 00000001
- +0 = 00000000

-3 = 11111101

-2 = 11111110

-1 = 11111111

-0 = 00000000

Conversion Between Lengths

- Positive number pack with leading zeros
- +18 = 00010010
- +18 = 00000000 00010010
- Negative numbers pack with leading ones
- -18 = 10010010
- -18 = 11111111 10010010
- i.e. pack with MSB (sign bit)

Benefits

- One representation of zero
- Arithmetic works easily (see later)
- Negating is fairly easy
  - 3 = 00000011
  - Boolean complement gives 11111100
  - Add 1 to LSB 11111101

Negation Special Case 1

- 0 = 00000000
- Bitwise NOT 11111111
- Add 1 to LSB +1
- Result 1 00000000
- Overflow is ignored, so:
- - 0 = 0 √

Negation Special Case 2

- -128 = 10000000
- bitwise NOT 01111111
- Add 1 to LSB +1
- Result 10000000
- So:
- -(-128) = -128 X
- Monitor MSB (sign bit)
- It should change during negation