CSCI 112
Introduction to computer Science -I

Instructor: Santanu Banerjee

Unsigned and Signed Integers

Standard Number Lengths

Storage Type	#bits	Capacity
Byte	8	256
Word	16	65536
Doubleword	32	4294967296
Quadword	64	18446744073709551616

Negative numbers

How to represent negative numbers?

- Signed magnitude
- One's complement
- Two's complement
- Excess (biased)

Signed magnitude

- +ve
  - All positions can be used with respective place values
  - 00110110 = ?
- Signed Representation
  - Most significant bit is reserved, 1 for –ve and (0 for +ve)
  - All other bit positions can be used with respective place values
  - 10110110 = ?

One's complement form

- Negate by flipping each bit
- Example (8-bits):

37 = 0010 0101

-37 = 1101 1010

Two's complement form

- Negate by flipping each bit *and adding 1*
- Example (8-bits):

37 = 0010 0101

1101 1010

\+ 1

-37 = 1101 1011

Excess(biased) form

- Add bias to two's complement
- Example (8-bit excess 128):

37 = 0010 0101

1101 1010 FLIP

\+ 1

1101 1011

+1000 0000

0101 1011 = -37

Convert and Compare format

- Signed Magnitude

  $123_{10} = 64 + 32 + 16 + 8 + 2 + 1 = 0111\ 1011_2$

  $-123_{10} \Rightarrow 1111\ 1011_2$

- One's Complement (flip the bits)

  $-123_{10} \Rightarrow 1000\ 0100_2$

- Two's Complement (add 1 to one's complement)

  $-123_{10} \Rightarrow 1000\ 0101_2$

- Excess 128 (add 128 to two's complement)

  $-123_{10} \Rightarrow 0000\ 0101_2$

Picking a format

Points for considerations

- Check for negative numbers?
- Test if a number is zero?
- Add & subtract positive & negative numbers?
- Determine if an overflow has occurred?
- Check if one number is larger than another?

Implemented in hardware: simpler => better

Representing 3 bit signed integer

Decimal	Umsigned	Signed Mag	1's Comp	2's Comp	Excess 4
7	111				
6	110				
5	101				
4	100				
3	011	011	011	011	111
2	010	010	010	010	110
1	001	001	001	001	101
0	000	000/100	000/111	000	100
-1		101	110	111	011
-2		110	101	110	010
-3		111	100	101	001
-4				100	000

Binary addition

- Simple, just a few rules
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 10

Binary addition (Signed Mag)

- Add two positives: 75 , 46

```
      111
  0   1001011
+ 0   0101110
-------------
  0   1111001
```

Binary addition (Signed Mag)

- Add two positives : 107 , 46

```
        1  111
  0   1101011
 +0   0101110
 ------------
  0   0011001   ➔ 25 (?)
```

- No room for carry => overflow

Binary addition (Signed Mag)

- Add two negatives :-46, -25

```
      111
  1  0101110
 +1  0011001
 ------------
  1  1000111
```

- Signs are same, just add the numbers

Binary addition (Signed Mag)

- Add mixed numbers: 46, -25

```
        02   02
   0  0101101
  +1  0011001
 ------------
   0  0010101
```

- Sign of the larger number becomes the sign of the result. Subtraction.

Binary addition (Signed Mag)

- Observations
- Signed magnitude representation is easy for people to understand, but it requires complicated computer hardware.
- Another disadvantage of signed magnitude is that it allows two different representations for zero: positive zero and negative zero.
- For these reasons (among others) computers systems employ complement systems for numeric value representation.

X86 -Unsigned Representation

- Just binary in one of the standard lengths
- E47A is the word-length unsigned representation for the decimal number 58490.

X86-Signed Representation

- 2's complement representation used in 80x86
- One of the standard lengths
- High-order (leading) bit gives sign
  - 0 for positive
  - 1 for negative
- For a negative number, you must perform the 2's complement operation to find the corresponding positive number.

2's Complement Operation

- +/- button on many calculators
- Manually by subtracting from 100…0
- E47A represents a negative word-length signed number since E = 1110

minus

- 10000 – E47A = 1B86 = 7046_{10}
  so E47A is the 2's complement signed representation for –7046

Multiple Interpretations

- One pattern of bits can have many different interpretations.
- The word FE89 can be interpreted as
  - An unsigned number whose decimal value is 65161
  - A signed number whose decimal value is –375