Numerical Representations On The Computer: Negative And Rational Numbers

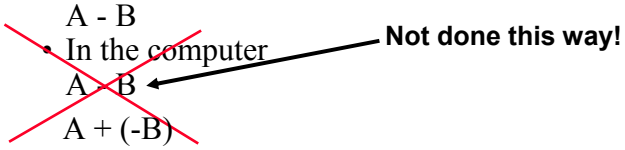
- **How are negative and rational numbers represented on the computer?**
- **How are subtractions performed by the computer?**

Subtraction

- In the real world
  A - B
- In the computer
  A - B

Subtraction

- In the real world
  A - B
- In the computer
  A - B
  A + (-B)



Representing Negative Numbers

- Real world
  - Negative numbers – same as the case of positive numbers but precede the number with a negative sign "-" e.g., -123456.
- Computer world
  - Negative numbers – employ *signed representations*.

representing negative and real numbers on the

Magnitude Of Non-Computer Representations

- All of the digits are used to represent the magnitude of the number
  - e.g., 175_{10}, 1001_2
- An explicit minus sign is needed to distinguish positive and negative numbers
  - e.g., 124_{10} vs. -124_{10} or 100_2 vs. -100_2

James Tam

Signed Binary: Magnitude

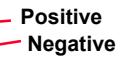
- One bit (most significant bit/MSB or the signed bit) is used to indicate the sign of the number.
- This bit cannot be used to represent the magnitude of the number
- If the MSB equals 0, then the number is positive
  - e.g. 0 bbb is a positive number (bbb stands for a binary number)
- If the MSB equals 1, then the number is negative
  - e.g. 1 bbb is a negative number (bbb stands for a binary number)
- Types of signed representations
  - One's complement
  - Two's complement

James Tam

representing negative and real numbers on the

Signed Binary: Magnitude

- One bit (most significant bit/MSB or the signed bit) is used to indicate the sign of the number.
- This bit cannot be used to represent the magnitude of the number
- If the MSB equals 0, then the number is positive
  - e.g. 0 bbb is a positive number (bbb stands for a binary number)
- If the MSB equals 1, then the number is negative
  - e.g. 1 bbb is a negative number (bbb stands for a binary number)
- Types of signed representations
  - One's complement
  - Two's complement



James Tam

Binary Subtraction

- Requires the complementing of a binary number
  - i.e., A – B becomes A + (-B)
- The complementing can be performed by representing the negative number as a One's or Two's complement value.

James Tam

presenting negative and real numbers on the

Complementing Regular Binary Using The One's Complement Representation

- For positive values there is no difference (no change is needed)
  - e.g., positive seven
    0111 (regular binary)
    0111 (1's complement equivalent)
- For negative values complement the number by negating the binary values: reversing (flipping) the bits (i.e., a 0 becomes 1 and 1 becomes 0).
  - e.g., minus six
    -0110 (regular binary)
    1001 (1's complement equivalent)

James Tam

Complementing Regular Binary Using The Two's Complement Representation

- For positive values there is no difference (no change is needed)
  - e.g., positive seven
    0111 (regular binary)
    0111 (2's complement equivalent)
- For negative values complement the number by negating the number: reversing (flipping) the bits (i.e., a 0 becomes 1 and 1 becomes 0) *and adding one to the result*.
  - e.g., minus six
    -0110 (regular binary)
    1010 (2's complement equivalent)

James Tam

presenting negative and real numbers on the

Interpreting The Pattern Of Bits

Bit pattern	Regular binary	1's complement	2's complement
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	-7	-8
1001	9	-6	-7
1010	10	-5	-6
1011	11	-4	-5
1100	12	-3	-4
1101	13	-2	-3
1110	14	-1	-2
1111	15	-0	-1

James Tam

Overflow: Regular Binary

- Occurs when you don't have enough bits to represent a value (wraps –around to zero)

Binary (1 bit)	Value
0	0
1	1
0	0
:	:

Binary (2 bits)	Value
00	0
01	1
10	2
11	3
00	0
:	:

Binary (3 bits)	Value
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7
000	0
:	:

James Tam

presenting negative and real numbers on the

Overflow: Signed

- In all cases it occurs do to a "shortage of bits"
- Subtraction – subtracting two negative numbers results in a positive number.

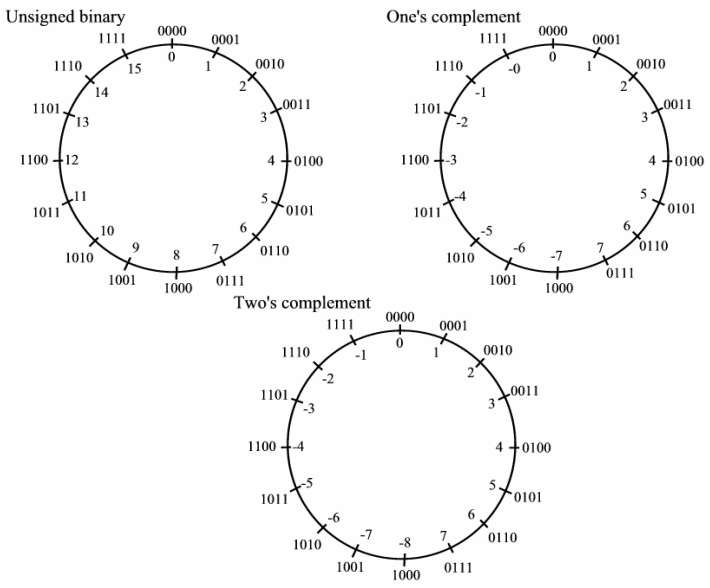
  e.g. - 7
  - 1
  + 7

- Addition – adding two positive numbers results in a negative number.

  e.g. 7
  + 1
  - 8

James Tam

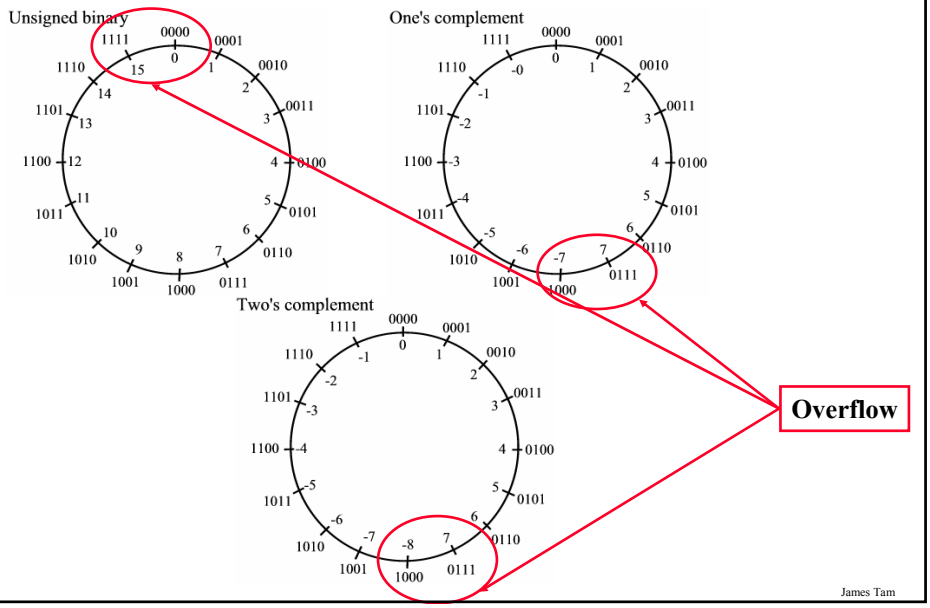
Summary Diagram Of The 3 Binary Representations



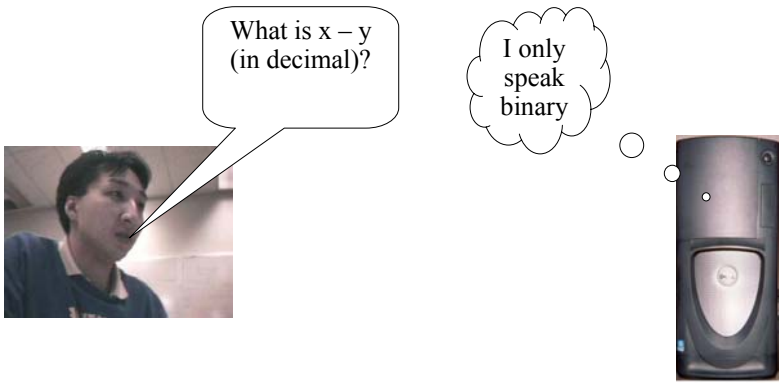
James Tam

presenting negative and real numbers on the

Summary Diagram Of The 3 Binary Representations

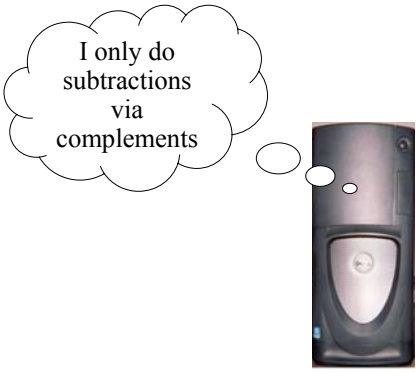


Binary Subtraction Via Complement And Add: A High-Level View



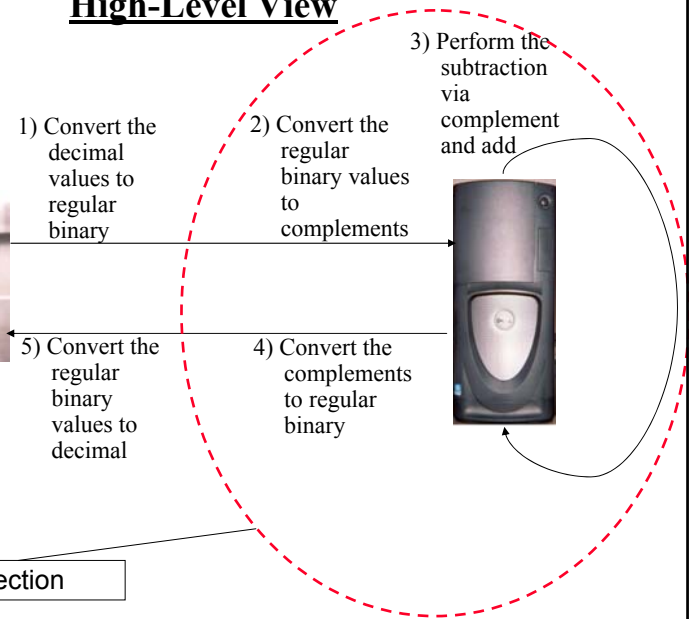
presenting negative and real numbers on the

Binary Subtraction Via Complement And Add: A High-Level View



James Tam

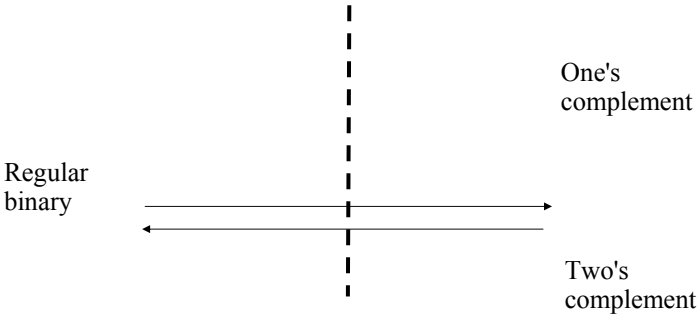
Binary Subtraction Via Complement And Add: A High-Level View



James Tam

presenting negative and real numbers on the

Crossing The Boundary Between Regular And Signed Binary



Each time that this boundary is crossed (steps 2 & 4 from the previous slide) apply the rule:

1) Positive numbers pass unchanged
2) Negative numbers must be converted (complemented)
   a. One's complement: negate the negative number
   b. Two's complement: negate and add one to the result

James Tam

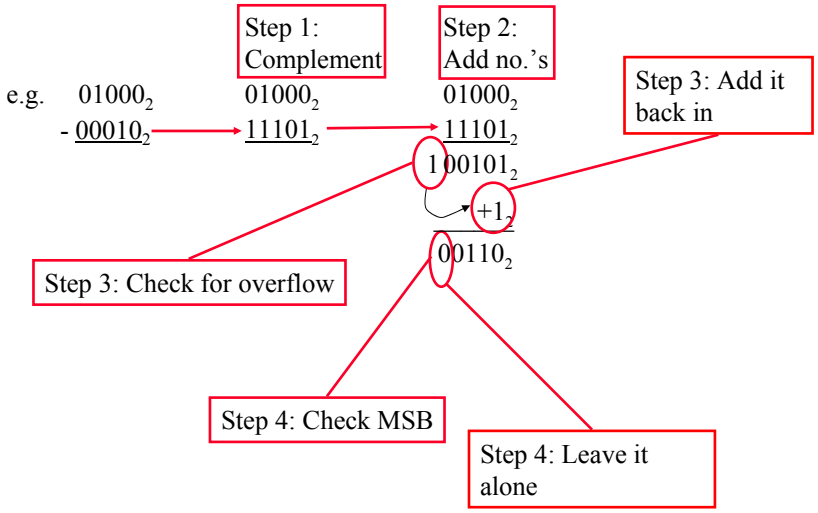
Binary Subtraction Through One's Complements

1) *Convert from regular binary to a 1's complement representation* (check if it is preceded by a minus sign).
   a. If the number is not preceded by a minus sign, it's positive (leave it alone).
   b. If the number is preceded by a minus sign, the number is negative (complement it by flipping the bits) and remove the minus sign.
2) Add the two binary numbers.
3) Check if there is overflow (a bit is carried out) and if so add it back.
4) *Convert the 1's complement value back to regular binary* (check the value of the MSB).
   a. If the MSB = 0, the number is positive (leave it alone)
   b. If the MSB = 1, the number is negative (complement it by flipping the bits) and precede the number with a minus sign

James Tam

presenting negative and real numbers on the

Binary Subtraction Through 1's Complements



James Tam

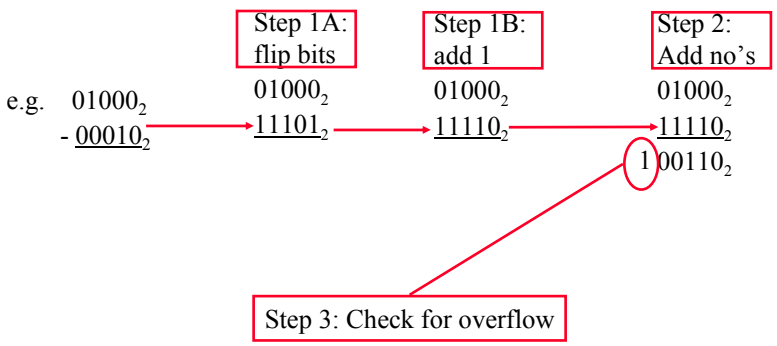
Binary Subtraction Through Two's Complements

1) *Convert from regular binary to a 2's complement representation* (check if it's preceded by a minus sign).
   a. If the number is not preceded by a minus sign, it's positive (leave it alone).
   b. If the number is preceded by a minus sign, the number is negative (complement it and discard the minus sign).
      i. Flip the bits.
      ii. Add one to the result.
2) Add the two binary numbers.
3) Check if there is overflow (a bit is carried out) and if so discard it.
4) *Convert the 2's complement value back to regular binary* (check the value of the MSB).
   a. If the MSB = 0, the number is positive (leave it alone).
   b. If the MSB = 1, the number is negative (complement it and precede the number with a negative sign).
      i. Flip the bits.
      ii. Add one to the result.

James Tam

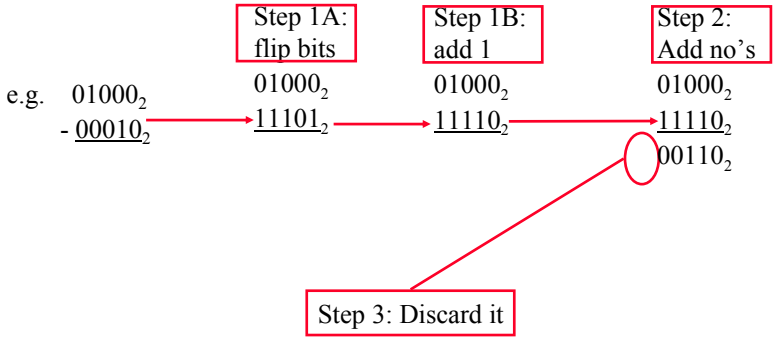
representing negative and real numbers on the

Binary Subtraction Through 2's Complements



James Tam

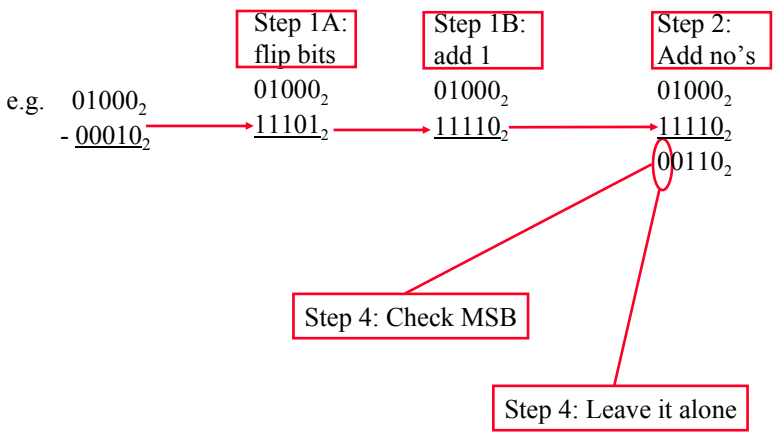
Binary Subtraction Through 2's Complements



James Tam

presenting negative and real numbers on the

Binary Subtraction Through 2's Complements



James Tam

Representing Real Numbers Via Floating Point

- Numbers are represented through a sign bit, a mantissa and an exponent

Sign	Mantissa	Exponent

Examples with 5 digits used to represent the mantissa:
- e.g. One: 123.45 is represented as $12345 * 10^{-2}$
- e.g. Two: 0.12 is represented as $12000 * 10^{-5}$
- e.g. Three: 123456 is represented as $12345 * 10^{1}$

Floating point numbers may result in a loss of accuracy!

James Tam

representing negative and real numbers on the

You Should Now Know

- How negative numbers are represented using 1's and 2's complement representations.
- How to convert regular binary to values into their 1's or 2's complement equivalent.
- What is signed overflow and why does it occur.
- How to perform binary subtractions via the negate and add technique.
- How are real numbers represented through floating point representations