CSD1100

Signed Binary Numbers

Vadim Surov

Signed Binary Number

- When counting, positive numbers can grow indefinitely towards +∞. In a similar way negative numbers can grow indefinitely towards -∞.

  $$-\infty, \ldots, -2 , -1 , 0 , +1 , +2 , \ldots, +\infty$$

- The '-' symbol is used to indicate negative numbers
- The '+' symbol is used to indicate positive numbers. In many cases the '+' symbol is omitted.
- Signed numbers has a single '0' (zero); having no sign.

Signed Binary Number

- The '+' symbol could be eliminated as it could be assumed, but the '-' symbol must still be represented somehow.
- Since computers do not understand anything but numbers, the ‘-' and the '+' symbols must be represented using two specific coded numbers.
- These codes could be represented using any two numbers as long as we understand them as the '-' and '+' signs when they occur at the beginning of any number.
- This means that the two above codes must be eliminated from the list of all possible numbers!

Signed Binary Number

- Over time many schemes to represent negative and positive numbers in computers were proposed.
- Some schemes seem to have been more successful than others.
- In all schemes a fixed number of bits is adopted when representing numbers.
- There are several commonly known negative number representations: **Signed Magnitude**, **Ones' Complement**, **Two's Complement.**
- Since the start of computer development in the early 1950's these methods have been adopted and used at various times.

Signed Magnitude Representation

Signed Magnitude Representation

- This early representation is intuitive since the sign and magnitude (absolute value) of a number are represented separately.
- A 4 bit binary system could represent 16 ($16 = 2^4$) different numbers ranging from 0 (0000_2) to 15 (1111_2):

	Bits	3	2	1	0
0	=	0	0	0	0
.		.	.	.	.
15	=	1	1	1	1

Signed Magnitude Representation

- Bit 3 is called the sign bit:
  - If bit 3 = 0 => number is positive.
  - If bit 3 = 1 => number is negative.

- Since the MSB is reserved for the sign, 3 bits (bits 0 to 2) are left to represent the number.

- The range of binary numbers represented in sign magnitude is: $-2^{n-1} - 1$ to $+ 2^{n-1} - 1$. For example, for 4 bits, the range is: -7 to + 7

Signed Magnitude Representation

- The above table would become:

	Bits	Sign bit	2	1	0
+ 7	=	0	1	1	1
+ 2	=	0	0	1	0
+ 1	=	0	0	0	1
+ 0	=	0	0	0	0
- 0	=	1	0	0	0
- 1	=	1	0	0	1
- 2	=	1	0	1	0
- 7	=	1	1	1	1

Addition/subtraction With Signed Magnitude Representation

- Addition and subtraction in signed binary numbers is not straight forward, why?
- Check the sign
- if (Signs are Equal)
  - Add magnitudes
  - Append the sign to the result
- else //Signs are different
  - Compare the results
    - Subtract the smaller from the larger
    - Append sign of the greater to result

Addition/subtraction With Signed Magnitude Representation

Example 1:

(-3 – 2 = -5) or 1011 + 1010

(010 + 011 = 101)

append sign (1) = 1101

Example 2:

(2 – 5 = -3) or 0010 + 1101

(101 – 010 = 011)

append sign (1) = 1011

Signed Magnitude Representation

- **Advantages**:
  - The method is simple and intuitive for humans to represent as many negative numbers as positive numbers

Signed Magnitude Representation

- **Disadvantages**:
    - The method requires several tests and decisions (such as switching the order of operands) when performing arithmetic.
        - It is harder to implement in computers, costing additional circuitry and execution time.
    - It has 2 representations for zero: +0 and -0!
        - Adding +1 to -1 leads to either 1000 or 0000 both representing 0.
    - Consequently it could be a poor choice for a computer system.

Complement Representation

Complement Representation

- In order to simplify computer arithmetic circuits it would be nice if subtraction is handled the same way as addition without the need to deal with borrowing.
  - 5-3 = 5 + (-3)
  - (-3) should be represented in a certain format.

Complement Representation

- In mathematics, the “method of complements” is a technique used to subtract one number from another using only addition of positive numbers.
- In brief, the number to be subtracted is first converted into its “complement”, and then added to the other number.
- The complement of a number M is a value that together with M makes the whole number.
- The whole number is determined by the base (number of digits) N.
- **Diminished radix complement** or (N-1)’s complement also can be used.

Ones’ Complement Representation

Ones’ Complement Representation

- It’s the diminished radix complement for binary numbers (radix 2).
- The ones' complement of a binary number is defined as the value obtained by inverting all the bits in the binary representation of the number (swapping 0s for 1s and vice versa).
- Positive numbers are still represented as in the sign magnitude method.
- The range of numbers represented when using n bits is –$(2^{n-1} - 1)$ to $(2^{n-1} - 1)$ for 8 bits: -127 to +127

Bits	Unsig. value	Ones' compl.
0111 1111	127	127
...	...	...
0000 0011	3	3
0000 0010	2	2
0000 0001	1	1
0000 0000	0	0
1111 1111	255	−0
1111 1110	254	−1
1111 1101	253	−2
1111 1100	252	−3
...	...	...
1000 0000	128	−127

Ones’ Complement Representation

- Disadvantages:
  - It has 2 representations for zero: +0 and -0! Even though they are the same algebraically. This causes problems when doing tests on arithmetic results.
  - Result of addition of 2 negative numbers must be incremented to get the correct result

- Not a good choice to represent negative numbers.

Ones' Complement Representation

- In 1964 a supercomputer (the CDC 6600) built by Semour Cray was based on ones' complement representation.
- Other computers such as the PDP-1 and UNIVAC 1100/2200 series also used the ones' complement representation for doing arithmetic.
- Nowadays, ones' complement representation is not used in modern computer systems.
- Most computers now use a variation of ones' complement (called two's complement) that eliminates the above problems

Two’s Complement Representation

Two's Complement Representation

- Positive numbers are still represented as in the sign magnitude method.
- Negative numbers are represented as Ones' complement plus 1.
- The range of numbers represented when using n bits is -2^{n-1} to $2^{n-1}-1$ for 8 bits: -128 to +127

Two's Complement Representation

	Bits	*Sign bit*	*2*	*1*	*0*
+ 7	=	0	1	1	1
+ 1	=	0	0	0	1
+ 0	=	0	0	0	0
- 1	=	1	1	1	1
- 8	=	1	0	0	0

Two's Complement Representation. Eliminating the borrowing

- Ex Dec: 778 - 89 = 779 + 999 - 89 +1 - 1000 = 689
- Ex Bin:

```
     1100001011          1100001011
 -      1011001      +   1111111111
 =   1010110010      -      1011001
                     +            1
                     -  10000000000
                     =   1010110010
```

Two’s Complement Representation

- **Advantages:**
  - It has a single representation of the zero: 0
  - Each positive number has a corresponding negative number that starts with a 1, except smallest negative (-8 for 4 bit) which has no corresponding positive number.
  - Simplifies the logic required for addition and subtraction, since can use the addition to add and subtract both negative and positive numbers the same way.
- Nowadays, almost all computer systems are based on the two's complement representation.

Binary Sequence	*2's Complement*	*1's Complement*	*Sign Magnitude*
0111	7	7	7
0110	6	6	6
0101	5	5	5
0100	4	4	4
0011	3	3	3
0010	2	2	2
0001	1	1	1
0000	0	0	0
1111	-1	-0	-7
1110	-2	-1	-6
1101	-3	-2	-5
1100	-4	-3	-4
1011	-5	-4	-3
1010	-6	-5	-2
1001	-7	-6	-1
1000	-8	-7	-0

References

- https://en.wikipedia.org/wiki/Method_of_complements
- https://en.wikipedia.org/wiki/Ones%27_complement