Faculty of Engineering

CSE115: Digital Design

Lecture 3:
Binary System Operations

Suggested Reading

- Sections 2.4-2.7

Binary Addition

Binary addition table:

carry in	X	Y	X+Y	carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Example:

16	8	4	2	1

```
   1 0 1 0 1     21
   0 1 1 0 0     12
  ----------
 1 0 0 0 0 1     33
```

Binary Subtraction

Binary subtraction table:

borrow in	X	Y	X-Y	borrow out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Example:

16	8	4	2	1	
1	1	0	1	0	26
0	1	1	1	1	15
0	1	0	1	1	11

Representation of Negative Numbers in Binary Systems

- Signed-magnitude Representation.
- Two’s-Complement Representation.
- One’s-Complement Representation.

Signed Magnitude Representation

- The MSB represents the sign bit (0 = positive, 1 = negative)
- The range for n-bit is from **$2^{n-1} - 1$** to **$+2^{n-1} - 1$**.
- Example: n=5, Range from -15 to 15
  - 00000 = 0 , 10000 = -0
  - 10011 = -3 , 01100 = +12
- Disadvantages:
  - **Complicated** digital adders
  - Two possible representations of zero

Two's Complement Representation

- The MSB represents the sign bit (0 = positive, 1 = negative)
- To calculate the negative number:
  1. **Complement all bits** of the positive number
  2. **Add 1**
- For n-bit number the decimal value =

$$B = \left(\sum_{i=0}^{n-2} b_i \cdot 2^i \right) - b_{n-1} \cdot 2^{n-1}$$

The range for n-bit is: from -2^{n-1} to $+2^{n-1} - 1$.

Advantages: Addition/subtraction performed directly and only one zero

Disadvantage: One extra negative number (not symmetric)

Two's Complement Example

N = 8: from -128(10000000) to 127 (01111111)

0_{10} = 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 step 1

\+ 1 step 2

0 0 0 0 0 0 0 0 = 0_{10}

Two's Complement Example

64	32	16	8	4	2	1

$+100_{10}$ = 0 1 1 0 0 1 0 0

step 1 — 1 0 0 1 1 0 1 1

step 2 — + 1

1 0 0 1 1 1 0 0 = -100_{10}

One's Complement Representation

- The MSB represents the sign bit (0 = positive, 1 = negative)
- To calculate the negative number, complement all bits of the positive number
- For n-bit number the decimal value =

$$B = \left(\sum_{i=0}^{n-2} b_i \cdot 2^i \right) - b_{n-1} \cdot (2^{n-1} - 1)$$

The range for n-bit is: from $2^{n-1} – 1$ to $+2^{n-1} – 1$.

<u>Advantages</u>: Symmetry, ease of complementation.

<u>Disadvantages</u>:

- ❑Two possible representations of zero.
- ❑Complicated digital adders.

One's-Complement Example

N = 8: form **-127**(10000000) to **127** (01111111)

$+100_{10}$ = 0 1 1 0 0 1 0 0

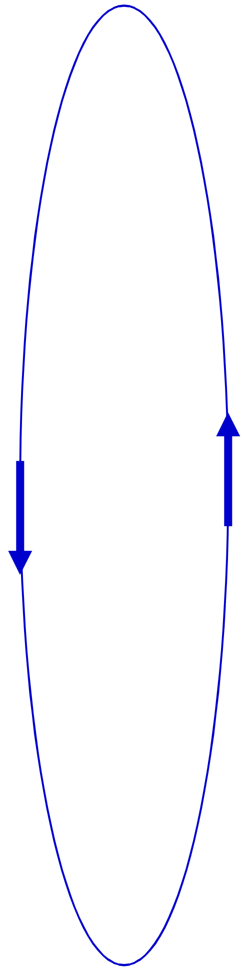
-100_{10} = 1 0 0 1 1 0 1 1

0_{10} = 0 0 0 0 0 0 0 0

-0_{10} = 1 1 1 1 1 1 1 1

Comparison (4-bit)

Decimal	Signed Magnitude	One's Compl.	Two's Compl.
−8	–	–	**1000**
−7	1111	1000	1001
−6	1110	1001	1010
−5	1101	1010	1011
−4	1100	1011	1100
−3	1011	1100	1101
−2	1010	1101	1110
−1	1001	1110	1111
0	0000 or 1000	0000 or 1111	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	**0111**



Exercise

What is the representation of +11, −11 in:

5-bit **signed** magnitude representation

+11 = **0 1 0 1 1** −11 = **1 1 0 1 1**

5-bit **one's** complement representation

+11 = **0 1 0 1 1** −11 = **1 0 1 0 0**

5-bit **two's** complement representation

+11 = **0 1 0 1 1** −11 = **1 0 1 0 1**

Two's Complement Addition (A+B)

1. Use binary addition rules
2. Ignore any carry beyond the sign bit

- If the range is not exceeded, addition result will be correct including the sign bit.

Examples:

```
   (−2)   1 1 1 0
+  (−4)   1 1 0 0
-----------------
  (−6)  1 1 0 1 0
```

```
   (−3)   1 1 0 1
+  (+3)   0 0 1 1
-----------------
  (0)   1 0 0 0 0
```

Addition Overflow

- Overflow detection rule:

- The **sign bit** of the sum is different from the sign bit of the two addends Or,

- The carry in (Cin) and the carry out (Cout) of the sign bit are different

Example:

(4)	0	1	0	0		(−4)		1	1 0 0	
+ (5)	0	1	0	1		+ (−5)		1	0 1 1	
(9)	1	0	1	0	Overflow		1	0	1 1 1	

Two's Complement Subtraction: Method 1 (a-b)

1. Use binary Subtraction rules
2. Ignore any borrow beyond the sign bit

Example:

```
    (2)   0 0 1 0
-   (4)   0 1 0 0
-----------------
   (-2)   1 1 1 0
```

Two's Complement Subtraction Method 2 (a+(-b))

- Add A to the Two's complement of B:

1. Take the One's complement of B
2. Add it to A with initial carry-in i.e. 1

Example: 2-4

```
          1        <- Initial Carry in
     0 0 1 0       <- 2
+    1 0 1 1       <- One's Complement of 4
-----------
(-2) 1 1 1 0
```

Two’s Complement Subtraction

- **Overflow detection** rule:

- The sign bit of the result is different from
  - the sign bit of A and
  - the two’s complement of B

Exercise

- Do the following in 7 bit **two's complement** arithmetic:

```
  (35)   0 1 0 0 0 1 1
+ (42)   0 1 0 1 0 1 0
-----------------------
 (-51)   1 0 0 1 1 0 1
```

Overflow

-12 - 56

```
0111000:  (56)
1000111
+      1
--------
1001000:  (-56)
```

```
  (-12)   1 1 1 0 1 0 0
+ (-56)   1 0 0 1 0 0 0
-----------------------
          0 1 1 1 1 0 0
```