# EEE104 – Digital Electronics (I)

## Lecture 4

Dr. Ming Xu

Dept of Electrical & Electronic Engineering

XJTLU

---

# In This Session

- Signed Numbers.
- Arithmetic Operations with Signed Numbers

---

# Signed Numbers

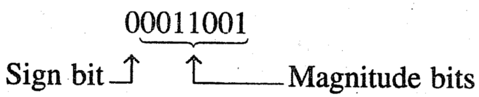
- The left-most bit in a signed binary number is the **sign bit**.
- The sign bit is 0 is for a positive number, and is 1 is for a negative number.
- Digital systems such as computers usually use **2's complement system** to represent signed numbers.
- The 2's complement of a number is calculated by inverting its bits and adding 1.

---

# Signed Numbers

In the 2's complement system

- A positive number is represented as a zero sign bit followed by true binary magnitude bits, e.g. +25 is



- Negative numbers are the 2's complements of the corresponding positive numbers, e.g. -25 is 11100111.
- Positive numbers are the 2's complements of the corresponding negative numbers.

# Signed Numbers

- We can add an infinite number of 0s to the left of a positive number and will not change its value, e.g. 011 (+3) = 00011 (+3).
- We can add an infinite number of 1s to the left of a negative number and will not change its value, e.g. 101 (-3) = 11101 (-3).

# Signed Numbers

The decimal value of signed binary numbers

- It is determined by summing the weights in all bit positions where there are 1s.
- The weight of a 1 sign bit (for a negative number) is calculated as that of a magnitude bit but given a negative value.

# Signed Numbers

The decimal value of signed binary numbers

Positive number

$$\begin{array}{cccccccc} -2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

$$64 + 16 + 4 + 2 = \mathbf{+86}$$

Negative number

$$\begin{array}{cccccccc} -2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

$$-128 + 32 + 8 + 2 = \mathbf{-86}$$

# Signed Numbers

**Range** of signed integer numbers

- The number of different combinations of n bits is

  $$\text{Total combinations} = 2^n$$

  e.g. 8 bits for 256 numbers.
- The range of values for n-bit numbers is

  $$-(2^{n-1}) \quad \text{to} \quad +(2^{n-1} - 1)$$

  e.g. 8 bits for -128 to +127.

# Arithmetic Operations with Signed Numbers

**Addition**

- Add the two numbers and **discard any final carry bit**.

Both numbers positive

$$\begin{array}{r} 00000111 \\ +\ 00000100 \\ \hline 00001011 \end{array} \qquad \begin{array}{r} 7 \\ +\ 4 \\ \hline 11 \end{array}$$

Positive number with magnitude larger than negative number

$$\begin{array}{lr} & 00001111 \\ & +\ 11111010 \\ \hline \text{Discard carry} \longrightarrow & 1\ \ 00001001 \end{array} \qquad \begin{array}{r} 15 \\ +\ -6 \\ \hline 9 \end{array}$$

# Arithmetic Operations with Signed Numbers

**Addition**

Positive number with magnitude larger than negative number

$$\begin{array}{r} 00010000 \\ +\ 11101000 \\ \hline 11111000 \end{array} \qquad \begin{array}{r} 16 \\ +\ -24 \\ \hline -8 \end{array}$$

Both numbers negative

$$\begin{array}{lr} & 11111011 \\ & +\ 11110111 \\ \hline \text{Discard carry} \longrightarrow & 1\ \ 11110010 \end{array} \qquad \begin{array}{r} -5 \\ +\ -9 \\ \hline -14 \end{array}$$

# Arithmetic Operations with Signed Numbers

**Addition**

- **Overflow**: when two numbers are added, the number of bits required to represent the sum exceeds the number of bits in the two numbers.
- It occurs only when both numbers are positive or negative.

$$\begin{array}{r} 01111101 \\ +\ 00111010 \\ \hline \underset{\uparrow}{1}\underbrace{0110111} \end{array} \qquad \begin{array}{r} 125 \\ +\ \ 58 \\ \hline 183 \end{array}$$

Sign incorrect

Magnitude incorrect

# Arithmetic Operations with Signed Numbers

**Subtraction**

- Subtraction is implemented through addition.
- Change the sign of the subtrahend and add to the minuend, e.g. subtracting +6 is equivalent to adding -6.
- The sign of a binary number is changed by taking its 2's complement.

$$\begin{array}{r} \text{minuend} \\ -\quad \text{subtrahend} \\ \hline \text{difference} \end{array}$$

# Arithmetic Operations with Signed Numbers

**Subtraction**

00001000 − 00000011

In this case, 8 − 3 = 8 + (−3) = 5.

| | | |
|---|---|---|
| | 00001000 | Minuend (+8) |
| | + 11111101 | 2's complement of subtrahend (−3) |
| Discard carry → | **1 00000101** | Difference (+5) |

00001100 − 11110111

In this case, 12 − (−9) = 12 + 9 = 21.

| | |
|---|---|
| 00001100 | Minuend (+12) |
| + 00001001 | 2's complement of subtrahend (+9) |
| 00010101 | Difference (+21) |

---

# Arithmetic Operations with Signed Numbers

**Multiplication — Partial Products Method**

1. Compute the magnitude product of corresponding positive numbers.
2. Attach a 0 sign bit. If the signs of the two numbers are different (negative product), take the 2's complement of the outcome.

---

# Arithmetic Operations with Signed Numbers

**Multiplication**

Multiply the signed number 01010011 (+83) and 11000101 (-59).

Step 1

11000101 ⟶ 00111011

Step 2

| | |
|---|---|
| 1010011 | Multiplicand |
| × 0111011 | Multiplier |
| 1010011 | 1st partial product |
| + 1010011 | 2nd partial product |
| 11111001 | Sum of 1st and 2nd |
| + 0000000 | 3rd partial product |
| 011111001 | Sum |
| + 1010011 | 4th partial product |
| 1110010001 | Sum |
| + 1010011 | 5th partial product |
| 100011000001 | Sum |
| + 1010011 | 6th partial product |
| 1001100100001 | Sum |
| + 0000000 | 7th partial product |
| 1001100100001 | Final product |

Step 3

0 1001100100001 (+4897)

↓

**1 0110011011111** (-4897)

---

# Arithmetic Operations with Signed Numbers

**Division** - accomplished using subtraction in computers:

1. Initialize the quotient to zero.
2. Subtract the divisor from the dividend or previous partial remainder. If the partial remainder is:
   - positive, add 1 to the quotient and repeat.
   - zero, add 1 to the quotient and finish.
   - negative, finish.
3. Determine the sign of the quotient by checking the signs of the dividend and divisor.

$$\frac{\text{dividend}}{\text{divisor}} = \text{quotient}$$