UNIVERSITI TEKNOLOGI MALAYSIA

www.utm.my

SCSR1013 DIGITAL LOGIC

MODULE 2c: ARITHMETIC OPERATIONS

FACULTY OF COMPUTING

Part 2: Arithmetic Operations

- Integer Numbers
  - Unsigned Numbers
  - Signed Numbers
- Addition
- Subtraction

Integer Representation

- Numbers can be represented as a combination of a value, or magnitude and sign, plus or minus
- Unsigned integer
- Signed integer

Unsigned Integer Data

- By unsigned integer, it is mean no negative values.
  - E.g. 0, 1, 2, ..., 254, 255, 256, 257, ... 65535, 65536, 65537, ..., 2000000000, 2000000001, ...
- A **bit** can store unsigned integers from 0 to 1 .
- A **byte** of 8 bits can store unsigned integers from 0 to 255 = $2^8 - 1$.

$$00000000_2 - 11111111_2$$

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$ $2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$

$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$

Integer: Unsigned Number

- In binary arithmetic, if the length of the number is restricted to 8 digits (0s and 1s), the largest value is $1111\ 1111_2$ = 255, and the smallest is 0.
- A **word** of 16 bits can store unsigned integers from 0 to 65535 = $2^{16} - 1$.
- In binary arithmetic, if the length of the number is restricted to 16 digits (0s and 1s), the largest value is $1111\ 1111\ 1111\ 1111_2$ = 65535, and the smallest is 0.

Integer: Unsigned Number

The range of number depend on the total number of bits used, *n*.

For positive number yang range is from 0 to 2^n-1.

Example:

Find the range of binary numbers that can be represented by 10 bits.

Number of bits, $n = 10$

$$00\ 0000\ 0000 \leq x \leq 11\ 1111\ 1111$$

$$0 \leq x \leq 2^{10}-1$$

$$0 \leq x \leq 1023$$

Upper and Lower Bound

No of Bits	Lower Bound	Upper Bound, $2^n - 1$	Range
4 bits	0	$2^4 - 1 = 15$	$0 \rightarrow 15$
8 bits	0	$2^8 - 1 = 255$	$0 \rightarrow 255$
10 bits	0	$2^{10} - 1 = 1023$	$0 \rightarrow 1023$

Integer: Unsigned Number

Example:

Find the lower and the upper bound of a 12-bit binary system.

-Lower bound = 0

-Upper bound $=2^n - 1 = 2^{12} - 1 = 4096 - 1 = 4095$

-Therefore the range is 0 → 4095

Signed Numbers

- However, integers can be positive and negative
  - +01000, +11101, -10001, -0111001
  - Need for a code to represent '-' and '+'.
- Positive and negative integers use a code system to indicate the sign.
  - Signed bit: 0 (+ve) or 1 (-ve) positioned at MSB
  - Positive numbers → 0 01000, 0 11101
  - Negative numbers → 1 10101, 1 0101001
  - This is referred as signed numbers.

(+ve) → 0 (-ve) → 1

Example:

Change the following decimal numbers to its binary representation.

i. +4 ii. -12

Example: Determine if the binary numbers is positive or negative.

i. **0** 010001 →

ii. **1** 0011 →

Value in decimal?

+17

- 3

Signed Numbers Representation

- Three representations:
  - Sign and magnitude (simple representation)
  - 1's complement
  - 2's complement

Sign and Magnitude Representation

- Simple and fast.
  - Lower bound: $-(2^{n-1} - 1)$
  - Upper bound: $2^{n-1} - 1$
  - Where *n* the total bit
- Example:

$+01110 = +(01110) = \boxed{0}\ 01110$

$-100100 = -(100100) = \boxed{1}\ 100100$

*Note:
A negative number has the same magnitude bits as the corresponding positive number but the sign bit is 1 rather than a 0.

*Resource: Floyd, Digital Fundamentals, 10th Edition, 2009

Lower bound < decimal < Upper bound

$-(2^{4-1}-1) < \text{decimal} < +(2^{4-1}-1)$

$-(2^{3}-1) < \text{decimal} < +(2^{3}-1)$

$-(8-1) < \text{decimal} < +(8-1)$

$-7 < \text{decimal} < +7$

Example: Integer 4 bits

Positive

Decimal	Binary	Sign & Mag
+7	+111	0 111
+6	+110	0 110
+5	+101	0 101
+4	+100	0 100
+3	+011	0 011
+2	+010	0 010
+1	+001	0 001
+0	+000	0 000

Negative

Decimal	Binary	Sign & Mag
-1	-001	1 001
-2	-010	1 010
-3	-011	1 011
-4	-100	1 100
-5	-101	1 101
-6	-110	1 110
-7	-111	1 111

Example:

Express the decimal number +25 and -25 as an 8-bit signed binary number in the sign-magnitude forms.

Solution: +25

= 1 1 0 0 1
= 0 0 0 1 1 0 0 1 (8-bit binary system)

(Sign bit) (Magnitude bits)

-25

= - (+25)
= - (0 0 0 1 1 0 0 1) 8-bit binary system)
= 1 0 0 1 1 0 0 1

Complement of a Number

- In base *B* arithmetic we can compute two complements:
  - *B*'s complement
  - (*B-1*)'s complement
- For binary numbers we can use 2's complement and 1's complement;
- In decimal arithmetic we have 10's complement and 9's complement.

- In general, complements are defined for numbers that have both integer and fractional parts. However in this discussion is restricted to complements of integers and binary numbers only.

1's complement

*Note:
Positive number represent the same way as the positive sign-magnitude numbers.

A negative number is the 1's complement of the corresponding positive number.

- Convert '0' to '1' and '1' to '0' For (-ve)
  - Lower bound: $-(2^{n-1}-1)$
  - Upper bound: $2^{n-1}-1$
  - Where *n* the total bit
- Example:

+01110 = + (01110) = 0 01110

-100100 = - (0100100) = 1011011

assume 7-bits binary system

UTM UNIVERSITI TEKNOLOGI MALAYSIA

www.utm.my

Example: Integer 4 bits

Decimal	Binary	1's Comp
+7	+111	0 111
+6	+110	0 110
+5	+101	0 101
+4	+100	0 100
+3	+011	0 011
+2	+010	0 010
+1	+001	0 001
+0	+000	0 000

Decimal	Binary	1's Comp
-1	-001	1 110
-2	-010	1 101
-3	-011	1 100
-4	-100	1 011
-5	-101	1 010
-6	-110	1 001
-7	-111	1 0 0 0

$$
\begin{aligned}
-7 &= -(+7) \\
&= -(0\ 1\ 1\ 1) \\
&= \ \ 1\ 0\ 0\ 0
\end{aligned}
$$

1's Complement

Example:

Express the decimal number +25 and -25 as an 8-bit signed binary number in the 1's complement forms.

Solution: +25 = 1 1 0 0 1

= (0 0 0 1 1 0 0 1) (8-bit binary system)

(Sign bit) (Magnitude bits)

-25 = - (+25)

= - (0 0 0 1 1 0 0 1) (8-bit binary system)

= 1 1 1 0 0 1 1 0 ← 1's Complement

2's complement

- Process:
  - Convert to 1's complement.
  - Add 1.
  - Lower bound: $-(2^{n-1})$
  - Upper bound: $2^{n-1}-1$
- Example:

+01110 = + (01110) = 0 01110

-100100 = - (0100100) = 1011011 (1's)

= 1011100 (2's)

*Note:
A negative number is the 2's complement of the corresponding positive number.

Example: Integer 4 bits

Decimal	Binary	2's Comp
+7	+111	0 111
+6	+110	0 110
+5	+101	0 101
+4	+100	0 100
+3	+011	0 011
+2	+010	0 010
+1	+001	0 001
+0	+000	0 000

Decimal	Binary	2's Comp
-1	-001	1 111
-2	-010	1 110
-3	-011	1 101
-4	-100	1 100
-5	-101	1 011
-6	-110	1 010
-7	-111	1 0 0 1
-8	-1000	1 000

1's Complement:

-7 = - (+7)
= - (0 1 1 1)
= 1 0 0 0

2's Complement:

```
  1 0 0 0
+       1
---------
  1 0 0 1
```

Example:

Express the decimal number +25 and -25 as an 8-bit signed binary number in the 2's complement forms.

Solution: +25 = 1 1 0 0 1

= 0 0 0 1 1 0 0 1 (8-bit binary system)

$$\underset{\text{(Sign bit)}}{0}\ \underbrace{0\ 0\ 1\ 1\ 0\ 0\ 1}_{\text{(Magnitude bits)}}$$

-25 = - (+25)

= - (0 0 0 1 1 0 0 1)

= 1 1 1 0 0 1 1 0 ← 1's Complement

= 1 1 1 0 0 1 1 1 ← 2's Complement

Example:

Compute -23_{10} to a 7-bit binary system using 1's complement representation.

-(+23) = -(00 10111) = **11** 01000

-23	= - (+23)
	= - (0 0 1 0 1 1 1)
	= 1 1 0 1 0 0 0

(7-bit binary system)

Example:

Compute -23_{10} to a 7-bit binary system using 2's complement representation.

-(+23) = -(**0**0 10111) = **11** 01001

2's Complement:

```
  1 1 0 1 0 0 0
+             1
-----------------
  1 1 0 1 0 0 1
```

Exercise 2c.1:

Determine the decimal value of this signed binary number (10010101) expressed in sign-magnitude.

Exercise 2c.2:

Express the decimal number -39 as an 8-bit number in the sign-magnitude, 1's complement, and 2's complement forms.

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Summarized of signed representation

Decimal	Sign & Magnitude	1's Complement	2's Complement
+7	0 111	0 111	0 111
+6	0 110	0 110	0 110
+5	0 101	0 101	0 101
+4	0 100	0 100	0 100
+3	0 011	0 011	0 011
+2	0 010	0 010	0 010
+1	0 001	0 001	0 001
+0	0 000	0 000	0 000
-1	1 111	1 110	1 111
-2	1 110	1 101	1 110
-3	1 101	1 100	1 101
-4	1 100	1 011	1 100
-5	1 011	1 010	1 011
-6	1 010	1 001	1 010
-7	1 001	1 000	1 001
-8	-	-	1 000

Arithmetic Operation: Addition

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

- Example:

$$\begin{array}{cccccc}
 & 1 & 0 & 0 & 1 & 0 \\
+ & 0 & 1 & 1 & 0 & 0 \\
\hline
 & 1 & 1 & 1 & 1 & 0 \\
\hline
\end{array}$$

$$\begin{array}{cccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & & & \\
 & 1 & 0 & 1 & 1 & 0 \\
+ & 0 & 1 & 1 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & 1 & 0 \\
\hline
\end{array}$$

Example:

Use 8-bit, 2's complement representation for the following operation:

i. 127 + 74

+127 = **0**111 1111

+74 = **0** 1001010

(+127)		0 111 1111
(+74)	+	0 100 1010
		1 100 1001

Result wrong

Example:

Use 8-bit, 2's complement representation for the following operation:

ii. 60 + (-30)

+60 = 0011 1100

-30 = -(+30) = - (0001 1110)

-(+30) →	- (0 0 0 1 1 1 1 0)	
	1 1 1 0 0 0 0 1	(1's)
	1	
	1 1 1 0 0 0 1 0	(2's)

(+60)		0 0 1 1 1 1 0 0
(-30)	+	1 1 1 0 0 0 1 0
(+30)	1	0 0 0 1 1 1 1 0

Carry bit is ignore !

Arithmetic Operation: Subtraction

- In digital system, subtraction is performed by using 2's complement and addition.
- Carry from the MSB (signed bit) is deleted.
- Example:

010011 – 001111 = 010011 + (– 001111)
= 010011 + (110001)
= 000100

	1			1	1	
	0	1	0	0	1	1
+	1	1	0	0	0	1
~~1~~	0	0	0	1	0	0

- (0 0 1 1 1 1)
→ 1 1 0 0 0 0 (1's)
1
1 1 0 0 0 1 (2's)

Example:

Perform the operations below using 6-bit 2's complement signed number.

(a) 24 – 17

-17 = - (+17)
= - (0 1 0 0 0 1) (6-bits)
= 1 0 1 1 1 0 (1's)
1
1 0 1 1 1 1 (2's)

(a) 24 – 17 = 24 + (-17) = +7

24 = + 24 = 0 11000

-17 = - (+ 17) = - (0 10001) = 1 01111

24 – 17 = 24 + (-17) = 0 11000 + 1 01111 = 0 00111 = +7

Example:

Perform the operations below using 6-bit 2's complement signed number.

(b) -9 – 15

(b) -9 – 15 = -9 + (-15) = -24

-9 = - (+9) = - (0 1001) = 1 0111 = 11 0111

-15 = - (+15) = - (0 1111) = 1 0001 = 11 0001

-9 – 15 = -9 + (-15) = 11 0111 + 11 0001 = **101000**

-9 = - (+9)
= - (0 0 1 0 0 1) (6-bits)
= 1 1 0 1 1 0 (1's)
1
1 1 0 1 1 1 (2's)

+

-15 = - (+15)
= - (0 0 1 1 1 1) (6-bits)
= 1 1 0 0 0 0 (1's)
1
1 1 0 0 0 1 (2's)