BCN 1043

COMPUTER ARCHITECTURE & ORGANIZATION

By
Dr. Mritha Ramalingam

Faculty of Computer Systems & Software Engineering
mritha@ump.edu.my

AUTHORS

- **Dr. Mohd Nizam Mohmad Kahar** (mnizam@ump.edu.my)
- **Jamaludin Sallim** (jamal@ump.edu.my)
- **Dr. Syafiq Fauzi Kamarulzaman** (syafiq29@ump.edu.my)
- **Dr. Mritha Ramalingam** (mritha@ump.edu.my)

Faculty of Computer Systems & Software Engineering

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

BCN1043

COMPUTER ARCHITECTURE & ORGANIZATION

Chapter 2 continues…

Number Conversion

Converting from Decimal to Binary

Remainder Method

Example: Convert 26d to base 2

26/2	=	13	0
13/2	=	6	1
6/2	=	3	0
3/2	=	1	1
1/2	=	0	1

=> 26d = 11010b

How about floating point number?

E.g.: Convert 0.875d into base 2 number

0.875	x 2	=	1.75	1	
0.75	x 2	=	1.5	1	
0.5	x 2	=	1.0	1	
0	x 2	=	0	0	=> **0.875d = 0.1110b**

Number Conversion

Converting from Binary to Decimal

101001b to decimal $(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

101001b =

$= 32 + 0 + 8 + 0 + 0 + 1 \quad = 41d$

Converting from Decimal to Hexadecimal

Remainder Method

Example: Convert 425d to base 16

425 / 16	=	26	9	->	9
26 / 16	=	1	10	->	A
1 / 16	=	0	1	->	1

=> 425d = 1A9h

Exercise

1. 345
2. 89
3. 622

Converting from Hexadecimal to Decimal

$A3F_{16}$ to decimal

$$A3F_{16} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$
$$= (10 \times 256) + (3 \times 16) + (15 \times 1)$$
$$= 2623_{10}$$

Exercise

1. 345
2. 2B67
3. EAD

Number Conversion

Converting from Hexadecimal to Binary

To convert a hex number to binary, we need only express each hex digit in binary

E.g.: Convert $DE1_{16}$ to binary

	D	E	1
=	1101	1110	0001
=	110111100001b		

Exercise:

1. 5DAB
2. 63ACE

Number Conversion

Converting from Binary to Hexadecimal

To convert from binary to hex, just reverse this process

E.g.

$10010001_2 = 1001\ 0001 = 91_{16}$

Exercise

1. 110011001
2. 111011100
3. 101010000001100

Number Conversion

Converting from Hexadecimal - Binary - Octal

$$1F0C_{16} = ?_8$$

1 → 0001, F → 1111, 0 → 0000, C → 1100

0001 1111 0000 1100

0|001 | 111|1 00|00 1|100

1 | 7 | 4 | 1 | 4

$$1F0C_{16} = 17414_8$$

Number Conversion

Octal to Decimal

$724_8 \Rightarrow$

$$4 \times 8^0 = 4$$
$$2 \times 8^1 = 16$$
$$7 \times 8^2 = 448$$
$$468_{10}$$

Hexadecimal to Decimal

$ABC_{16} \Rightarrow$

$$C \times 16^0 = 12 \times 1 = 12$$
$$B \times 16^1 = 11 \times 16 = 176$$
$$A \times 16^2 = 10 \times 256 = 2560$$
$$2748_{10}$$

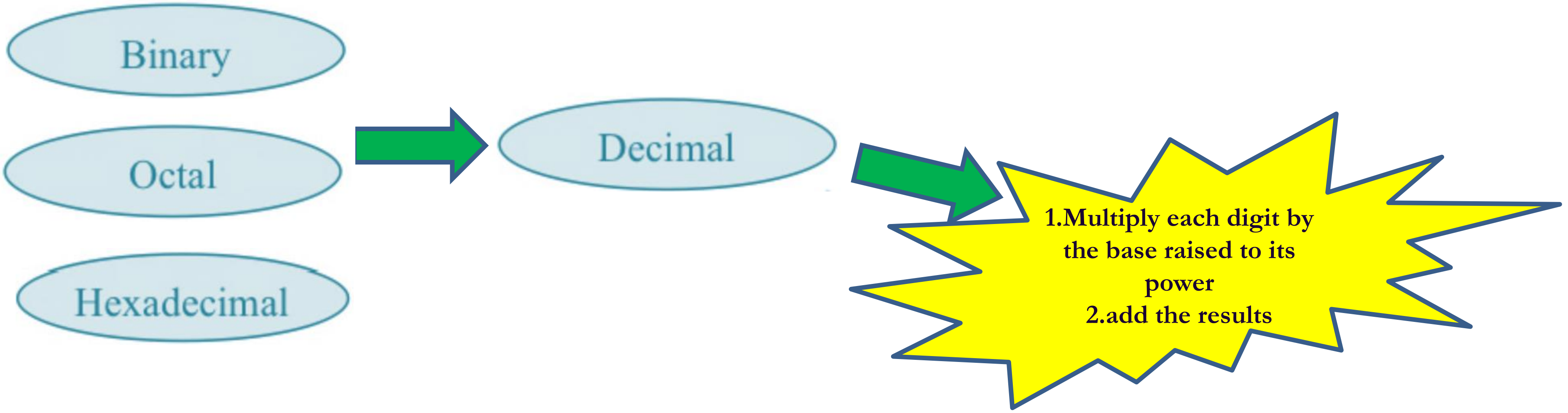
Binary to Decimal

Bit "0"

$101011_2 \Rightarrow$

$$1 \times 2^0 = 1$$
$$1 \times 2^1 = 2$$
$$0 \times 2^2 = 0$$
$$1 \times 2^3 = 8$$
$$0 \times 2^4 = 0$$
$$1 \times 2^5 = 32$$
$$43_{10}$$

Number Conversion



Number Conversion

Decimal to octal

$1234_{10} = ?_8$

8	1234	
8	154	2
8	19	2
8	2	3
	0	2

$1234_{10} = 2322_8$

Decimal to hexadecimal

$1234_{10} = ?_{16}$

16	1234	
16	77	2
16	4	13 = D
	0	4

$1234_{10} = 4D2_{16}$

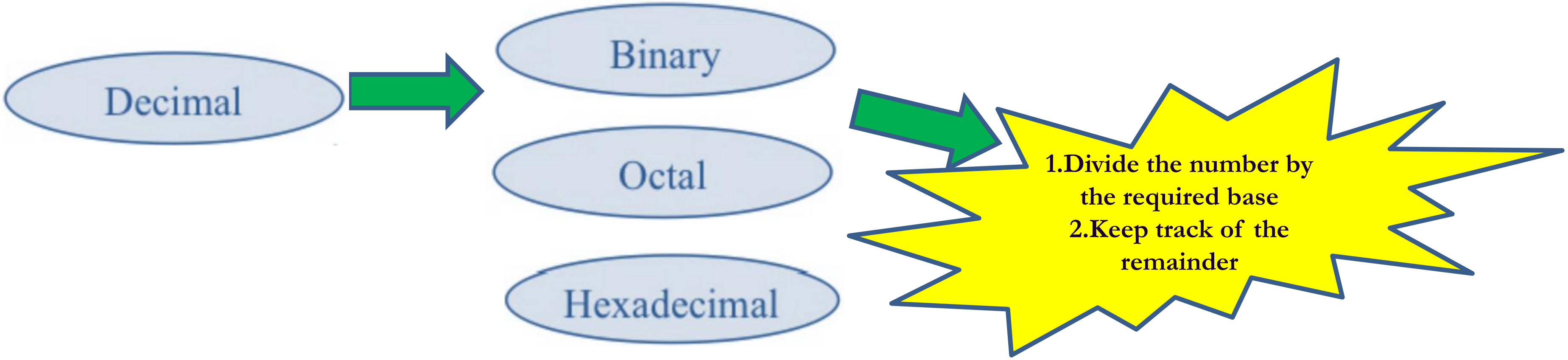
Decimal to Binary

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Number Conversion



Chapter 2

Machine Level Representation of data

- Bits, bytes, and words
- Numeric data representation and number bases
- **Fixed- and floating-point systems**
- Signed and twos-complement representations

Fixed and floating point systems

Number system

Integers

Represents whole numbers

Fractional numbers

Represents numbers with fractions

Fractional numbers

Fixed point **Floating point**

Fixed point systems

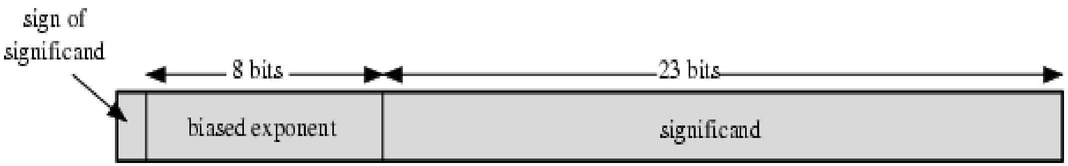
Fixed point

- numbers with fractions
- with fixed points
- E.g.1001.1010 = $2^4 + 2^0 + 2^{-1} + 2^{-3}$

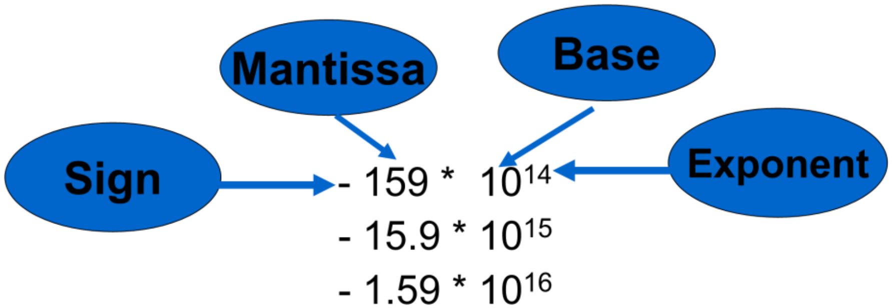
  =9.625

Floating point systems

fractional numbers with floating (movable) points



(a) Format



- This number is stored in a binary word with three fields:
  - Sign: plus or minus
  - Significand S (Mantissa)
  - Exponent E

Chapter 2

Machine Level Representation of data

- Bits, bytes, and words
- Numeric data representation and number bases
- Fixed- and floating-point systems
- **Signed and twos-complement representations**

Signed magnitude Representation

- Left most bit is sign bit representing sign of the number
- 0 means positive
- 1 means negative
- Rest of the bits represent magnitude of the number

0 → Positive Number

Sign ← (left most bit)

Magnitude	Decimal value
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	+0

1 → Negative Number

Sign-magnitude	Decimal value
1111	-7
1110	-6
1101	-5
1100	-4
1011	-3
1010	-2
1001	-1
1000	-0

Signed magnitude Representation

- Examples

  +18 = 00010010

  -18 = 10010010

```
0 10010011 10100010000000000000000 =  1.638125 × 2^20
1 10010011 10100010000000000000000 = -1.638125 × 2^20
0 01101011 10100010000000000000000 =  1.638125 × 2^-20
1 01101011 10100010000000000000000 = -1.638125 × 2^-20
```

1's Complement

- Positive integers are similar to sign-magnitude notation.
- Negative integer is denoted as 1's complement of the positive number in sign-magnitude notation. Ones complement of any number is obtained by complementing each one of the bits,
- i.e., 1 - replaced by 0, and 0 - replaced by 1.
  - 18_{10} = 00010010
  - -18_{10} = 1's complement = 11101101

$n = 4$

$(2n - 1) = 1111$

$\overline{N}$	Decimal value
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	+0

$\overline{N}$	Decimal value
1000	-7
1001	-6
1010	-5
1011	-4
1100	-3
1101	-2
1110	-1
1111	-0

Convert to 1's complement (Q&A)

- Find the 1's complement of:

  - 11111111_2
  - 11111_2
  - 00000000_2
  - 00000_2
  - 10001010_2
  - 11010111_2
  - 11110011_2

Convert to 1's complement (Q&A)

Answer

- 11111111_2: 1's complement = 00000000_2
- 11111_2: 1's complement = 00000_2
- 00000000_2: 1's complement = 11111111_2
- 00000_2: 1's complement = 11111_2
- 10001010_2: 1's complement = 01110101_2
- 11010111_2: 1's complement = 00101000_2
- 11110011_2: 1's complement = 00001100_2

Ones Complement Addition

ex. 1.

```
 +1   0001
 -6   1001
     --------
 -5   1010
```

ex. 2.

```
 +3   0011
 -3   1100
     --------
 -0   1111
```

One’s Complement Addition

- An example of one’s complement integer addition with an end-around carry:

$$
\begin{array}{rl}
1\,0\,0\,1\,1 & (-12)_{10} \\
+\,0\,1\,1\,0\,1 & (+13)_{10} \\
\hline
1\,0\,0\,0\,0\,0 & \\
 & \text{End-around carry} \\
+\qquad\quad 1 & \\
\hline
0\,0\,0\,0\,1 & (+1)_{10}
\end{array}
$$

Ones Complement End Around Carry

```
-2     1101
-4     1011
       --------
    (1)1000
        --> 1
     --------
       1001   = -6
```

carry

Twos compliment Representation

- All positive numbers begin with 0
- All negative numbers begin with 1

1. Perform the 1's complement operation.
2. add 1 to LSB.

Positive (+)		Negative (-)	
Binary Pattern	Decimal Value	Binary Pattern	Decimal Value
		1 0 0 0	-8
0 1 1 1	7	1 0 0 1	-7
0 1 1 0	6	1 0 1 0	-6
0 1 0 1	5	1 0 1 1	-5
0 1 0 0	4	1 1 0 0	-4
0 0 1 1	3	1 1 0 1	-3
0 0 1 0	2	1 1 1 0	-2
0 0 0 1	1	1 1 1 1	-1
0 0 0 0	0		

Twos compliment Representation

Advantages

- One representation of zero
- Negating is fairly easy
  - 3 = 00000011
  - Boolean complement gives 11111100
  - Add 1 to LSB 11111101

Twos compliment Representation

Examples

- +3 = 00000011
- +2 = 00000010
- +1 = 00000001
- +0 = 00000000
- -1 = 11111111
- -2 = 11111110
- -3 = 11111101

Question

	1s compliment	2s complement?
59	00111011	00111011
102	01100110	
65		01000001
−65	10111110	10111111
−125	10000010	10000011
125	01111101	
84		
−83		
83		
91		

Chapter 2 Review
Machine Level Representation of data

- Bits, bytes, and words
- Numeric data representation and number bases
- Fixed- and floating-point systems
- Signed and twos-complement representations

Chapter 2 Ends!