Logic Design - Introduction

Prepared by： Eng.Amani Safwat

What is Logic Design?

Logic Design is a branch of computer engineering that uses logic circuits to process binary data (0s and 1s).

Why do we need logical design?

- To create RAM (CPU): the cores that execute every instruction in a computer.
- To create memory: the place where the ones and zeros are stored.
- To create any digital device: such as phones, controllers, and smart cars.

Number Systems

- Decimal (10) : 0-9 “used by humans”
- Binary (2) :0-1 “used by computers”
- Octal (8) :0-7 “shortcut for binary”
- Hexadecimal (16):0-9 A-F “used in programming”

A = 10 , B = 11 , C =12 , D = 13 , E = 14 , F = 15

Mapping

Binary Systems

- Represented: 0,1

Mapping from binary to decimal:

Weights method :

$2^n = m$ $2^0=1$ $2^1=2$ $2^2=4$ $2^3=8$ $2^4=16$ $2^5=32$

EX:

$(110110)_2 \rightarrow (?)_{10}$

32	16	8	4	2	1
1	1	0	1	1	0

We collect what is above one

32+16+4+2=54

So **$(110110)_2 \rightarrow (54)_{10}$**

Binary Systems

Mapping from binary to Octal :

We divide the number into 3 digits then We collect what is above one

EX:

$(10010)_2$ ->$(?)_8$

0	1	0		0	1	0		
4	2	1		4	2	1		
	2				2			

So **$(10010)_2$ ->$(22)_8$**

Binary Systems

Mapping from binary to Hexadecimal :

We divide the number into 4 digits then We collect what is above one

EX:

$(111010)_2$ ->$(?)_{16}$

0 0 1 1	1 0 1 0
8 4 2 1	8 4 2 1
3	A

So **$(111010)_2$ ->$(3A)_{16}$**

decimal Systems

Mapping from decimal to Binary :

We use the weights method, and the last number is the smallest and closest number to the decimal number. Then we make one for the numbers that are equal to the decimal number, otherwise zero.

EX:

$(55)_{10}$ ->$(?)_2$

32	16	8	4	2	1
1	1	0	1	1	1

So **$(55)_{10}$ ->$(110111)_2$**

decimal Systems

Mapping from decimal to Hexadecimal :

Convert from decimal to binary and from binary to Hexadecimal

EX:

$(210)_{10}$ ->$(?)_{16}$

10 ->2

128	64	32	16	8	4	2	1
1	1	0	1	0	0	1	0

2 ->16

1 1 0 1	0 0 1 0
8 4 2 1	8 4 2 1
D	3

So **$(210)_{10}$ ->$(D3)_{16}$**

decimal Systems

Mapping from decimal to Octal :

Convert from decimal to binary and from binary to Octal

EX:

$(22)_{10}$ ->$(?)_8$

10 ->2

16	8	4	2	1
1	0	1	1	0

2 ->8

0	1	0		1	1	0
4	2	1		4	2	1
	2				6	

So **$(22)_{10}$ ->$(26)_8$**

Octal Systems

Mapping from Octal to binary :

EX:

$(67)_8$ ->$(?)_2$

6			7		
4	2	1	4	2	1
1	1	0	1	1	1

So **$(67)_8$ ->$(110111)_2$**

Octal Systems

Mapping from Octal to decimal :

Convert from Octal to binary and from binary to decimal

EX:

$(43)_8$ ->$(?)_{10}$

8 ->2

	4			3	
4	2	1	4	2	1
1	0	0	0	1	1

2 ->10

32	16	8	4	2	1
1	0	0	0	1	1

32+2+1+35

So **$(43)_8$ ->$(35)_{10}$**

Octal Systems

Mapping from Octal to Hexadecimal :

Convert from Octal to binary and from binary to Hexadecimal

EX:

$(36)_8$ ->$(?)_{16}$

8 ->2

3	6
4 2 1	4 2 1
0 1 1	1 1 0

2 ->16

0 0 0 1	1 1 1 0
8 4 2 1	8 4 2 1
1	E

So **$(36)_8$ ->$(1E)_{16}$**

Hexadecimal Systems

Mapping from Hexadecimal to binary :

EX:

$(AFF)_{16}$ ->$(?)_2$

A	F	F
8 4 2 1	8 4 2 1	8 4 2 1
1 0 1 0	1 1 1 1	1 1 1 1

So **$(AFF)_{16}$ ->$(101011111111)_2$**

Hexadecimal Systems

Mapping from Hexadecimal to decimal :

Convert from Hexadecimal to binary and from binary to decimal

EX:

$(AC)_{16}$ ->$(?)_{10}$

16 ->2

A				C			
8	4	2	1	8	4	2	1
1	0	1	0	1	1	0	0

2 ->10

128	64	32	16	8	4	2	1
1	0	1	0	1	1	0	0

128+32 +8+4=172

So **$(AC)_{16}$ ->$(172)_{10}$**

Hexadecimal Systems

Mapping from Hexadecimal to Octal :

Convert from Hexadecimal to binary and from binary to Octal

EX:

$(101)_{16}$ ->$(?)_{8}$

16 ->2

1	0	1
8 4 2 1	8 4 2 1	8 4 2 1
0 0 0 1	0 0 0 0	0 0 0 1

2 -> 8

0 0 0	1 0 0	0 0 0	0 0 1
4 2 1	4 2 1	4 2 1	4 2 1
0	4	0	1

So **$(101)_{16}$ ->$(0401)_{8}$**