

Computer System Architecture

(THIRD EDITION)



M. Morris Mano

PRENTICE HALL

2-1 Integrated Circuits

2 / 21

■ Integrated Circuits(IC)

- ◆ Digital circuits are constructed with **Integrated Circuits**
- ◆ An Integrated Circuits is a small silicon semiconductor crystal, called **chip**
- ◆ The various gates are interconnected inside the chip to form the required circuit
- ◆ The chip is mounted in a ceramic or plastic container, and connections are welded by thin gold wires to external pins to form the integrated circuits
- ◆ The number of pins may range from 14 in a small IC package to 100 or more in a larger package
- ◆ Each IC has a numeric designation printed on the surface of the package for identification

■ Logic Family에 의한 분류

- ◆ MOSFET { MOS Family(N or P Channel) : High Component Density
CMOS Family(N or P Channel) : Low Power
현재 TTL 과 CMOS
주로 사용
- ◆ BJT { TTL Family : 일반적으로 많이 사용
ECL Family : 고속을 요구하는 특수용도

2-2 Decoder/Encoder

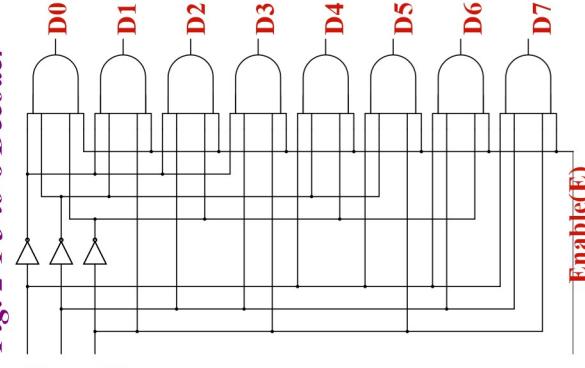
3 / 21

■ Decoder

- ◆ A combinational circuit that converts binary information from the **n** coded inputs to a maximum of 2^n unique outputs
- ◆ **n-to-m** line decoder = **n × m** decoder
 - **n** inputs, **m** outputs
 - If the n-bit coded information has unused bit combinations, the decoder may have less than 2^n outputs

$$\bullet m \leq 2^n$$

■ 3-to-8 Decoder

- ◆ A Binary-to-octal conversion
 - ◆ Logic Diagram : *Fig. 2-1*
 - ◆ Truth Table : *Tab. 2-1*
 - ◆ Commercial decoders include one or more Enable Input(E)
- 
- | Enable | A2 | A1 | A0 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|--------|----|----|----|----|----|----|----|----|----|----|----|
| 0 | x | x | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
- Fig. 2-1 3-to-8 Decoder*
- Tab. 2-1 Truth table for 3-to-8 Decoder*

2-2 Decoder/Encoder

■ NAND Gate Decoder

◆ Constructed with NAND instead of AND gates

◆ Logic Diagram/Truth Table : *Fig. 2-2*

Enable <input type="checkbox"/> Input		Output				
E	A1 A0	D0	D1	D2	D3	
0	0 0	0	1	1	1	
0	0 1	1	0	1	1	
0	1 0	1	1	0	1	
0	1 1	1	1	1	0	
1	x x	1	1	1	1	

◆ Decoder Expansion

◆ Constructed decoder : *Fig. 2-3*

◆ 3 X 8 Decoder constructed with two 2 X 4 Decoder

■ Encoder

◆ Inverse Operation of a decoder

◆ 2^n input, n output

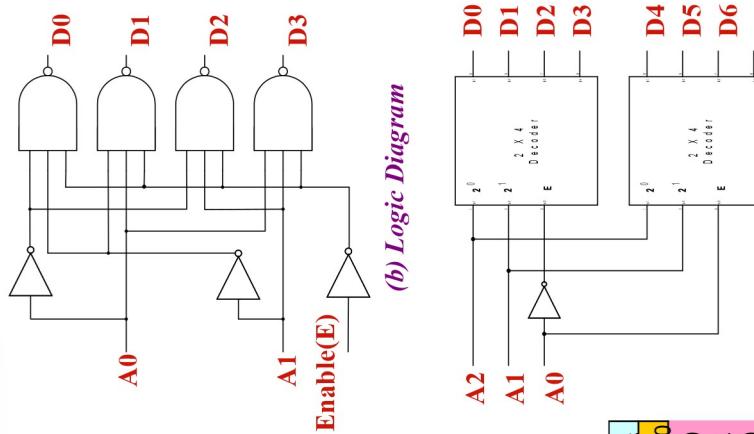
◆ Truth Table : *Tab. 2-2*

● 3 OR Gates Implementation

$$\gg A0 = D1 + D3 + D5 + D7$$

$$\gg A1 = D2 + D3 + D6 + D7$$

$$\gg A2 = D4 + D5 + D6 + D7$$



(a) Truth Table

(b) Logic Diagram

Inputs		Outputs								
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0

Fig. 2-3 A 3-to-8 Decoder
constructed with two
with 2-to-4 Decoder

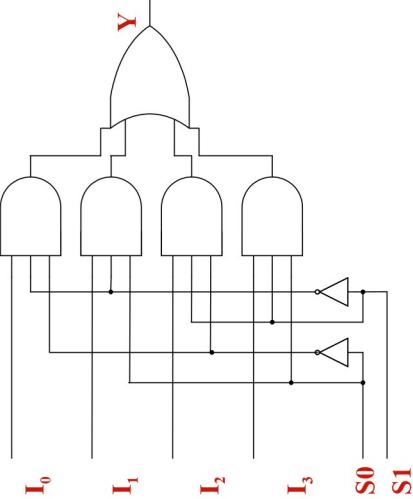
Chap. 2 Digital Components

2-3 Multiplexers

5 / 21

■ Multiplexer(Mux)

- ◆ A combinational circuit that receives binary information from one of 2^n input data lines and directs it to a single output line
- ◆ A 2^n -to-1 multiplexer has **2^n input data lines** and **1 output selection lines**(Data Selector)
- ◆ 4-to-1 multiplexer Diagram : *Fig. 2-4*
- ◆ 4-to-1 multiplexer Function Table : *Tab. 2-3*

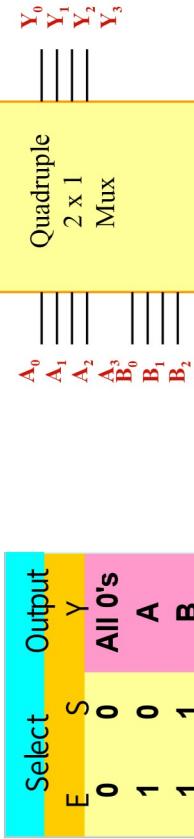


Select	Output	I ₀	I ₁	I ₂	I ₃
S ₁	Y	0	I ₀	I ₁	I ₂
0	0	0	1	I ₃	
0	1	1	0		
1	0	1	0	I ₂	I ₃
1	1	1	1		

Tab. 2-3 Function Table for
4-to-1 line Multiplexer

■ Quadruple 2-to-1 Multiplexer

- ◆ Quadruple 2-to-1 Multiplexer : *Fig. 2-5*



(a) Function Table

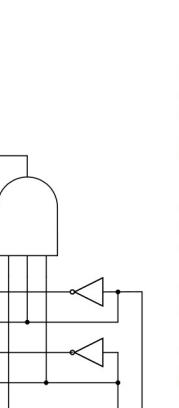


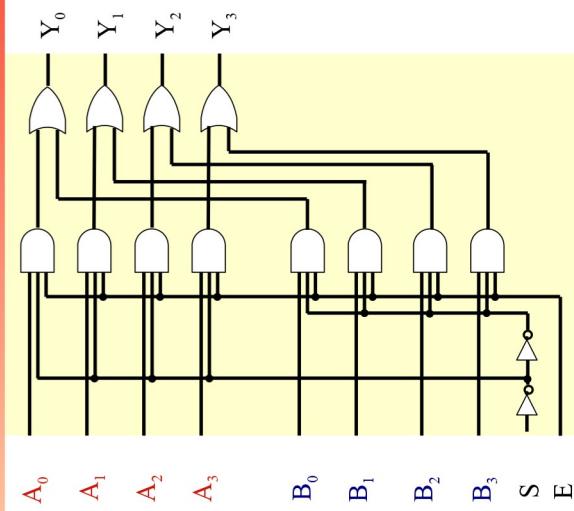
Fig. 2-4 4-to-1 Line Multiplexer

Chap. 2 Digital Components

(b) Block Diagram

2-3 Multiplexers

☞ Demultiplexer – multiplexer 와 반대 되는
연산을 수행하는 Digital combinational circuit



*Fig A. Combinational logic diagram with
four 2 \times 1 multiplexer*

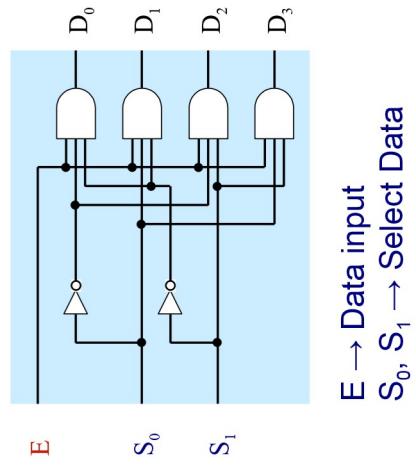


Fig B. Demultiplexer

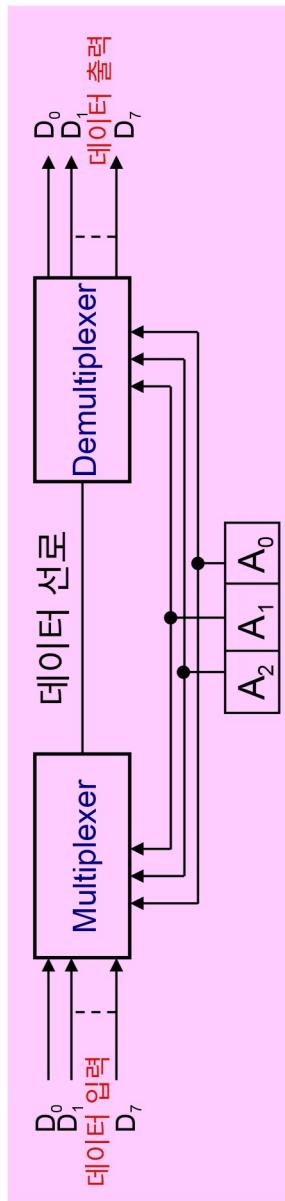


Fig C. Data transfer system

2-4 Registers

7 / 21

- Register
 - ◆ A group of flip-flops with each flip-flop capable of storing one bit of information
 - ◆ An n-bit register has a group of n flip-flops and is capable of storing any binary information of n bits
 - ◆ The simplest register consists only of flip-flops, with no external gate : *Fig. 2-6*
 - ◆ A clock input C will load all four inputs in parallel
 - The clock must be *inhibited* if the content of the register must be left unchanged

- Register with Parallel Load
 - ◆ A 4-bit register with a load control input : *Fig. 2-7*
 - ◆ The clock inputs receive clock pulses at all times
 - ◆ The buffer gate in the clock input will increase “fan-out”
 - ◆ Load Input
 - 1 : Four input transfer
 - 0 : Input inhibited, Feedback from output to input(*no change*)

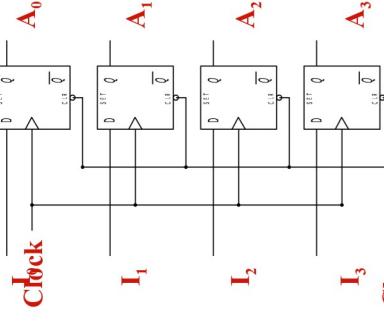


Fig. 2-6 4-bit register

2-5 Shift Registers

■ Shift Register

- ◆ A register capable of shifting its binary information in one or both directions
- ◆ The logical configuration of a shift register consists of a chain of flip-flops in cascade
- ◆ The simplest possible shift register uses only flip-flops : *Fig. 2-8*
- ◆ The **serial input** determines what goes into the leftmost position during the shift
- ◆ The **serial output** is taken from the output of the rightmost flip-flop

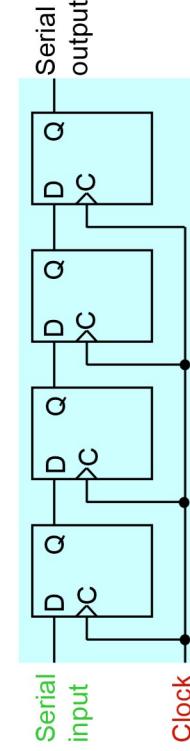


Fig. 2-8 4-bit shift register

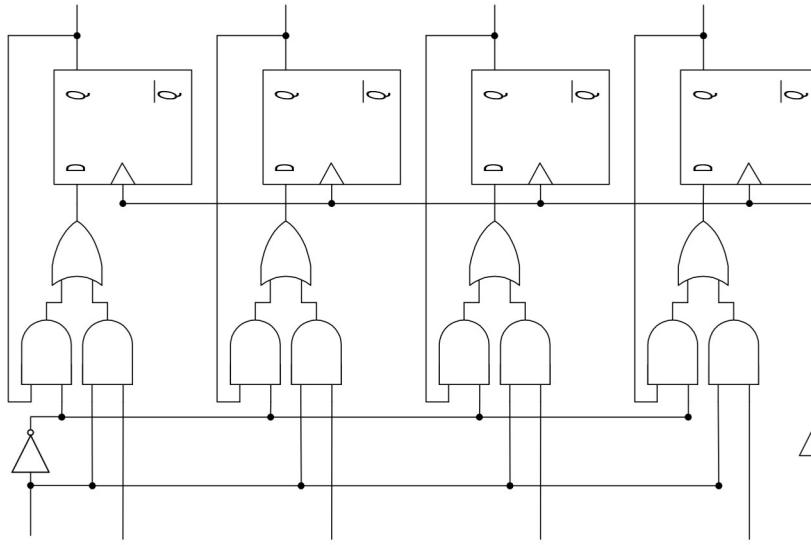


Fig. 2-7 4-bit register with parallel load

2-5 Shift Registers

☞ 보충설명

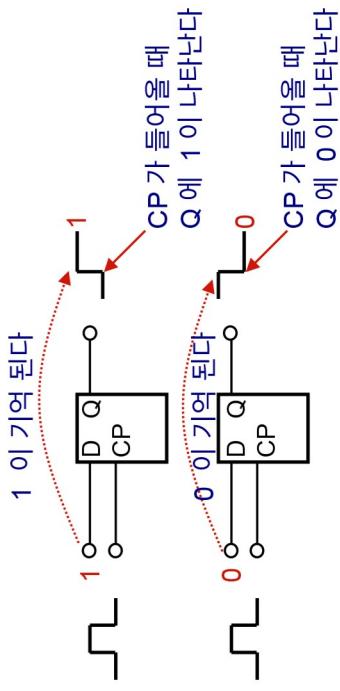


Fig a. 기억소자로서의 Flip-Flop

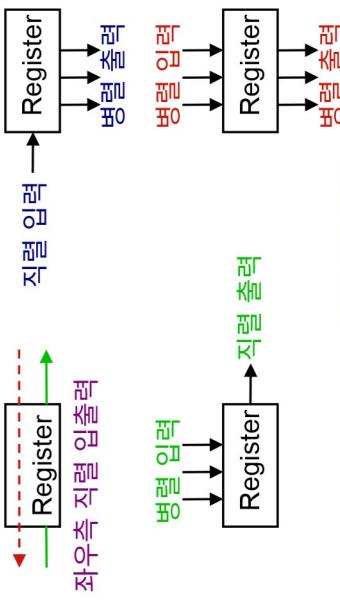


Fig b. Register에서 자료 이동

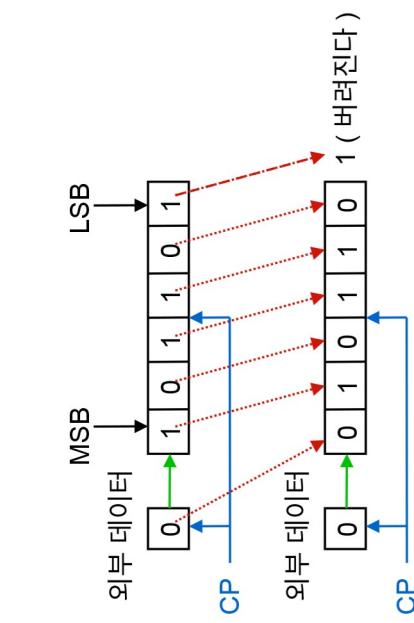


Fig c. 6bit Shift Register의 동작

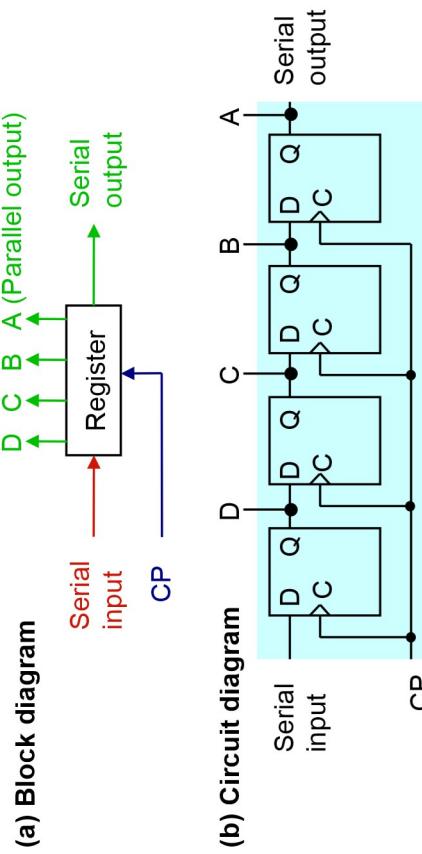


Fig d. D Flip-Flop을 사용한 Shift Register

2-5 Shift Registers

- Bidirectional Shift Register with Parallel Load
 - ◆ A register capable of shifting in *one direction only* is called a ***unidirectional shift register***
 - ◆ A register that can shift in *both directions* is called a ***bidirectional shift register***

- ◆ The most general shift register has all the capabilities listed below:

- An input clock pulse to synchronize all operations
- A shift-right /left (serial output/input)
- A parallel load, n parallel output lines
- The register unchanged even though clock pulses are applied continuously

- ◆ 4-bit bidirectional shift register with parallel load :

Fig. 2-9

Mode	Operation
S1 S0	
0 0	No change
0 1	Shift right(down)
1 0	shift left(up)
1 1	Parallel load

*Tab. 2-4 Function Table for Register
of Fig. 2-9*

2-5 Shift Registers

11 / 21

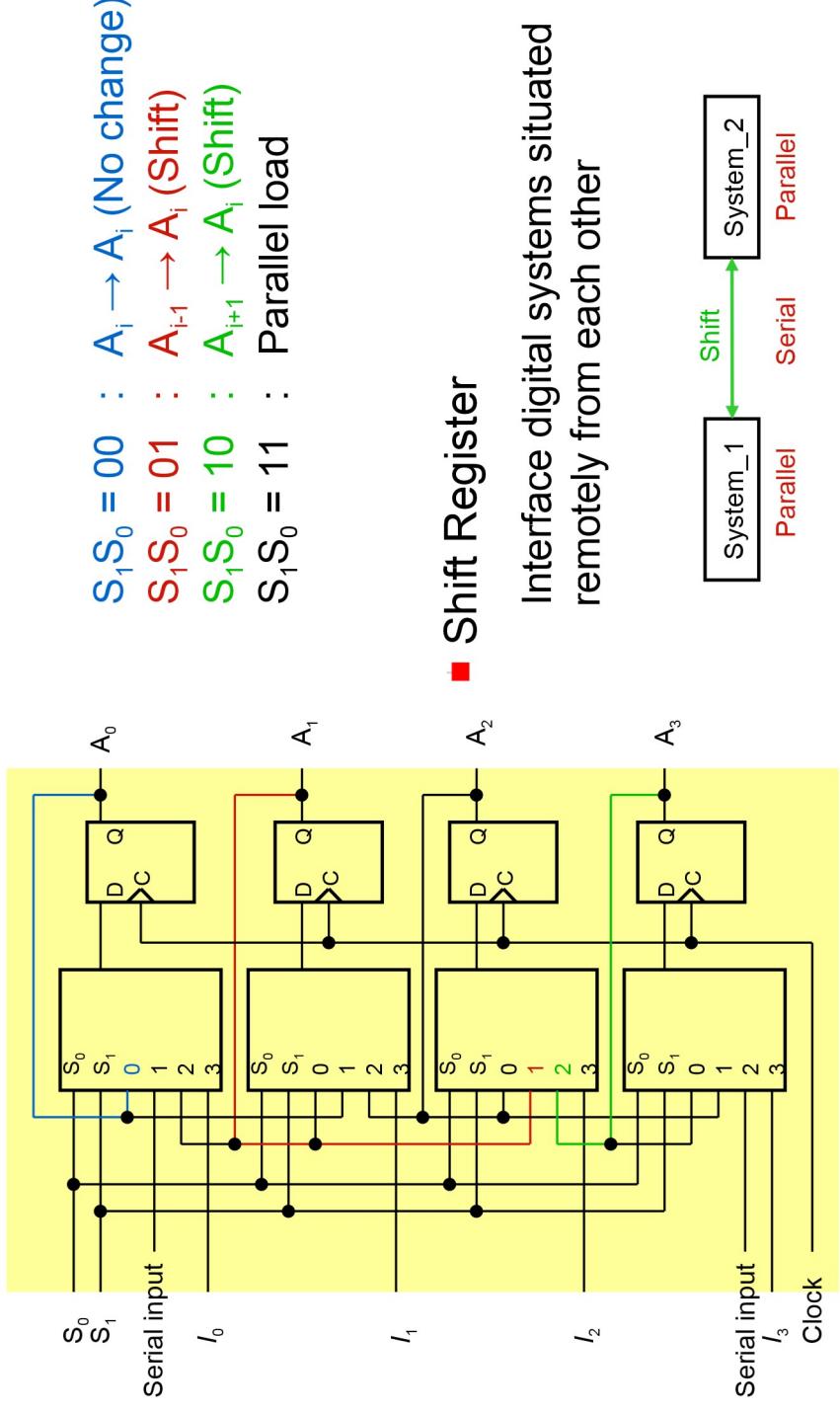


Fig. 2-9 Bidirectional shift register

2-6 Binary Counter

12 / 21

- Counter
 - ◆ A register goes through a predetermined sequence of state(Upon the application of input pulses)
 - ◆ Used for counting the number of occurrences of an event and **useful for generating timing signals to control/ the sequence of operations** in digital computers
 - ◆ An n-bit binary counter is a register of n flip-flop(count from 0 to $2^n - 1$)
- 4 bit Synchronous Binary Counter
 - **J K Q(t+1) 0 0 Q(t) 1 1 Q(t)'** A counter circuit will usually employ F/F with complementing capabilities(T and J-K F/F)
 - ◆ 4 bit Synchronous Binary Counter :

Fig. 2-10

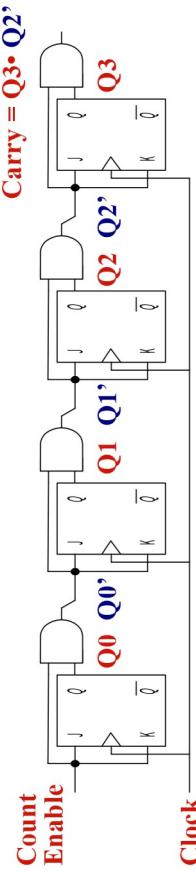
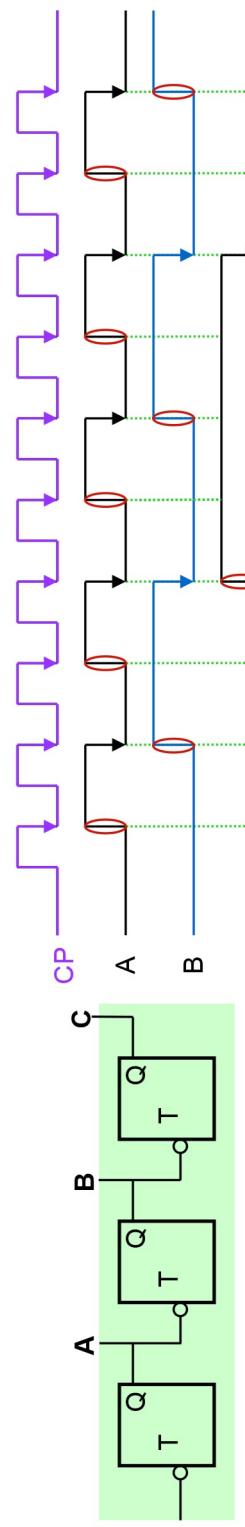


Fig. 2-10 4-bit Synchronous binary counter

2-6 Binary Counter

13 / 21

■ 3 bit asynchronous Counter Circuit

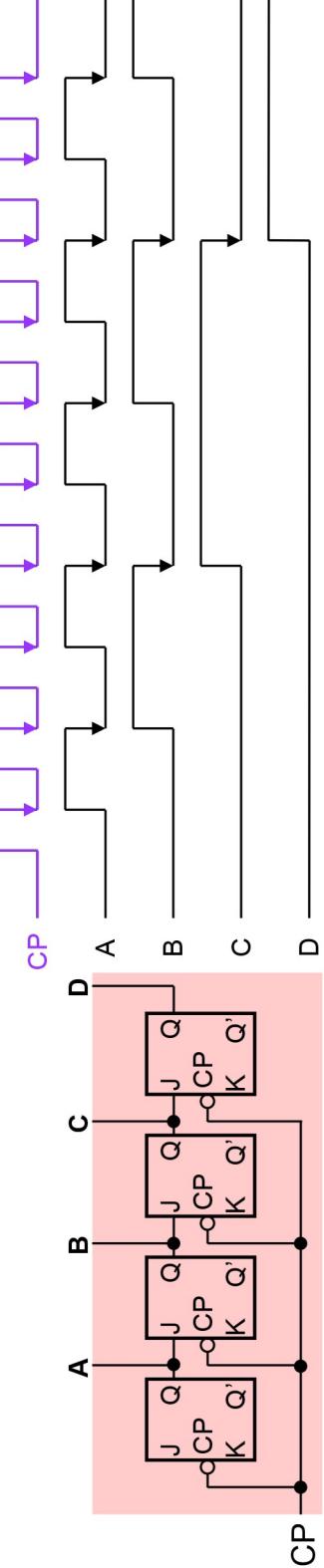


(a) Consists of T Flip-Flop

Count 0 1 2 3 4 5 6 7 8 9 10

(b) Timing chart

■ 4 bit synchronous Counter



(a) Consists of JK Flip-Flop

Count 0 1 2 3 4 5 6 7 8 9 10

(b) Timing chart

2-6 Binary Counter

14 / 21

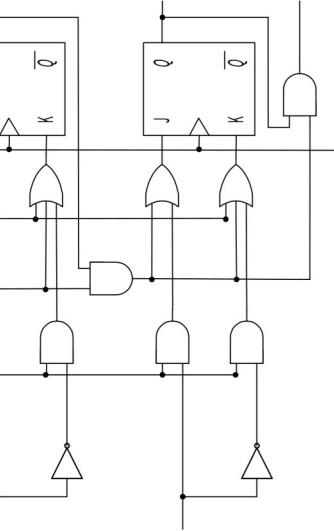
■ Binary Counter with Parallel Load

- ◆ Counters employed in digital systems(**CPU Register**) require a parallel load capability for transferring an initial binary number prior to the count operation

- ◆ 4-bit binary counter with Clear, Parallel Load, and Increment(Counter) :

Fig. 2-11

◆ Function Table : *Tab. 2-5*



- Increment : $1 \rightarrow J=K=1$ (Toggle)
(Clear, Load=0)

Fig. 2-11 4-bit binary counter with parallel load

- Clear : $1 \rightarrow K=X, J=0 \rightarrow \text{Clear}(Q=0)$
(Clear, Load=X)

J	K	$\alpha(t+1)$	$\alpha(t)$
0	0	Q(t)	Q(t)
0	1	0	1
1	0	0	1
1	1	1	1

- Load : $1 \rightarrow \begin{cases} I=1 \rightarrow J=1, K=0 \\ (Clear=0) \end{cases}$
(Load=X)

Chap. 2 Digital Components

2-7 Memory Unit

15 / 21

■ Memory Unit

- ◆ A collection of storage cells together with associated circuits needed to transfer information in and out of storage
- ◆ The memory stores binary information in groups of bits called **words**
- ◆ Word
 - A group of binary information that is processed in one simultaneous operation
- ◆ Byte
 - A group of eight bits (nibble : four bits)
 - ◆ The number of address line = **K**
 - Address(Identification number) : 0, 1, 2, 3, ... up to $2^k - 1$
 - The selection of specific word inside memory : k bit binary address
 - 1 Kilo= 2^{10} , 1 Mega= 2^{20} , 1 Giga= 2^{30}
 - 16 bit address line 예제 : $2^{16} = 64\text{K}$
 - ◆ Solid State Memory(IC Memory)
 - RAM(Volatile Memory)
 - ROM(Non-volatile Memory)

Dec	Hex
0	0000
1	0001
2	0010
3	0011
.	.
65535	FFFF

2-7 Memory Unit

16 / 21

■ Random Access Memory

- ◆ The memory cells can be accessed for information transfer from any desired random location
- ◆ Communication between a memory and its environment is achieved through **data input and output lines**, **address selection lines**, and **control lines** : **Fig. 2-12**
- ◆ The two operations that a random-access memory can perform are the **write** and **read** operations
- ◆ Memory Write

 - 1) Apply the binary address
 - 2) Apply the data bits
 - 3) Activate the write input

- ◆ Memory Read

 - 1) Apply the binary address
 - 2) Activate the read input

 - » The content of the selected word does not change after reading

◆ NV-RAM : battery back-up CMOS RAM

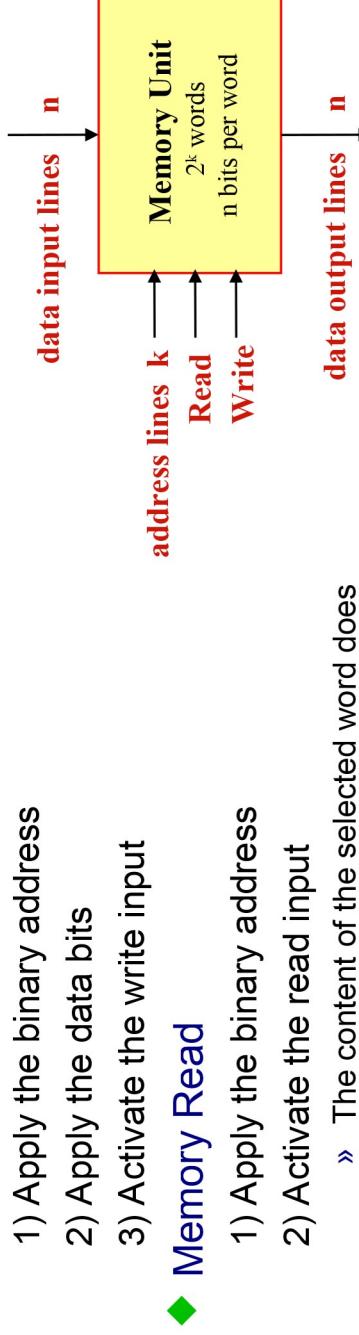


Fig. 2-12 Block diagram of RAM

2-7 Memory Unit

17 / 21

■ Read-Only Memory

- ◆ A memory unit that performs the read operation **only**; it does not have a write capability
- ◆ ROM comes with special internal electronic **fuses** that can be “*programmed*” for a specific configuration
- ◆ $m \times n$ ROM : *Fig. 2-13*
- k address input lines to select one of $2^k = m$ **words** of memory, and n output lines(n bits word)
- ◆ ROM is classified as a **combinational circuit**, because the outputs are a function of only the present inputs(address lines)
 - There is no need for providing storage capabilities as in a RAM
- ◆ ROM is also employed in the design of **control units** for digital computers
 - A Control Unit that utilizes a ROM to store binary control information is called a **micro-programmed control**
- ◆ Types of ROMs
 - UVEPROM(Chip level erase), EEPROM(Byte level erase), Flash ROM(Page or block level erase), OTPROM, Mask ROM

