

Digital Logic & Digital Systems

Part B

Functional Units from Logic Gates
&
Sequential Logic, Counters and Shift Registers

4004CEM

Computer Architecture & Networks

Today

- *Adders & Subtractors*
- *Comparators*
- *Decoder & Encoder*
- *Multiplexer and Demultiplexer*
- *Registers & Shift Registers*

Digital Circuits

Digital circuits are classified into two categories:

- **Combinational circuit**

The input values explicitly determine the output (**the output depends only on the current state of the inputs**)

- **Sequential circuit**

The output is a function of the input values and the existing state of the circuit (**the output depends on the current state of the inputs and the previous state of the outputs**)

Digital Circuits

Digital circuit operations are described using:

- Boolean expressions
- Logic diagrams
- Truth tables

Digital Circuits

- **Boolean expressions:** An algebraic notation which is used demonstrate the components and activity of electrical circuits.
- **Logic diagram:** A graphical representation of a circuit. Each type of gate is represented by a specific graphical symbol.
- **Truth table:** A table showing all possible input values and the associated output values.

Adders

At the digital logic level, addition is performed in binary. Addition operations are carried out by special circuits called, appropriately, **adders**.

- Addition rules:

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

Add Carry to
next bit (if not
MSB)

Half Adder

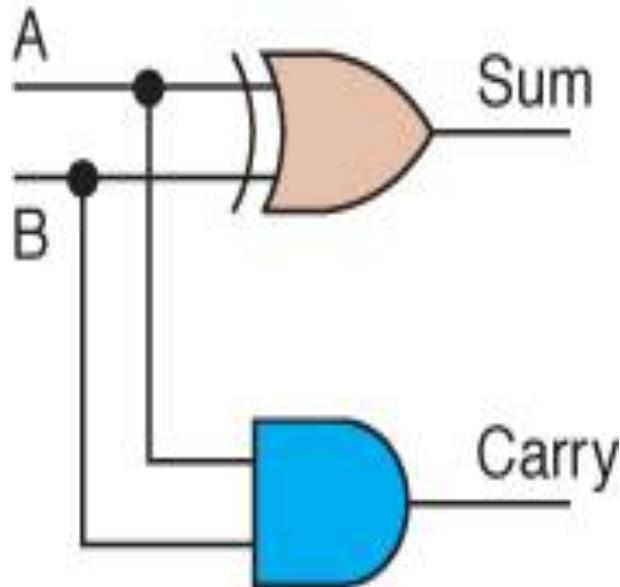
A circuit that computes the sum of two bits

- Examine the half adder's truth table carefully.
 - The **Sum** column has the same results as the **XOR** gate.
 - The **Carry** column has the same results as the **AND** gate.

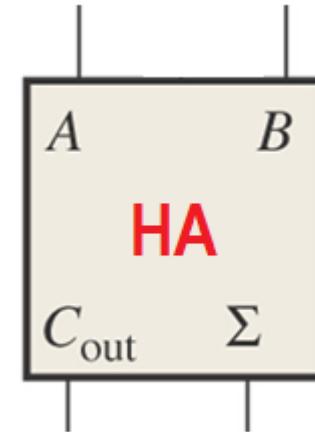
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table

Half Adder



Circuit diagram



Logic symbol

Boolean expressions:

$$\text{Sum} = A \oplus B$$

$$\text{Carry} = AB$$

Binary Addition

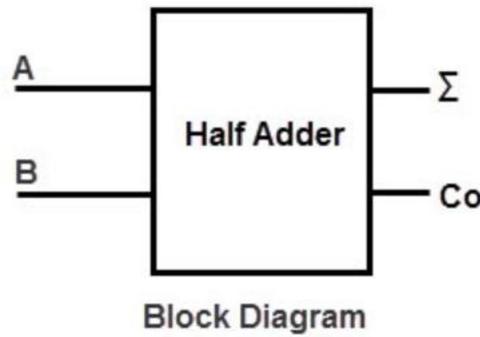
- Conceptually similar to decimal addition

$$\begin{array}{r} & \text{(carry)} \\ & 1 \\ 1 & 0 & 1 & 0 \\ + & 1 & 1 \\ \hline 1 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{r} & \text{(carry)(carry)} \\ & 1 \\ & 1 \\ 1 & 0 & 1 & 0 \\ + & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 1 & 0 \end{array}$$

$$\begin{array}{cccc} & 1 & 1 & 0 & 0 \\ & \underline{+1} & \underline{+0} & \underline{+1} & \underline{+0} \\ (\text{carry}) & 1 & 0 & 1 & 1 & 0 \end{array}$$

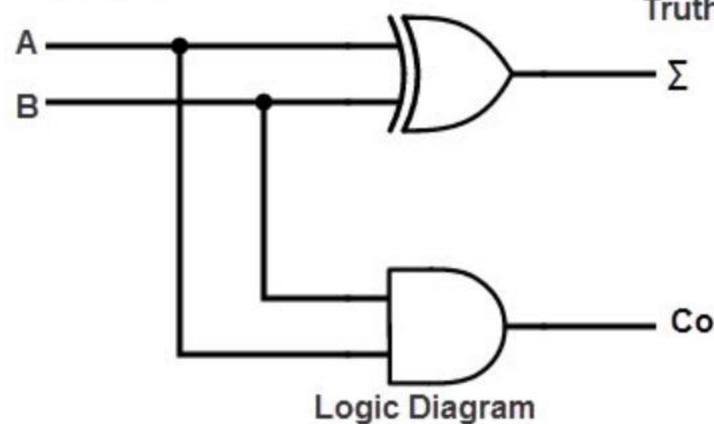
INPUT		OUTPUT	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Block Diagram

A	B	Sum (Σ)	Carry Out (Co)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

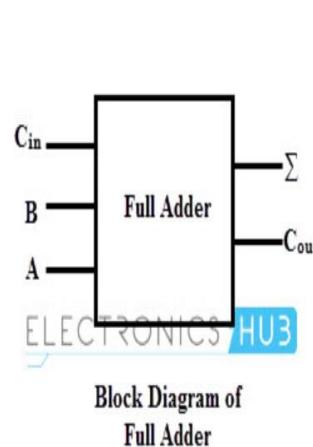
Truth Table



Logic Diagram

Full Adder

A circuit that takes the carry-in value into account



C _{in}	B	A	Σ	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

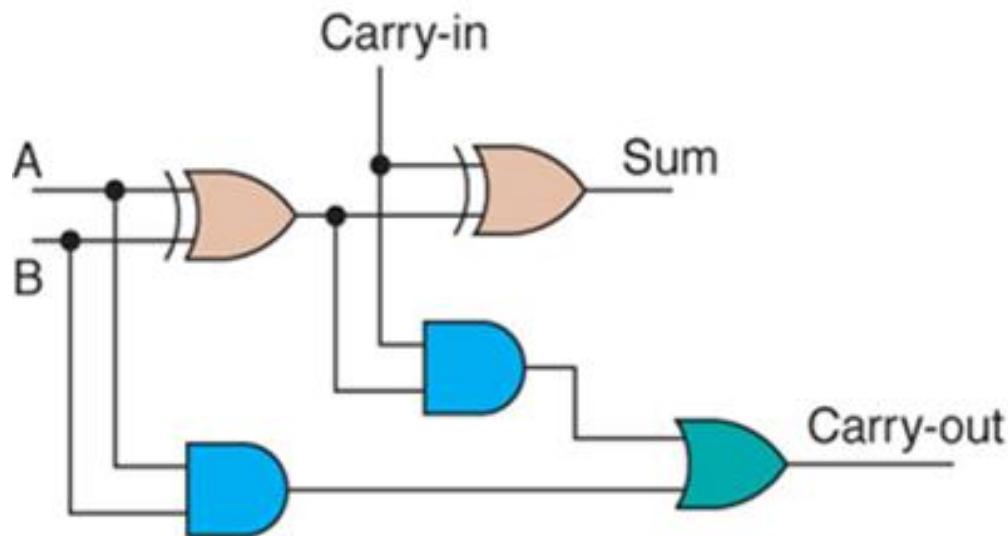
Truth Table

Truth Table

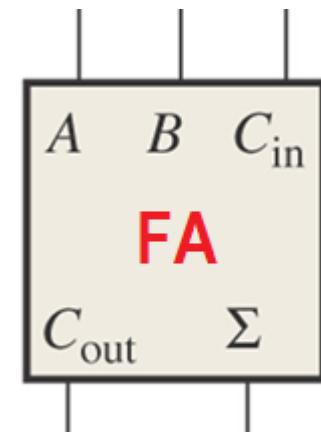
A	B	Carry-in	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Adder

A circuit that takes the carry-in value into account



Circuit diagram



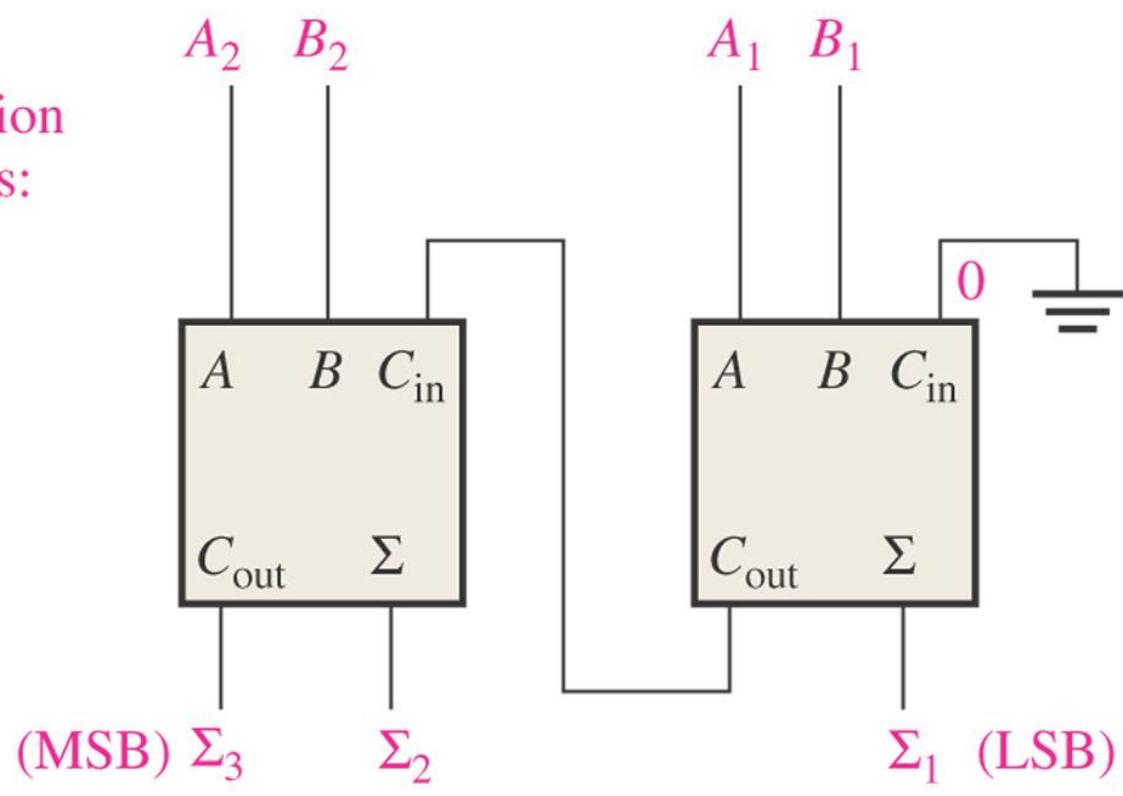
Logic symbol

Parallel Binary Adder

Two or more full adders are connected to form parallel binary adders

General format, addition
of two 2-bit numbers:

$$\begin{array}{r} A_2 A_1 \\ + B_2 B_1 \\ \hline \Sigma_3 \Sigma_2 \Sigma_1 \end{array}$$



2-bit parallel binary adder

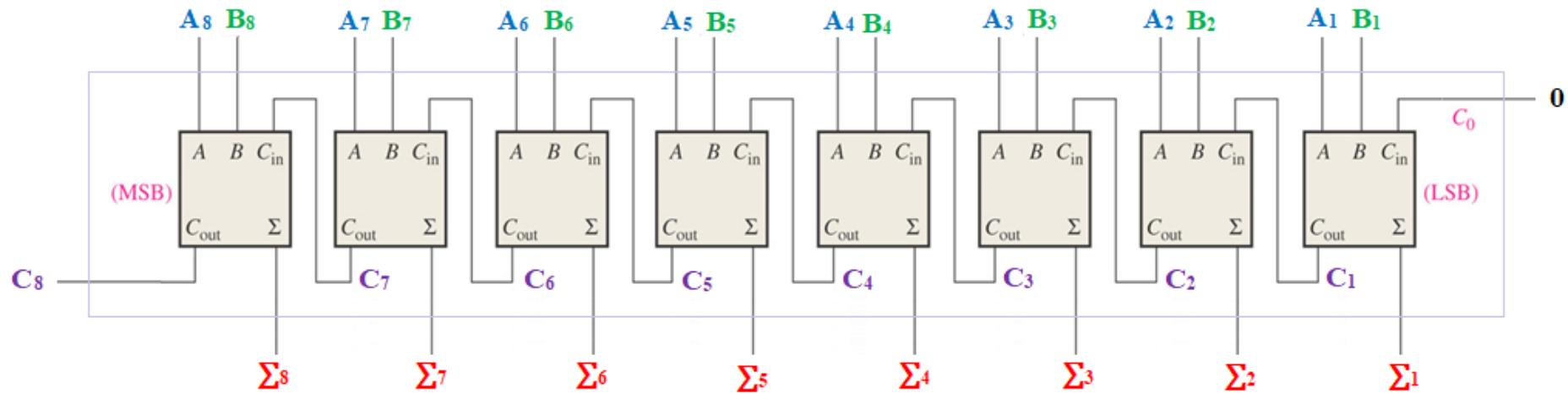
Parallel Binary Adder

To add two 8-bit numbers ($A + B$), the carry input to the first stage should be 0 and the sum size will be 9-bit at most

$$0 \leftarrow C_0$$

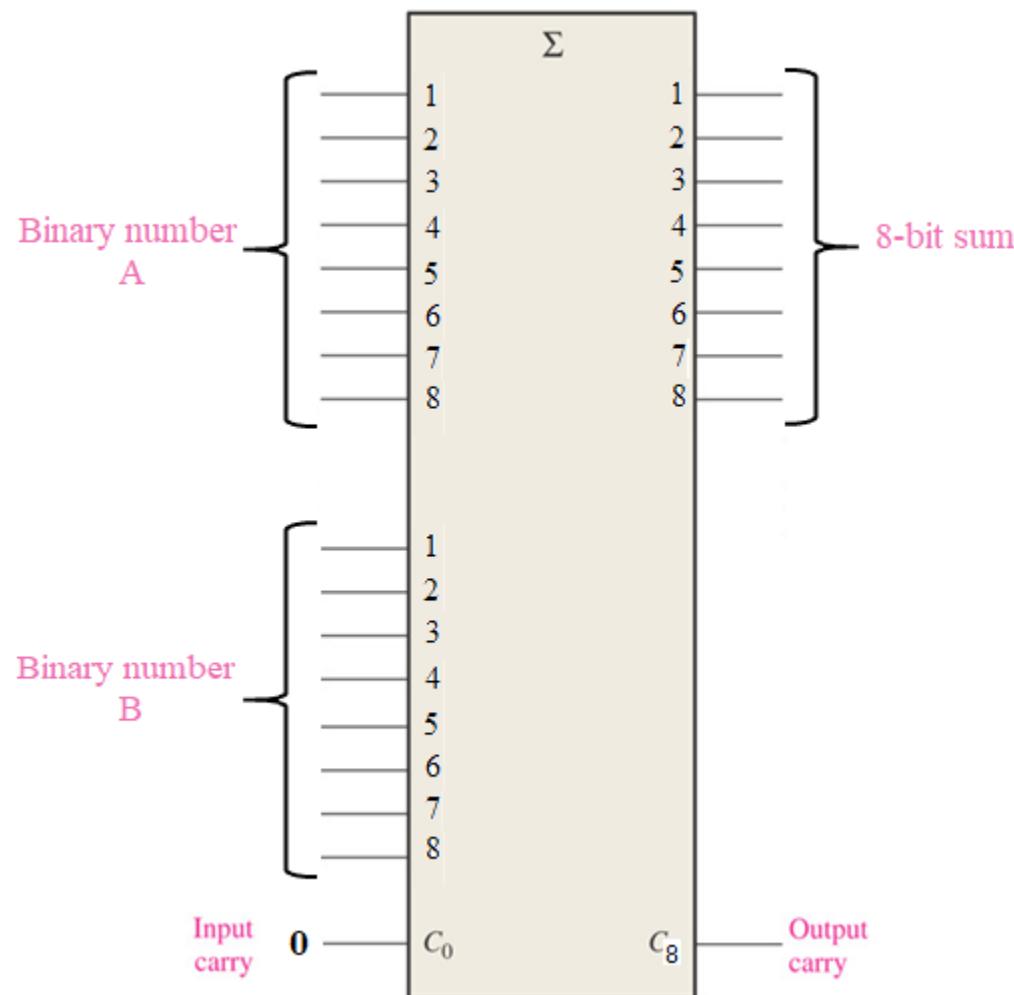
$$\begin{array}{r} A_8 \ A_7 \ A_6 \ A_5 \ A_4 \ A_3 \ A_2 \ A_1 \\ B_8 \ B_7 \ B_6 \ B_5 \ B_4 \ B_3 \ B_2 \ B_1 \\ + \end{array}$$

$$C_8 \ \Sigma_8 \ \Sigma_7 \ \Sigma_6 \ \Sigma_5 \ \Sigma_4 \ \Sigma_3 \ \Sigma_2 \ \Sigma_1$$



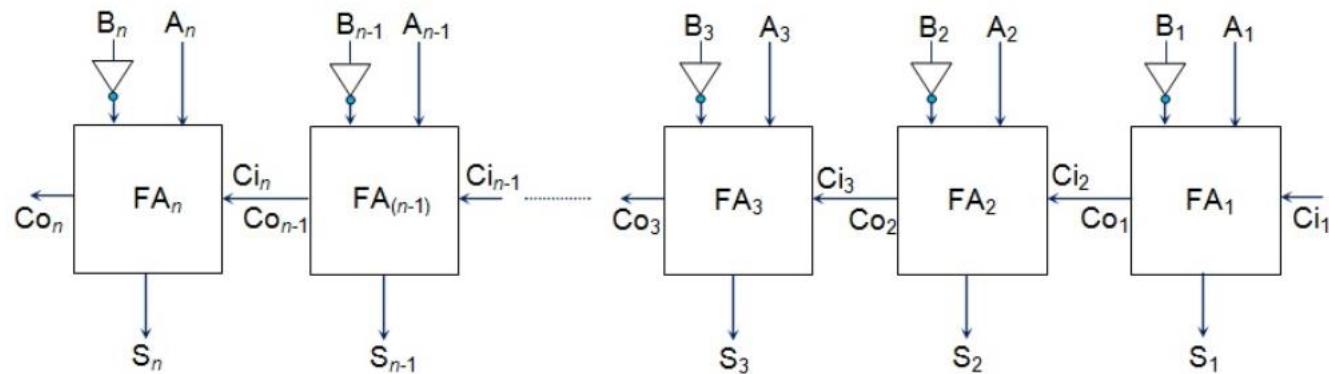
Block diagram for 8-bit parallel binary adder

Parallel Binary Adder



8-bit parallel binary adder

Parallel Subtractor

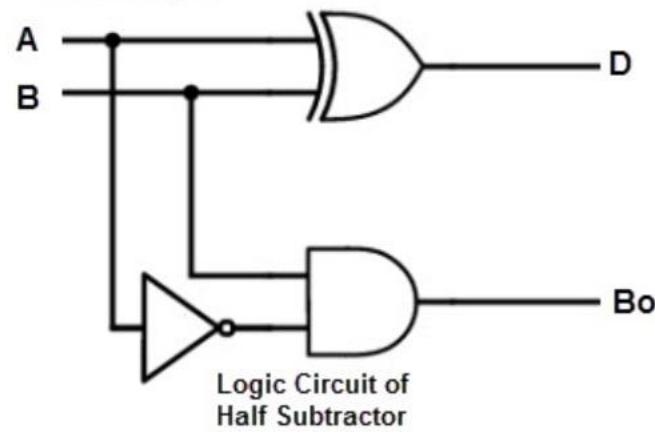
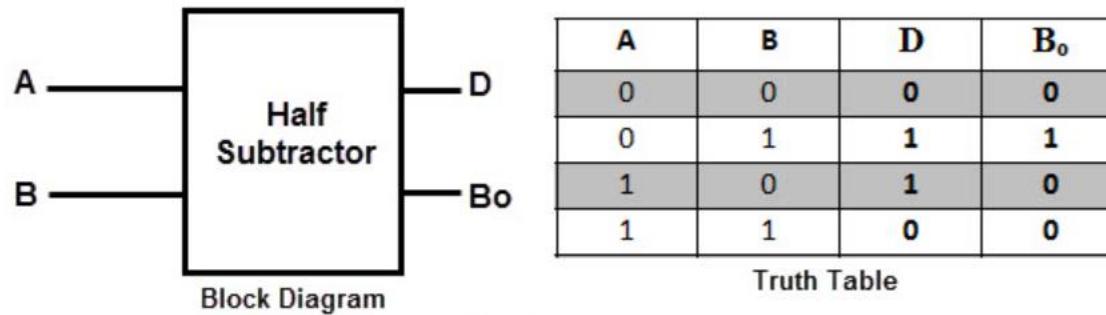


HALF SUBTRACTOR TRUTH TABLE

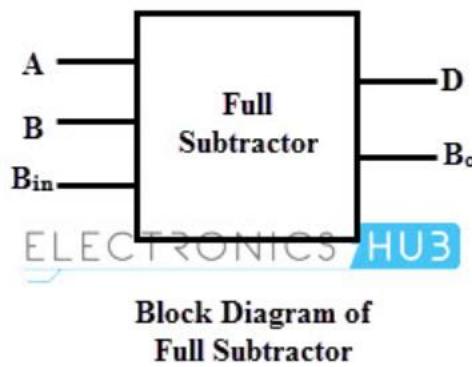
INPUT		OUTPUT	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Difference similar to XOR output

Borrow = NOR + AND

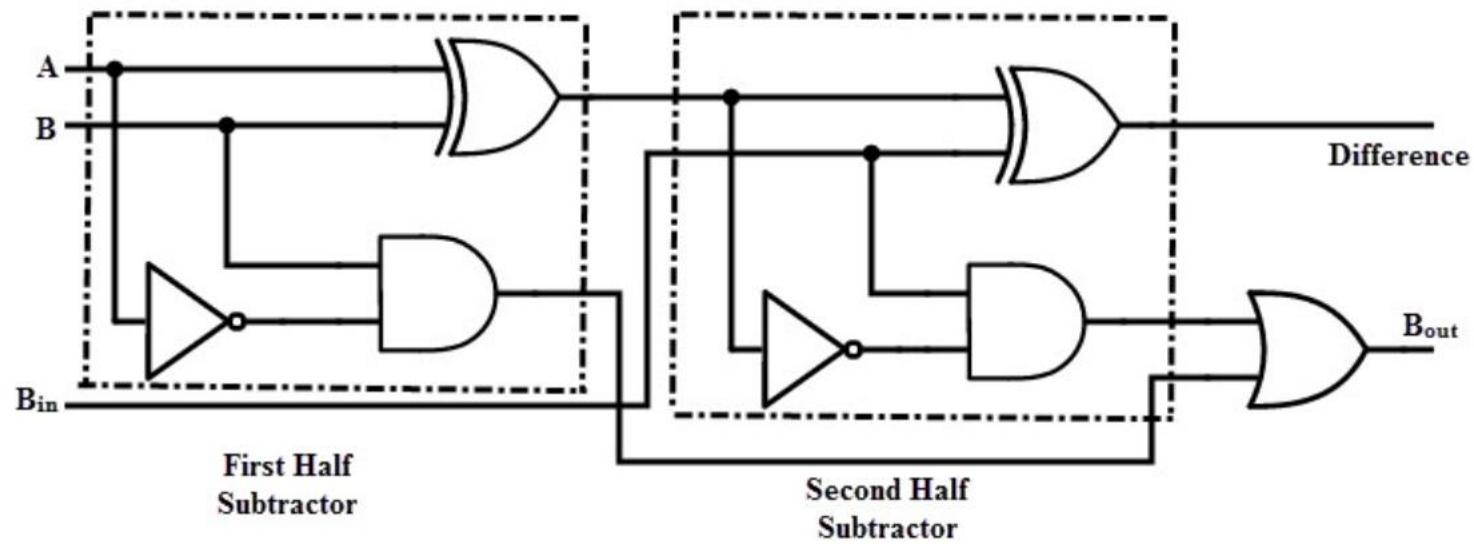


Full subtractor



A	B	B _{in}	D	B _o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table



Parallel Binary Subtractor

- The subtraction $A - B$ can be done by taking the 2's complement of B and adding it to A because $\textcolor{blue}{A - B = A + (-B)}$
- The 2's complement of B can be calculated by adding **1** to the 1's complement (convert **0** → **1** and **1** → **0**) of binary number B
$$\textcolor{red}{-B = \bar{B} + 1}$$
- Thus, the subtraction $A - B$ can be calculated using the parallel binary adder circuit by complementing all the bits of B and add 1 to the least significant bit (by setting carry C_0 to 1)

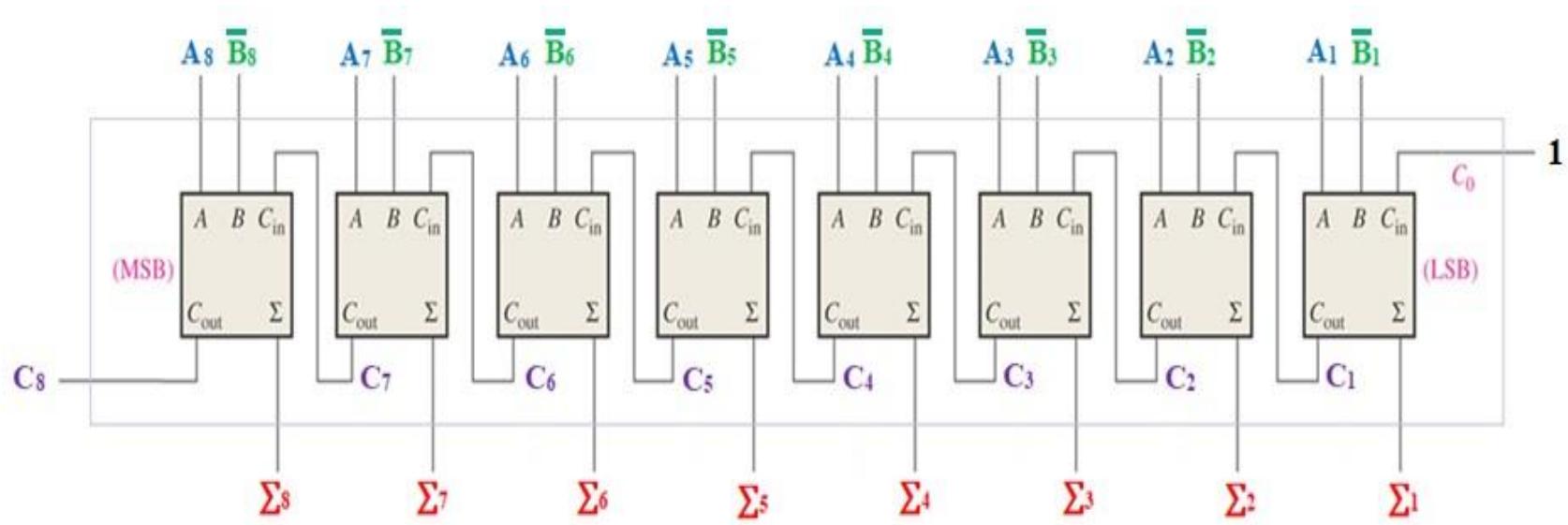
Parallel Binary Subtractor

To subtract two 8-bit numbers ($A - B$),
the first number A should be added to
the 2's complement of the second
number ($\bar{B} + 1$)

$$1 \leftarrow C_0$$

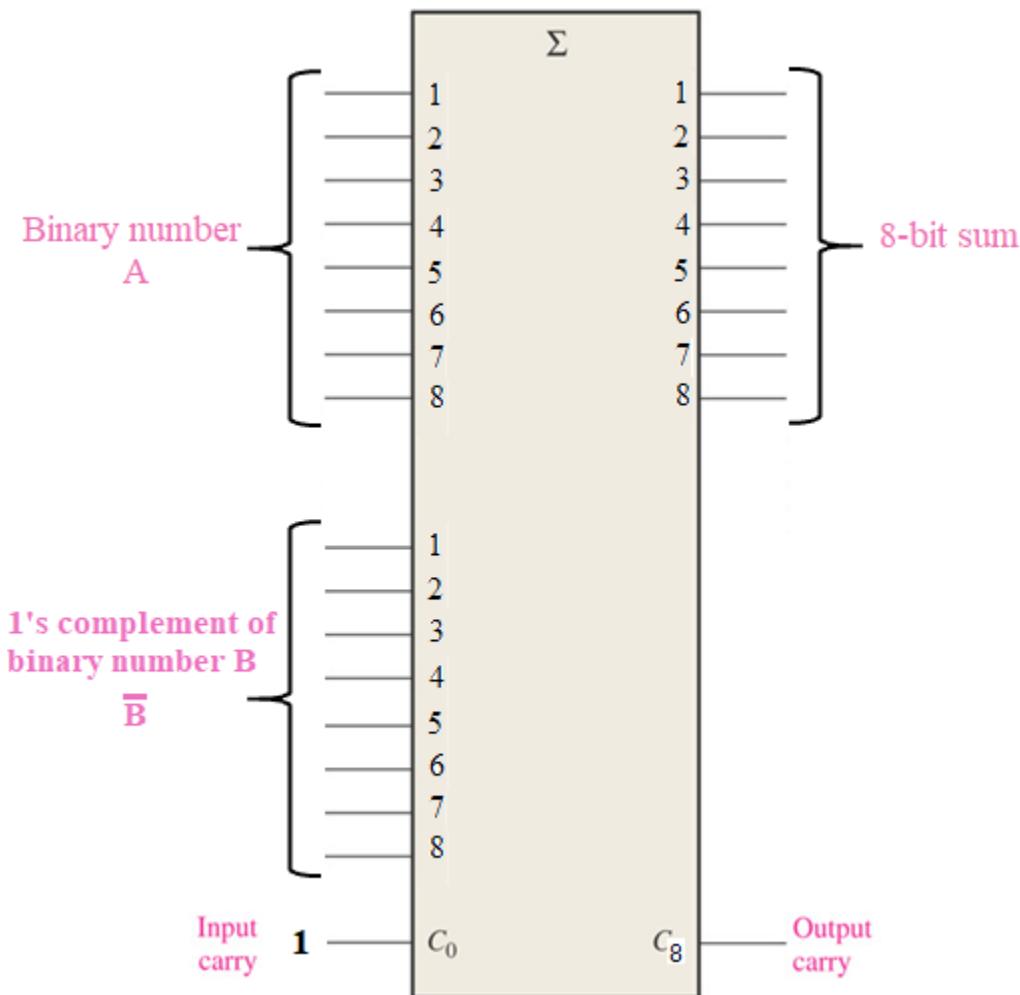
$$\begin{array}{cccccccc} A_8 & A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 \\ \bar{B}_8 & \bar{B}_7 & \bar{B}_6 & \bar{B}_5 & \bar{B}_4 & \bar{B}_3 & \bar{B}_2 & \bar{B}_1 \end{array} +$$

$$C_8 \Sigma_8 \Sigma_7 \Sigma_6 \Sigma_5 \Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1$$



Block diagram for 8-bit parallel binary Subtractor

Parallel Binary Subtractor

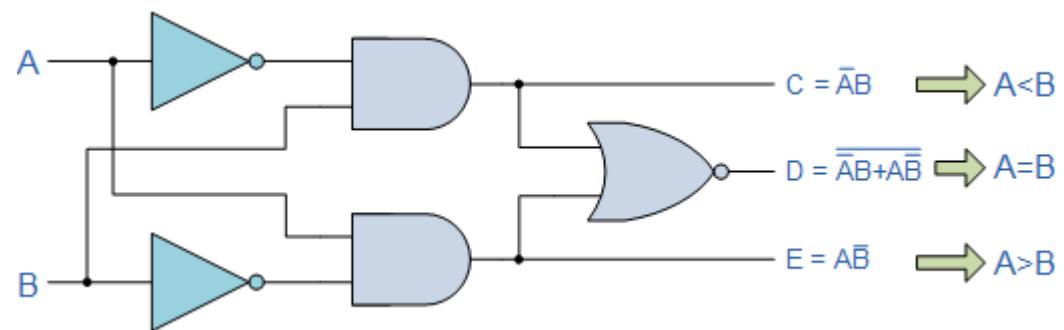


8-bit parallel binary subtractor

Comparator

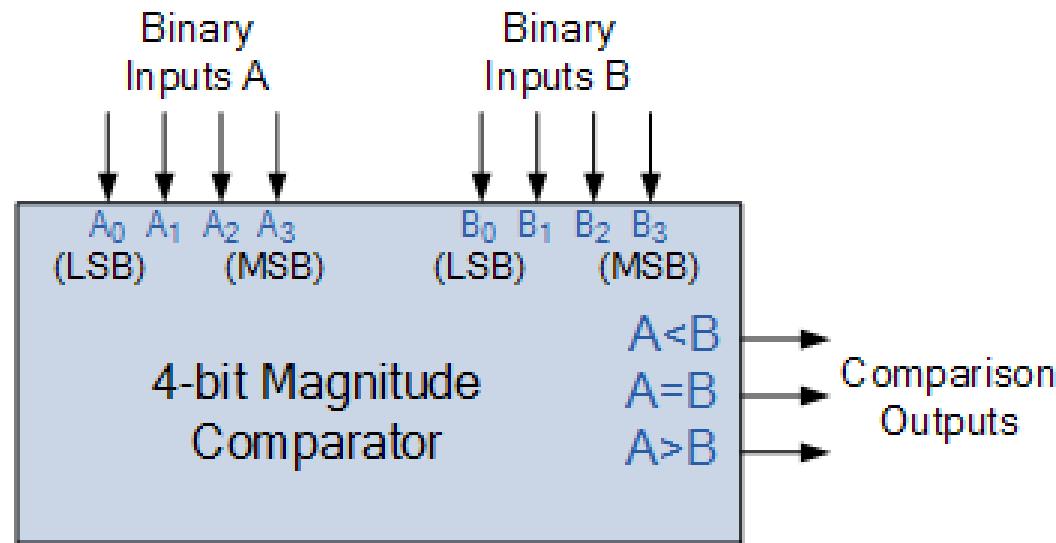
- Magnitude Comparator has three output terminals, one each for equality, $A = B$ greater than, $A > B$ and less than $A < B$.

Inputs		Outputs		
B	A	$A > B$	$A = B$	$A < B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



1-bit Digital Comparator

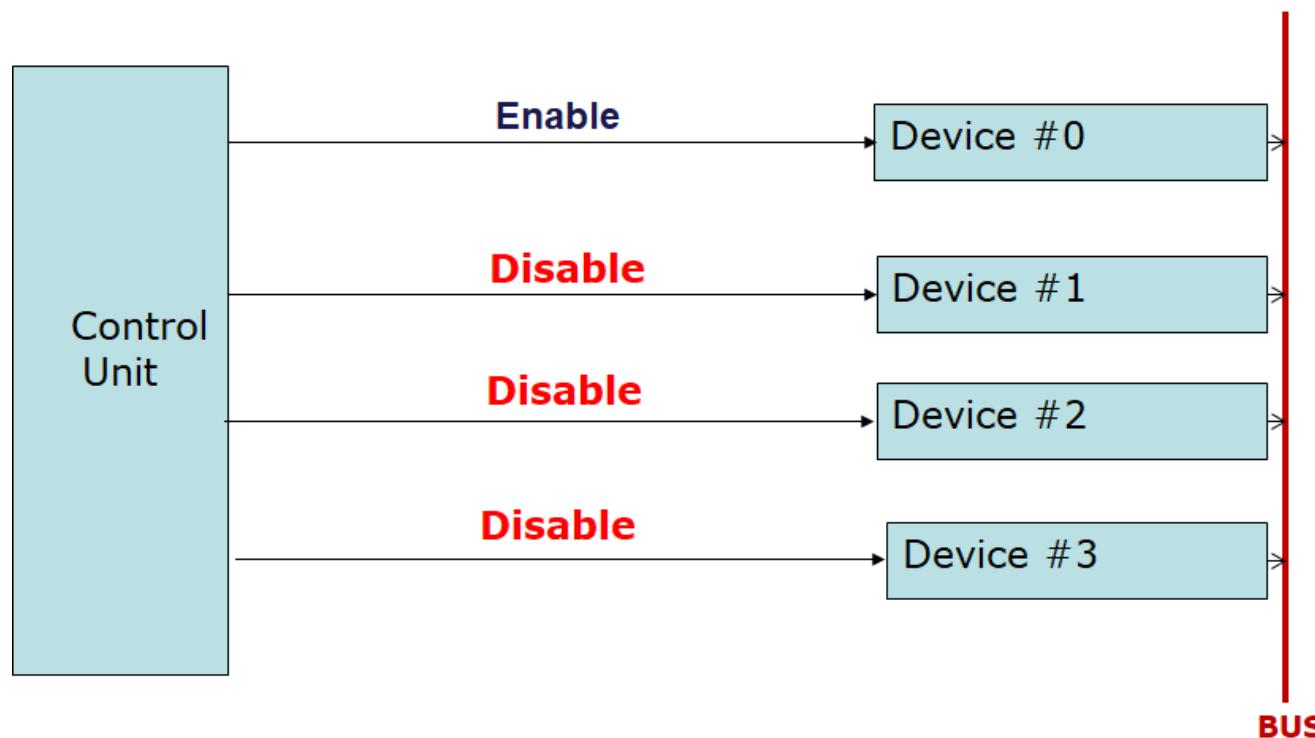
Comparator



4-bit Magnitude Comparator

Decoder

- If it is required to read or write data from 4 devices, only **one** device must be active at a time. To be able to control these 4 devices, each device should be connected to the **Control Unit** via an individual wire.
- The more devices are connected, the more the number of wires!!

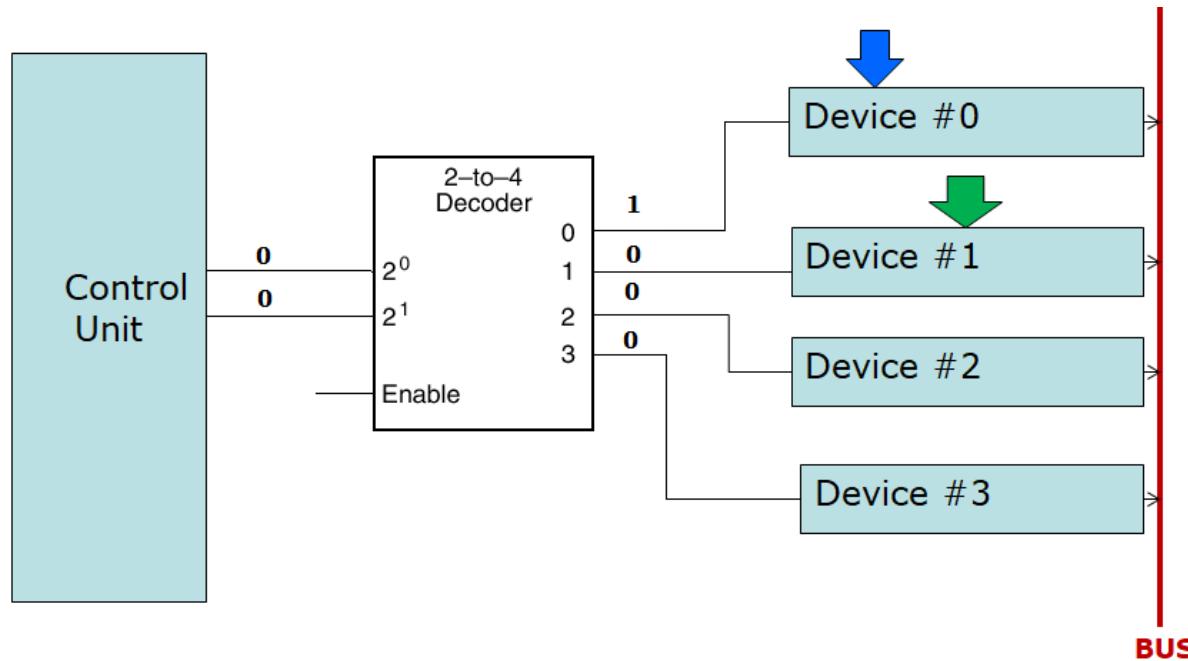


Decoder

- Address decoding provides a solution to control the 4 devices using only **2 wires** from my **Control Unit**.

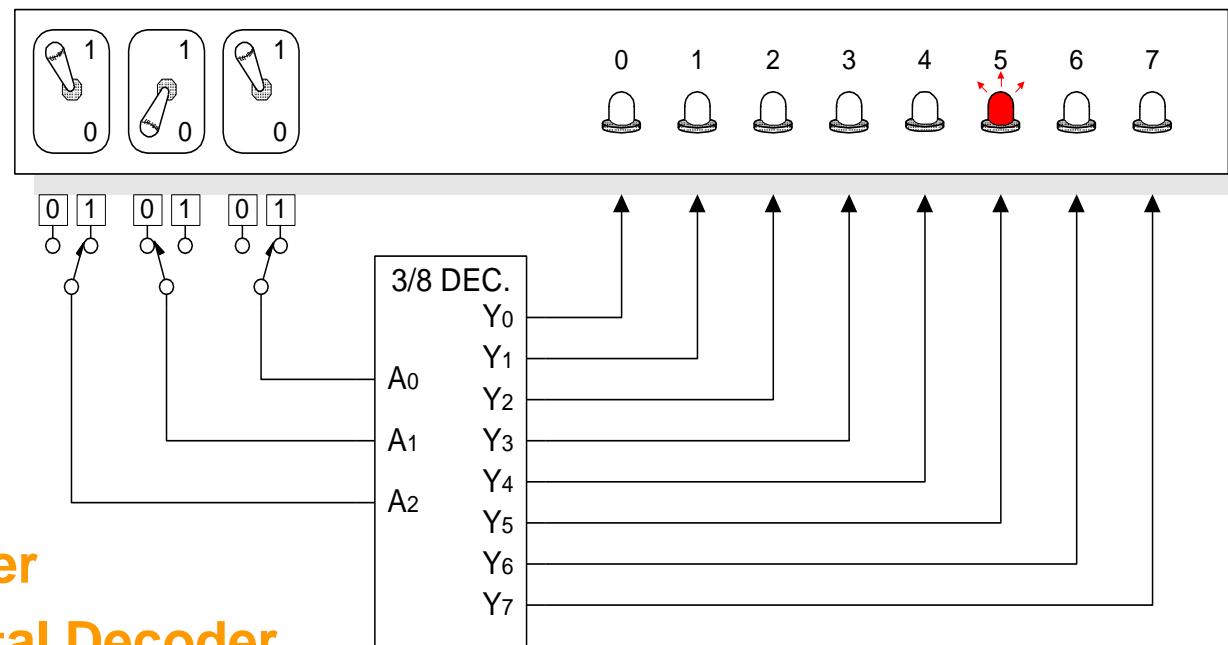


A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



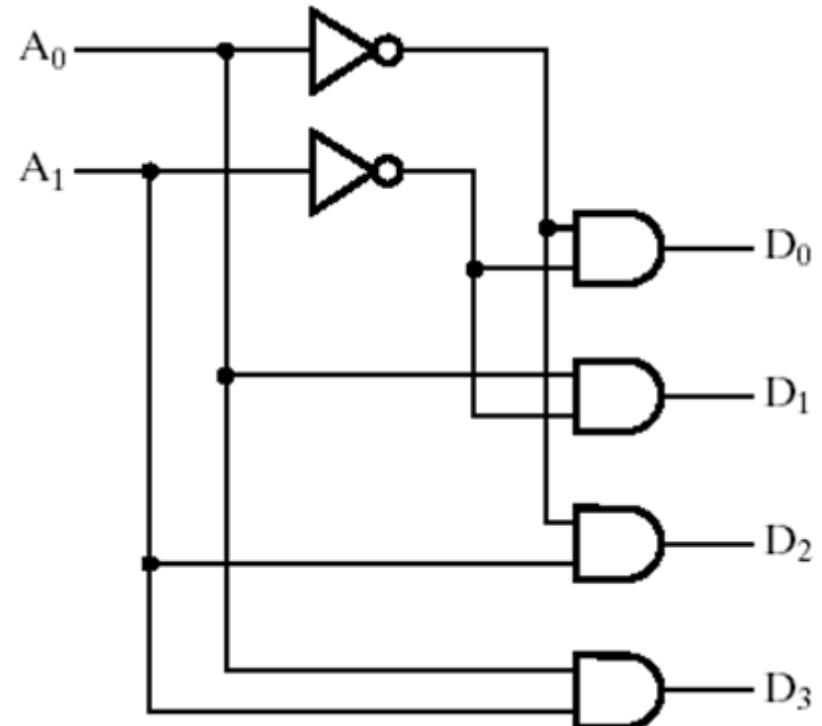
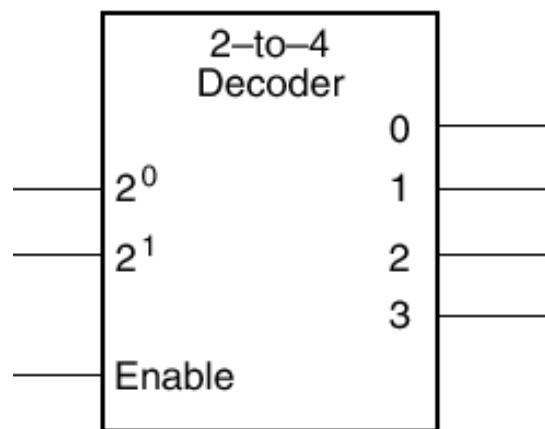
Decoder

- A decoder is a combinational digital circuit with a number of inputs n and a number of outputs 2^n .
- Only **one** of the outputs is enabled at a time. The output enabled is the one specified by the binary number formed at the inputs of the decoder.



Decoder

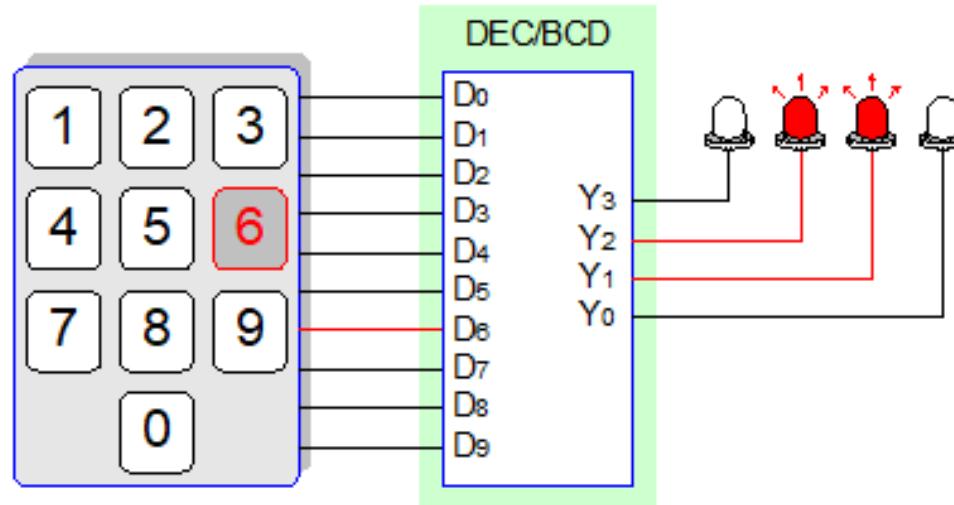
A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



2-to-4 Decoder

Encoder

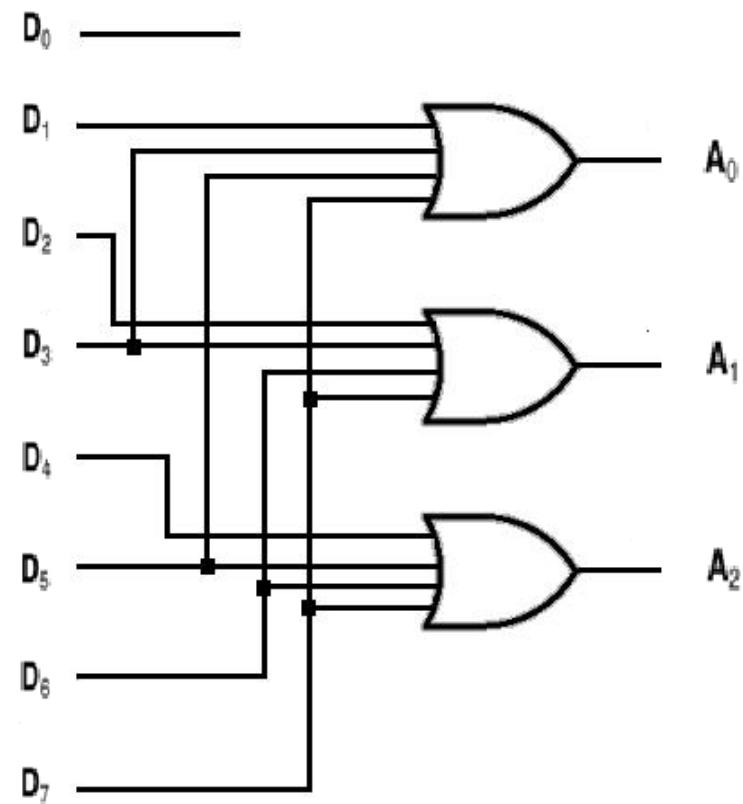
- The opposite of the decoding process.
- An encoder has a number of input lines, **only one** of which is activated at a given time.
- Take **2^n** input lines and generate **n** output lines.



Decimal-to-BCD Encoder

Encoder

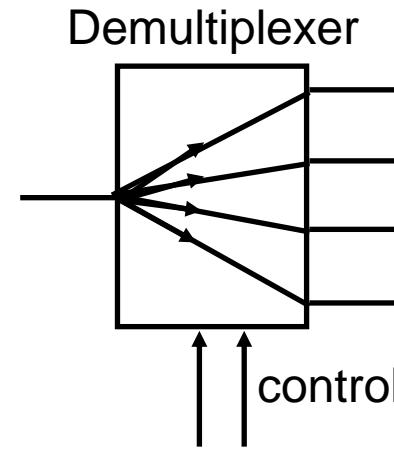
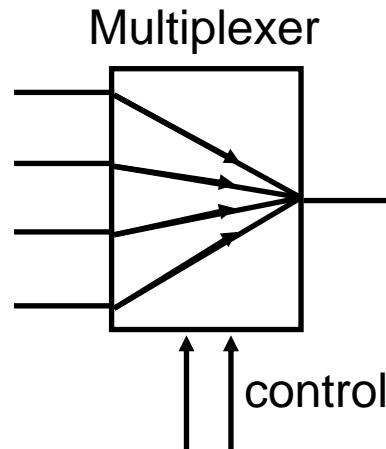
Inputs							Outputs			
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	A ₂	A ₁	A ₀
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



Octal-to-Binary Encoder

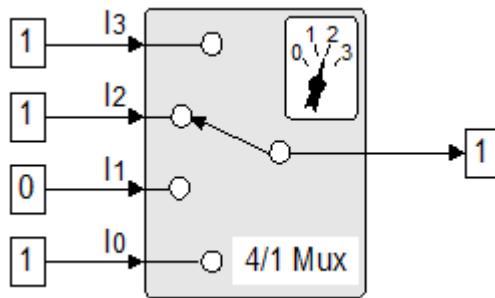
Switching Network

- **Multiplexer (MUX)**
 - Routes one of many inputs to a single output
 - Also called a *selector*
- **Demultiplexer (DEMUX)**
 - Routes a single input to one of many outputs
 - Also called a *distributer*

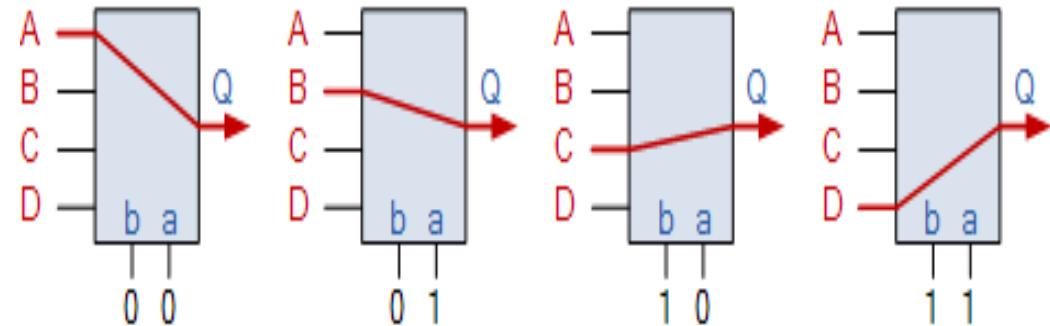


Multiplexers

- A multiplexer (MUX) has 2^n data inputs, n control inputs and one output. The output has always the same value as the data input specified by the binary number at the control inputs.
- A television producer wishes to select the output of one of three cameras to go to a monitor.
- MUX is used to select which register from the CPU set of registers is connected to the ALU.

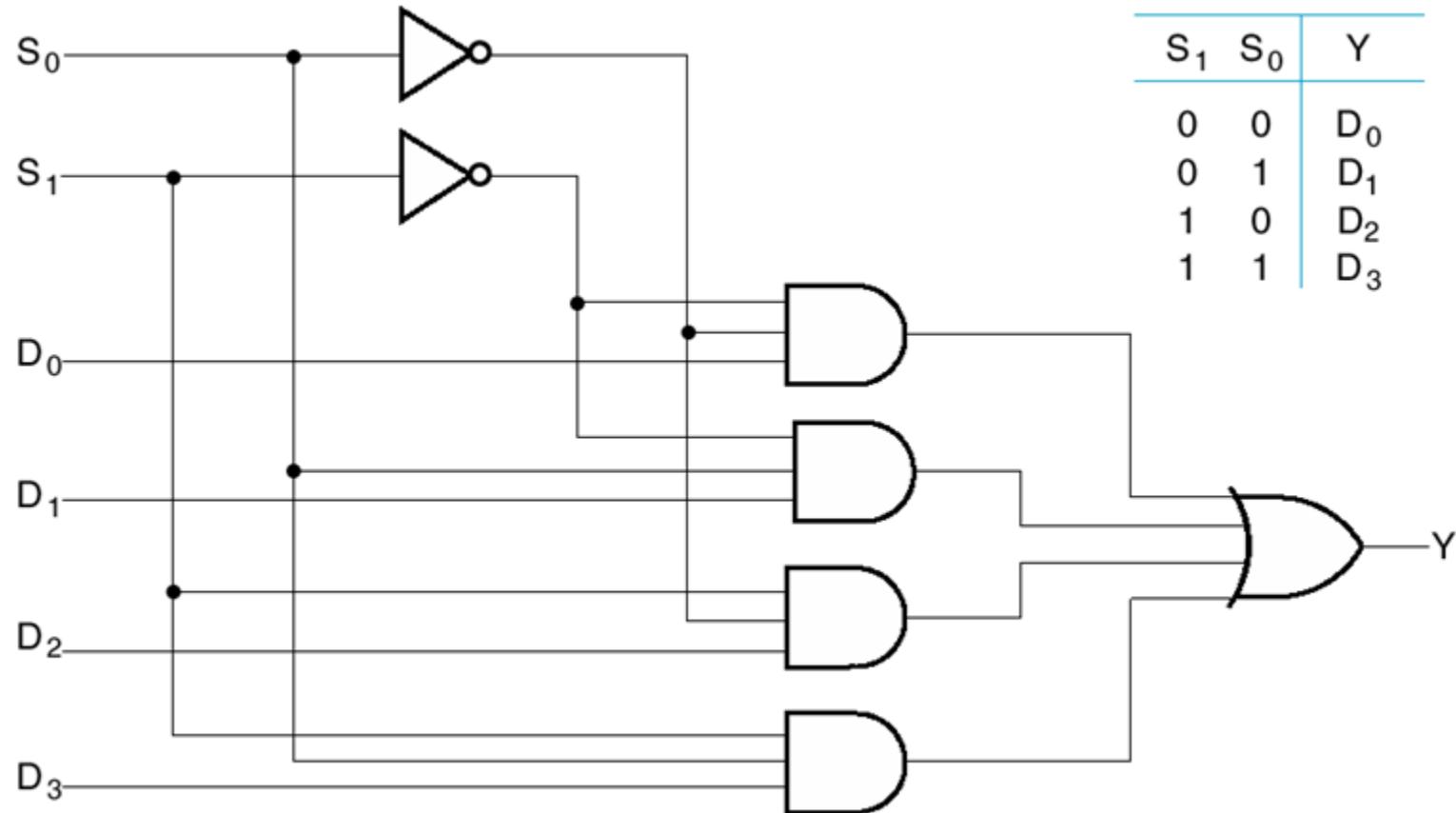


4-to-1 Multiplexer



Multiplexer Input Line Selection

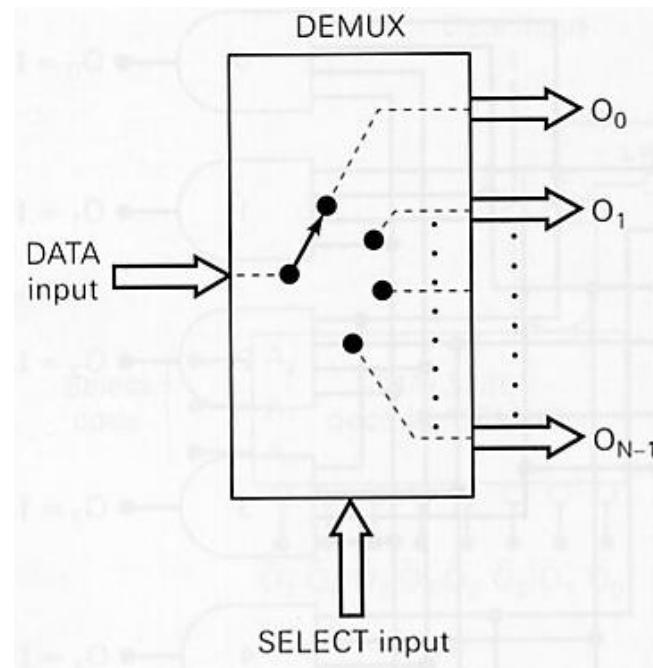
4-to-1 Multiplexer



4-to-1 Multiplexer

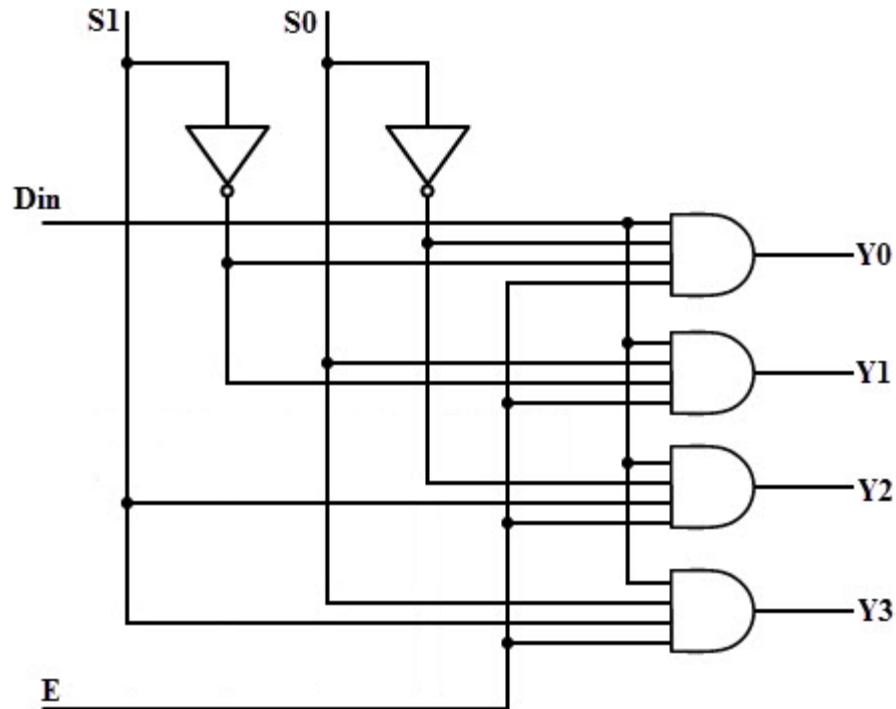
Demultiplexer

- A demultiplexer (DEMUX) has a single input and distributes it over 2^n outputs using n control inputs.



Demultiplexer

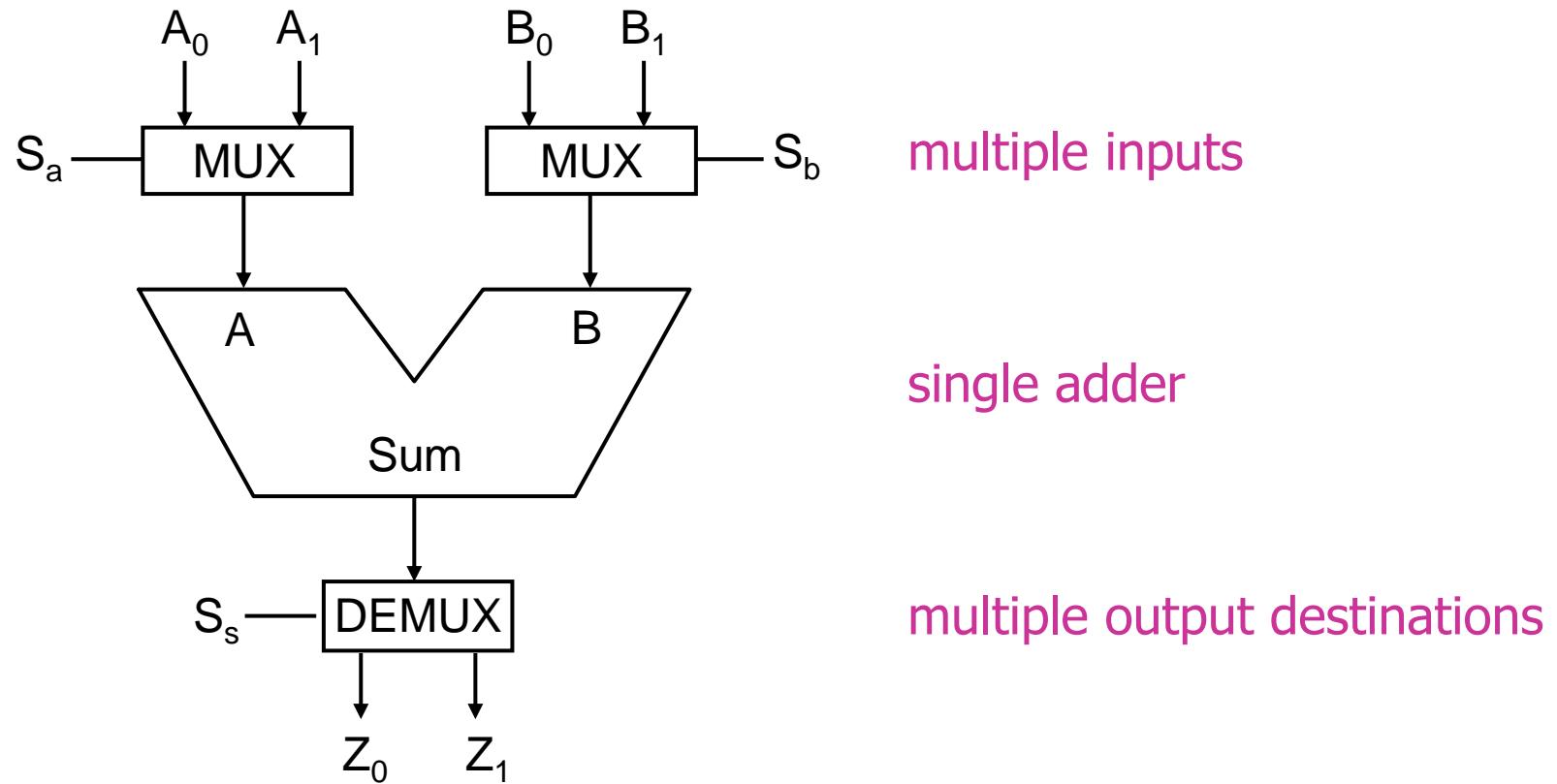
Data Input	Select Inputs		Outputs			
D	S ₁	S ₀	Y ₃	Y ₂	Y ₁	Y ₀
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0



1-to-4 Demultiplexer

MUX/DEMUX Applications

- Sharing complex logic functions such as sharing an adder: MUX select inputs and DEMUX distribute the sum



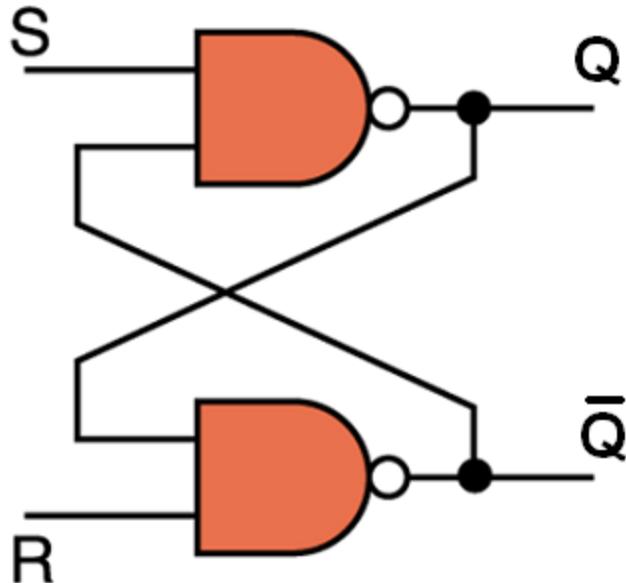
Sequential Circuits



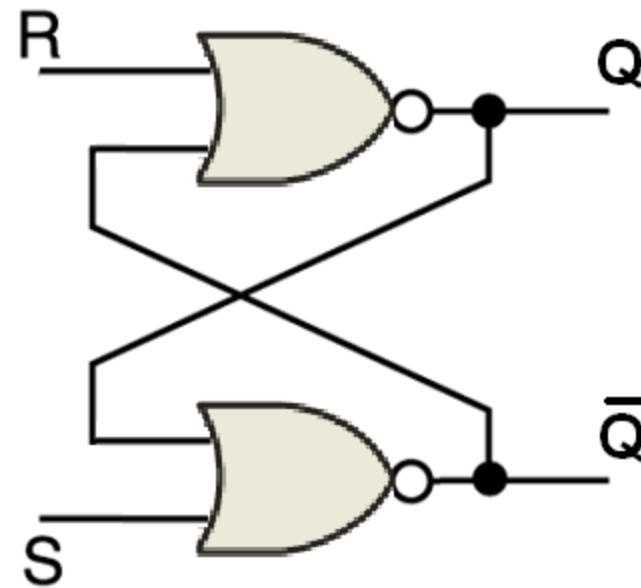
Sequential Circuits

- Digital circuits can also be used to store information.
- This application employs **sequential circuits**, because the output of the circuit is also used as input to the circuit. These circuits are the basic form for designing memory.

Circuits as Memory



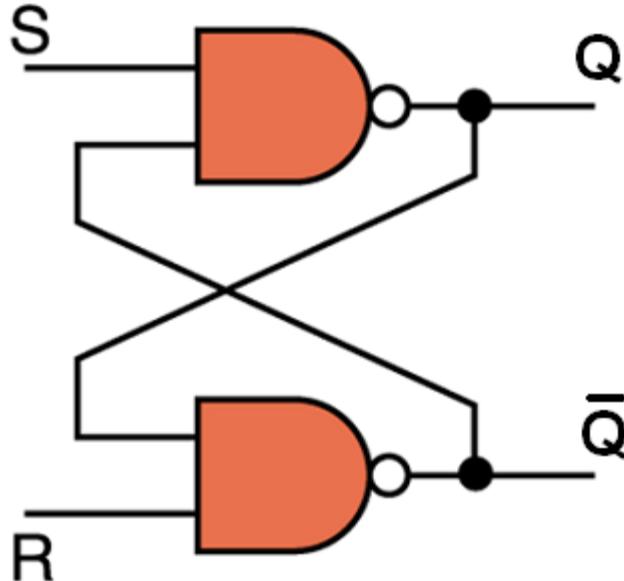
NAND Active-LOW Latch



NOR Active-HIGH Latch

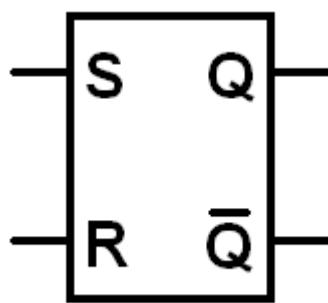
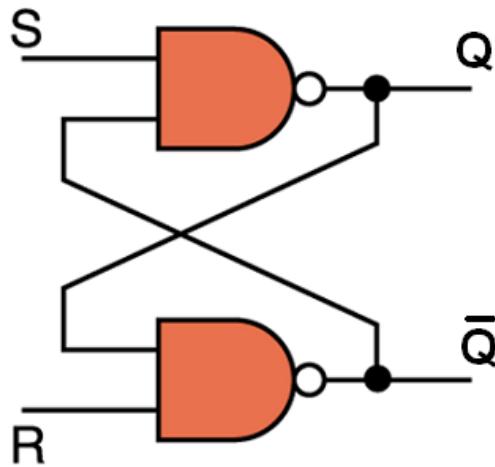
The S-R (Set-Reset) latch is the most basic type. It can be constructed from NOR gates or NAND gates.

SR Latch



- An S-R latch stores a single binary digit (1 or 0)
- There are several ways an S-R latch circuit can be designed using various kinds of gates

SR Latch



- The design of this circuit guarantees that the two outputs Q and \bar{Q} are always complements of each other.
- The value of Q at any point in time is considered to be the current state of the circuit.
- Therefore, if Q is 1, the circuit is storing a 1; if Q is 0, the circuit is storing a 0.

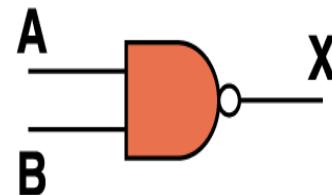
NAND Gate Recap

The **NAND** gate accepts **more than one** input signal. If all inputs are **1**, the output is **0**; otherwise, the output is **1**

Boolean Expression

$$X = (A \cdot B)'$$

Logic Diagram Symbol



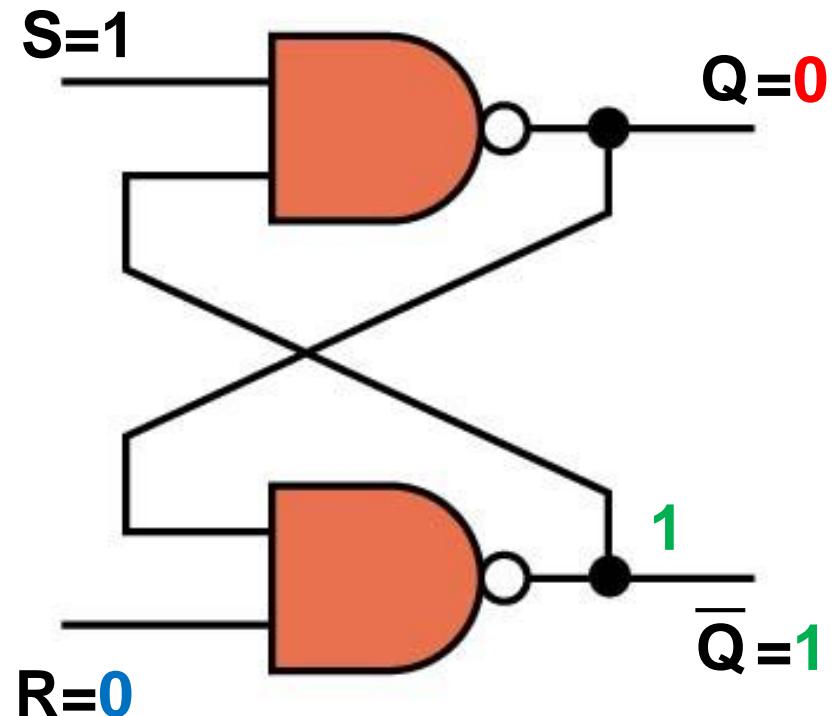
Truth Table

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

SR Latch

If $R=0$ and $S=1$

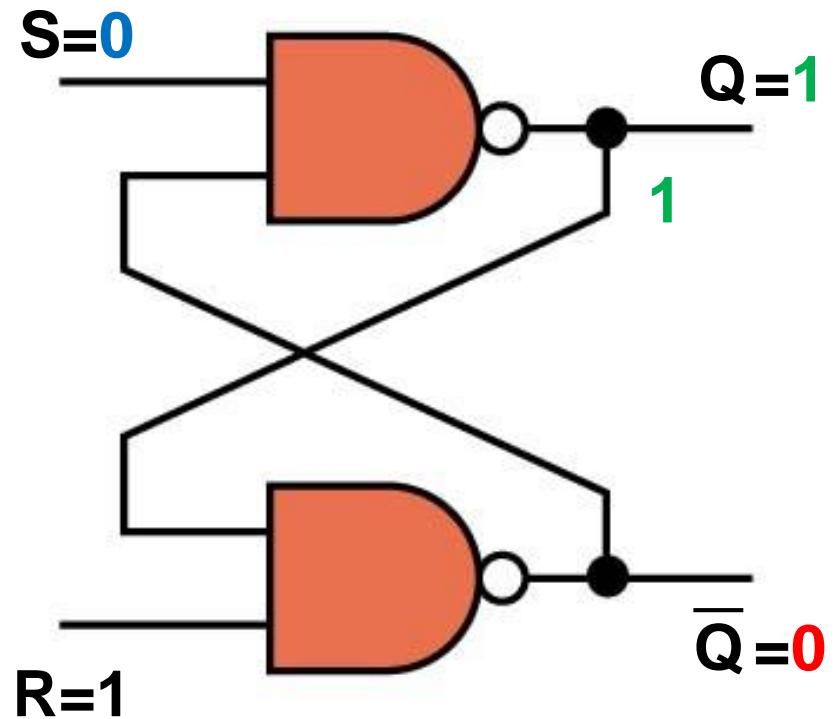
The latch is **Reset ($Q = 0$)**



SR Latch

If $R=1$ and $S=0$

The latch is Set ($Q = 1$)



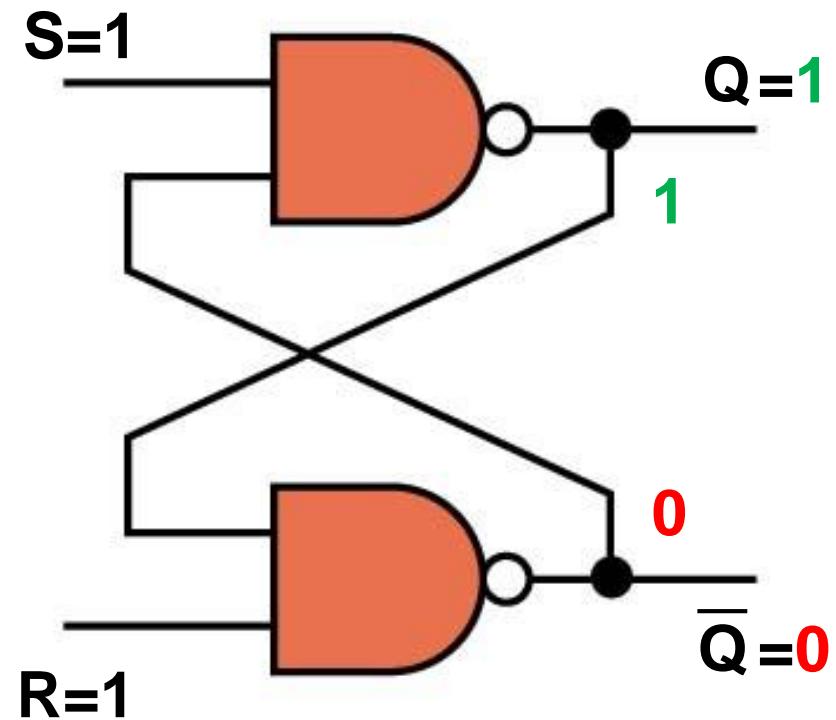
SR Latch

If $R=1$ and $S=1$

1 at one of the inputs does not guarantee the value of output. So, the second input has to be tested.

If $Q=1$

The latch stores the old value of Q ($Q=1$)



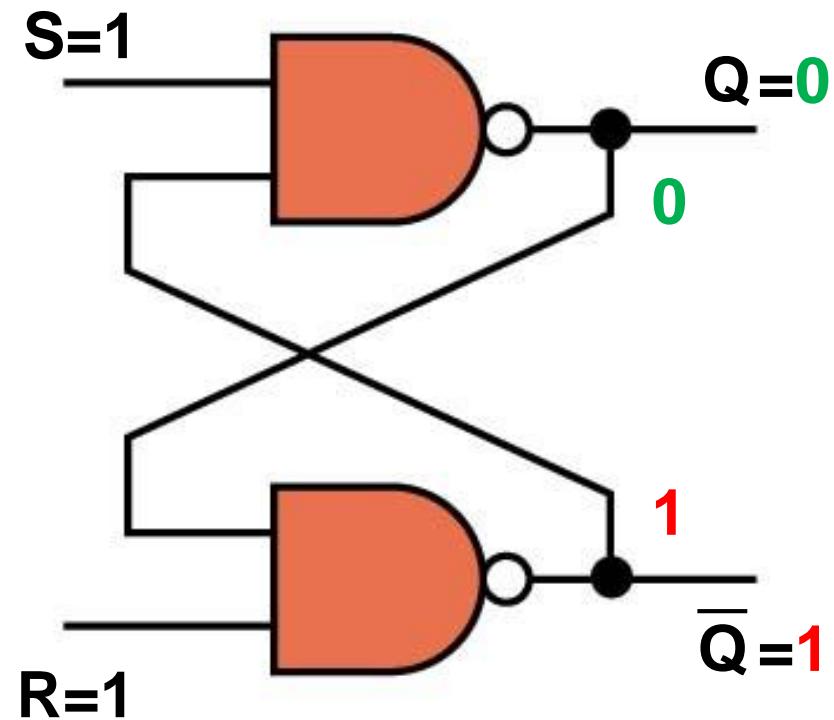
SR Latch

If $R=1$ and $S=1$

1 at one of the inputs does not guarantee the value of output. So, the second input has to be tested.

If $Q=0 \Rightarrow \bar{Q}=1$

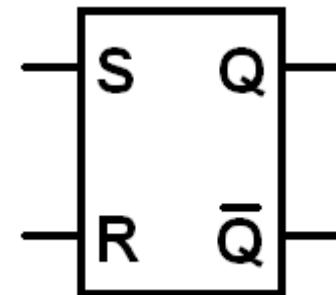
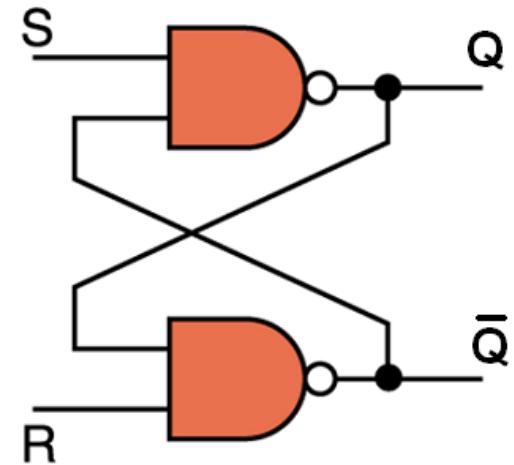
The latch stores the old value of Q ($Q=0$)



SR Latch

S	R	Q	Q'	
1	1	NC	NC	No change. Latch remained in present state.
0	1	1	0	Latch SET.
1	0	0	1	Latch RESET.
0	0	0	0	Invalid condition.

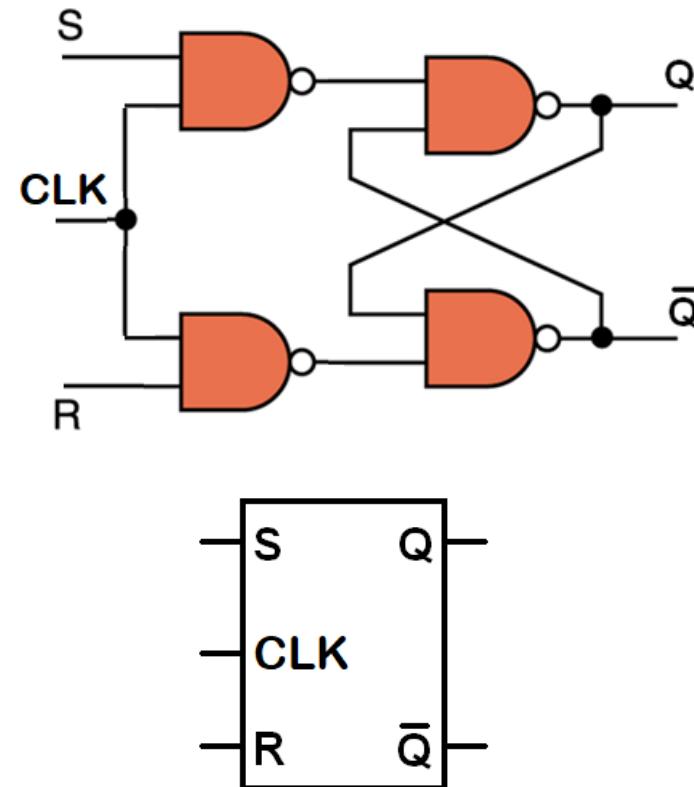
Truth Table



Gated SR Latch

- The S and R inputs are only enabled (i.e. affect the circuit) when the clock input is **1**. When the clock input is **0**, it stores the previous state.

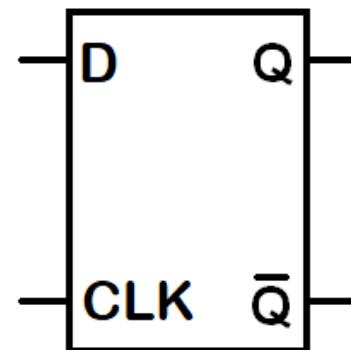
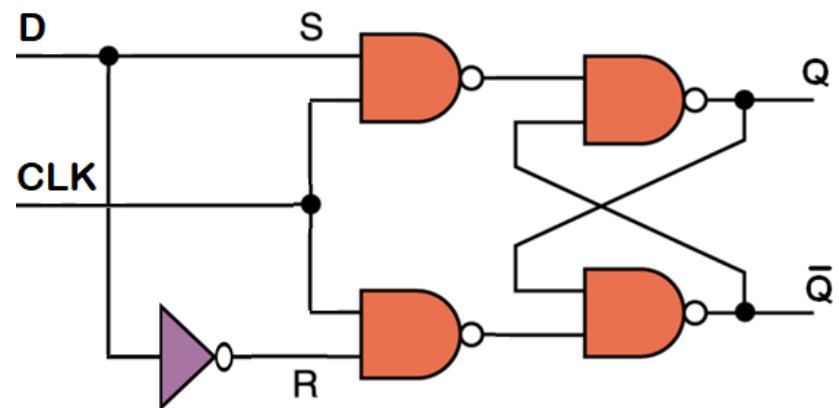
clk	S	R	Q
1	0	0	NC
1	0	1	0
1	1	0	1
1	1	1	X
0	X	X	NC



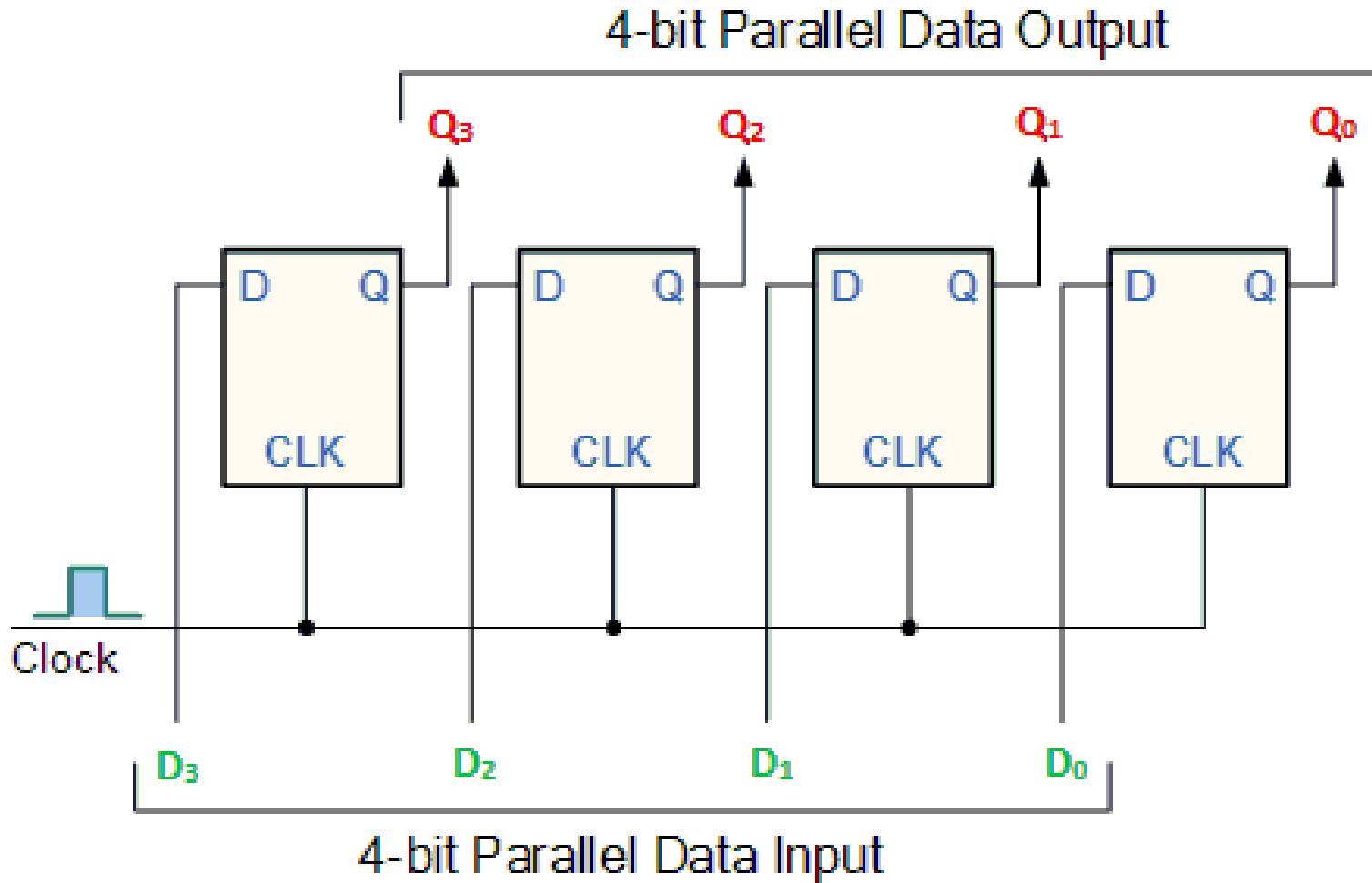
D-Type Latch

- It ensures that inputs S and R are never equal to 1 at the same time

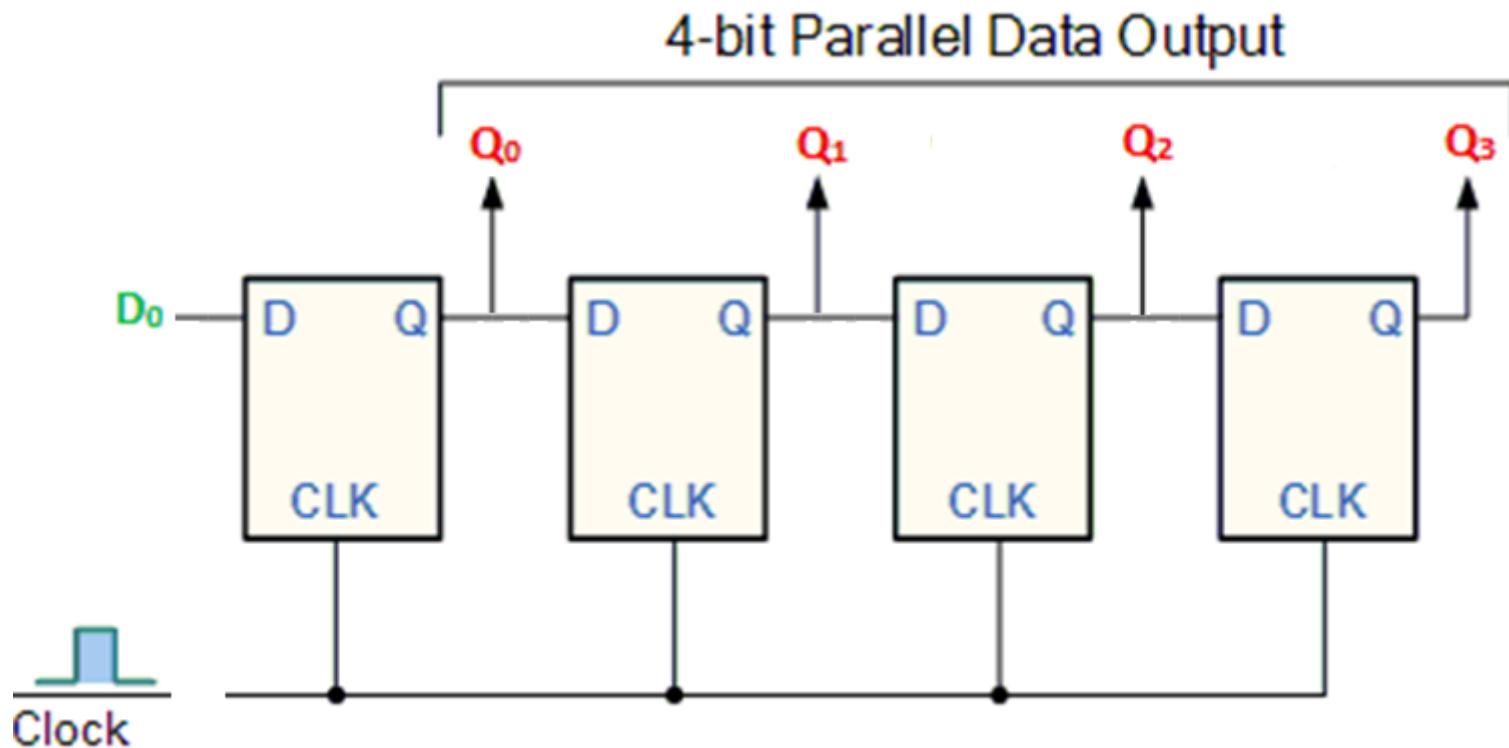
Clk	D	Q
0	0	NC
0	1	NC
1	0	0
1	1	1



4-bit Register



4-bit Shift Register



Serial in – Parallel out Shift Register

Further Reading

- Computer Organization and Architecture – Designing for Performance (10th Edition), William Stallings [Chapters: 11]
- Digital Fundamentals (11th Edition), Thomas L. Floyd [Chapters: 6, 7, 8, 9]
- Computer Organization and Design – The Hardware/Software Interface (5th Edition), David A. Patterson & John L. Hennessy [Appendix: A]
- Fundamentals of Computer Architecture, Mark Burrell [Chapter: 4, 5]