Computer Fundamentals

Lecture 9 : Logic & Control - Part III

Objectives

- After completing this module you will be able to:
  - Understand Combinational Circuits
    - Decoder, Multiplexer, Comparator, Arithmetic Circuits
  - Understand Sequential Circuits
    - SR, JK, D, T Flip-flops
  - Understand the arrangement of the ALU

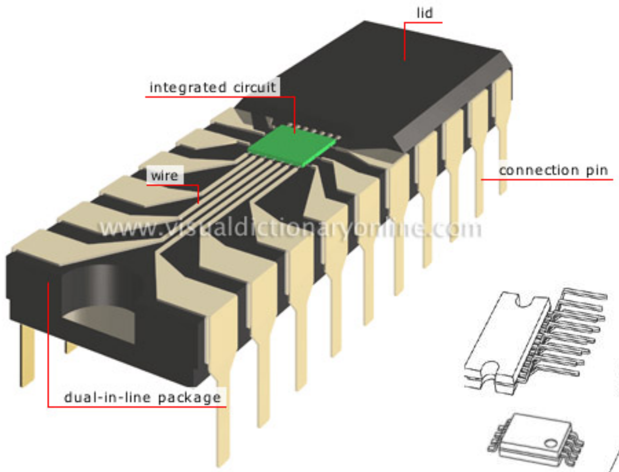
Lecture Outline

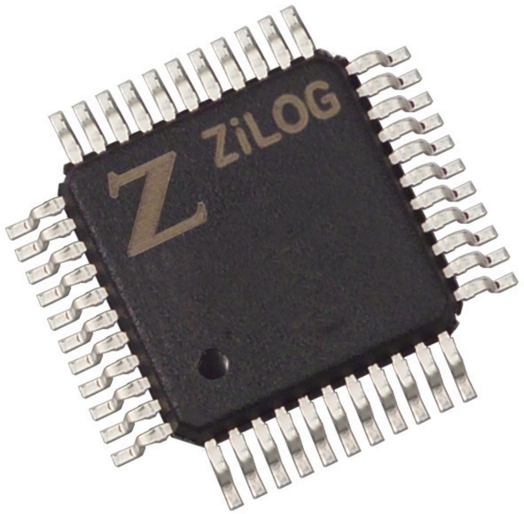
- Integrated Circuits
- Combinational Circuits
  - Half adder & Full adder
  - The use of the Decoder, Multiplexer, De-multiplexer.
  - Enable input of a digital device.
- Sequential Circuits (Fundamental Building Blocks of Memory)
  - Introduction to SR Latch (Level triggered Flip-flops).
  - Concept of the Flip-flop (Edge Triggered Flip-flops).

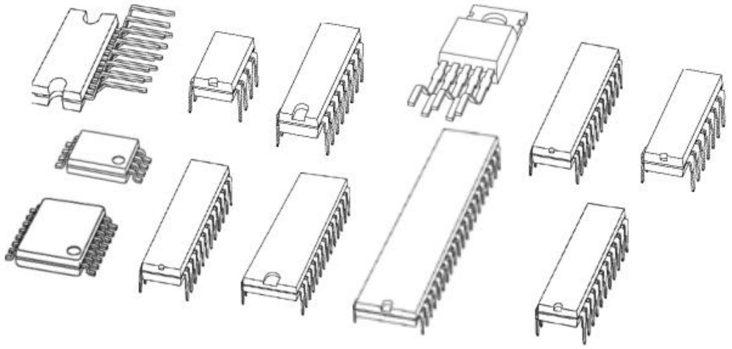
Integrated Circuits (ICs)

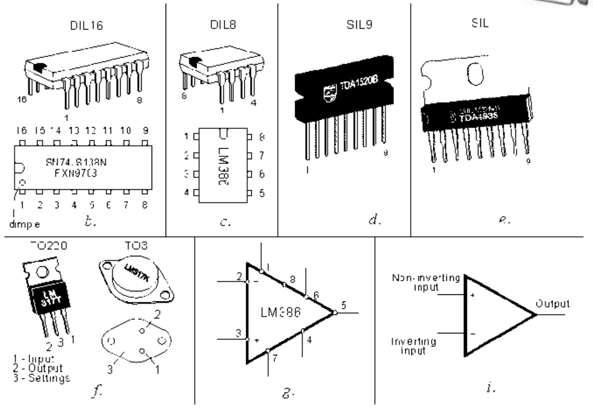
- An Integrated Circuit is a small silicon semiconductor crystal(chip), containing the electronic components for digital gates.
- Various gates are interconnected inside the chip to form the required circuits.
- Chip is mounted in a ceramic or plastic container and connections are welded by thin gold wires to the external pins to form the integrated circuit.

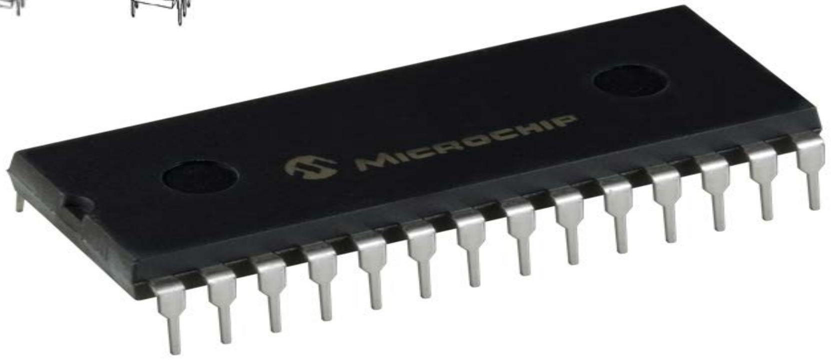
Integrated Circuits











Integrated Circuits (ICs)

- The IC pins provide access to the input and output terminals of individual gates as well as power supply to the whole device.

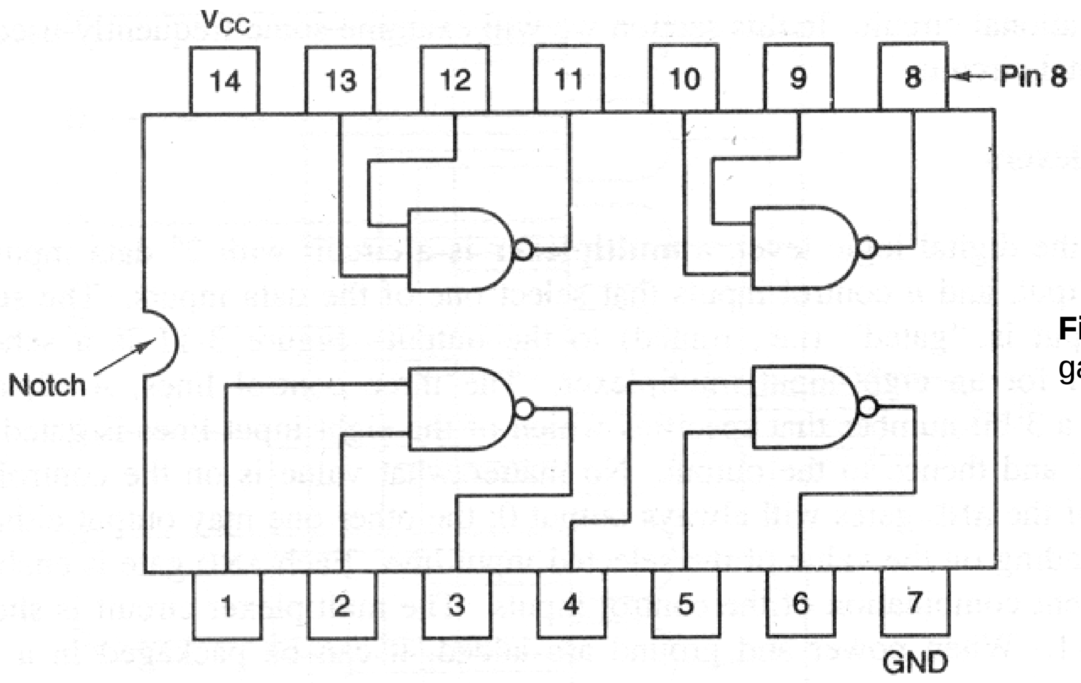


Figure 4-17. An SSI chip containing four gates.

Integrated Circuits (Contd.)

- ICs are roughly classified based on the number of gates they contain:
  - **SSI (Small Scale Integrated) circuit**: contain several independent gates in a single package (1 to 10 gates.)
  - **MS I (Medium Scale Integrated) circuit**: 10 to 200 gates are in a single package. Usually perform specific elementary digital functions such as adders, decoders and registers.

Integrated Circuits (Contd.)

- **LSI (Large Scale Integrated) circuit**: has 200 to few thousand gates in a single package. They include digital systems such as processors, memory chips and programmable modules.

- **VLSI (Very Large Scale Integrated) circuit**: >100,000 gates. Examples : large memory arrays and complex microcomputer chips.

  characteristics – size is small and low cost and this is revolutionized the computer system design technology by designing economical computers.

Combinational Circuits

- A connected arrangement of *logic gates* with a set of inputs and outputs
- At any given time, the binary values of the outputs are a function of the binary combination of the inputs.
- This circuits are employed in digital computers for generating binary control decisions and for providing digital components required for data processing.

Combinational Circuits

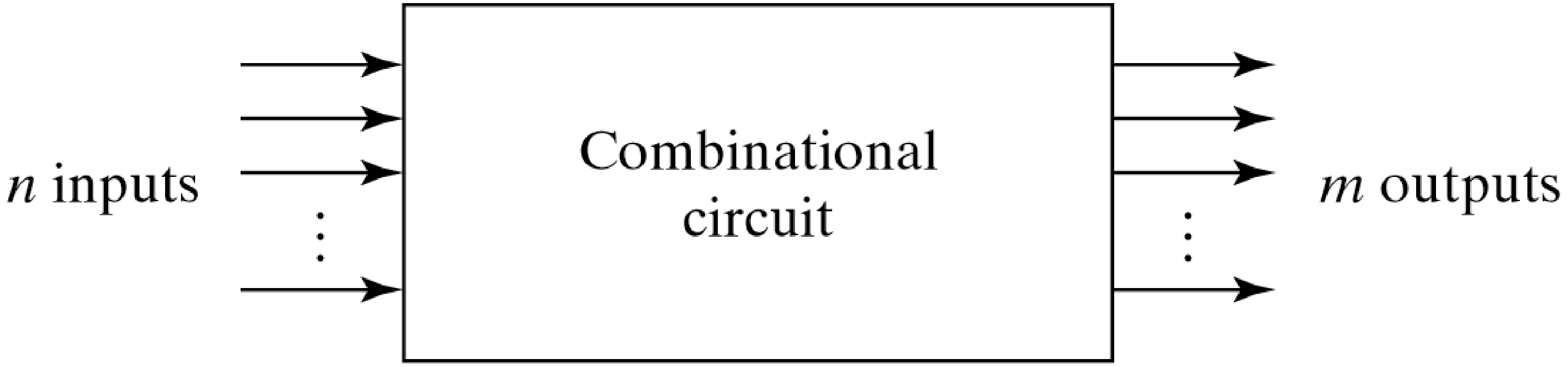


Fig. 4-1 Block Diagram of Combinational Circuit

Designing Combinational Circuits

In general we have to do following steps:

1. Problem description
2. Input/output of the circuit
3. Define truth table
4. Simplification for each output
5. Draw the circuit

Combinational Circuits

- To demonstrate the design of combinational circuits, two simple arithmetic circuits can be given
  - Half-Adder
  - Full-Adder

- Half-Adder
  - A combinational circuit that performs the arithmetic addition of two bits is called half adder.

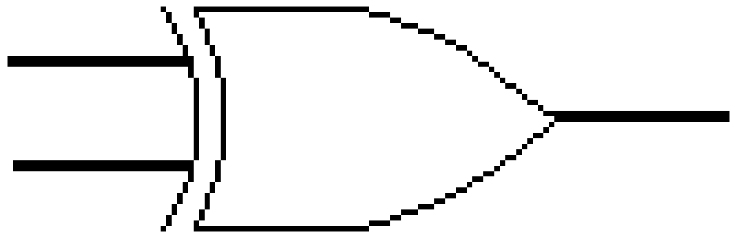
Combinational Circuits – Half-Adder

- Input variables of the Half-Adder
  - Augend and addend bits.
- The Output variables
  - sum and carry
- It is necessary to specify the two output variables because sum of 1 +1 is binary 10. which is two digits.

Exclusive-OR

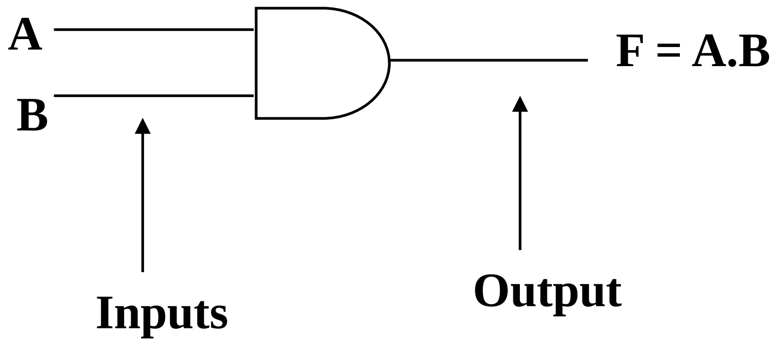
- The *XOR* (*exclusive-OR*) *gate* acts in the same way as the logical "either/or."
- The output is "true" if either, but not both, of the inputs are "true."
- The output is "false" if both inputs are "false" or if both inputs are "true."
- Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same.

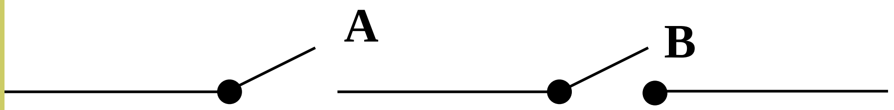
Exclusive-OR



Truth table for XOR gate		
INPUT		OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

AND Gate

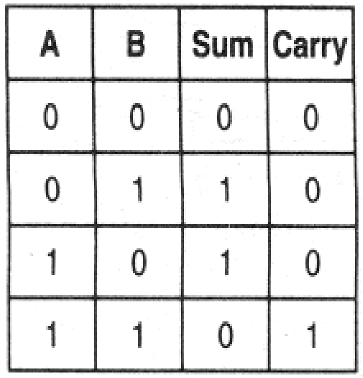




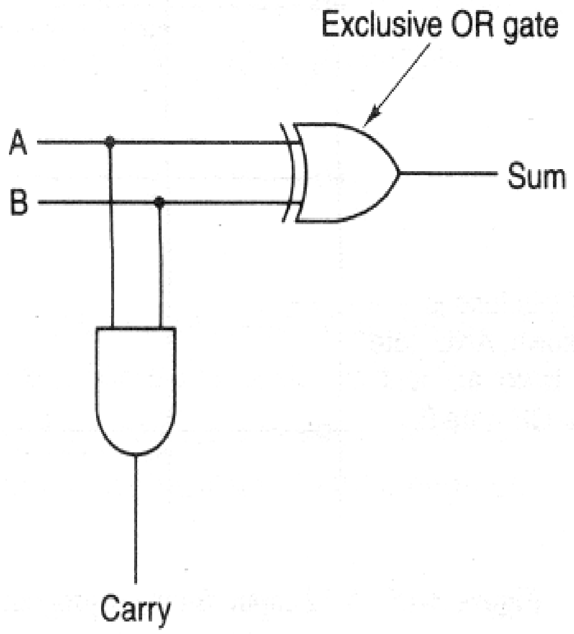
INPUT		OUTPUT	
A	B	F=A.B	
0	0	0	False as one or more inputs are false
0	1	0	
1	0	0	
1	1	1	True as all inputs are true

Combinational Circuits: Half Adder

- Half Adder : The sum is **XOR** operation and the carry an **AND**



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



Sum= A’B+AB’

$A \oplus B$

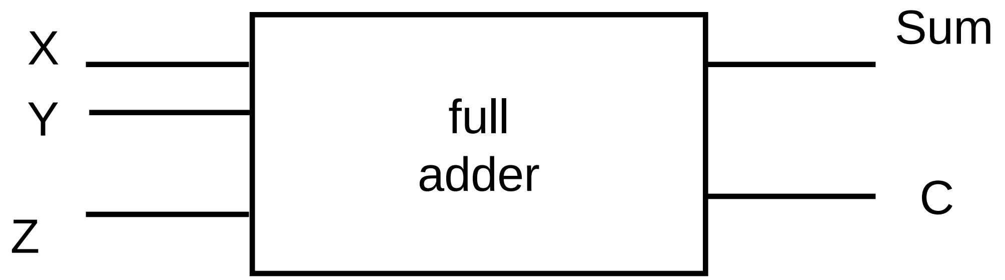
Carry=AB

Figure 4-23 (a) Truth table for 1 bit addition. (b) A circuit for a half adder

Combinational Circuits – Full Adder

1. A full adder is a combinational circuit that forms the arithmetic sum of three input bits.
2. It consists three inputs and two outputs.
3. Define a truth table.

Input			Output	
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Combinational Circuits – Full Adder

4. Simplification for each output

Input			Output	
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$C = xy'z + x'yz + xy$

$= z(xy' + x'y) + xy$

$= z(x \oplus y) + xy$

$S = xy'z' + x'y'z + xyz + x'yz'$

$= z'(xy' + x'y) + z(x'y' + xy)$

$= z'(x \oplus y) + z(x \oplus y)'$

$= a'b \bigcirc + ab'$ (let $a=z$, $b=x \oplus y$)

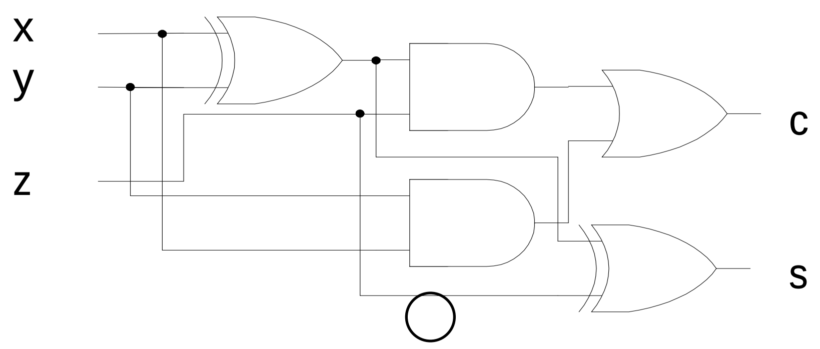
$= x \oplus y \oplus z$

$(x \oplus y)' = (xy' + x'y)'$
$= x'y'' + x''y'$
$= x'y + xy'$
$= (x'+y)(x+y')$ Duality
$= x'x + x'y' + xy + yy'$ Identity
$= x'y' + xy$

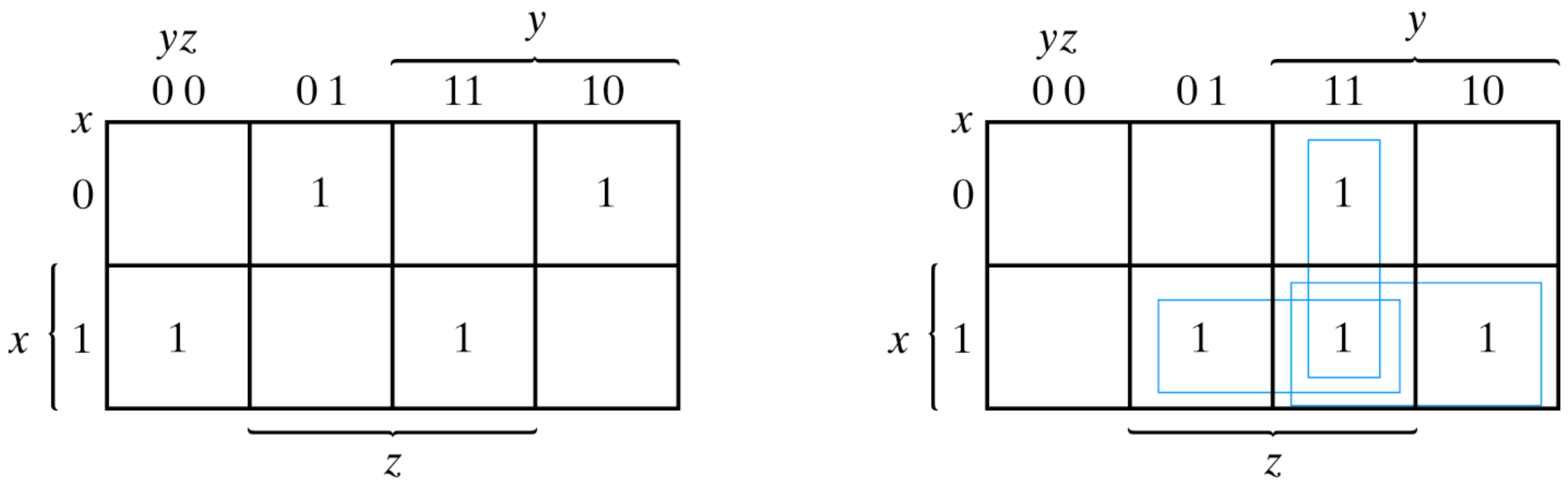
Combinational Circuits: Full Adder

5. Draw a diagram

Input			Output	
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Maps of Full-Adder

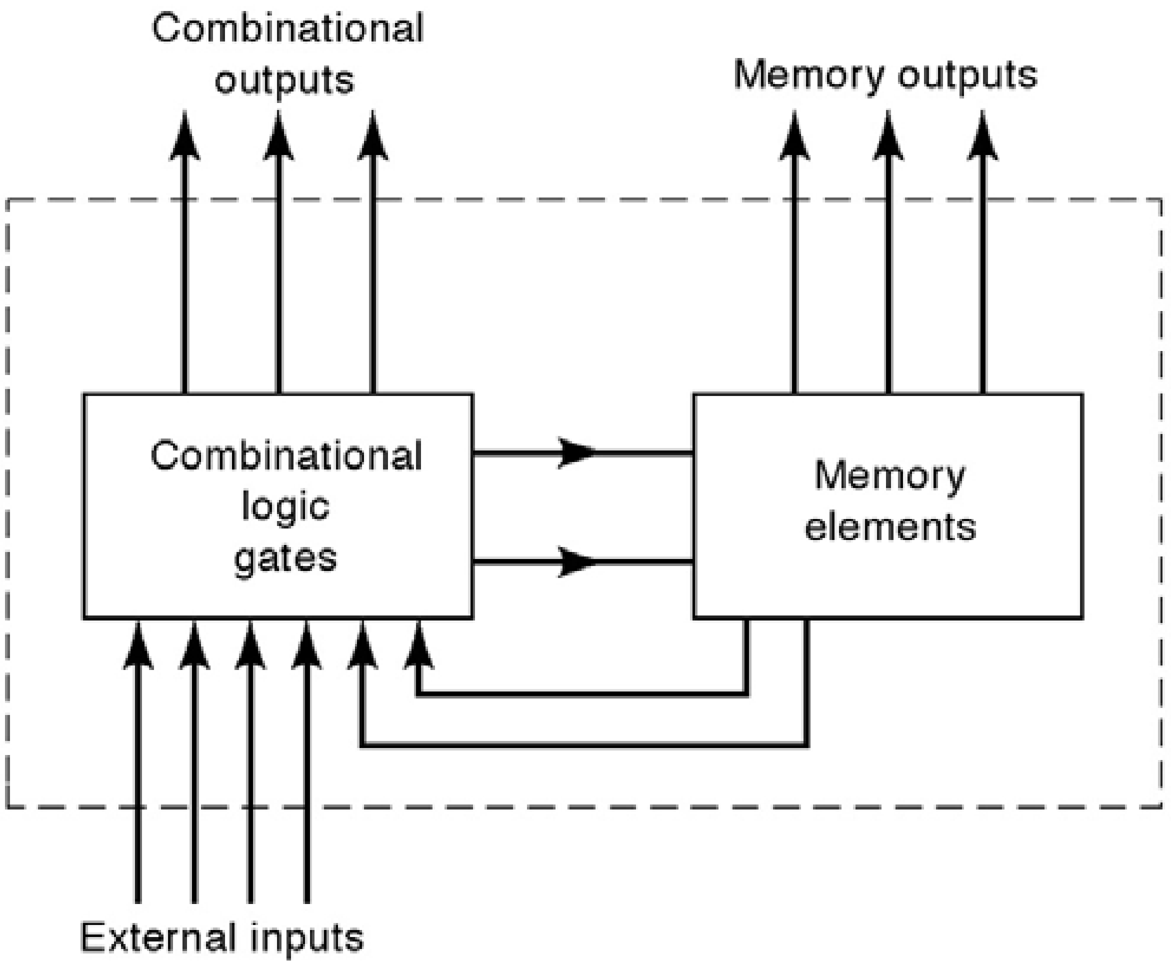


$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$\begin{aligned} S &= xy + xz + yz \\ &= xy + xy'z + x'yz \end{aligned}$$

Fig. 4-6 Maps for Full Adder

General Digital System Diagram



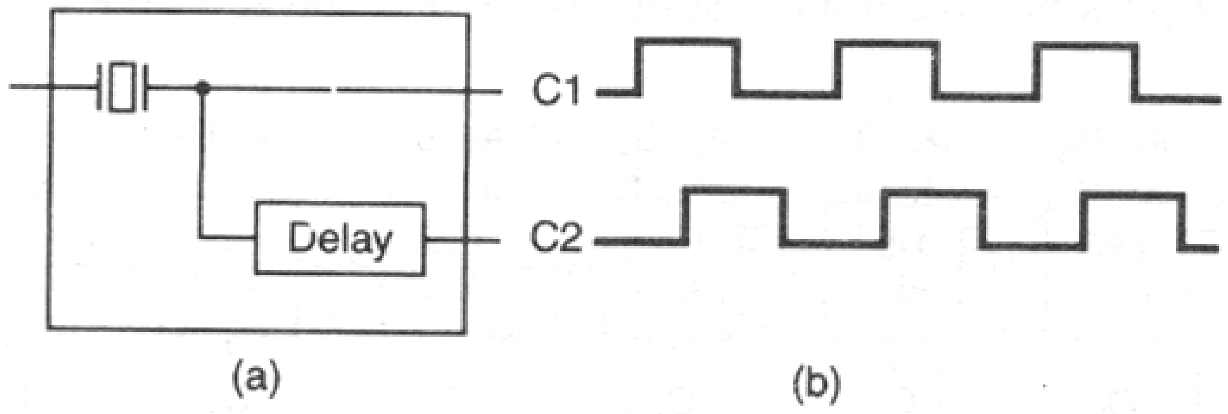
Synchronous and Asynchronous Sequential Logic

- **Synchronous**
  - the timing of all state transitions is controlled by a common clock
  - changes in all variables occur simultaneously
- **Asynchronous**
  - state transitions occur independently of any clock and normally dependent on the timing of transitions in the input variables
  - changes in more than one output do not necessarily occur simultaneously

Synchronous and Asynchronous Sequential Logic (Contd.)

- **Clock**
  - A clock signal is a square wave of fixed frequency Often, transitions will occur on one of the edges of clock pulses
  - Clock signals are used to maintain the desired timing in the circuits.
  - Clock circuits emit pulse trains of precise repetition interval and width.
  - Sometimes it is necessary to have one clock pulse train trail another by a fixed time.
  - A circuit with the appropriate delay may be inserted to achieve the desired phase shift

Clock Signals



Clock circuit and the clock waveforms

Flip-Flops (Latch)

- A flip-flop or latch is a circuit that has two stable states and can be used to store state information.
- This circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs
- Flip-flops are the fundamental element of sequential circuits.
  - (gates are the fundamental element for combinational circuits)
- Flip-flops is a binary cell capable of storing one bit information and basic storage element in sequential logic

Flip-Flops

- Flip-flops have two outputs
  - One for the normal value (Q)
  - One for the complement value the bit stored it ($\overline{Q}$).
- Flip-flops can be either simple (transparent or asynchronous) or clocked (synchronous)
- the transparent ones are commonly called latches.[
- The word *latch* is mainly used for storage elements
- The clocked devices are described as *flip-flops*.

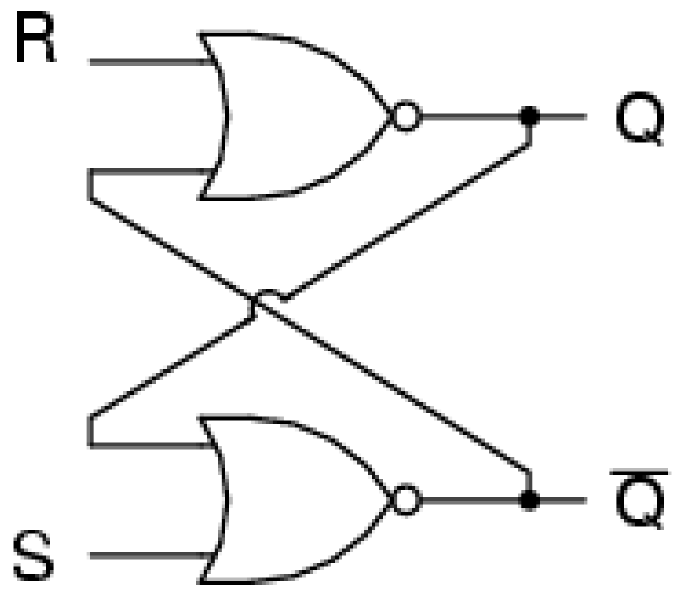
Flip-Flops(Contd.)

- **Usage:** Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

- **Main types of flip-flops**
  - S-R(Set-Reset) Flip-Flop
  - D (Data or Delay) Flip Flop
  - J-K Flip-Flop

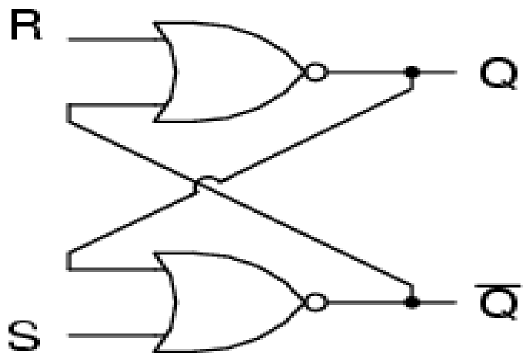
S-R Flip-Flops

- A RS-flip-flop is the simplest possible memory element.
- It is constructed by feeding the outputs of two **NOR** gates back to the other **NOR** gates input.



S	R	Action
0	0	Keep state
0	1	Q=0
1	0	Q=1
1	1	Restricted Combination

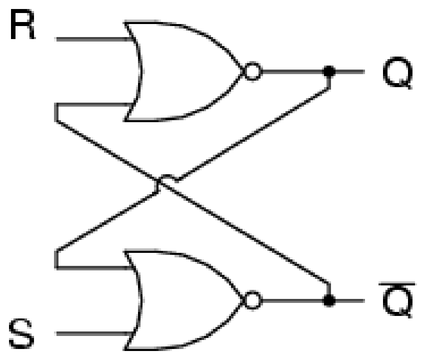
S-R Flip-Flops



S	R	Action
0	0	Keep state
0	1	Q=0
1	0	Q=1
1	1	Restricted Combination

- **S=0 and R=0:** Assume the flip flop is set (Q=0 and $\overline{Q}$=1), then the output of the top NOR gate remains at Q=1 and the bottom NOR gate stays at $\overline{Q}$=0.
- Similarly, when the flip flop is in a reset state (Q=1 and $\overline{Q}$=0), it will remain there with this input combination.
- Therefore, with inputs S=0 and R=0, the flip flop remains in its state.

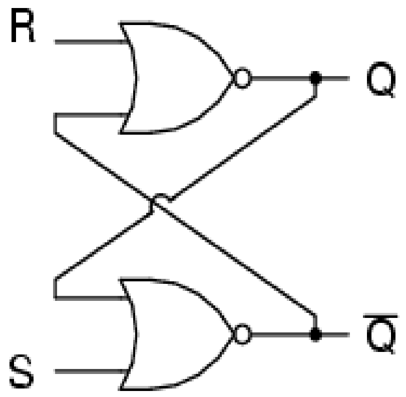
S-R Flip-Flops



S	R	Action
0	0	Keep state
0	1	Q=0
1	0	Q=1
1	1	Restricted Combination

- **S=0 and R=1:** Similar to the arguments above, the outputs become Q=0 and $\overline{Q}$=1.
- We say that the flip flop is reset.

S-R Flip-Flops



S	R	Action
0	0	Keep state
0	1	Q=0
1	0	Q=1
1	1	Restricted Combination

- **S=0 and R=0:** Assume the flip flop is set (Q=0 and $\overline{Q}$=1), then the output of the top NOR gate remains at Q=1 and the bottom NOR gate stays at $\overline{Q}$'=0.
- Similarly, when the flipflop is in a reset state (Q=1 and $\overline{Q}$'=0), it will remain there with this input combination.
- Therefore, with inputs S=0 and R=0, the flipflop remains in its state.
- **S=1 and R=1:** This input combination must be avoided.