CS 221
Logic Design

Fall 2021

By Wessam El-Behaidy & Salwa Osama

Registers & Counters

Lecture 9

Sequential Circuit

- A circuit with flip-flops is considered a sequential circuit even in the absence of combinational gates
- Circuits that include flip-flops are usually classified by the function they perform rather than by the name of the sequential circuit.
- Two such circuits are **registers** and **counters**

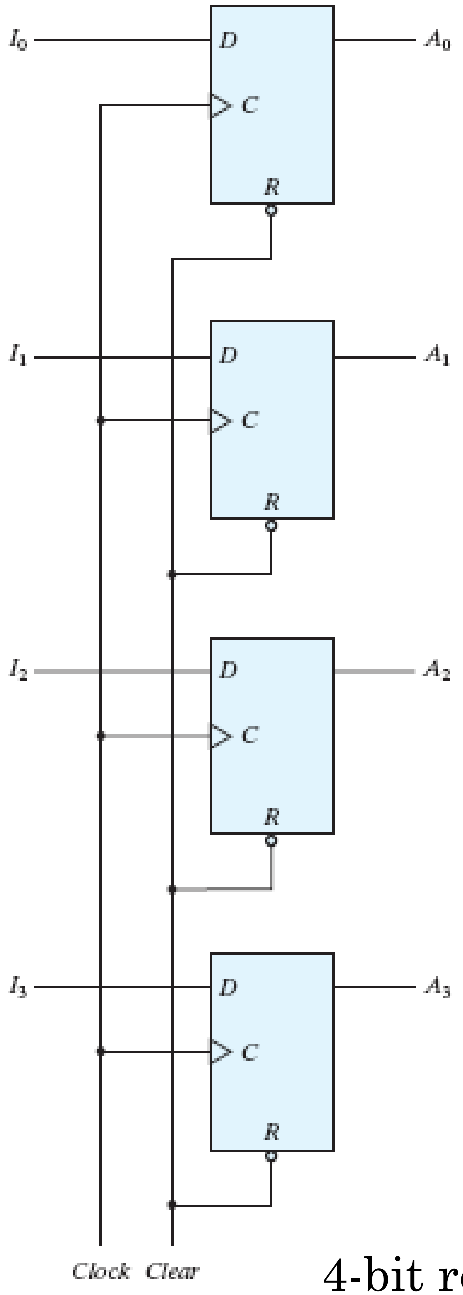
Register

- A *register* is a group of flip-flops.
- An *n-bit* register consists of:
  - A group of *n* flip-flops capable of storing *n* bits of binary information
- Its broadest definition,
  - A register consists of a group of flip-flops together with gates that affect their operation.
    - Flip-flops hold the binary information and
    - The gates determine how the information is transferred into the register

Simplest register

- 4-bit register consists of only

4 D-type flip-flops
  - Common clock input triggers all flip-flops on the positive edge
  - When input **Clear** (**R**eset) goes to **active low(zero)➔** all flip-flops are reset asynchronously.
  - R inputs must be at logic 1 during normal clocked operation



4-bit register

Register with parallel load

- The transfer of new information into a register is referred to as ***loading*** or ***updating*** the register
- If all the bits of the register are loaded simultaneously with a common clock pulse, we say ***loading*** is done in ***parallel.***
- Previous 4-bit register (fig.6.1) will load the 4 inputs in parallel
  - A problem , with this configuration, is when the contents of the register must be unchanged.

BASIC COMPUTER REGISTERS

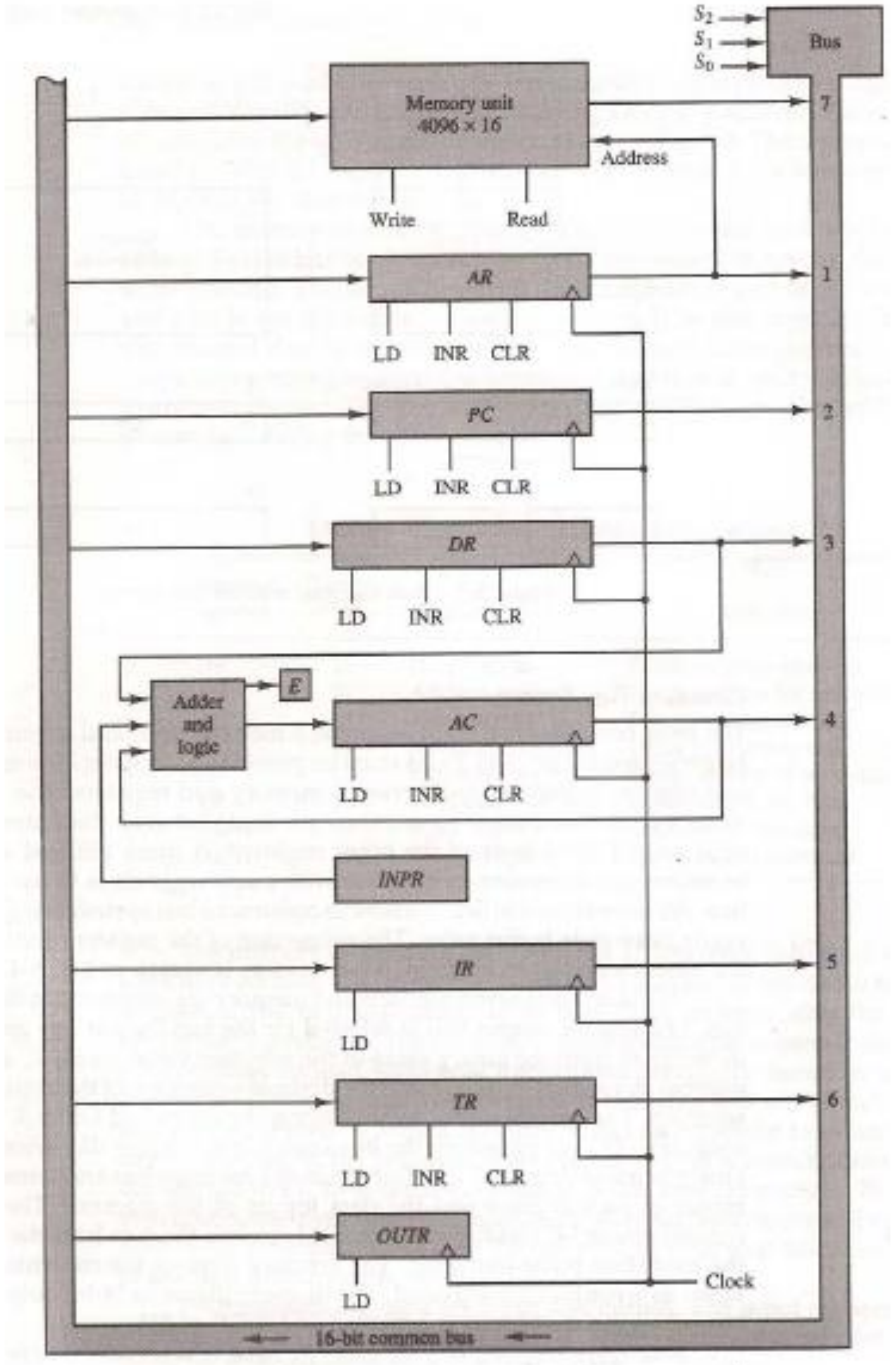


Figure 5-4 Basic computer registers connected to a common bus.

Register with parallel load (Cont.)

- Unchanged the content of the 4-bit register needs:

a) The inputs must be held constant

➔ the data bus would be unavailable for other traffic

b) The clock must be inhibited from reaching the register by controlling the clock input signal with an enabling gate

➔ inserting gates into the clock path is ill advised because it produces uneven propagation delays between the master clock and the inputs of flip-flops

To fully synchronize the system, we must ensure that all clock pulses arrive at the same time anywhere in the system, so that all flip-flops trigger simultaneously

Master clock generator & Control signal

- ***Master clock generator***
  - It supplies a continuous train of clock pulses
  - The pulses are applied to all flip-flops and registers in the system
- ***Control Signal***
  - A separate ***control signal*** must be used to decide which register operation will execute at each clock pulse

REGISTER WITH PARALLEL LOAD (CONT.)

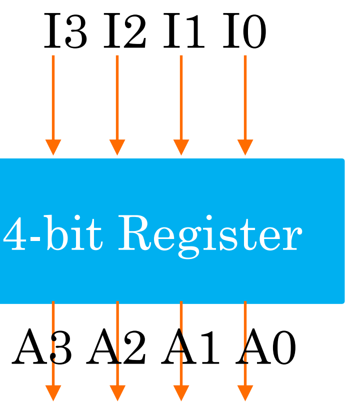
- For this reason, it is advisable to:
  - control the operation of the register with the D inputs
- A 4-bit register with a load control input
  - The load input determines the action to be taken with each clock pulse
    - When load =1, four external inputs are transferred into register with next positive transition
    - When load =0, the output of the flip-flops are connected to their respective inputs ("no change" condition)

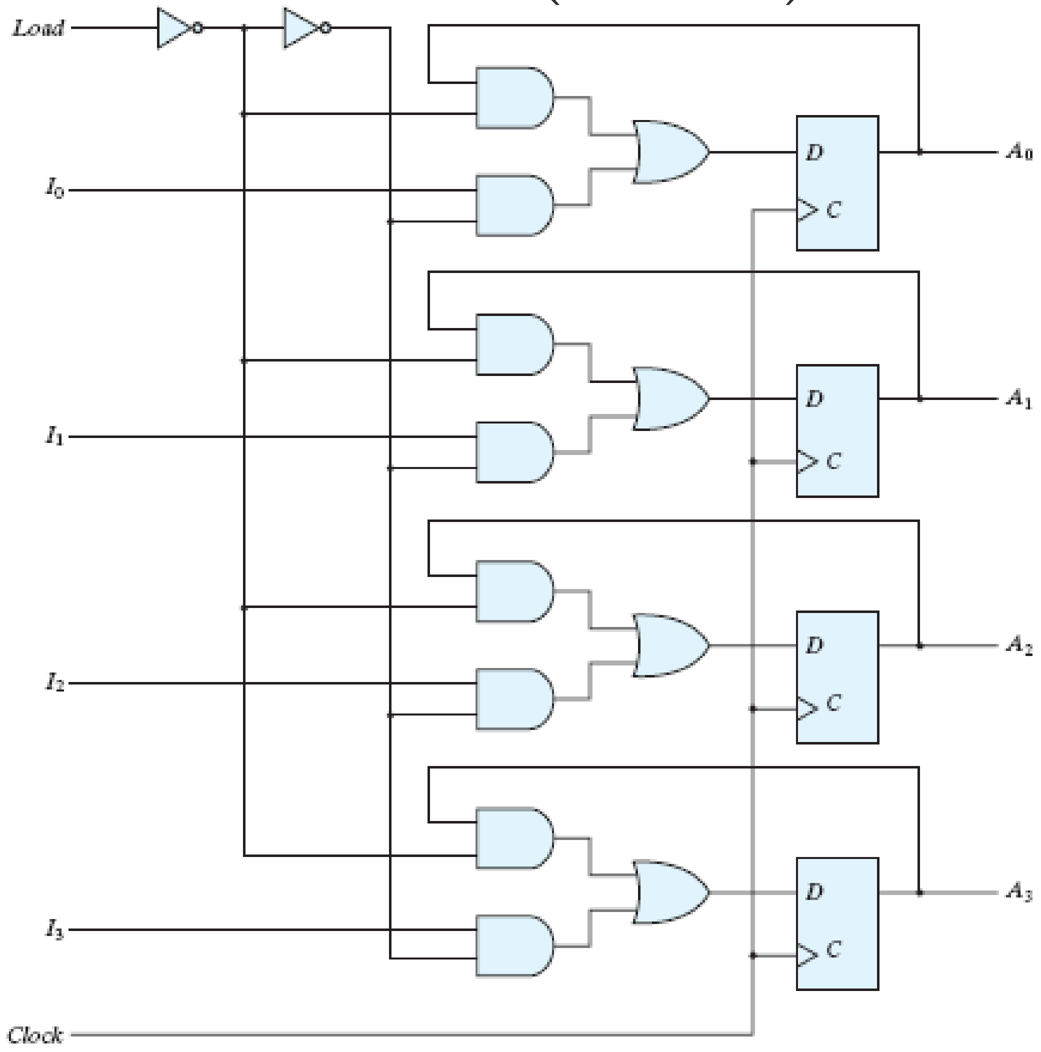
L(oad)	D_A
0	A(t)
1	I(nputs)

Register with parallel load (Cont.)

L(oad)	D_A
0	A(t)
1	I(nputs)

D_A= L'A +LI

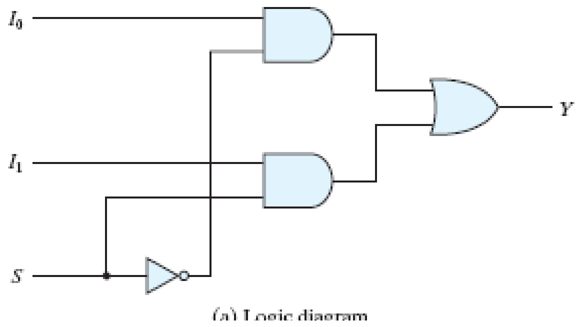




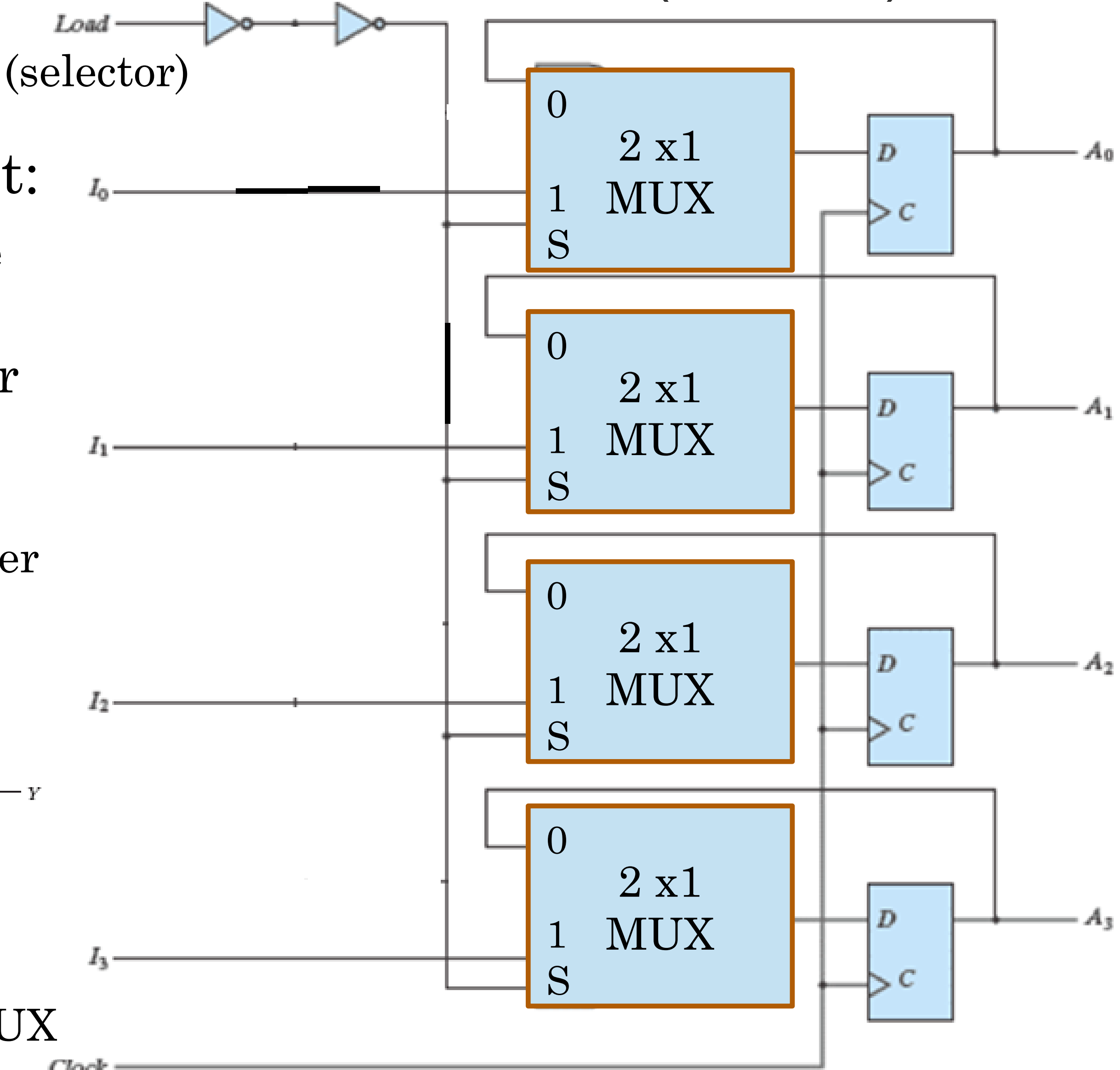
Register with parallel load (Cont.)

- The additional gates implement:
  - A 2 x 1 Mux whose output drives the input to the register with either:
    - the data bus or the output of the register

D_A= L'A +LI



Internal design of 2x1 MUX



Shift Register

- A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction
- The logical configuration of a shift register:
  - A chain of flip-flops in cascade, with the output of one flip-flop connected to the input to the next flip-flop

Shift Register (Cont.)

- The simplest shift register, shown in fig. 6.3
  - This shift is unidirectional
  - Each clock pulse shifts the contents of the register one bit position to the right
  - Serial input (SI): determines what goes into the leftmost flip-flop
  - Serial output (SO): is the output from the rightmost flip-flop

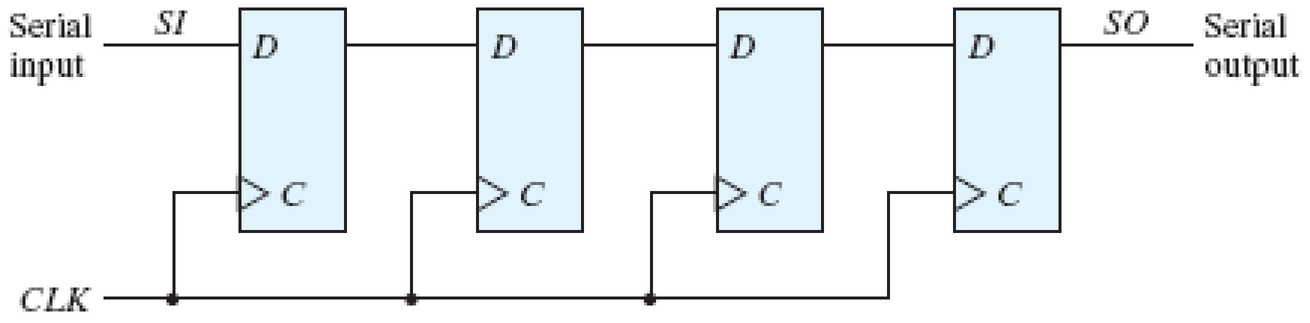


fig. 6.3: 4-bit shift register

Serial & parallel Mode

Serial Mode

The register have a single serial output.

Information is transferred one bit at a time while the registers are shifted in the same direction

Parallel Mode

Information is available from all bits of a register

and

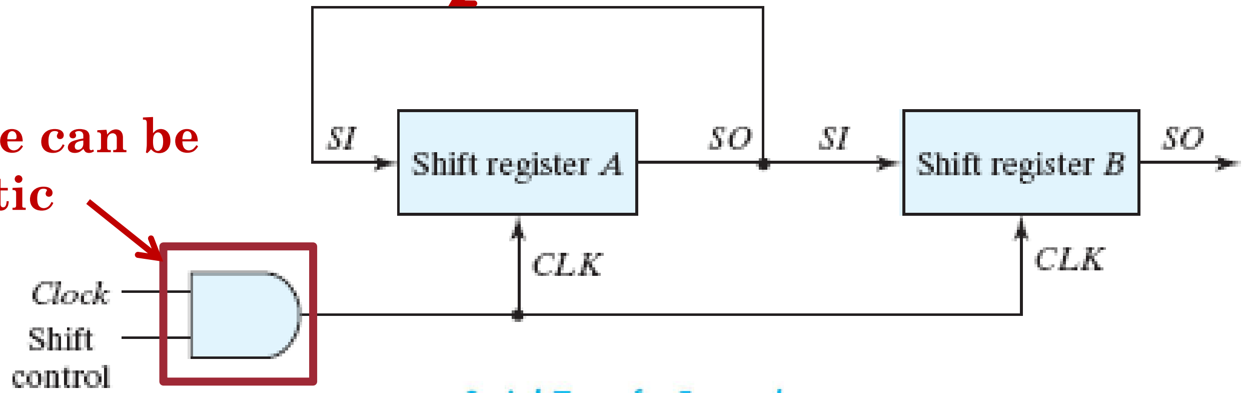
all bits can be transferred simultaneously during one clock pulse

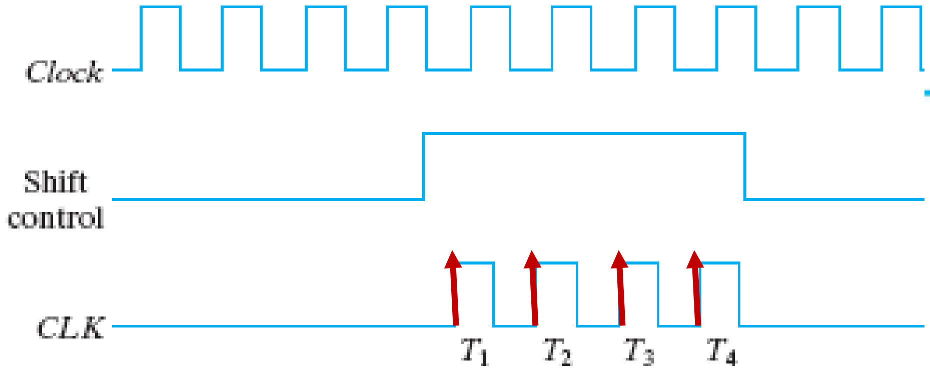
Serial transfer from register A to B

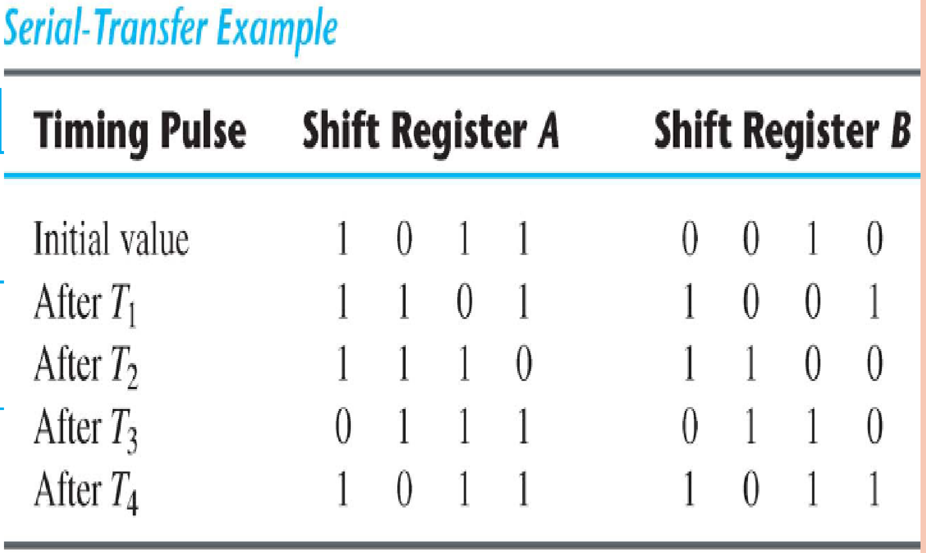
To prevent loss of information stored in source register

In practice can be problematic

control unit

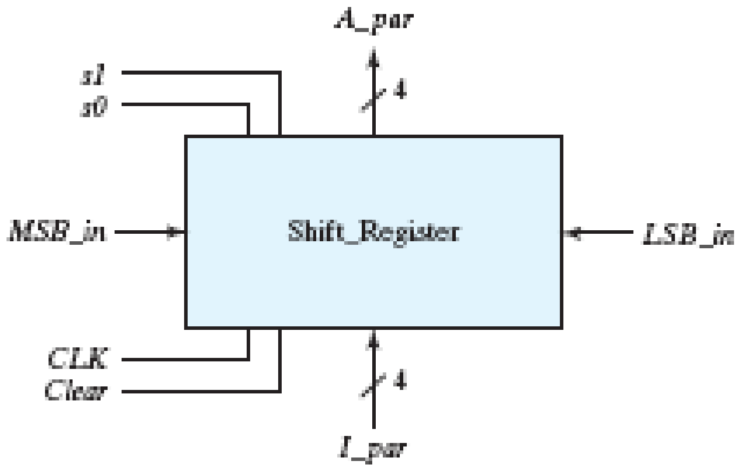





Serial-Transfer Example

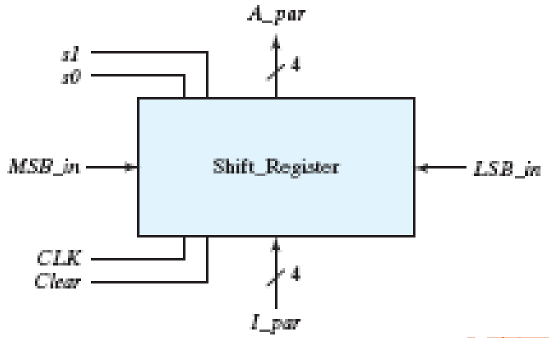
Timing Pulse	Shift Register A				Shift Register B			
Initial value	1	0	1	1	0	0	1	0
After T_1	1	1	0	1	1	0	0	1
After T_2	1	1	1	0	1	1	0	0
After T_3	0	1	1	1	0	1	1	0
After T_4	1	0	1	1	1	0	1	1

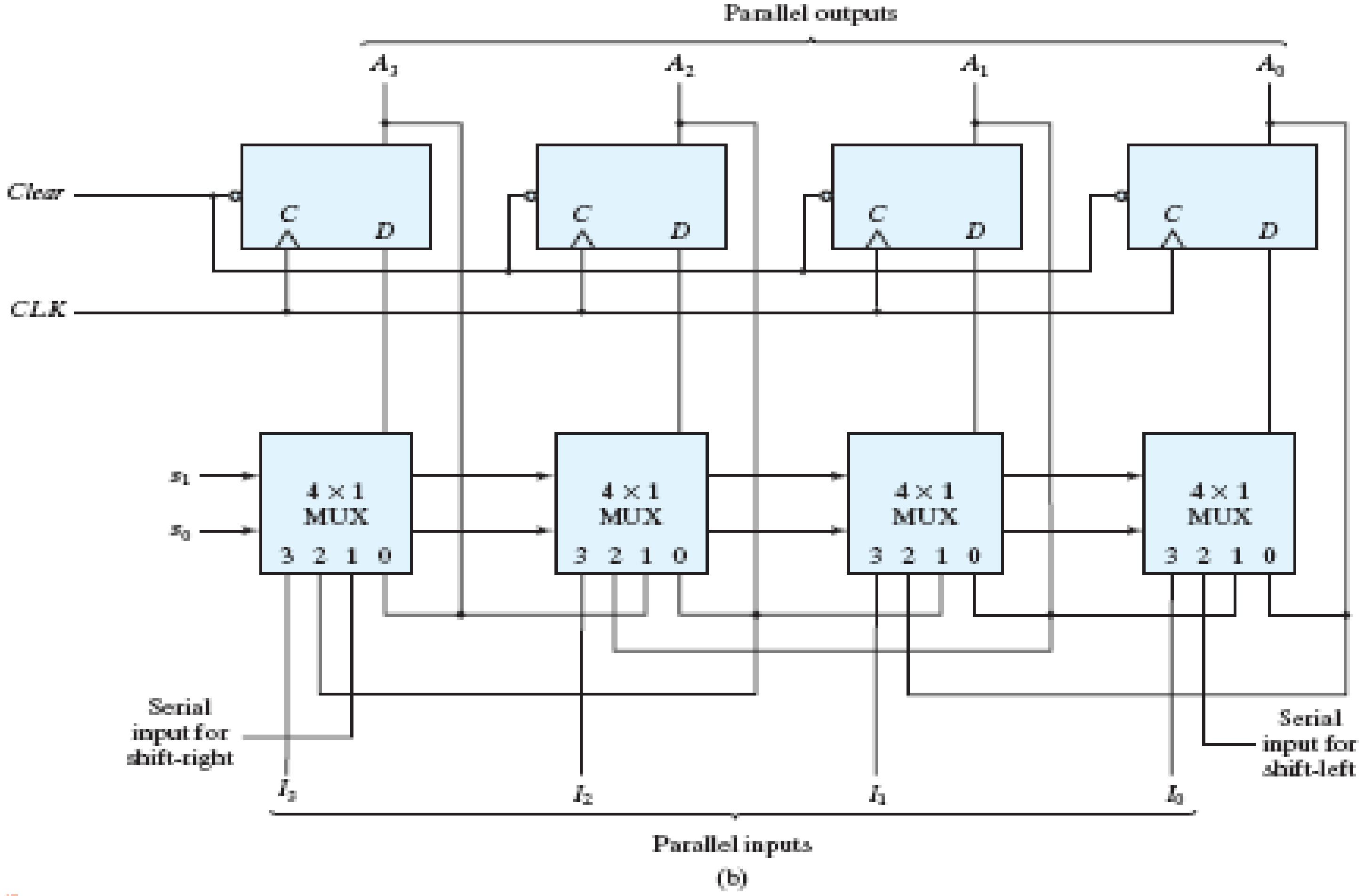
UNIVERSAL SHIFT REGISTER



- It has the following capabilities:
  - A *clear* control to clear the register to 0
  - A *clock* input to synchronize the operations
  - A *shift-right* control to enable the shift-right operation and the serial input and output lines associated with the shift right
  - A *shift-left* control to enable the shift-left operation and the serial input and output lines associated with the shift left
  - A *parallel-load* control to enable a parallel transfer and the *n* input lines associated with the parallel transfer
  - *n* parallel output lines
  - A *control state* that leaves the information in the register unchanged in response to the clock.

4-bit Universal shift register





Mode Control

s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

PARALLEL-TO-SERIAL & SERIAL-TO-PARALLEL CONVERSION

- Shift register are often used to interface digital systems situated remotely from each other

Ex: to transmit an n bits between 2 points

- If the distance is far,

  ➔ it will be expensive to transmit n-bit in parallel

  ➔ it is economical to use a single line and transmit the information serially

- ***The transmitter:***
  - Accepts the n-bit data in parallel into a shift register and transmit it serially. “parallel to serial conversion”
- ***The receiver:***
  - Accepts the data serially into a shift register.
  - When all n-bits are received, they are taken from the output of the register in parallel . “ serial-to-parallel conversion”

COUNTER

- A *counter* is essentially a register (*special type of register*) that goes through a predetermined sequence of binary states.
- The gates in the counter are connected in such a way as to produce the prescribed sequence of states.
- Counters are available in 2 categories:
  - Ripple counters
    - C input of some or all flip-flops are triggered, not by the common clock , but rather by the transition occurs in other flip-flop outputs.
  - Synchronous counters
    - C input of all flip-flops receive common clock

Count enable=1, the counter counts
Count enable=0, no change state

Synchronous Counters: binary counter (count up)

- In synchronous binary counter,
  - Flip-flop in the *least significant* position is complemented
    - with every pulse
  - Flip-flop in *any other position* is complemented when
    - all the bits in the lower significant positions are equal to 1
  - Ex:

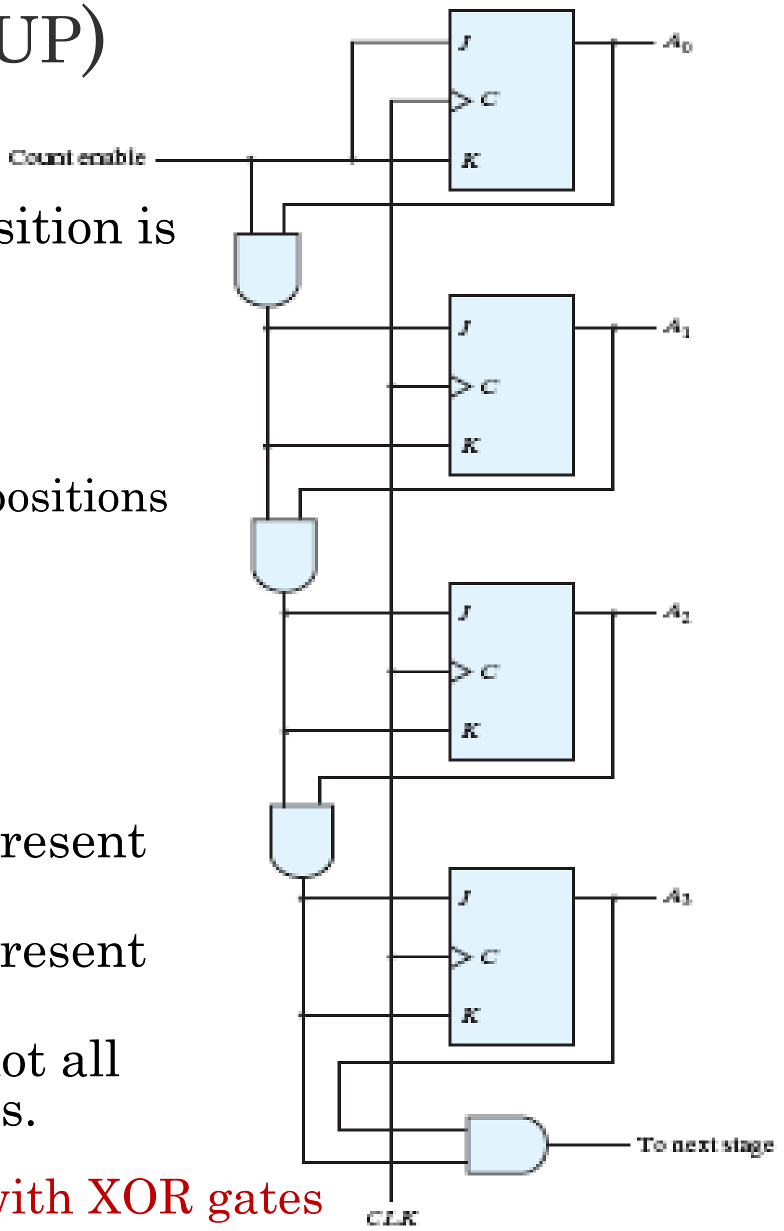
If present state (A3A2A1A0)=0011,
the next state=0100

→ A0 is always complemented

→ A1 is complemented because the present state of A0=1

→ A2 is complemented because the present state of A1A0=11

→ A3 is not complemented because not all bits in their lower significant are 1s.



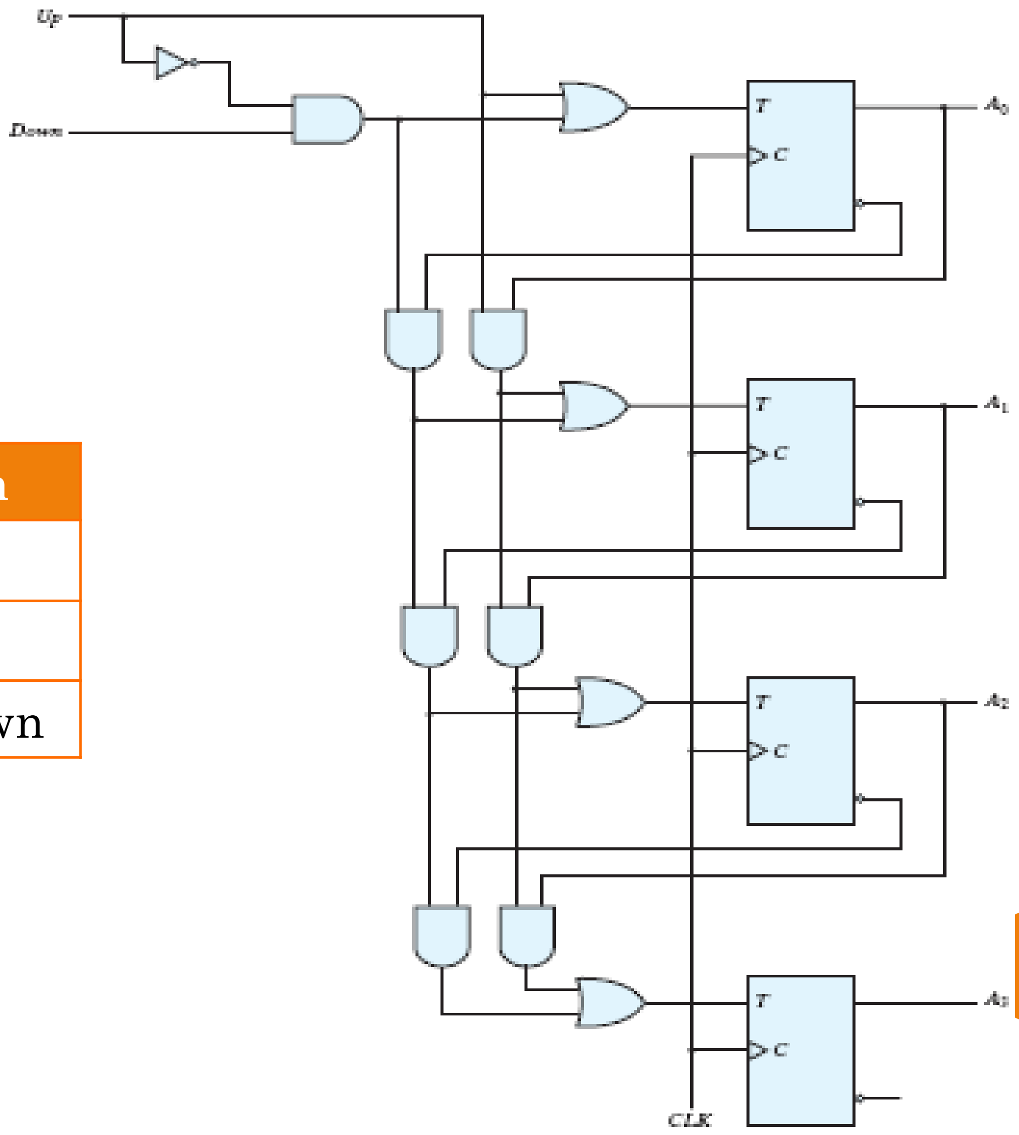
Note: we can use JK type, T type or D type with XOR gates

COUNTDOWN BINARY COUNTER

- It's the reverse of up binary counter
- It counts from 1111 to 0000 and back to 1111
- Its procedure:
  - The least significant position is complemented with each pulse
  - A bit in any other position is complemented if all lower significant bits are equal to 0

➔ Its design is as count up except that the inputs to the AND gates must come from the complemented outputs (A's), not normal outputs (As)

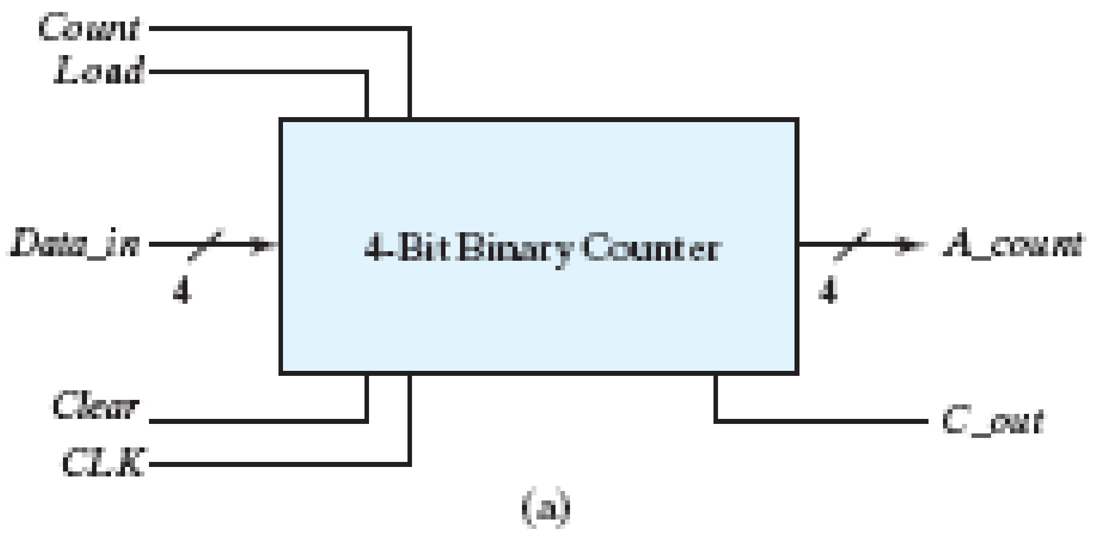
Up-Down binary counter



Up	Down	Operation
0	0	No change
1	x	Counts up
0	1	Counts down

BINARY COUNTER WITH PARALLEL LOAD

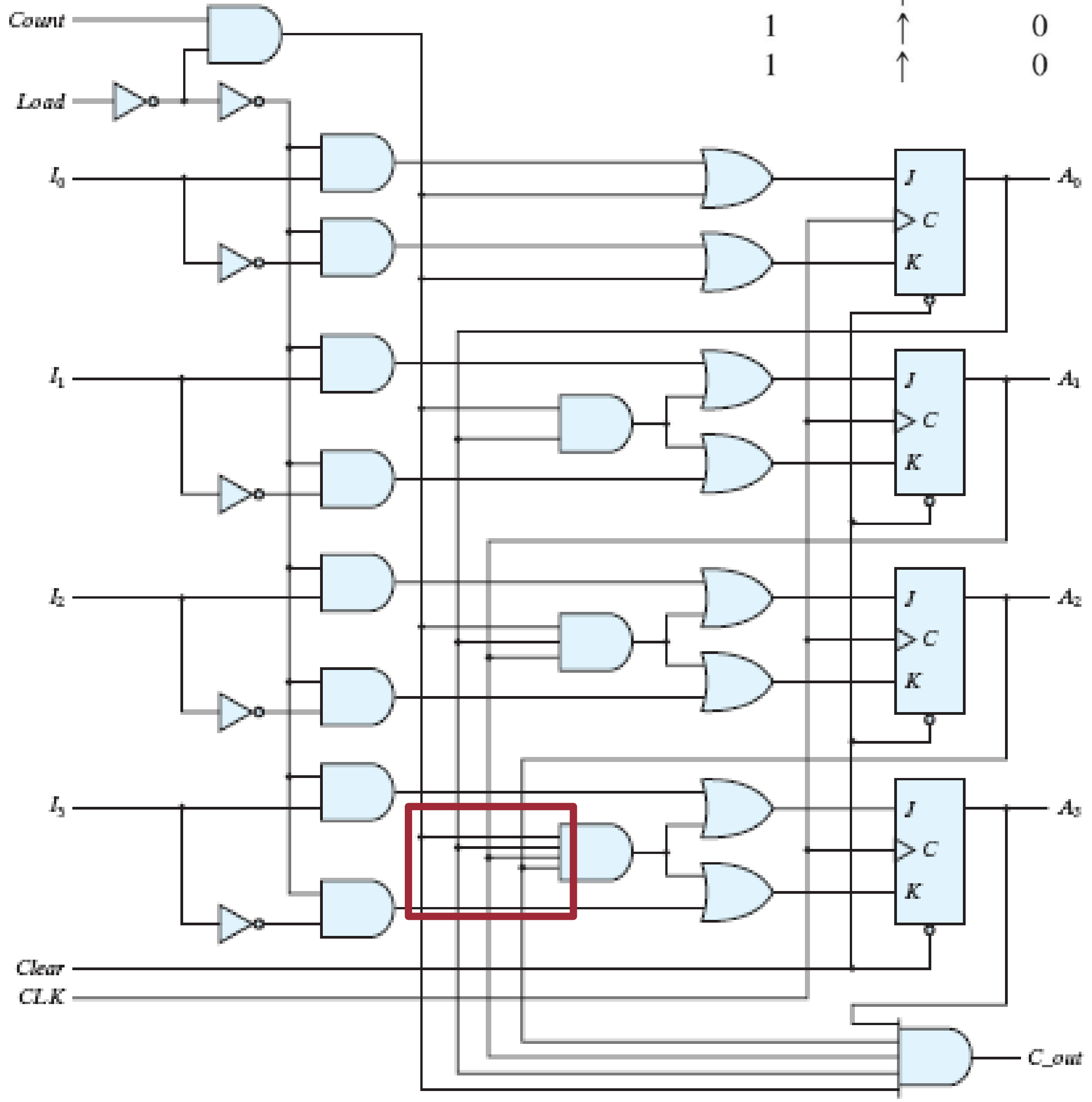
- Counters often required a parallel load capability for transferring an initial binary number into the counter prior to count.



Binary counter with parallel load Cont.)

Function Table for the Counter of Fig. 6.14

Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change



Reduce delay for generating the carry

THANKS