Digital Logic and Circuit
Paper Code: CS-102

Outline

- **Sequential Circuit**
  - **Design Procedure**

- **Register**
  - **Shift Register**
  - **Serial Transfer**

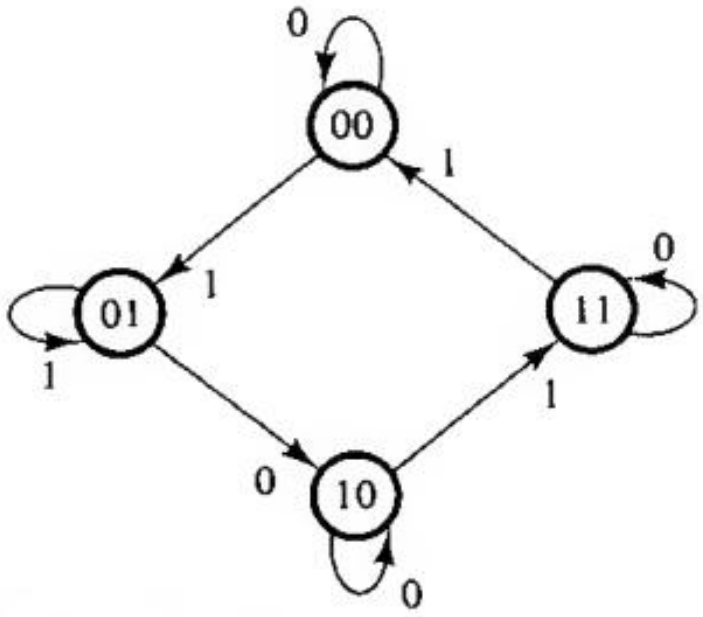
DESIGN PROCEDURE

- A synchronous sequential circuit is made up of flip-flops and combinational gates.
- The design of the circuit consists of choosing the flip-flops and then finding a combinational gate structure that, together with the flip-flops, produces a circuit that fulfills the stated specifications.
- The number of flip-flops is determined from the number of states needed in the circuit.
- In fact, once the type and number of flip-flops are determined, the design process involves a transformation from the sequential- circuit problem into a combinational-circuit problem.
- In this way, the techniques of combinational-circuit design can be applied.

The procedure is first summarized by a list of consecutive recommended steps:

1. The word description of the circuit behavior is stated. This may be accompanied by a state diagram, a timing diagram, or other pertinent information.
2. From the given information about the circuit, obtain the state table.
3. The number of states may be reduced by state-reduction methods if the sequential circuit can be characterized by input-output relationships independent of the number of states.
4. Assign binary values to each state if the state table obtained in step 2 or 3 contains letter symbols.
5. Determine the number of flip-flops needed and assign a letter symbol to each.
6. Choose the type of flip-flop to be used.
7. From the state table, derive the circuit excitation and output tables.
8. Using the map or any other simplification method, derive the circuit output functions and the flip-flop input functions.
9. Draw the logic diagram.

Consider the following state diagram

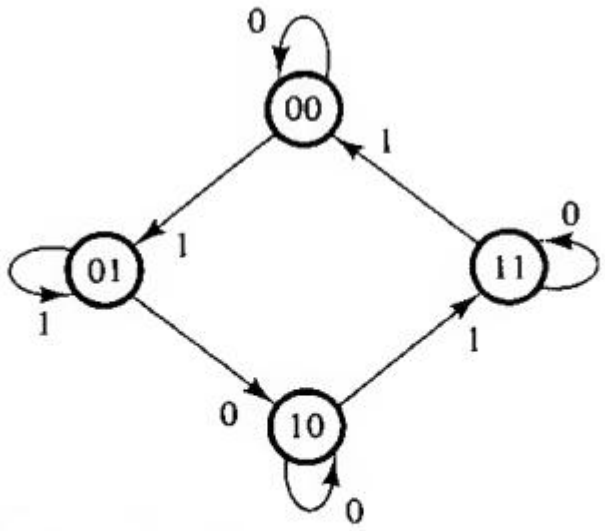


State diagram for design example

- The reduction of the number of states and the assignment of binary values to the states were discussed already.
- The examples that follow assume that the number of states and the binary assignment for the states are known. As a consequence, steps 3 and 4 of the design will not be considered in subsequent discussions.
- It has already been mentioned that m flip-flops can represent up to 2^m distinct states. A circuit may have unused binary states if the total number of states is less than 2^m .
- The unused states are taken as don't-care conditions during the design of the combinational circuit part of the circuit.

- The type of flip-flop to be used may be included in the design specifications or may depend on what is available to the designer.
- Many digital systems are constructed entirely with JK flip-flops because they are the most versatile available.
- When many types of flip-flops are available, it is advisable to use the D flip-flop for applications requiring transfer of data (such as shift registers), the T type for applications involving complementation (such as binary counters), and the JK type for general applications.

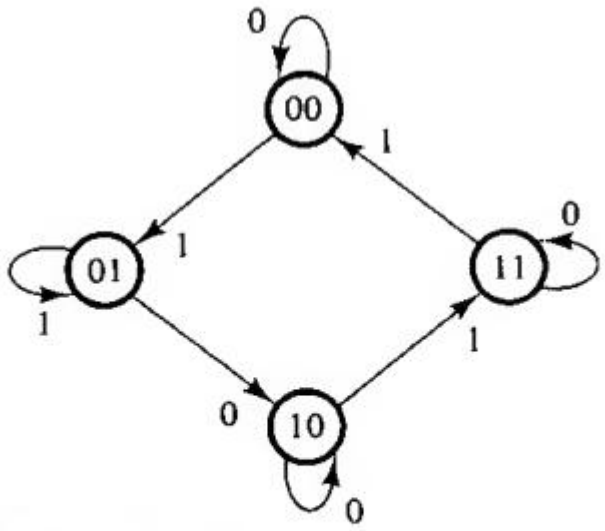
Consider the state diagram



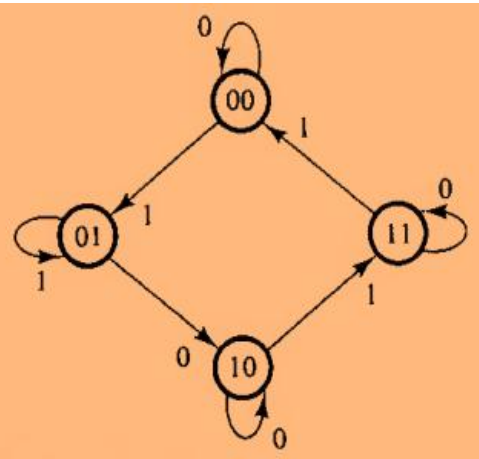
State diagram for design example

- We wish to design the clocked sequential circuit whose state diagram is given in the above Figure. The type of flip-flop to be used is JK.
- The state diagram consists of four states with binary values already assigned. Since the directed lines are marked with a single binary digit without a slash, we conclude that there is one input variable and no output variables.
- (The state of the flip-flops may be considered the outputs of the circuit). The two flip-flops needed to represent the four states are designated A and B. The input variable is designated x.

Consider the state diagram



State diagram for design example



State diagram for design example

Excitation Table

Inputs of Combinational Circuit			Next State		Outputs of Combinational Circuit			
Present State		Input			Flip-Flop Inputs			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Q(t)	Q(t + 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation table JK Flipflop

The procedure is first summarized by a list of consecutive recommended steps:

1. The word description of the circuit behavior is stated. This may be accompanied by a state diagram, a timing diagram, or other pertinent information.

2. From the given information about the circuit, obtain the state table.

3. The number of states may be reduced by state-reduction methods if the sequential circuit can be characterized by input-output relationships independent of the number of states.

4. Assign binary values to each state if the state table obtained in step 2 or 3 contains letter symbols.

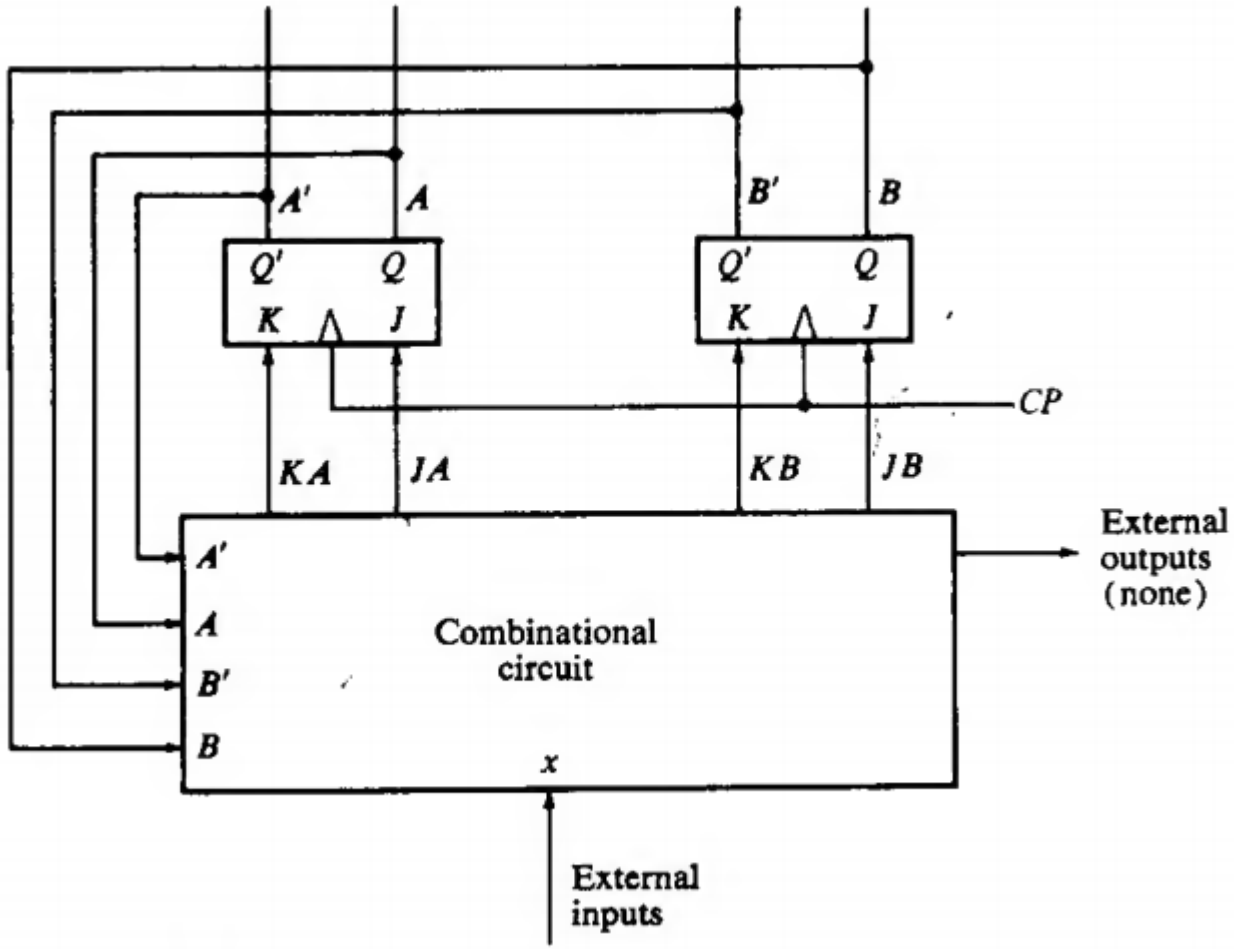
5. Determine the number of flip-flops needed and assign a letter symbol to each.

6. Choose the type of flip-flop to be used.

7. From the state table, derive the circuit excitation and output tables.

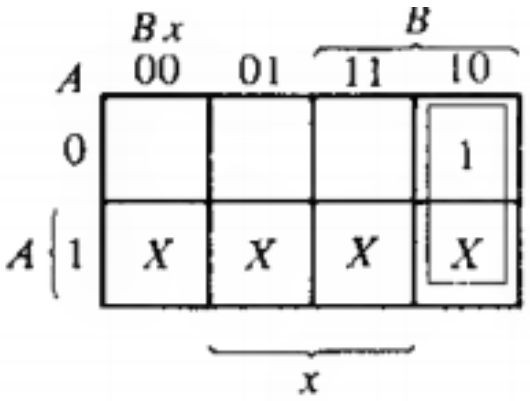
8. Using the map or any other simplification method, derive the circuit output functions and the flip-flop input functions.

9. Draw the logic diagram.

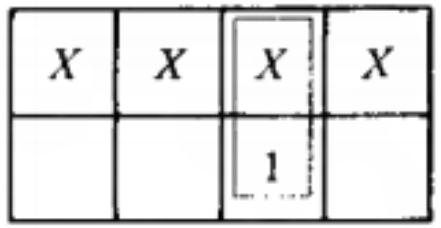


Excitation Table

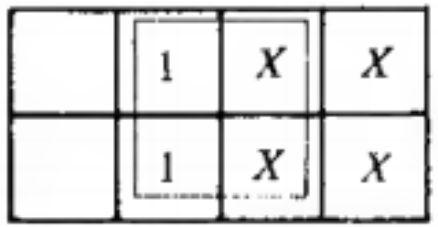
Inputs of Combinational Circuit			Next State		Outputs of Combinational Circuit			
Present State		Input			Flip-Flop Inputs			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



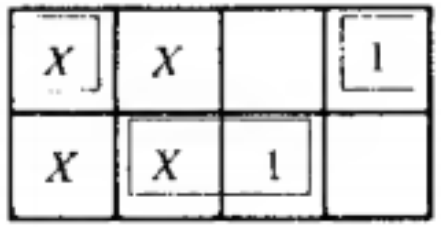
$JA = Bx'$



$KA = Bx$



$JB = x$



$KB = (A \oplus x)'$

Excitation Table

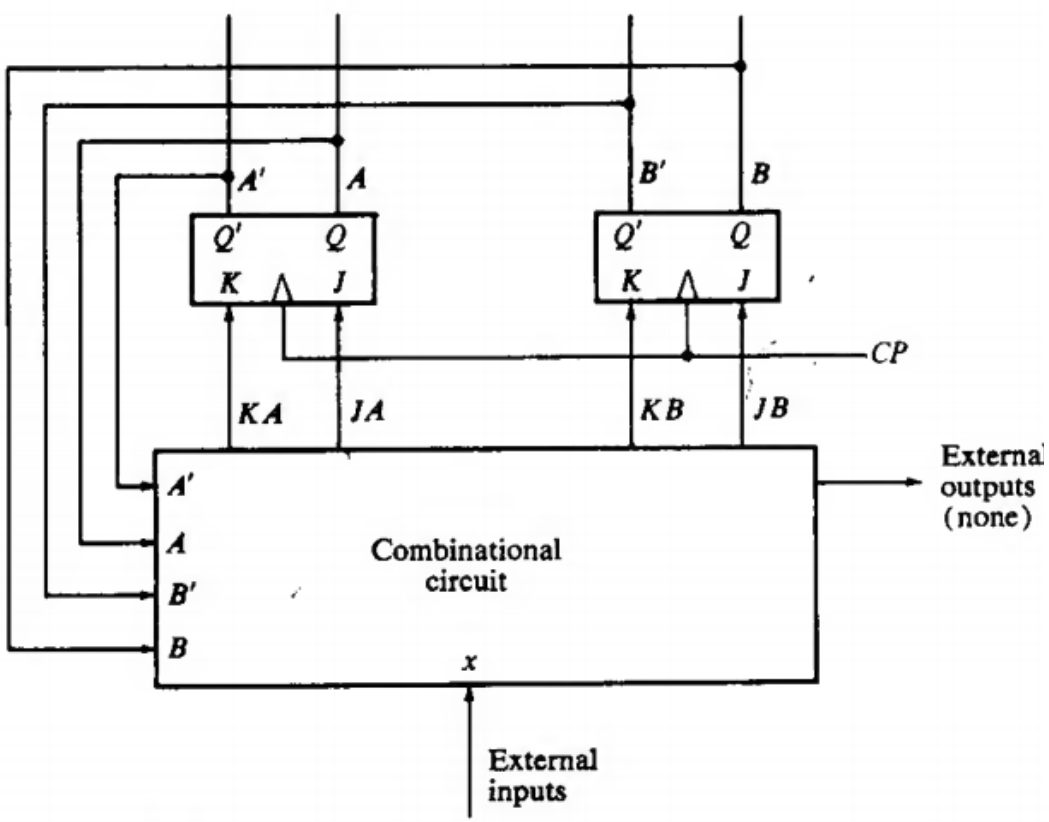
Inputs of Combinational Circuit			Next State		Outputs of Combinational Circuit			
Present State		Input			Flip-Flop Inputs			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

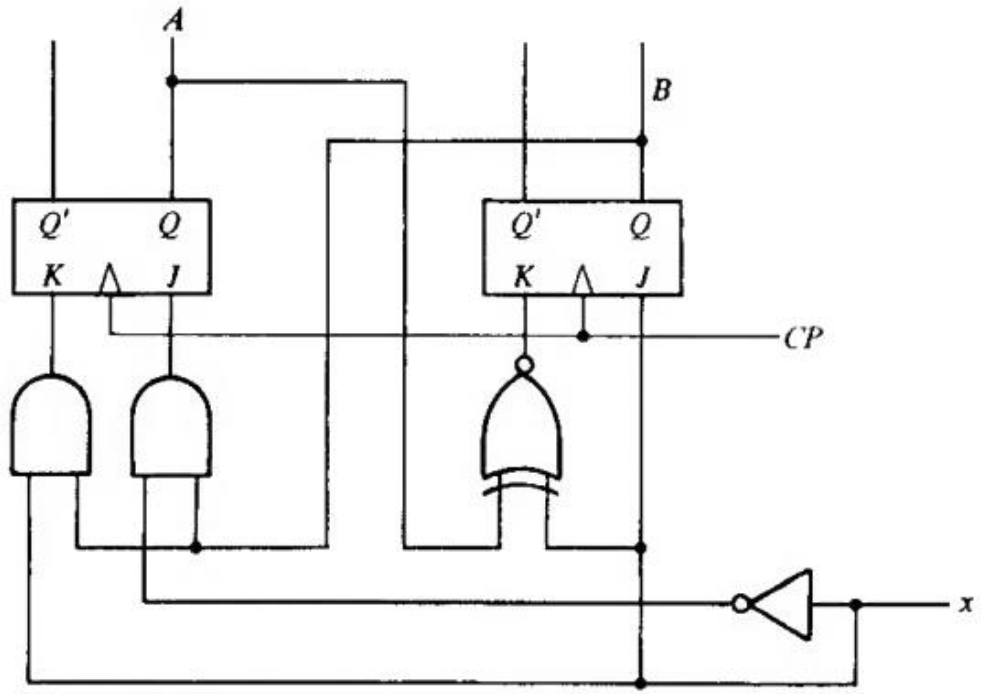
$$JA = Bx' \qquad KA = Bx$$

$$JB = x \qquad KB = (A \oplus x)'$$

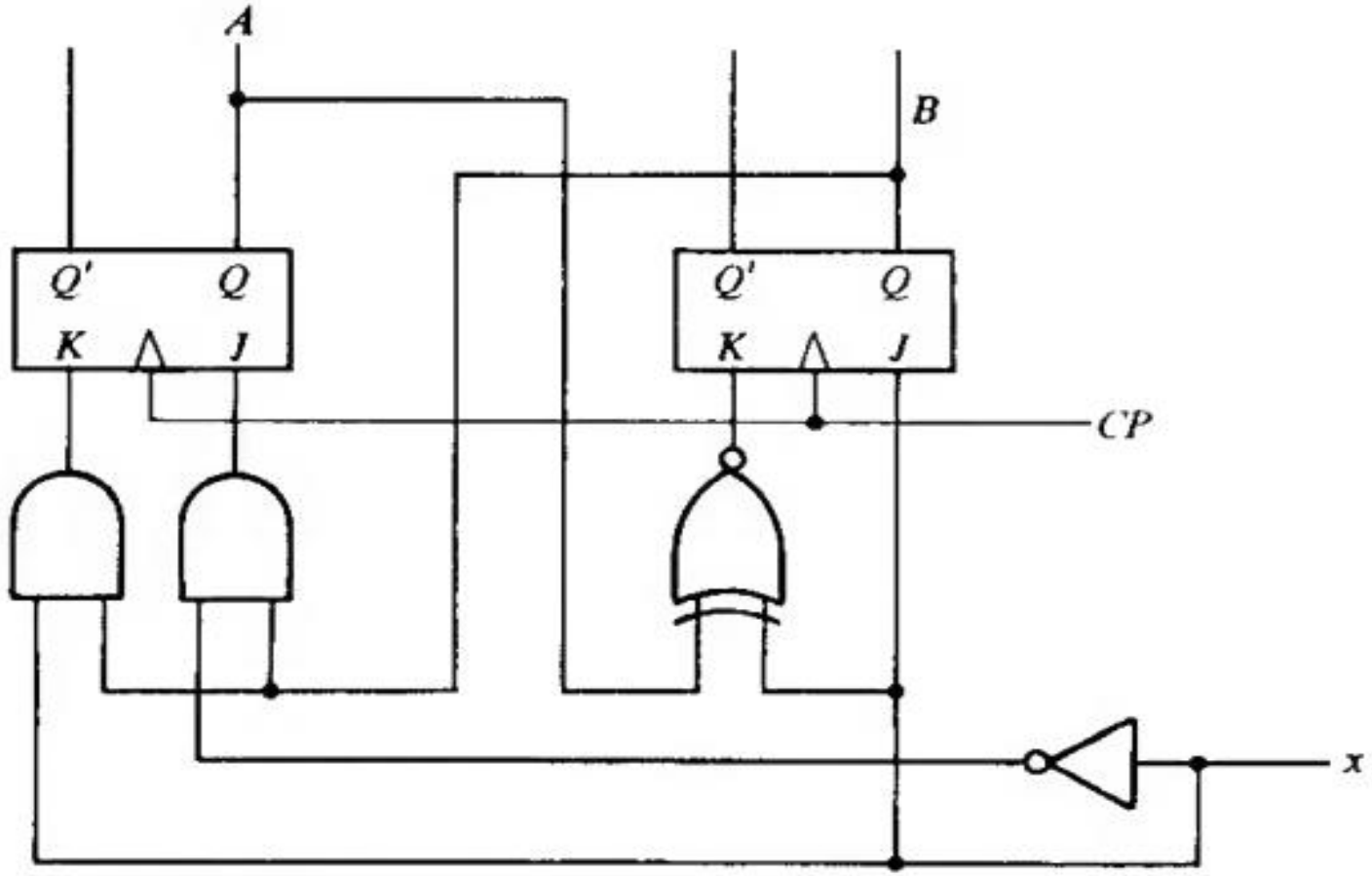
$$JA = Bx' \qquad KA = Bx$$

$$JB = x \qquad KB = (A \oplus x)'$$





Logic diagram of sequential circuit



Logic diagram of sequential circuit

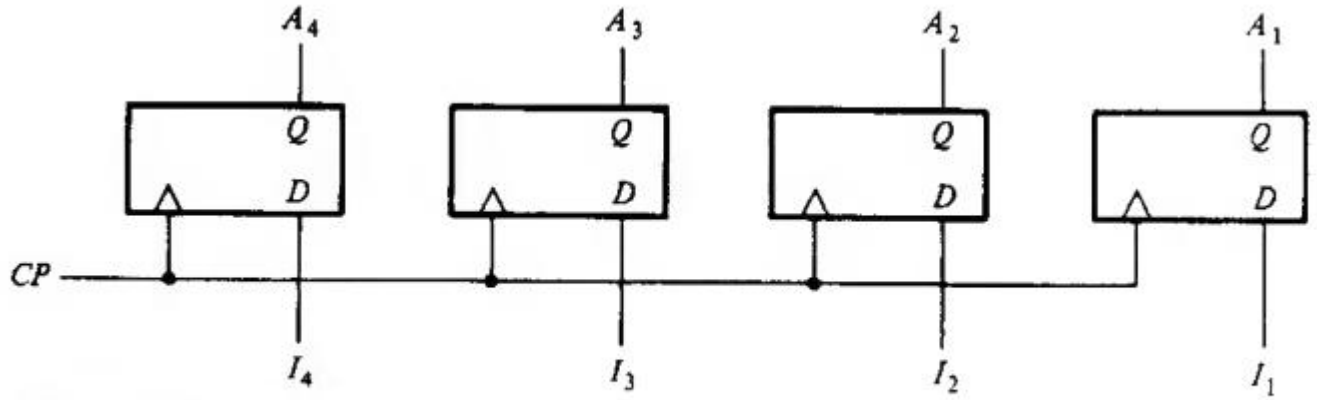
Register

Register

- A register is a group of binary cells suitable for holding binary information.
- A group of flip-flops constitutes a register, since each flip-flop is a binary cell 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 containing n bits.
- In addition to the flip-flops, a register may have combinational gates that perform certain data-processing tasks.
- A register consists of a group of flip-flops and gates that affect their transition.
- The flip-flops hold binary information and the gates control when and how new information is transferred into the register.

- Various types of registers are available in MSI circuits.
- The simplest possible register is one that consists of only flip-flops without any external gates. Figure shows such a register constructed with four D-type flip-flops and a common clock-pulse input.

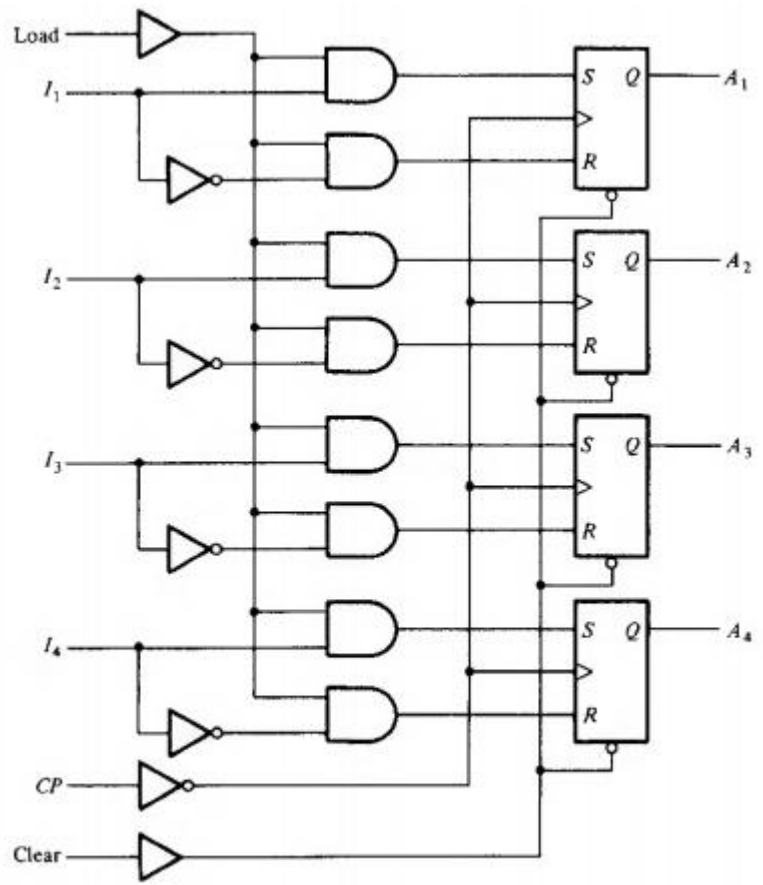
- The clock pulse input, CP, enables all flip-flops, so that the information presently available at the four inputs can be transferred into the 4-bit register.
- The four outputs can be sampled to obtain the information presently stored in the register.



4-bit register

Register with Parallel Load

- The transfer of new information into a register is referred to as loading the register.
- If all the bits of the register are loaded simultaneously with a single clock pulse, we say that the loading is done in parallel.
- A pulse applied to the CP input of the register of Figure will load all four inputs in parallel.
- In this configuration, the clock pulse must be inhibited from the CP terminal if the content of the register must be left unchanged. In other words, the CP input acts as an enable signal that controls the loading of new information into the register.
- When CP goes to 1 , the input information is loaded into the register. If CP remains at 0, the content of the register is not changed.
- Note that the change of state in the outputs occurs at the positive edge of the pulse.
- If a flip-flop changes state at the negative edge, there will be a small circle under the triangle symbol in the CP input of the flip-flop.



4-bit register with parallel load

A 4-bit register with a load control input using RS flip-flops is shown in Figure.
The CP input of the register receives continuous synchronized pulses, which are applied to all flip-flops.

The inverter in the CP path causes all flip-flops to be triggered by the negative edge of the incoming pulses.

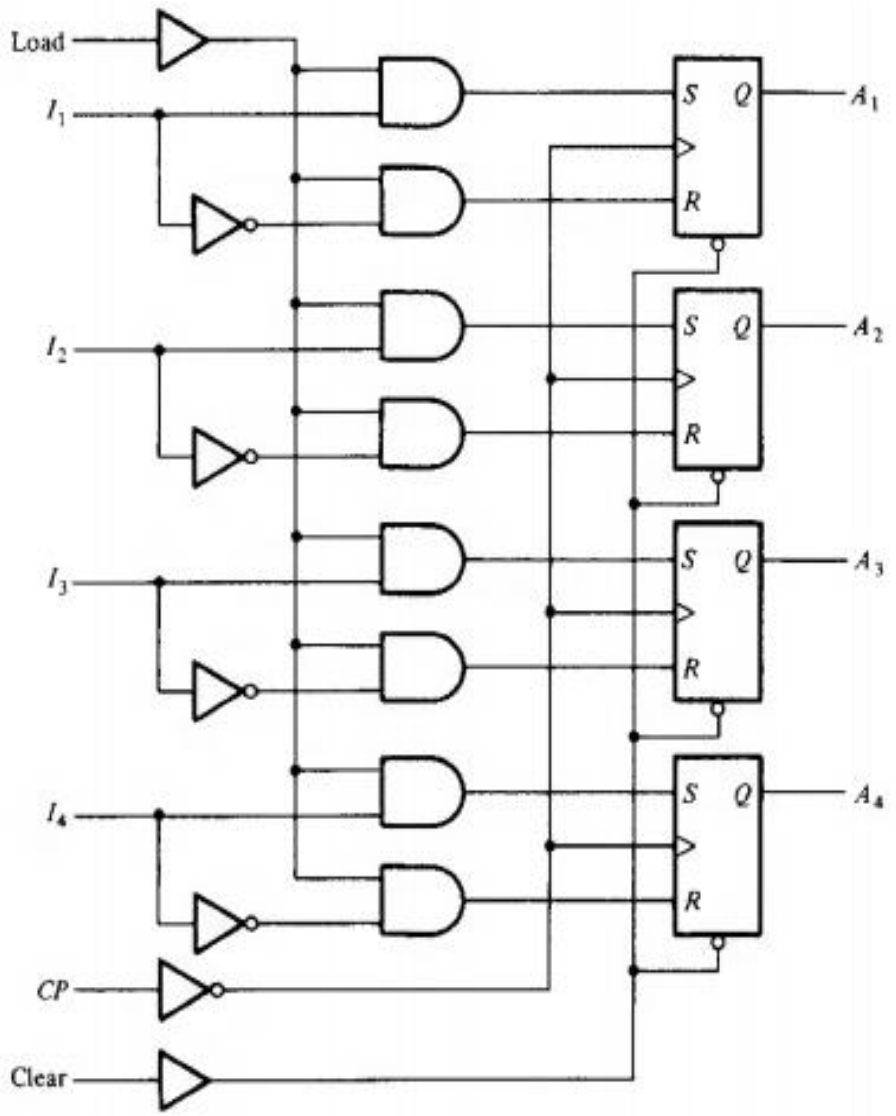
The clear input goes to a special terminal in each flip-flop through a noninverting buffer gate.

When this terminal goes to 0, the flip-flop is cleared asynchronously.

The clear input is useful for clearing the register to all 0's prior to its docked operation.

The clear input must be maintained at 1 during normal clocked operations .

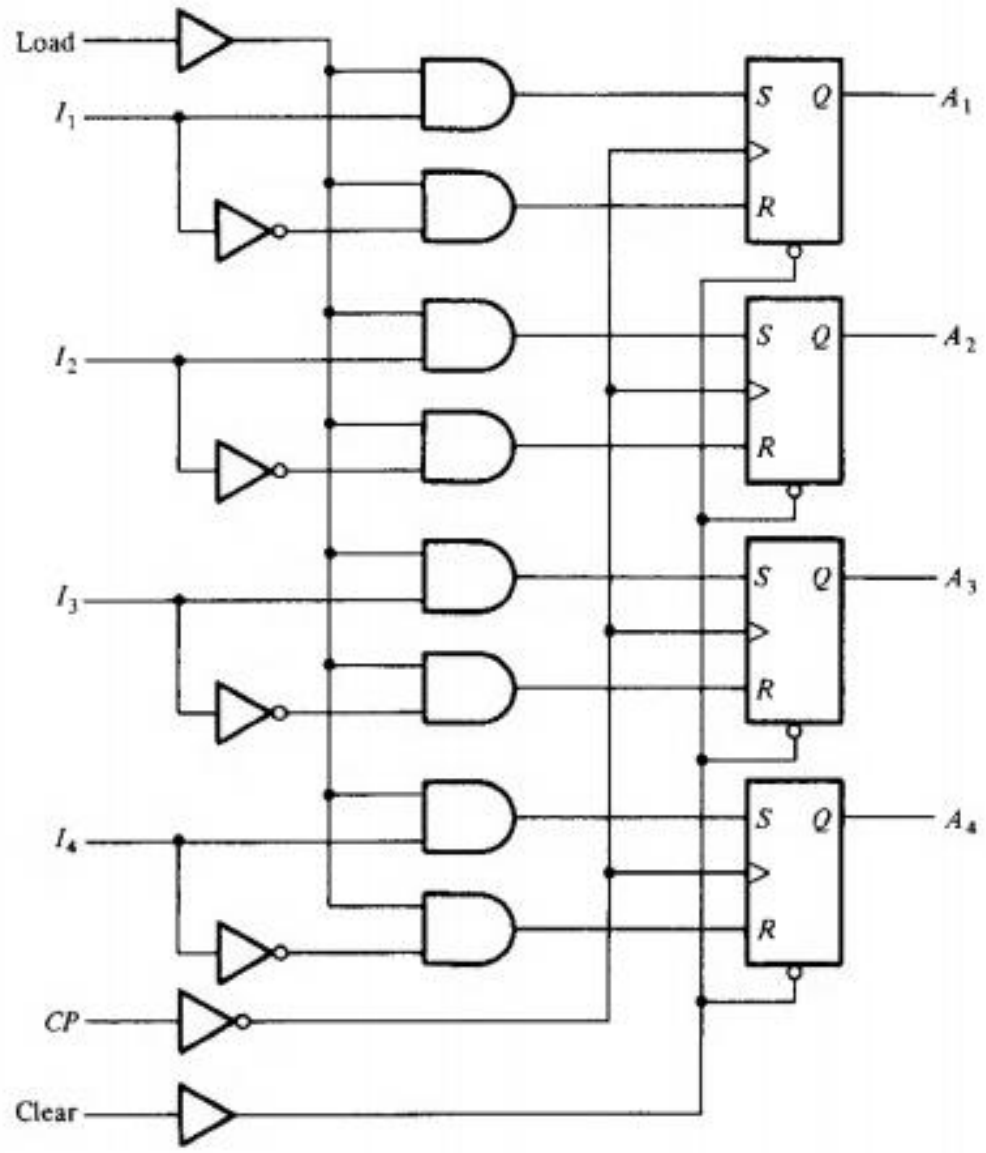
The load input goes through a buffer gate (to reduce loading) and through a series of AND gates to the R and S inputs of each flip-flop.



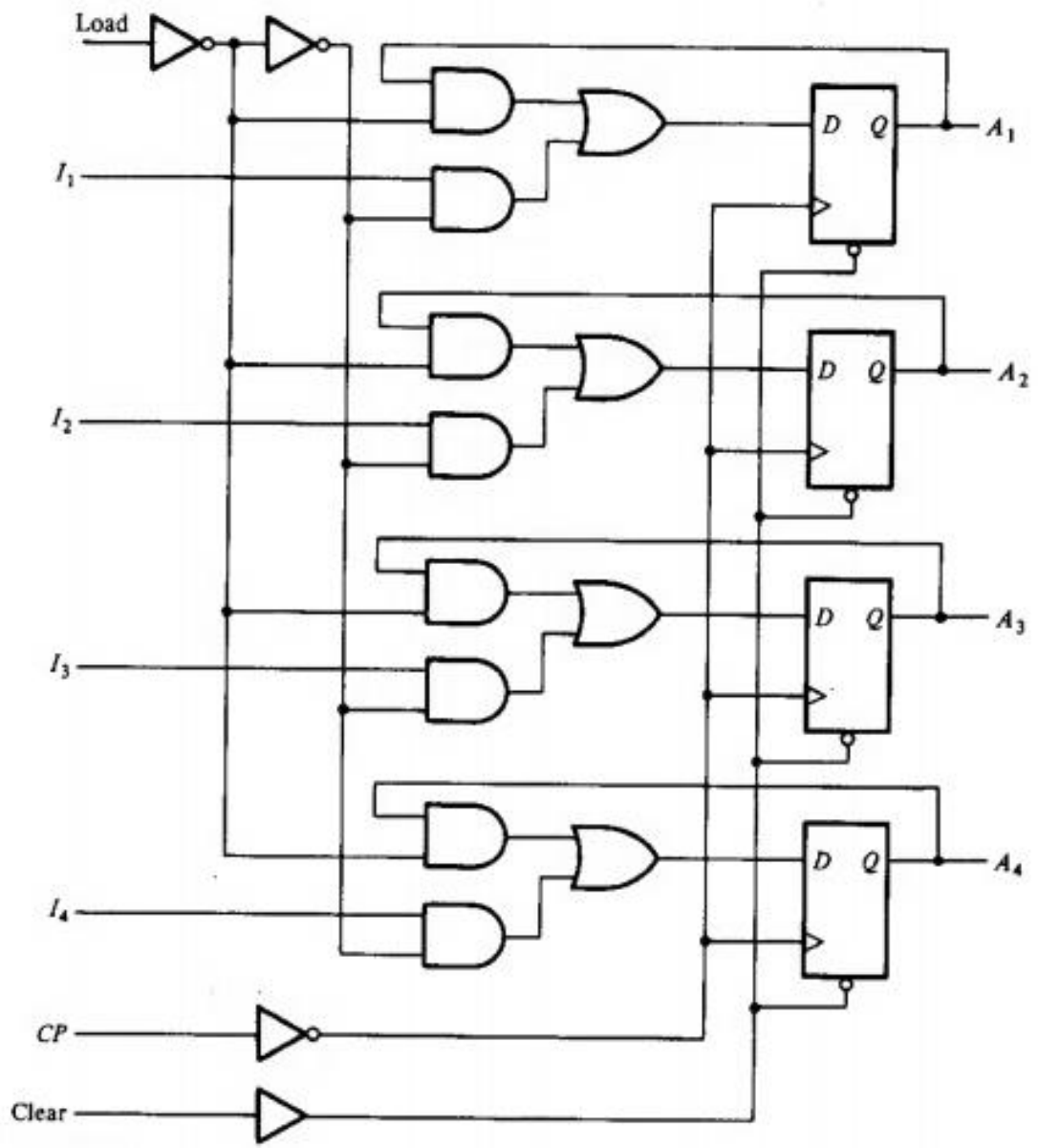
4-bit register with parallel load

- Although clock pulses are continuously present, it is the load input that controls the operation of the register.
- The two AND gates and the inverter associated with each input I determine the values of R and S.

- If the load input is 0, both R and S are 0, and no change of state occurs with any clock pulse.
- Thus, the load input is a control variable that can prevent any information change in the register as long as its input is 0.
- When the load control goes to 1, inputs I1 through I4 specify what binary information is loaded into the register on the next clock pulse.
- For each I that is equal to 1, the corresponding flip-flop inputs are S = 1, R = 0. For each I that is equal to 0, the corresponding flip-flop inputs are S = 0, R = 1.
- Thus, the input value is transferred into the register provided the load input is 1, the clear input is 1 , and a clock pulse goes from 1 to 0.
- This type of transfer is called a parallel- load transfer because all bits of the register are loaded simultaneously.



4-bit register with parallel load


Load
I_1
I_2
I_3
I_4
CP
Clear
D Q
A_1
A_2
A_3
A_4

Design the sequential circuit for the circuit whose state table is

- The state table specifies two flip-flops, A1 and A2
- one input, x, and one output y
- The next-state and output information is obtained directly from the table:

$A1(t + 1) = \sum(4, 6)$
$A2(t + 1) = \sum(1,2, 5, 6)$
$y(A1, A2 , x) = \sum (3, 7)$

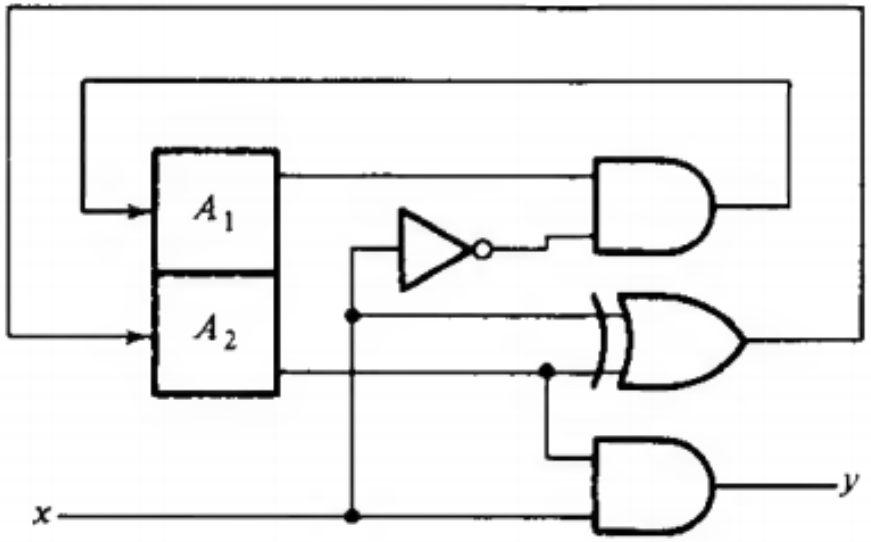
Present state		Input	Next state		Output
A_1	A_2	x	A_1	A_2	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

- The minterm values are for variables A1, A 2 and x , which are the present-state and input variables.
- The functions for the next state and output can be simplified by means of maps to give

$$A_1(t+1) = A_1x'$$
$$A_2(t+1) = A_2 \oplus x$$
$$y = A_2x$$

Present state		Input	Next state		Output
A_1	A_2	x	A_1	A_2	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

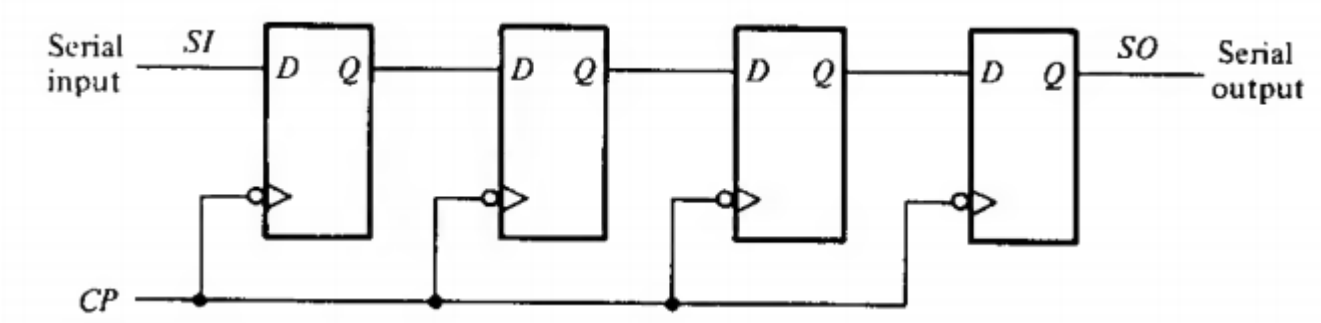
(a) State table



(b) Logic diagram

Shift Register

- A register capable of shifting its binary information either to the right or to the left is called a shift register.
- The logical configuration of a shift register consists of a chain of flip-flops connected in cascade, with the output of one flip-flop connected to the input of the next flip-flop.
- All flip-flops receive a common clock pulse that causes the shift from one stage to the next.

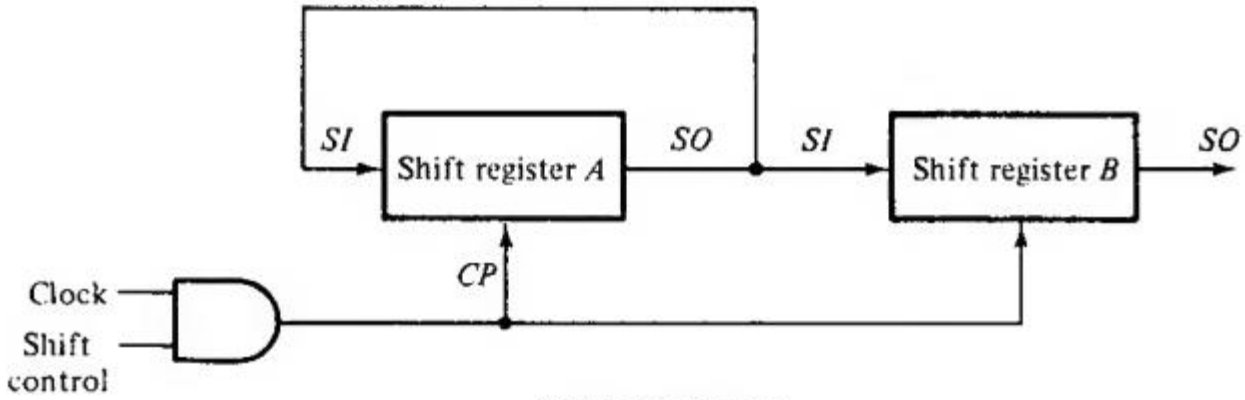


Serial Transfer

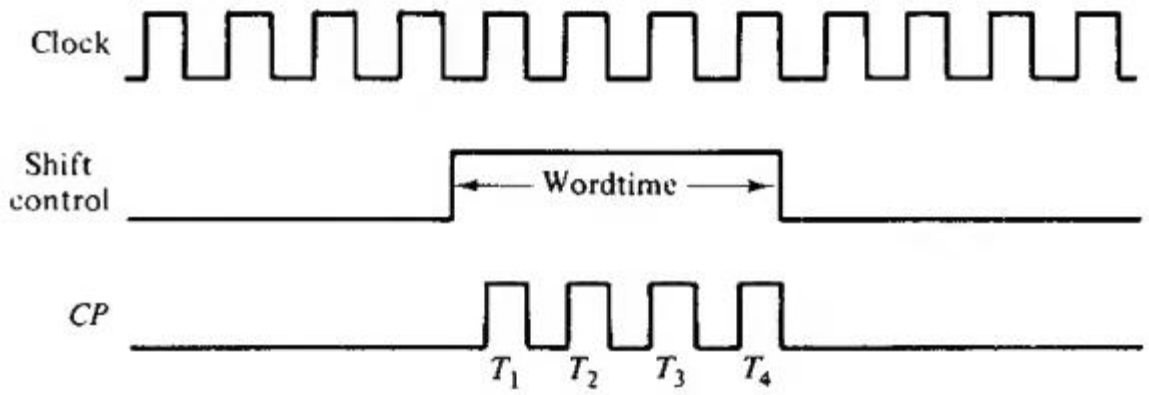
- A digital system is said to operate in a serial mode when information is transferred and manipulated one bit at a time.
- The content of one register is transferred to another by shifting the bits from one register to the other.
- The information is transferred one bit at a time by shifting the bits out of the source register into the destination register.

- The serial transfer of information from register A to register B is done with shift registers, as shown in the block diagram.
- The serial output (SO) of register A goes to the serial input (SI) of register B.
- To prevent the loss of information stored in the source register, the A register is made to circulate its information by connecting the serial output to its serial input terminal.
- The initial content of register B is shifted out through its serial output and is' lost unless it is transferred to a third shift register.
- The shift-control input determines when and by how many times the registers are shifted. This is done by the AND gate that allows clock pulses to pass into the CP terminals only when the shift control is 1

Suppose the shift registers have four bits each.



(a) Block diagram

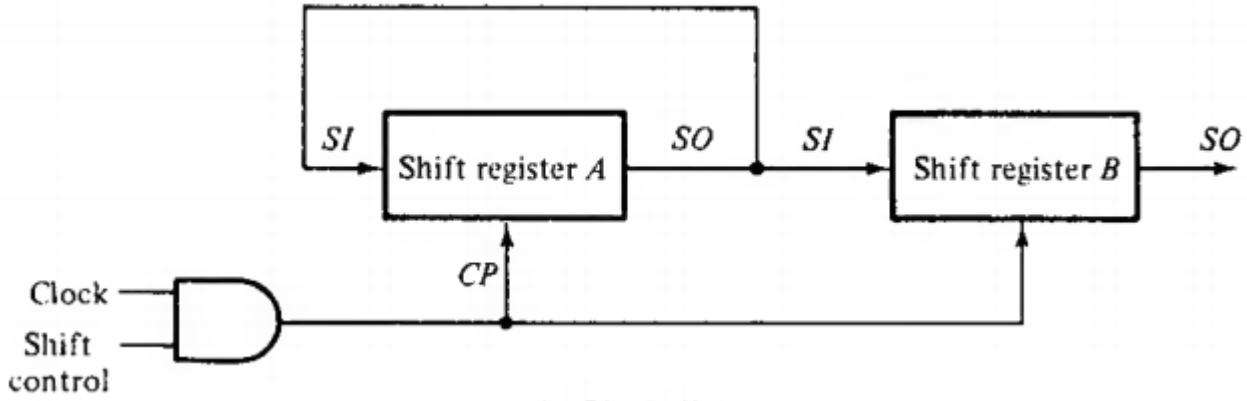


Serial Transfer from Register A to Register B

- Assume that the binary content of A before the shift is 1011 and that of B, 0010.
- After the first pulse, T1 the rightmost bit of A is shifted into the leftmost bit of B and, at the same time, this bit is circulated into the leftmost position of A.
- The other bits of A and B are shifted once to the right. The previous serial output from B is lost and its value changes from 0 to 1.

Serial-Transfer Example

Timing Pulse	Shift Register *A*	Shift Register *B*	Serial Output of *B*
Initial value	1 0 1 1	0 0 1 0	0
After T_1	1 1 0 1	1 0 0 1	1
After T_2	1 1 1 0	1 1 0 0	0
After T_3	0 1 1 1	0 1 1 0	0
After T_4	1 0 1 1	1 0 1 1	1



Serial-Transfer Example

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

- The next three pulses perform identical operations, shifting the bits of A into B, one at a time.
- After the fourth shift, the shift control goes to 0 and both registers A and B have the value 1011. Thus, the content of A is transferred into B, while the content of A remains unchanged.

- Computers may operate in a serial mode, a parallel mode, or in a combination of both.
- Serial operations are slower because of the time it takes to transfer information in and out of shift registers.

- Serial computers, however, require less hardware to perform operations because one common circuit can be used over and over again to manipulate the bits coming out of shift registers in a sequential manner.
- The time interval between clock pulses is called the bit time , and the time required to shift the entire contents of a shift register is called the word time .
- These timing sequences are generated by the control section of the system. In a parallel computer, control signals are enabled during one clock-pulse interval. Transfers into registers are in parallel, and they occur upon application of a single clock pulse, in a serial computer, control signals must be maintained for a period equal to one word time.
- The pulse applied every bit time transfers the result of the operation, one at a time, into a shift register.
- Most computers operate in a parallel mode because this is a faster mode of operation.

Suggested Reading

- M. Morris Mano, Digital Logic and Computer Design, PHI.

Thank you