SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

16EC303–VLSI DESIGN

III YEAR/ V SEMESTER

UNIT 3 –SEQUENTIAL LOGIC CIRCUITS

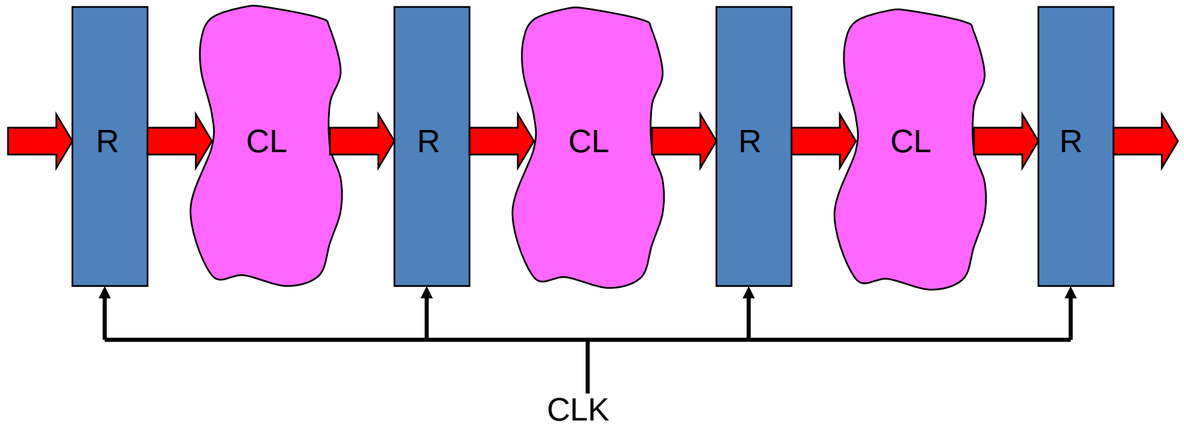
TOPIC 8,9 –SYNCHRONOUS & ASYNCHRONOUS DESIGN

OUTLINE

- INTRODUCTION-SYNCHRONOUS & ASYNCHRONOUS SEQUENTIAL CIRCUIT
- ASYNCHRONOUS SEQUENTIAL CIRCUIT-BASICS
- SYNCHRONOUS CLOCKED SEQUENTIAL CIRCUIT
- SR LATCH
- D LATCH
- CLOCK RESPONSES-LATCH & FF
- JK FLIP-FLOP
- T FLIP-FLOP
- CHARACTERISTICS EQUATIONS OF ALL FF
- ACTIVITY
- D FLIP-FLOP WITH ASYNCHRONOUS RESET
- MEALY AND MOORE MODELS
- SHIFT REGISTERS
- COUNTERS
- RIPPLE COUNTERS
- SYNCHRONOUS COUNTERS
- ASSESSMENT
- SUMMARY-OTHER COUNTERS

SYNCHRONOUS SEQUENTIAL CIRCUIT

- The change of internal state occurs in response to the synchronized clock pulses.
- Input changes occur between clock pulses
- Data are read during the clock pulse
- It is supposed to wait long enough after the external input changes for all flip-flop inputs to reach a steady value before applying the new clock pulse
- Unsuitable Situations:
  - Inputs change at any time and cannot be synchronized with a clock
  - Circuit is large, Clock skew can not be avoided
  - High performance design

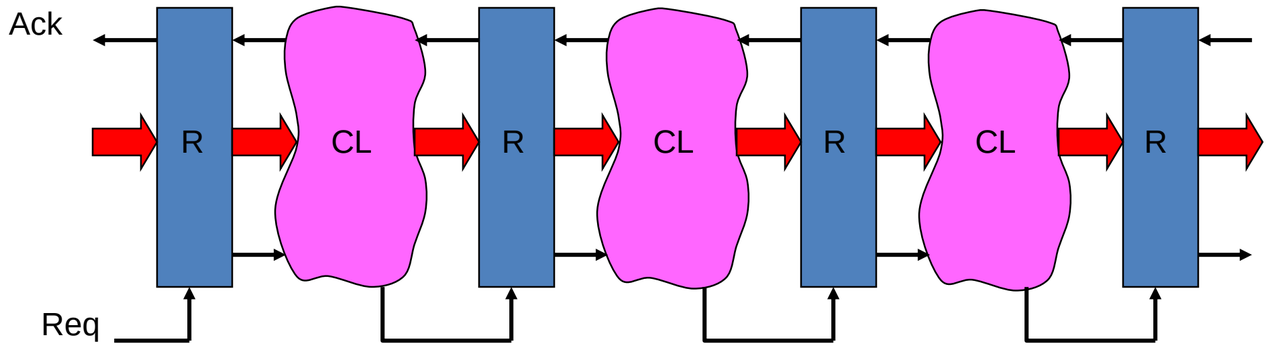


Implicit (global) synchronization between blocks
Clock Period > Max Delay (CL)

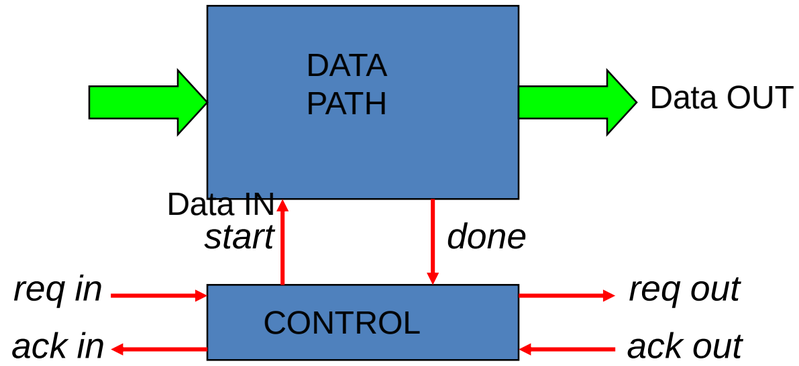
ASYNCHRONOUS CIRCUITS

- Not synchronized by a common clock, States change immediately after input changes
- The circuit reaches a steady-state condition when $y_i = Y_i$ for all i.
- For a given value of input variables, the system is stable if the circuit reaches a steady state condition.
- A transition from one stable state to another occurs only in response to a change in an input variable
- Fundamental-mode operation
  - The input signals change only when the circuit is in a **stable condition**
  - The input signals change **one at a time**
- The time between two input changes must be longer than the time it takes the circuit to reach a stable state.
- Timing is a Major Problem because of
  - **Unequal delays** through various paths in the circuit

ASYNCHRONOUS SEQUENTIAL CIRCUITS



Explicit (Local) synchronization: **Req/Ack handshakes**



ASYNCHRONOUS SEQUENTIAL CIRCUITS BASICS

- No clock signal is required
- Internal states can change at any instant of time when there is a change in the input variables
- Have better performance but hard to design due to timing problems

•Why Asynchronous Circuits?

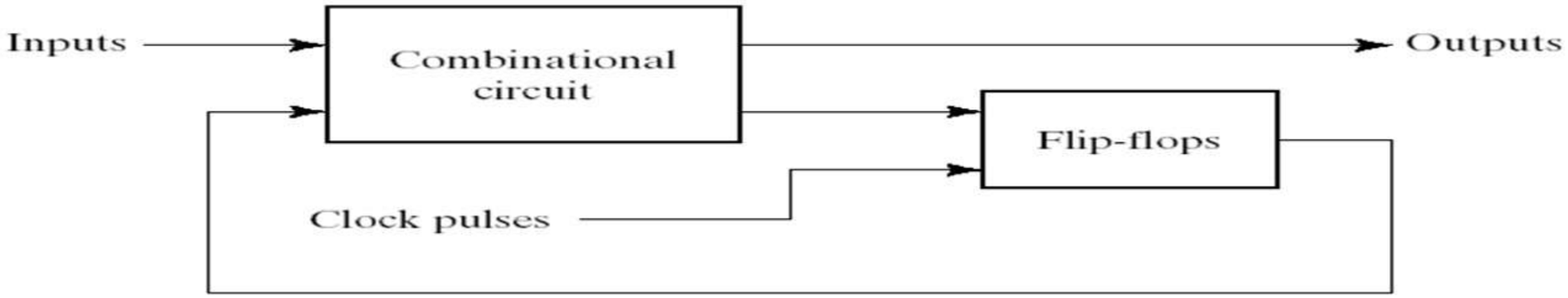
- Accelerate the speed of the machine (no need to wait for the next clock pulse).
- Simplify the circuit in the small independent gates.
- Necessary when having multi circuits each having its own clock.

•Analysis Procedure

- The analysis consists of obtaining a table or a diagram that describes the sequence of internal states and outputs as a function of changes in the input variables.

SYNCHRONOUS CLOCKED SEQUENTIAL CIRCUIT

A sequential circuit may use many flip-flops to store as many bits as necessary. The outputs can come either from the combinational circuit or from the flip-flops or both.

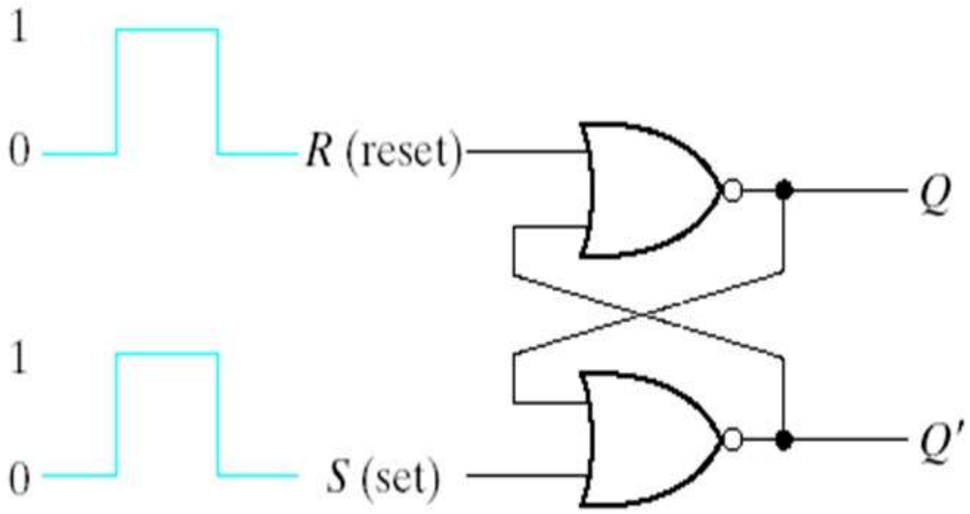


(a) Block diagram

(b) Timing diagram of clock pulses

SR LATCH

The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates. It has two inputs labeled S for set and R for reset.

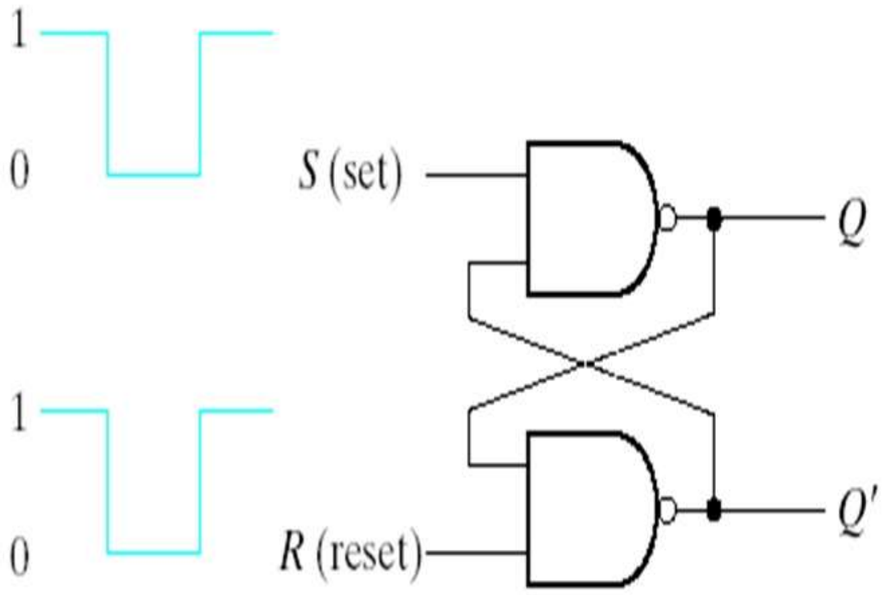


(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

(b) Function table

SR LATCH WITH NAND GATES



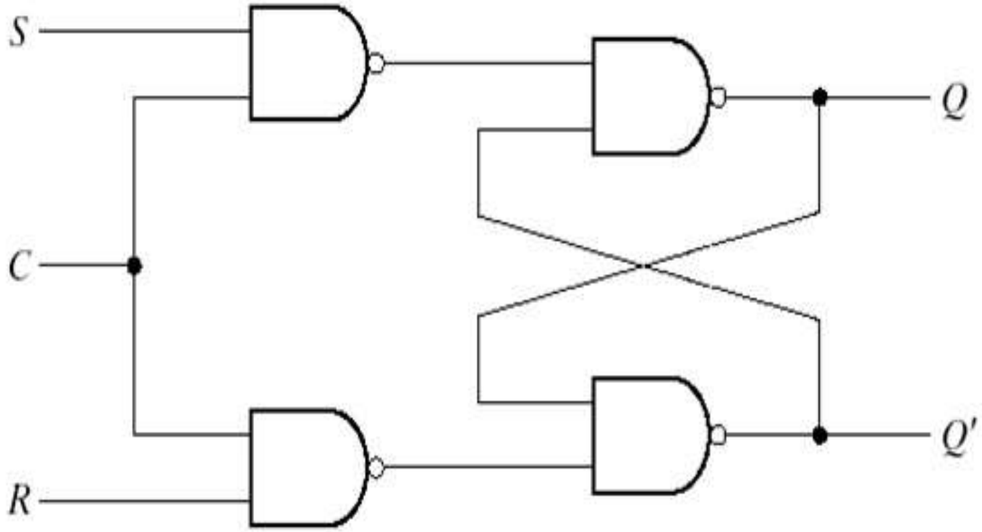
(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(b) Function table

SR LATCH WITH CONTROL INPUT

The operation of the basic SR latch can be modified by providing an additional control input that determines when the state of the latch can be changed., it consists of the basic SR latch and two additional NAND gates.



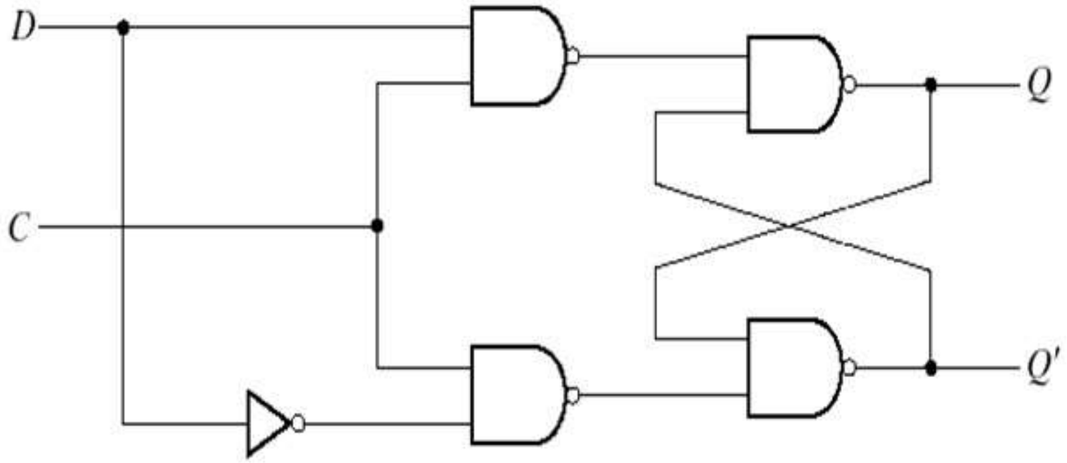
(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; set state
1	1	1	Indeterminate

(b) Function table

D LATCH

One way to eliminate the undesirable condition of the indeterminate state in SR latch is to ensure that inputs S and R are never equal to 1 at the same time in Fig 5-5. This is done in the D latch.



(a) Logic diagram

C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

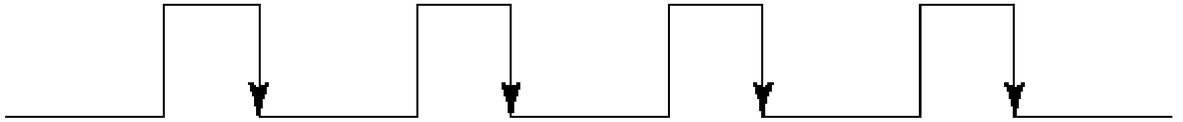
(b) Function table

CLOCK RESPONSES: A.LATCH B& C -FLIPFLOP

(a) Response to positive level

a positive level response in the control input allows changes, in the output when the D input changes while the clock pulse stays at logic 1.

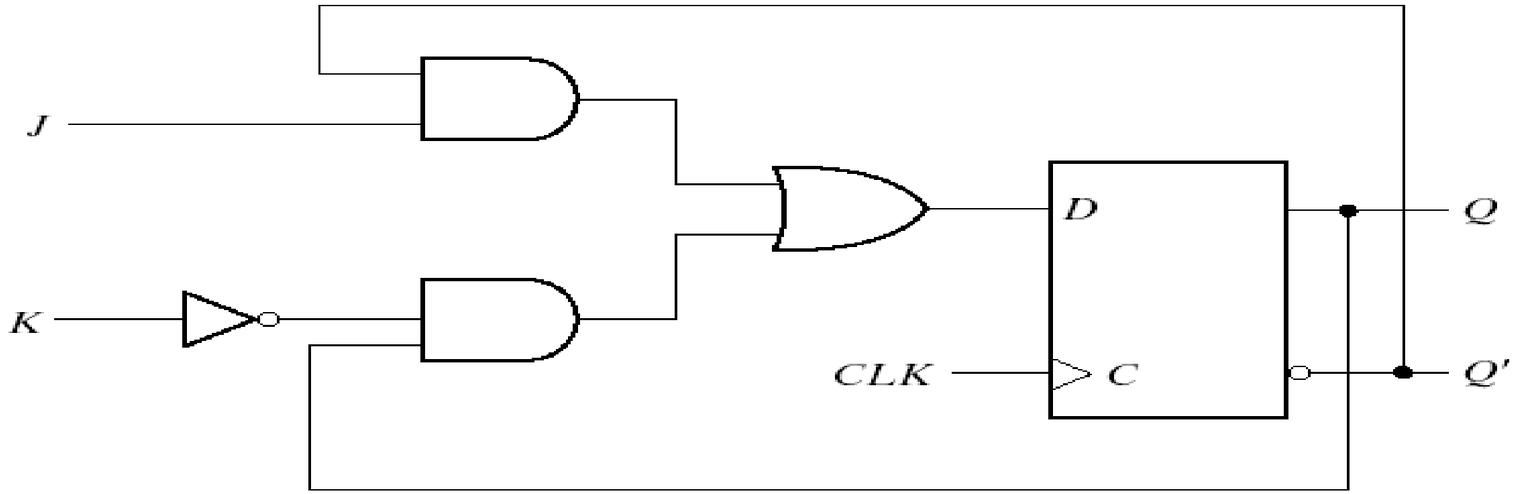
(b) Positive-edge response



(c) Negative-edge response

JK FLIP-FLOP

There are three operations that can be performed with a flip-flop: set it to 1, reset it to 0, or complement its output. The JK flip-flop performs all three operations. The circuit diagram of a JK flip-flop constructed with a D flip-flop and gates.

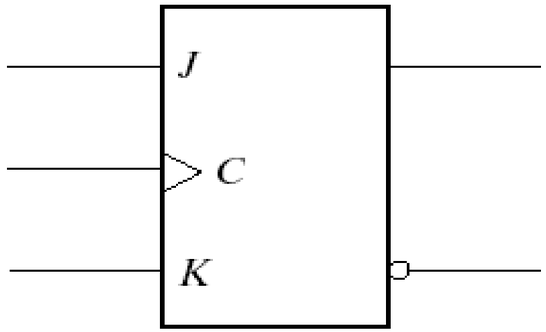


(a) Circuit diagram

JK FLIP-FLOP

The J input sets the flip-flop to 1, the K input resets it to 0, and when both inputs are enabled, the output is complemented. This can be verified by investigating the circuit applied to the D input:

D = J Q` + K` Q

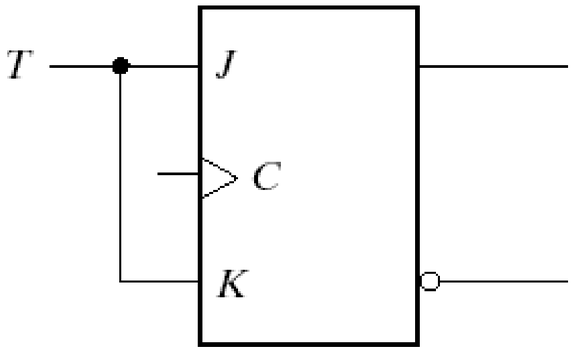


(b) Graphic symbol

J	K	Q(t+1)	State
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q”(t)	Complement

T FLIP-FLOP

The T(toggle) flip-flop is a complementing flip-flop and can be obtained from a JK flip-flop when inputs J and K are tied together.



(a) From *JK* flip-flop

D	Q(t+1)	D FF	T FF
0	0	Reset	No change
0	1	set	Complement

CHARACTERISTIC EQUATIONS & ACTIVITY

D flip-flop Characteristic Equations

$$Q(t + 1) = D$$

JK flip-flop Characteristic Equations

$$Q(t + 1) = JQ` + K`Q$$

T flip-flop Characteristic Equations

$$Q(t + 1) = T \qquad Q = TQ` + T`Q$$

ACTIVITY- CAN ANYONE SAYS Z TO A

DIRECT INPUTS

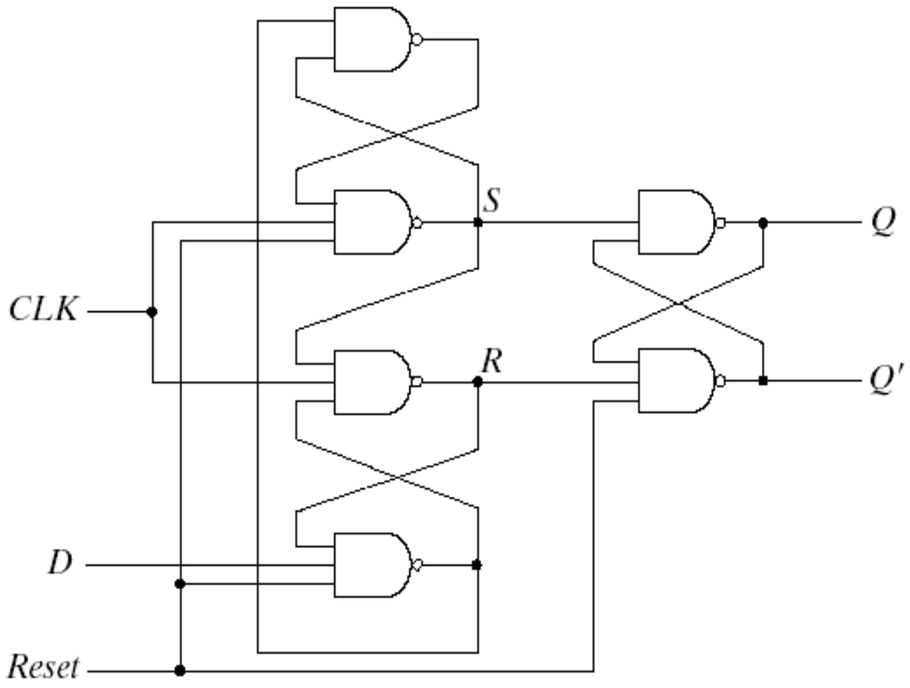
- Some flip-flops have asynchronous inputs that are used to force the flip-flop to a particular state independent of the clock.

•The input that sets the flip-flop to 1 is called present or direct set. The input that clears the flip-flop to 0 is called clear or direct reset.

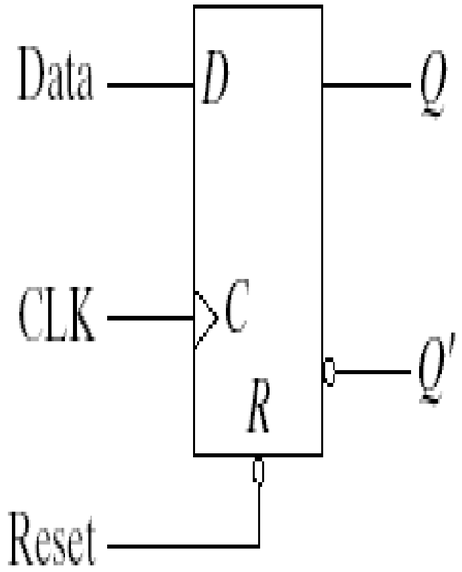
•When power is turned on a digital system, the state of the flip-flops is unknown. The direct inputs are useful for bringing all flip-flops in the system to a known starting state prior to the clocked operation.

D FLIP-FLOP WITH ASYNCHRONOUS RESET

A positive-edge-triggered D flip-flop with asynchronous reset.



(a) Circuit diagram



(b) Graphic symbol

R	C	D	Q	Q′
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(b) Function table

MEALY AND MOORE MODELS

• The most general model of a sequential circuit has inputs, outputs, and internal states. It is customary to distinguish between two models of sequential circuits:

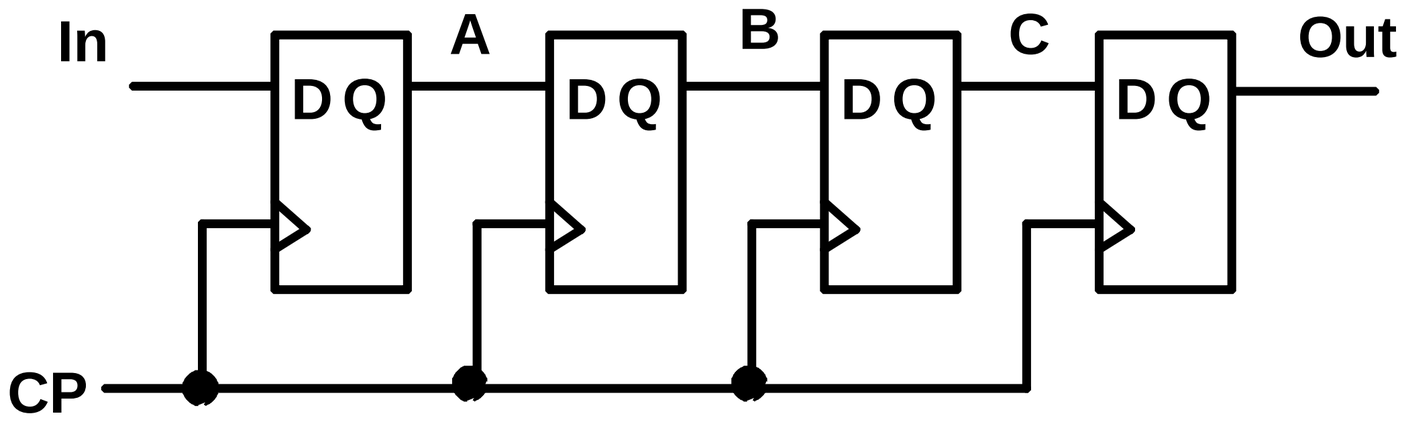
Mealy model and the Moore model

• They differ in the way the output is generated.

- In the **Mealy** model, the **output is a function of both the present state and input.**
- In the **Moore** model, the **output is a function of the present state only.**

SHIFT REGISTERS

- Shift Registers move data laterally within the register toward its MSB or LSB position
- In the simplest case, the shift register is simply a set of D flip-flops connected in a row like this:
- Data input, In, is called a serial input or the shift right input.
- Data output, Out, is often called the serial output.
- The vector (A, B, C, Out) is called the parallel output.

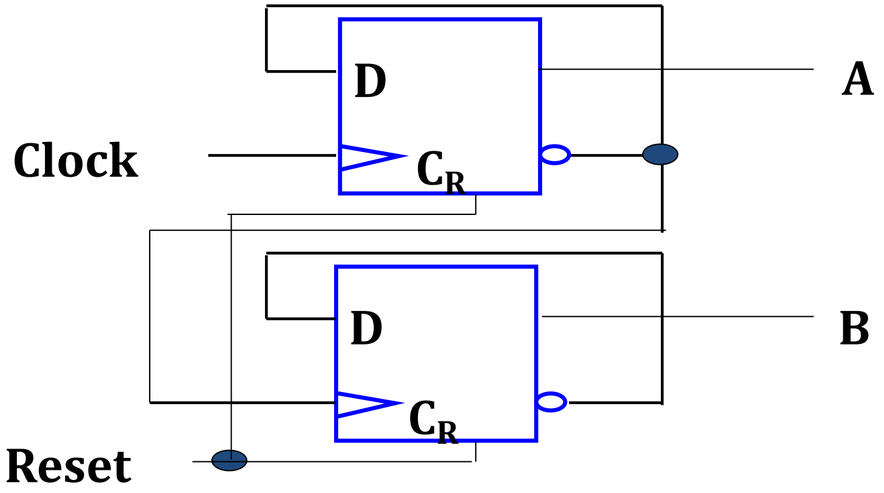


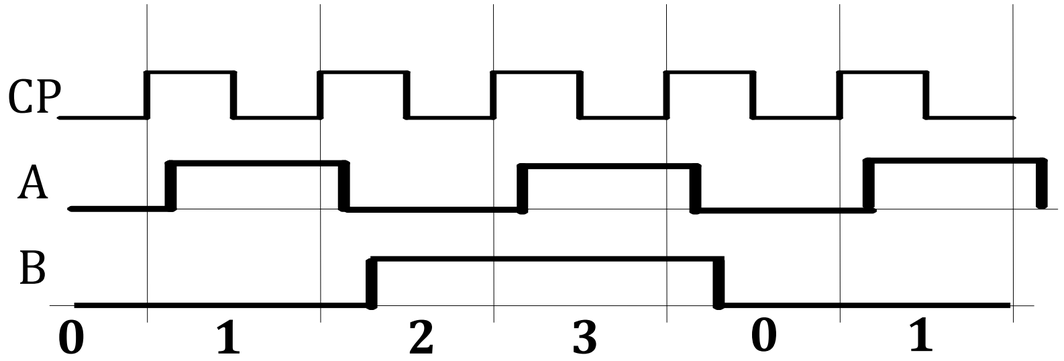
COUNTERS

- Counters are sequential circuits which "count" through a specific state sequence.
- It can count up, count down, or count through other fixed sequences.
- Two distinct types are in common usage:
- **Ripple Counters**
  - Clock is connected to the flip-flop clock input on the LSB bit flip-flop
  - For all other bits, a flip-flop output is connected to the clock input, thus circuit is not truly synchronous
  - Output change is delayed more for each bit toward the MSB.
  - Resurgent because of low power consumption
- **Synchronous Counters**
  - Clock is directly connected to the flip-flop clock inputs
  - Logic is used to implement the desired state sequencing

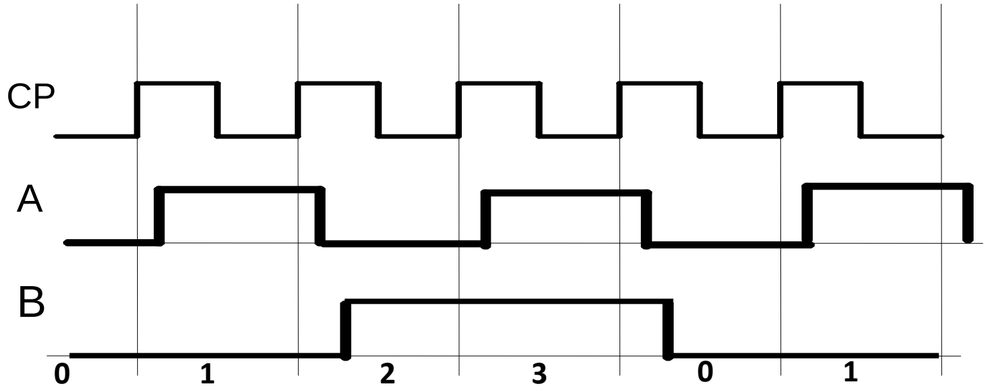
RIPPLE COUNTER

- How does it work?
  - When there is a positive edge on the clock input of A, A complements
  - The clock input for flip- flop B is the complemented output of flip-flop A
  - When flip A changes from 1 to 0, there is a positive edge on the clock input of B causing B to complement





RIPPLE COUNTER -1



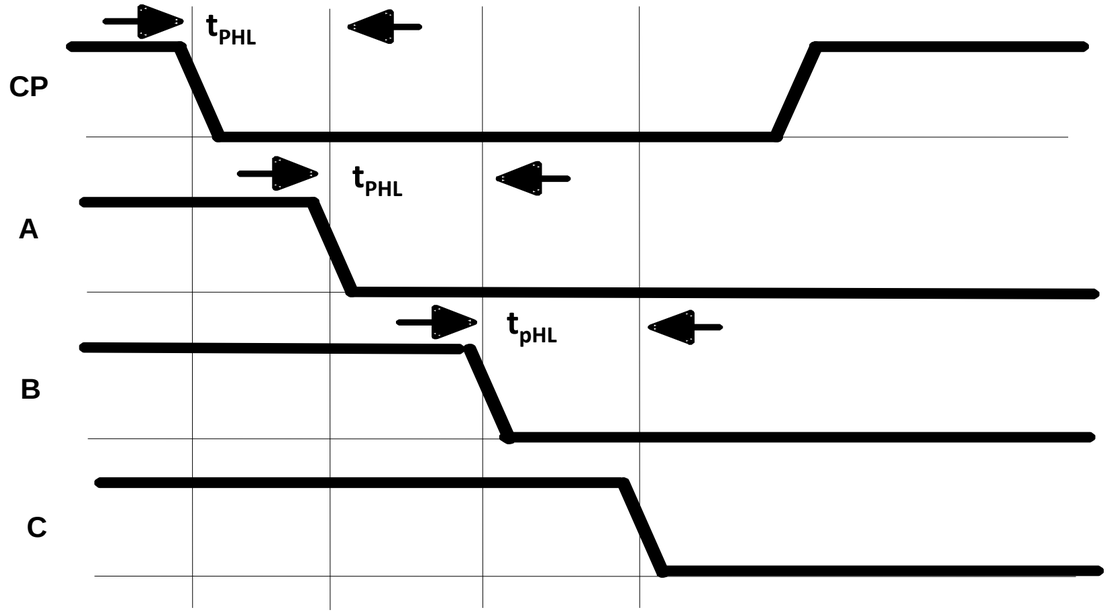
- The arrows show the cause-effect relation-ship from the prior slide =>
- The corresponding sequence of states => (B,A) = (0,0),
- Each additional bit, C, D, …behaves like bit B, changing half as frequently as the bit before it.
- For 3 bits: (C,B,A) = (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1), (0,0,0), …

RIPPLE COUNTER -2

- These circuits are called *ripple counters* because each edge sensitive transition (positive in the example) causes a change in the next flip-flop's state.
- The changes "ripple" upward through the chain of flip-flops, i. e., each transition occurs after a clock-to-output delay from the stage before.
- To see this effect in detail look at the waveforms on the next slide.

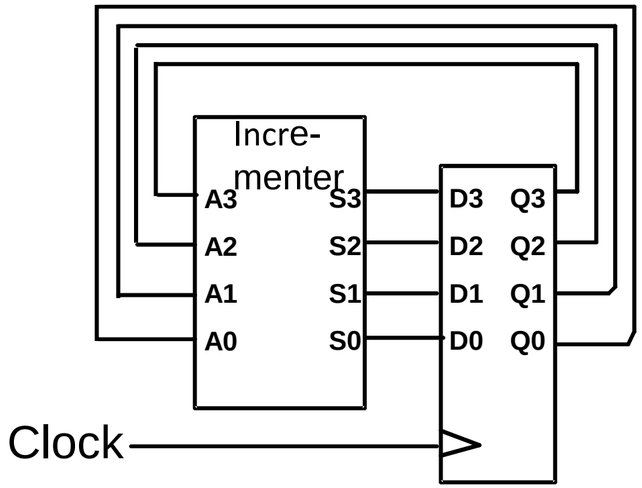
RIPPLE COUNTER -3

- Starting with C = B = A = 1, equivalent to (C,B,A) = 7 base 10, the next clock increments the count to (C,B,A) = 0 base 10. In fine timing detail:
  - The clock to output delay t_{PHL} causes an increasing delay from clock edge for each stage transition.
  - Thus, the count “ripples” from least to most significant bit.
  - For *n* bits, total worst case delay is *n* t_{PHL}.



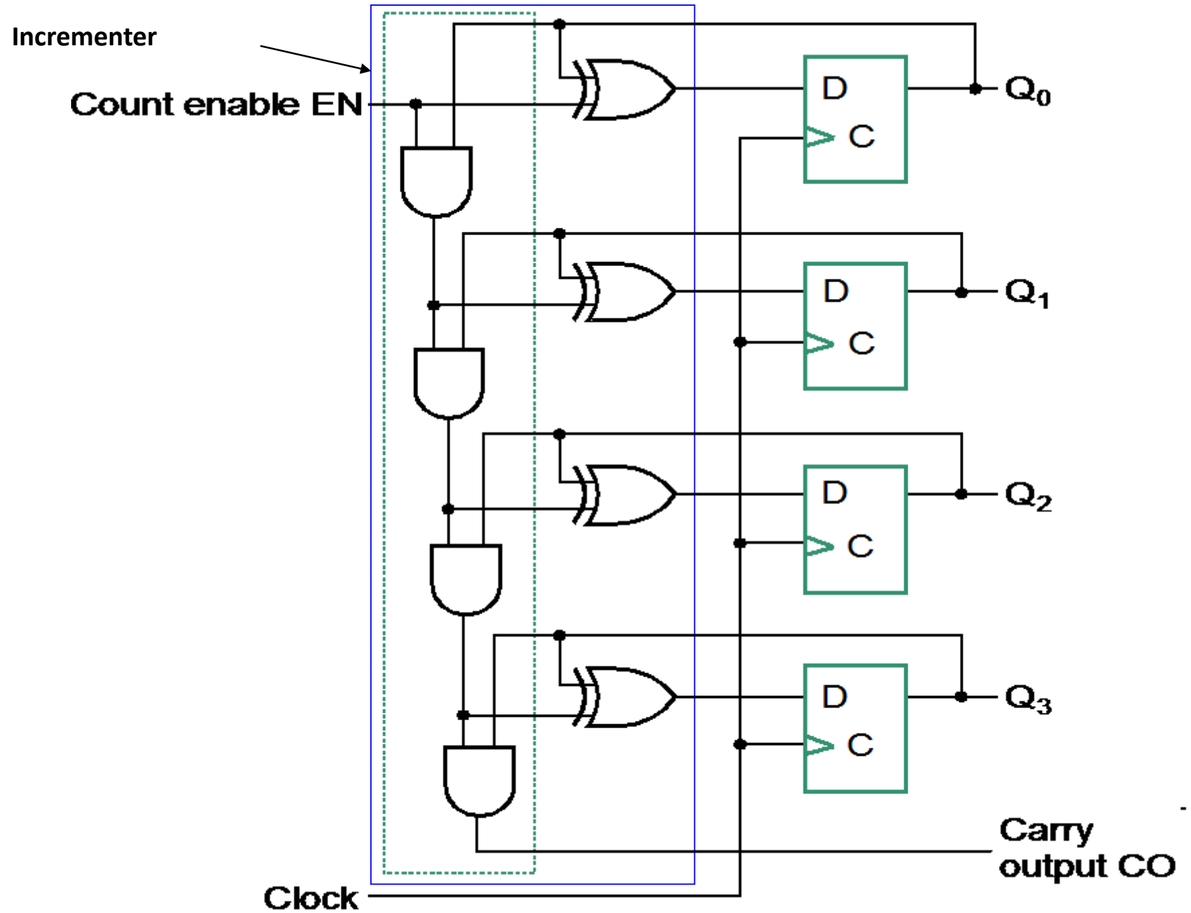
SYNCHRONOUS COUNTERS-1

- To eliminate the "ripple" effects, use a common clock for each flip-flop and a combinational circuit to generate the next state.
- For an up-counter,
  use an incrementer =>



SYNCHRONOUS COUNTERS-2

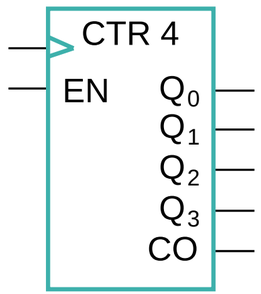
- Internal details =>
- Internal Logic
  - XOR complements each bit
  - AND chain causes complement of a bit if all bits toward LSB from it equal 1
- Count Enable
  - Forces all outputs of AND chain to 0 to “hold” the state
- Carry Out
  - Added as part of incrementer
  - Connect to Count Enable of additional 4-bit counters to form larger counters



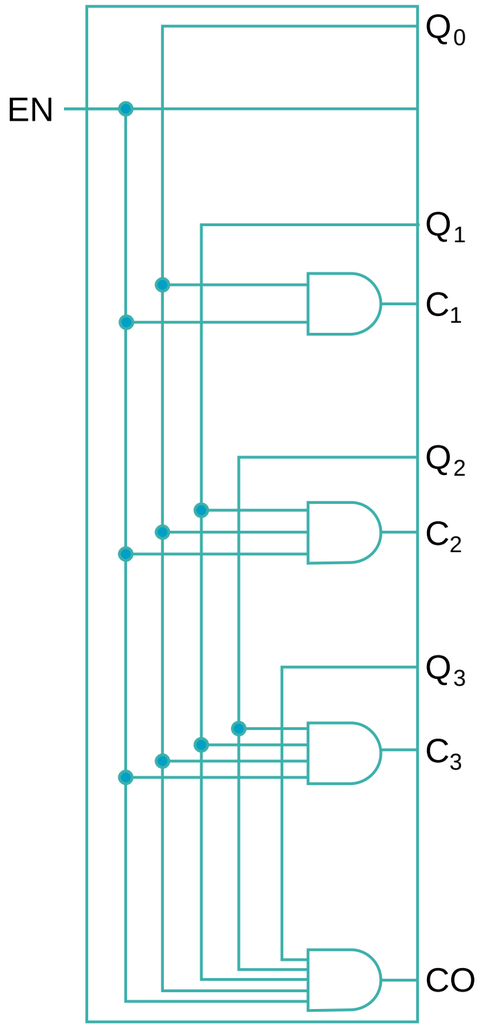
(a) Logic Diagram-Serial Gating

SYNCHRONOUS COUNTERS-3

- Carry chain
  - series of AND gates through which the carry “ripples”
  - Yields long path delays
  - Called *serial gating*
- Replace AND carry chain with ANDs => in parallel
  - Reduces path delays
  - Called *parallel gating*
  - Like carry lookahead
  - Lookahead can be used on COs and ENs to prevent long paths in large counters
- Symbol for Synchronous Counter



Symbol



Logic Diagram-Parallel Gating

ASSESSMENT

1. Compare synchronous & asynchronous sequential circuits.
2. Differentiate SR latch D latch
3. write down the characteristic equations of Flip Flop (FF)
4. What are the needs of direct inputs
5. Compare Mealy model and the Moore model
6. Draw & explain about Synchronous Counters
7. Draw & explain about Ripple Counters

SUMMARY & THANK YOU

- **OTHER COUNTERS**:
  - **Down Counter** - counts downward instead of upward
  - **Up-Down Counter** - counts up or down depending on value a control input such as Up/Down
  - **Parallel Load Counter** - Has parallel load of values available depending on control input such as Load
- **Divide-by-n (Modulo n) Counter**
  - Count is remainder of division by n which n may not be a power of 2 or
  - Count is arbitrary sequence of n states specifically designed state-by-state
  - Includes modulo 10 which is the **BCD counter**