Application of Flip flops

Application of Flip flops

- Registers
- Counters
- Shift registers

Registers

- A register is a memory device that can be used to store more than one bit of information.
- A register is usually realized as several flip-flops with common control signals that control the movement of data to and from the register.

- Common refers to the property that the control signals apply to all flip-flops in the same way
- A register is a generalization of a flip-flop.
- A flip flop stores one bit, a register stores several bits
- The main operations on a register are the same as for any storage devices, namely
    - oad or Store: Put new data into the register
    - ead: Retrieve the data stored in the register (usually without changing the stored data

Counters

- Counters are a specific type of sequential circuit or registers.
- Counters are sequential circuits which "count" through a specific state sequence. They can count up, count down, or count through other fixed sequences.

Shift Registers

- They are used for storage and transfer of data.
- Shift registers have no specific sequence of states unlike counters.

Benefits of counters

- Counters can act as simple clocks to keep track of time.
  - How many bits have been sent or received?
  - How many steps have been performed in some computation?
- All processors contain a program counter, or PC.
  - Programs consist of a list of instructions that are to be executed one after another (for the most part).
  - The PC keeps track of the instruction currently being executed.
  - The PC increments once on each clock cycle, and the next program instruction is then executed.

- Counters are important components in computers
  - The increment or decrement by one in response to input
- A counter that follows the binary number sequence is called a binary counter
  - n-bit binary counter: n flip-flops, count in binary from 0 to 2^n-1

Modulus

- Each of the counts of the counter is called state of the counter.
- A counter with *n* flip-flops can have 2^n states and of same modulus.
- The number of states in a counter is known as its mod (modulo) number.
- Thus a 2-bit counter is a mod-4 counter.
- A mod-n counter may also described as a divide-by-*n* counter.

- Counter is a Clocked sequential circuit whose state diagram contains a single cycle.
- The number of flip-flops determines the count limit or number of states.
- For example, a Modulus-12 counter would count from 0 (0000) to 11 (1011) and requires four flip-flops (16 states - 12 used).

- Counters with non-power of 2 modulus has unused states

Types of counter

- Counters are available in two types
  - Synchronous Counters(parallel counters)
  - Asynchronous Counters (Ripple Counters)

- Synchronous Counters:
  - A common clock signal is connected to the C input of each flip-flop
- Ripple Counters
  - Clock 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.

- Ripple counter let some flip-flop output to be used as clock signal source for other flip-flop
- Synchronous counter use the same clock signal for all flip-flop
  - Synchronous counters are faster than asynchronous counters because of the simultaneous clocking.
  - Synchronous counters are an example of *state machine* design because they have a set of states and a set of transition rules for moving between those states after each clocked event.

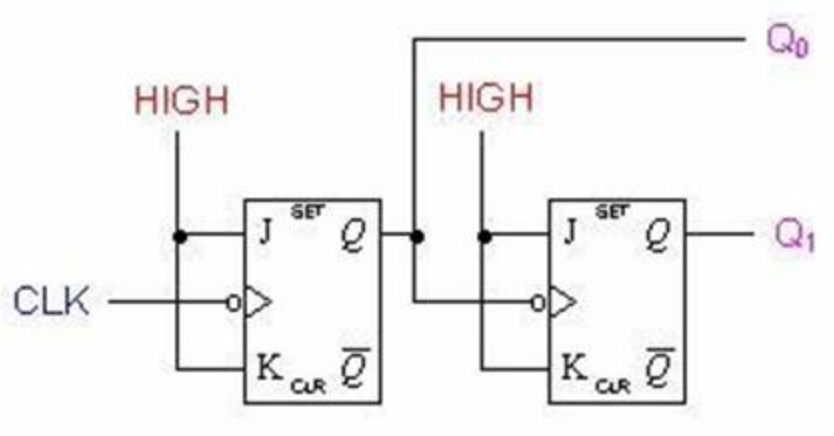
Other Counters

- Binary counter
  - Counter that follows a binary sequence
  - N bit binary counter counts in binary from n to 2^{n-1}
- 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; n may not be a power of 2
- Count is arbitrary sequence of n states specifically designed state-by-state
- Includes modulo 10 which is the BCD counter

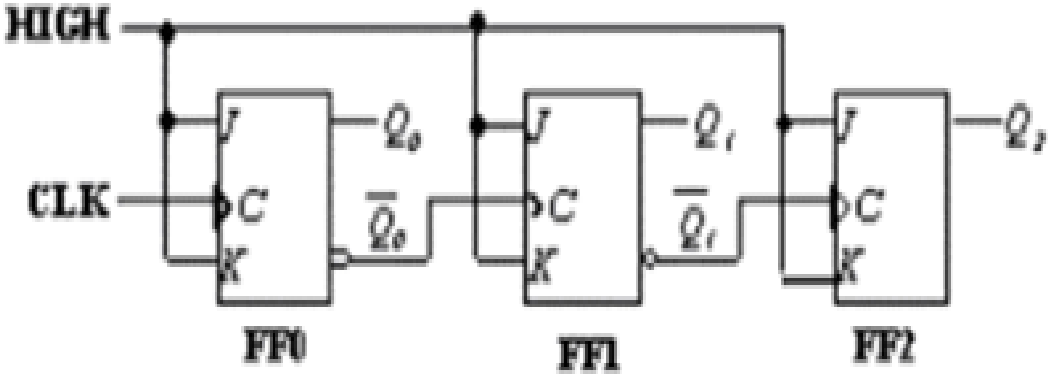
Two-bit asynchronous counter

Output from one flip-flop is connected to clock input for the next flip-flop MSB



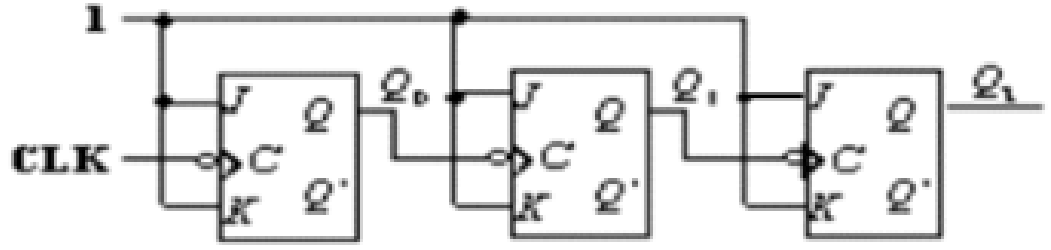
$00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$

3-bit ripple/asynchronous counter

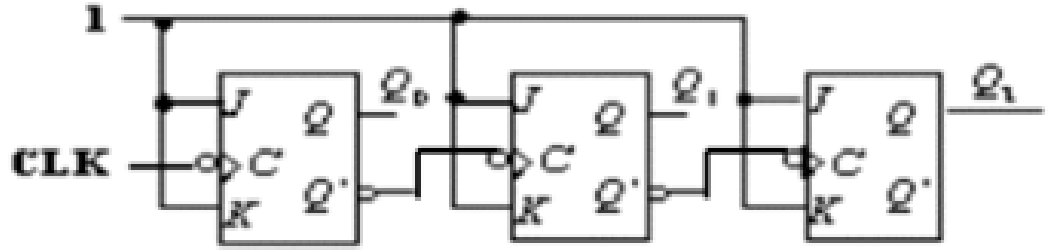


Asynchronous Down Counter

- Down asynchronous counter count from large to zero and repeat
- Example: 3-bit binary down counter



3-bit binary up counter

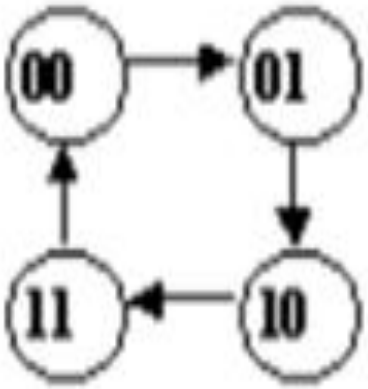


3-bit binary down counter

Synchronous Counter (Parallel)

- Synchronous counter: flip-flop with the same synchronous clock signal
- We can build synchronous counter using process to design sequential circuit
- Example: 2-bit synchronous binary counter (using T flip-flop or JK)

- Example: 2-bit synchronous binary counter (using T flip-flop or JK)



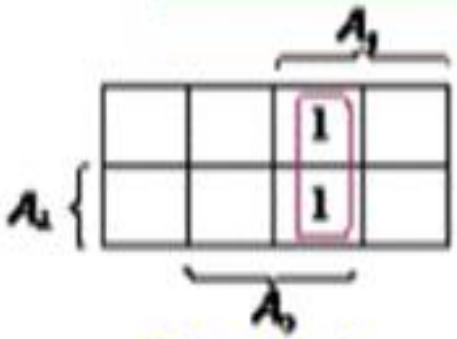
Present state		Next state		Flip-flop inputs	
A_1	A_0	A_1^+	A_0^+	TA_1	TA_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

$$TA_1 = A_0$$

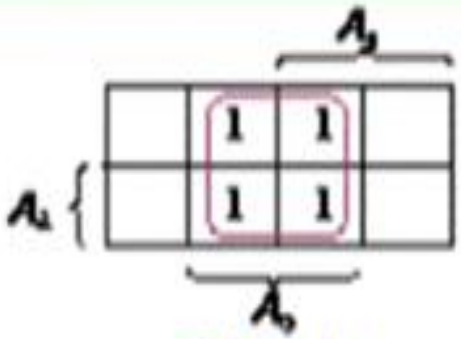
$$TA_0 = 1$$

- Example: 3-bit synchronous binary counter (using T flip)

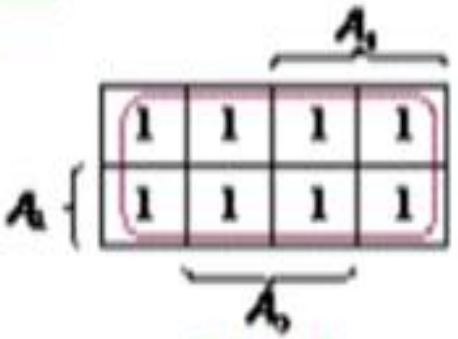
Present state			Next state			Flip-flop inputs		
A_2	A_1	A_0	A_2^*	A_1^*	A_0^*	TA_2	TA_1	TA_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1



$TA_2 = A_1 . A_0$



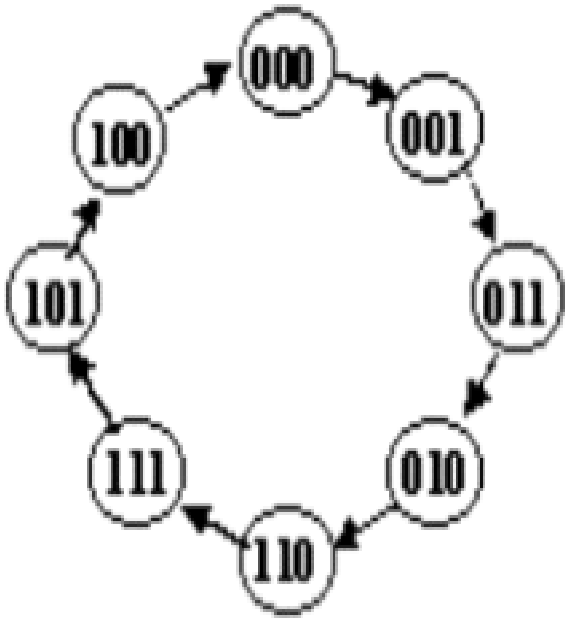
$TA_1 = A_0$



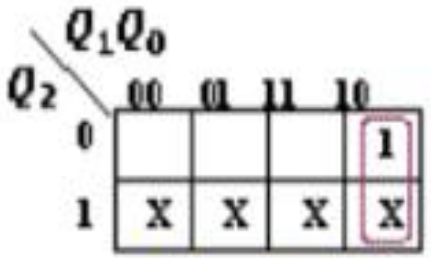
$TA_0 = 1$

Designing Synchronous Counter

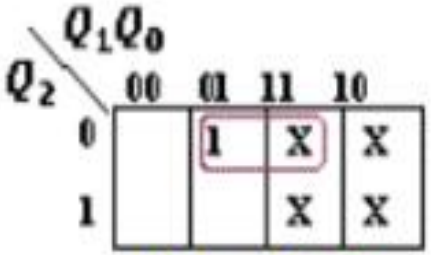
- Example: 3-bit Gray Code Counter (using JK flip-flop)



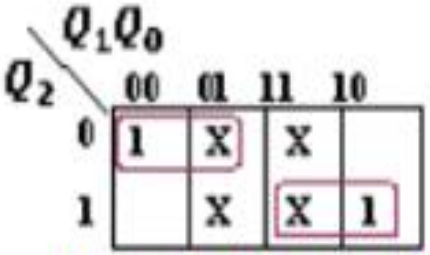
Present state			Next state			Flip-flop inputs					
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	JQ_2	KQ_2	JQ_1	KQ_1	JQ_0	KQ_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	0	1	1	0	1	X	X	0	0	X
0	1	1	0	1	0	0	X	X	0	X	1
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	1	X	0	X	1	X	0



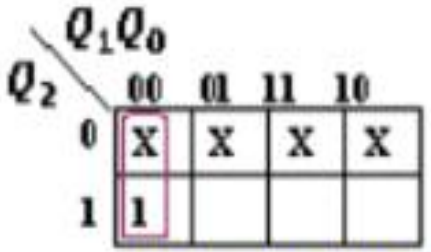
$JQ_2 = Q_1 \cdot Q_0'$



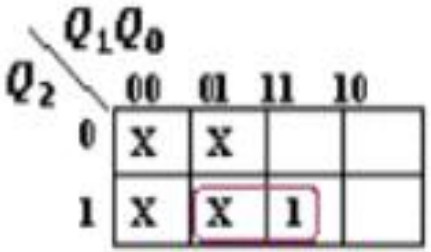
$JQ_1 = Q_2' \cdot Q_0$



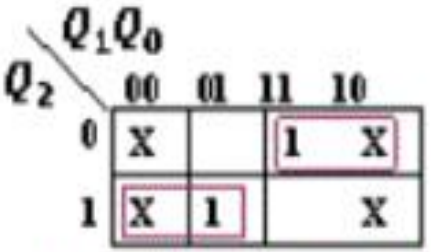
$JQ_0 = Q_2 Q_1 + Q_2' \cdot Q_1'$
$= (Q_2 \oplus Q_1)'$



$KQ_2 = Q_1' \cdot Q_0'$



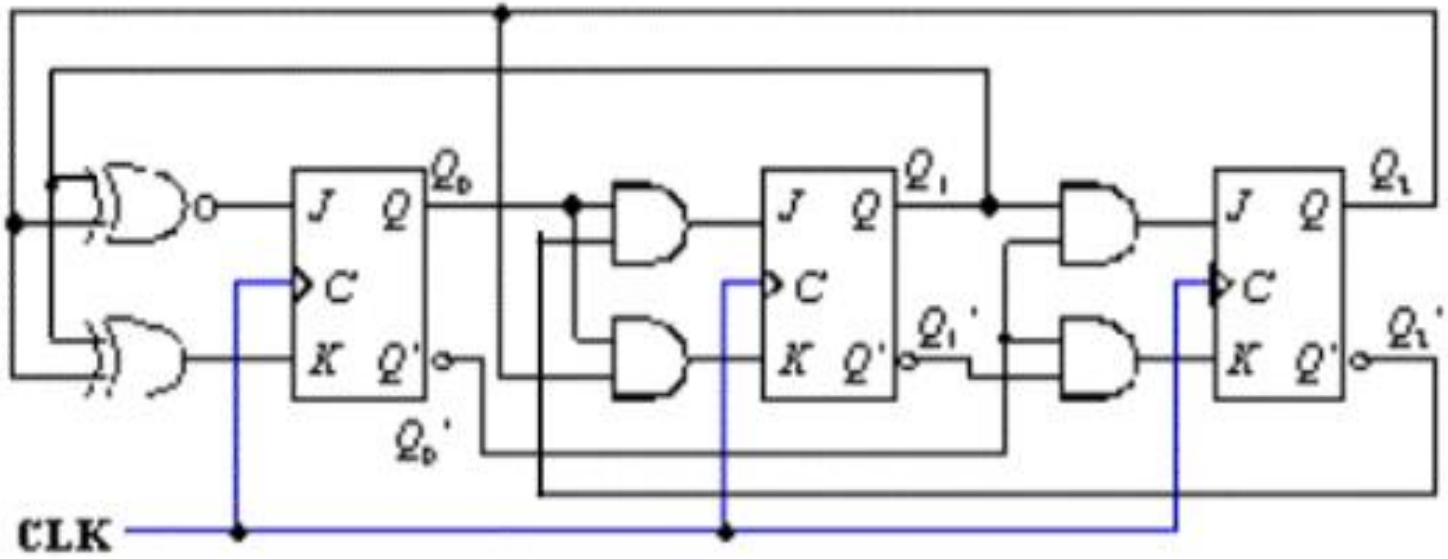
$KQ_1 = Q_2 \cdot Q_0$



$KQ_0 = Q_2 Q_1' + Q_2' \cdot Q_1$
$= Q_2 \oplus Q_1$

$JQ_2 = Q_1.Q_0'$ $JQ_1 = Q_2'.Q_0$ $JQ_0 = (Q_2 \oplus Q_1)'$

$KQ_2 = Q_1'.Q_0'$ $KQ_1 = Q_2.Q_0$ $KQ_0 = Q_2 \oplus Q_1$



Example 1

- Design a 3 bit synchronous counter using JK flip flops

Example 2

- Design a 3 bit UP/DOWN [bidirectional]counter using JK Flip flops

Design of synchronous counter with unused states

- Self starting counter
  - Whenever a counter goes to an invalid state, it comes back to a valid state in 1 or 2 clock pulses and then starts counting in a normal sequence, it is called self starting.
- Lock out condition
  - If a counter enters an invalid state, it keeps on moving from one invalid state to another when clock pulses are applied.
  - It never returns to a valid state.

Example 1

- Design a mod 6 synchronous counter using JK flip flops Check whether the counter is self starting.

Example 2

- Design a synchronous BCD counter using JK flip flops. Check whether the counter is self starting.

Example 3

- Design a synchronous counter using T flip flops that goes through states 0,3,5,6,0.. Check whether the counter is self starting. Check whether any lock out condition arises.

Design of Asynchronous Counter

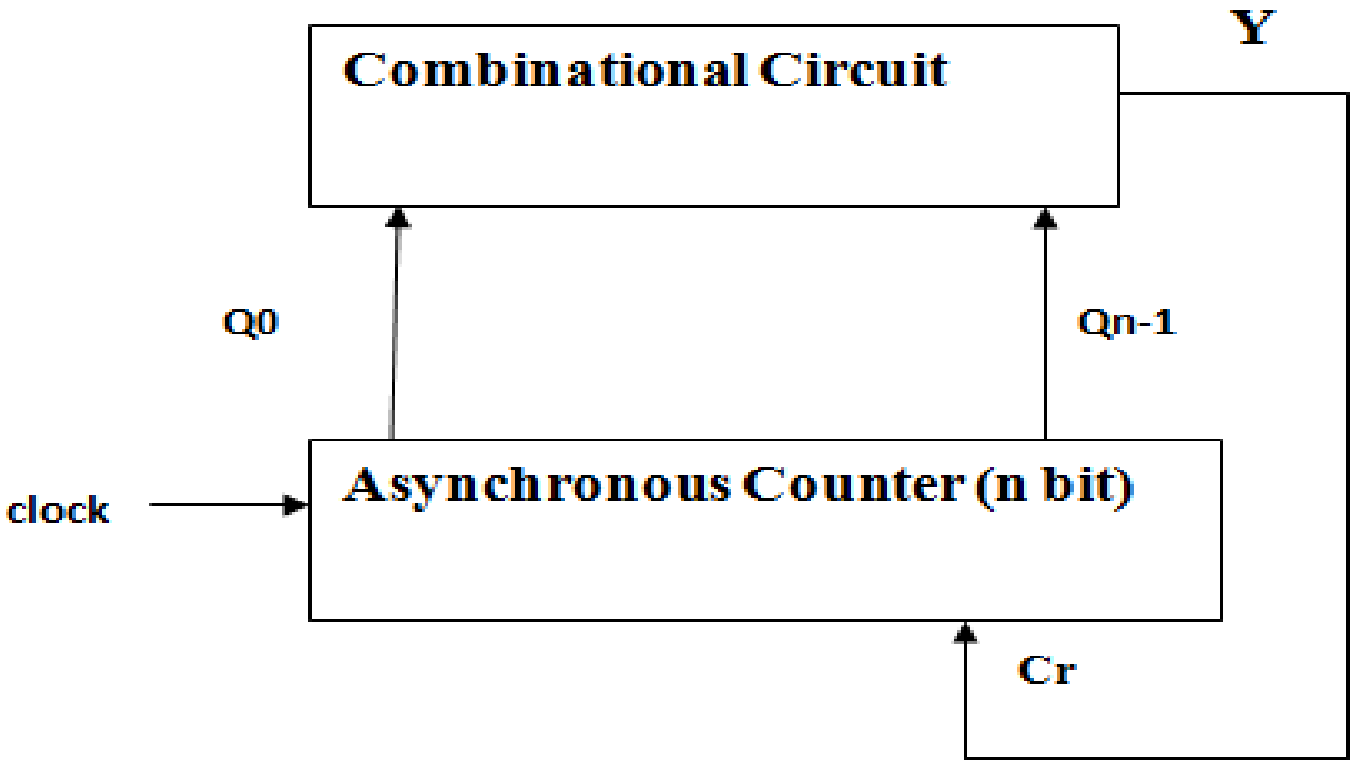
Asynchronous Counter

- Can be designed using T or JK flip flops
- These flip flops are used in toggle mode
- The flip flops are not clocked simultaneously
- The out put of previous flip flop becomes input to the next flip flop, hence asynchronous.
- The inputs J and K of the flip flops are 1, this acts as a toggle switch and out put of flip flop is complemented.

Mod M Asynchronous Counter

- An n bit asynchronous counter counts $N = 2^n$ clock pulses.
- To count M clock pulses, which is less than N, the Reset terminal of Flip flop is used.
- A combinational circuit is designed such that all the flip flops are reset after the count M.
- The out put of the combinational circuit must be zero after M clock pulses

Block diagram of Mod M asynchronous counter



Procedure to design a Mod M Asynchronous Counter

- Find the minimum number of flip flops required(n: M< 2^n) .
- Prepare the sequence.
- Design the combinational circuit such that all the flip flops are reset after M clock pulses.
  - Draw the truth table of the asynchronous counter with out put of the combinational circuit (Y) such that Y=1 for valid states and Y=0 for invalid states
  - Find out simplified expression for Y using K map
  - Draw the logic diagram

Example 1

- Design a Mod 4 ripple counter using JK flip flops.

Example 2

- Design a Mod 9 ripple counter using T flip flops.

Synthesis of synchronous Sequential Circuits

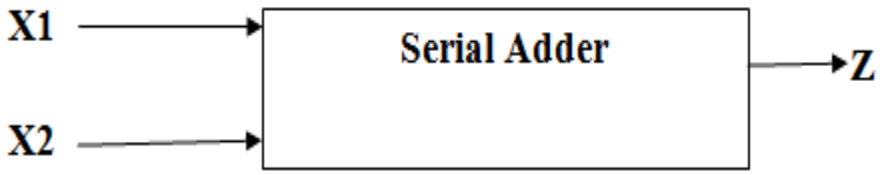
Synthesis of synchronous Sequential Circuits

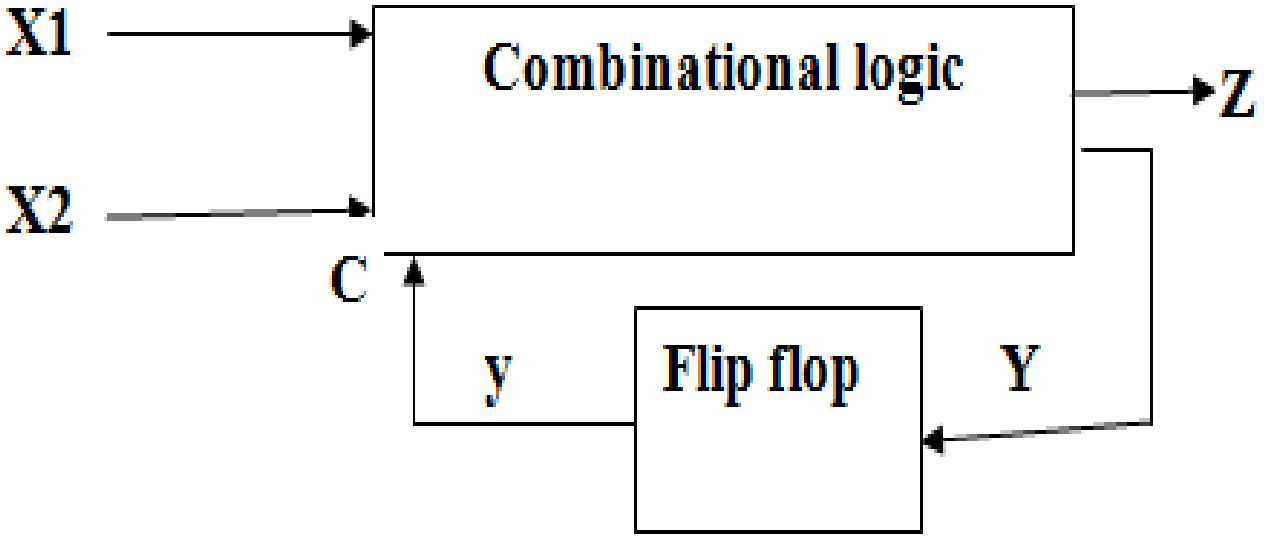
- Steps are
  - Form a description of the problem through a state table/diagram.
  - If possible, reduce states.
  - Make state assignment and derive the transition or out put table
  - Select the type and number of memory elements to be used
  - Using k map, obtain minimal expression for the flip flop inputs
  - Draw the logic diagram

Synthesis of synchronous Sequential Circuits

- Serial binary adder
- Sequence generator
- Sequence detector

Serial binary adder

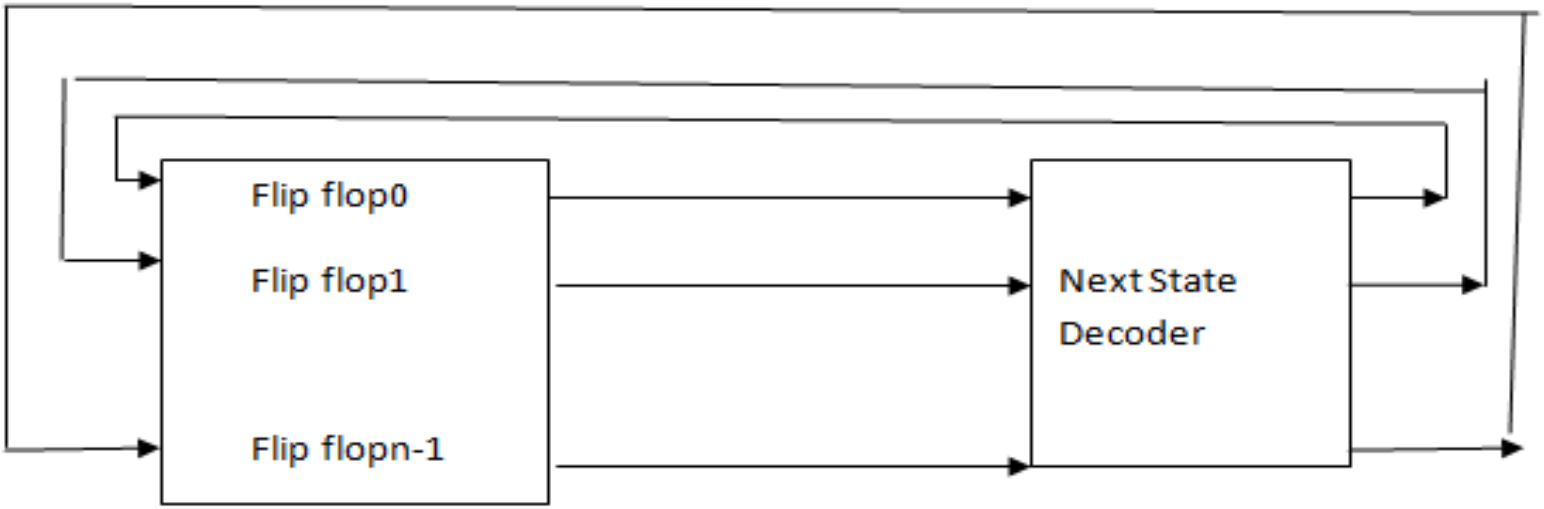




Sequence Generator

- This is a sequential circuit which generates a desired sequence of bits in synchronous with a clock.
- Also called pulse generator, in which a particular repetitive series of pulses are generated

Block Diagram



To design a sequence generator,

- first calculate the number of flip flops required
- Then, design the next state decoder to generate the desired sequence

Design procedure

1) Find the minimum number of flip flops required (n).
   - It depends on the nature of sequence
   - Can be calculated using the equation

$$Max(C_0, C_1) \leq 2^{n-1}$$

Here, C_0 is the count of zeros and C_1 is the count of ones.

2) Draw the state table as per the following procedure
   - Starting from LSB, assign the desired sequence to the out put of the flip flops, which produces the out put
   - Assign the out put of the other flip flops 0 or 1, such that all the states are different.

3) Draw the state diagram
4) Compute the inputs to the flip flops and simplify
5) Draw the logic diagram

Example 1

- Design a sequence generator using D flip flops to generate the sequence 101100110

Sequence Detector

- It is a sequential machine which produces an out put of 1 every time the desired sequence is detected, otherwise, gives output as 0.

Example 1

- Design a sequence detector to detect the sequence 1010 and overlapping is permitted. Assume the input sequence is 01101010 and the corresponding out put sequence is 00000101