Analog and Digital Electronics

(Second Year BTech Computer)

(2019-20)

(Faculty: Mr. Y. K. Sharma)

Courtesy: R. P. Jain (Modern Digital Electronics)

UNIT- 4

Counters

Courtesy: R. P. Jain & J. S. Katre

Counters

- **Counters:** Asynchronous counter. Synchronous counter, ring counters, Johnson Counter, Modulus of the counter (IC 7490).

- **Synchronous Sequential Circuit Design:** Models – Moore and Mealy, State diagram and State Tables, Design Procedure, Sequence generator and detector.

- **Asynchronous Sequential Circuit Design:** Difference with synchronous circuit design, design principles and procedure, applications.

Counters

- There are two types of counters Asynchronous (ripple) & Synchronous counters.
- Asynchronous counter is easy to design & requires the least amount of homework.
- Asynchronous counter is not triggered simultaneously & is also called as serial or series counter.
- Design of Synchronous counter requires some amount of homework & each Flip-Flop is triggered simultaneously.
- Each count of the counter is called as a state of the counter.
- The number of states through which the counter passes before returning to the starting state is called the modulus of the counter.
- E.g. A 2-bit counter has 4 states, it is called a mod-4 counter & as it divides the clock signal frequency by 4, it is called as divide-by-4 counter.
- An n-bit counter will have n FF's & 2^n states, and divides the input frequency by 2^n. Hence it is called as divide-by-2^n counter. The LSB of ripple counters is the Q output of the FF to which the external clock is applied.

Counters

Classification of Sequential Circuits:

- Synchronous sequential circuits: Contents of memory elements can be changed only at the rising or falling edges of clock signal.
- Asynchronous sequential circuits: Contents of memory elements can be changed at any instant of time.

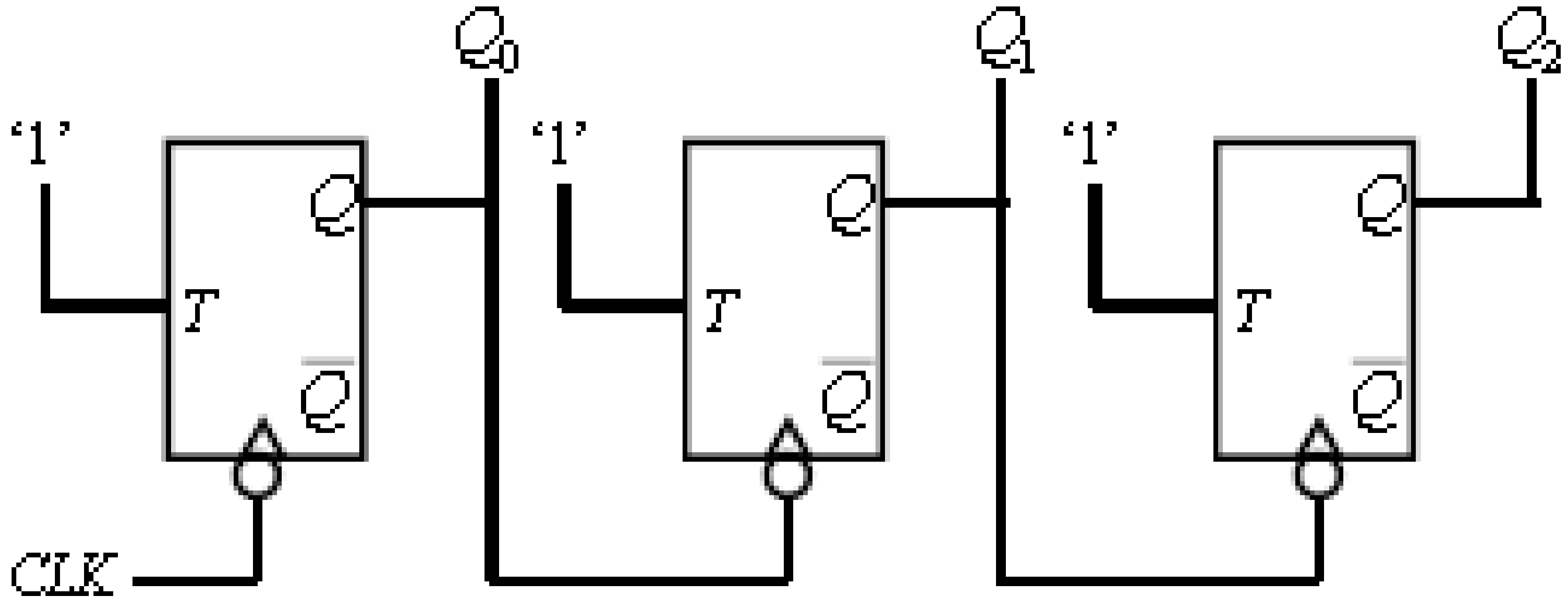
Asynchronous	Synchronous
1. Depends upon the sequence in which the input signals change.	1. Behaviour can be defined from the knowledge of its signal at discrete instants of time.
2. Commonly used memory elements are time delays.	2. Memory elements used are Flip-Flops.
3. Design is tedious.	3. Design is easy.
4. They are combinational circuit with feedback.	4. Synchronization is achieved by system clock.
	5. Also called as clocked sequential circuit.

Asynchronous Inputs

- It is common for the FF types we have mentioned to also have additional so called ‘asynchronous’ inputs
- They are called asynchronous since they take effect independently of any clock or enable inputs.
- Reset/Clear – force *Q* to 0
- Preset/Set – force *Q* to 1
- Often used to force a synchronous circuit into a known state, say at start-up.

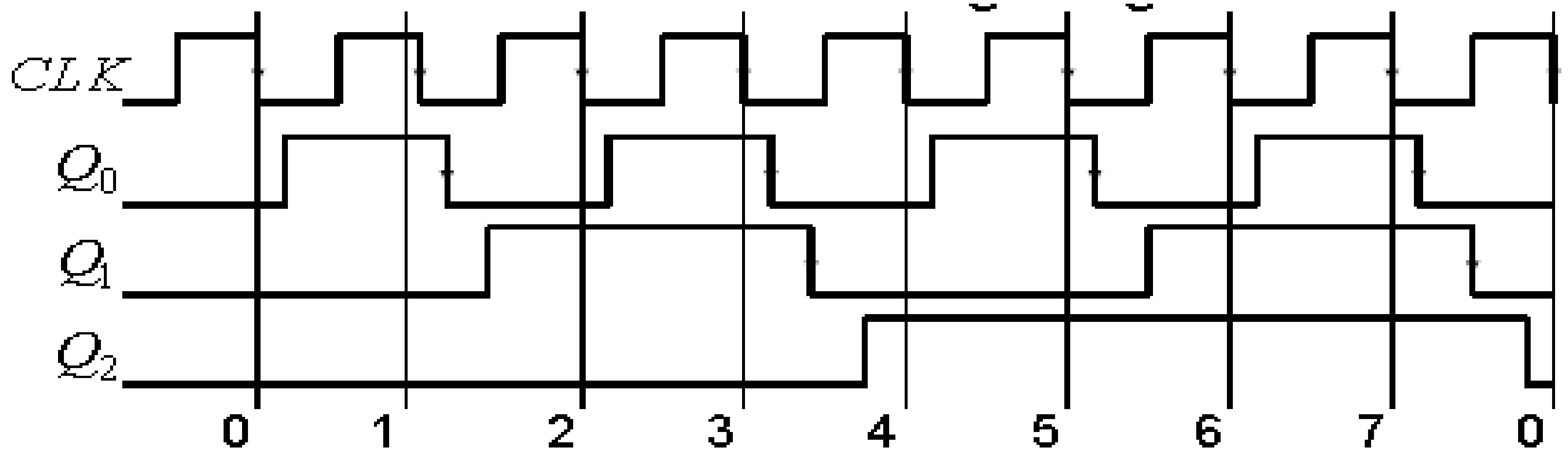
Ripple Counters

- It can be designed by using negative edge triggered T-type FF’s operating in toggle mode, i.e T=1.
- Since FF’s are not clocked using the same clock therefore it is called as asynchronous or ripple counters.



Ripple Counters

- Timing diagram for ripple counter.
- Outputs do not change synchronously, so hard to know when count output is actually valid.
- Propagation delay builds up from stage to stage, limiting maximum clock speed before miscounting occurs.



Synchronous Counters

- All flip flop clock inputs are directly connected to the clock signal and so all FF outputs change at the same time, i.e. synchronously.
- More complex combinational logic is now needed to generate the appropriate FF input signals(Which will be different depending upon the type of FF chosen.)
- We can design synchronous counters using any type of Flip-Flops.
- We will consider using D-type FFs to design 0-7 up counter.
- While designing counter we will make use of a modified state transition table. This table has additional columns that defines the required FF inputs(or excitation table values).
- We make use of excitation table for a particular type of FF.
- Firstly investigate the so called characteristic table and characteristic equation for a FF.
- For D-FFs it is not necessary to write out the FF input columns, since we know they are identical to those for the next state.
- To design the circuit we need to determine appropriate combinational logic circuits which will generate the required FF inputs from the current states.
- This can be done using Boolean algebra or using K-maps.

Characteristic Table

In general, a characteristic table for a FF gives the next state of the output, i.e.,Q' in terms of its current state Q and current inputs

Q	D	Q'
0	0	0
0	1	1
1	0	0
1	1	1

Which gives the characteristic equation,

$$Q' = D$$

i.e., the next output state is equal to the current input value

Since Q' is independent of Q the characteristic table can be rewritten as

D	Q'
0	0
1	1

Excitation Table

The characteristic table can be modified to give the excitation table. This table tells us the required FF input value required to achieve a particular next state from a given current state

Q	Q'	D
0	0	0
0	1	1
1	0	0
1	1	1

As with the characteristic table it can be seen that Q', does not depend upon, Q, however this is not generally true for other FF types, in which case, the excitation table is more useful. Clearly for a D-FF, $D = Q'$

Characteristic and Excitation Tables

- Characteristic and excitation tables can be determined for other FF types.
- These should be used in the design process if D-type FFs are not used
- We will now determine the modified state transition table for the example 0 to 7 up-counter

Modified State Transition Table

In addition to columns representing the current and desired next states (as in a conventional state transition table), the modified table has additional columns representing the required FF inputs to achieve the next desired FF states

Modified State Transitation Table

For a 0 to 7 counter, 3 D-type FFs are needed

Current state			Next state			FF inputs		
Q_2	Q_1	Q_0	$Q_2^{'}$	$Q_1^{'}$	$Q_0^{'}$	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

The procedure is to:

Write down the desired count sequence in the current state columns

Write down the required next states in the next state columns

Fill in the FF inputs required to give the defined next state

Note: Since $Q' = D$ (or $D = Q'$) for a D-FF, the required FF inputs are identical to the Next state

Synchronous Counter Example

- Also note that if we are using D-type FFs, it is not necessary to explicitly write out the FF input columns, since we know they are identical to those for the next state
- To complete the design we now have to determine appropriate combinational logic circuits which will generate the required FF inputs from the current states
- We can do this from inspection, using Boolean algebra or using K-maps.

Synchronous Counter Example

Current state $Q_2 Q_1 Q_0$			Next state $Q_2' Q_1' Q_0'$			FF inputs $D_2 D_1 D_0$		
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

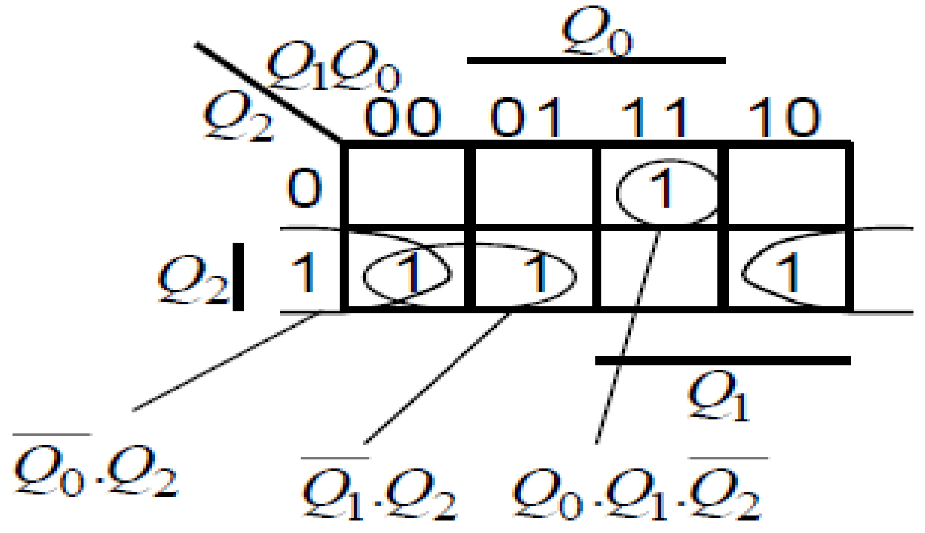
By inspection,

$$D_0 = \overline{Q_0}$$

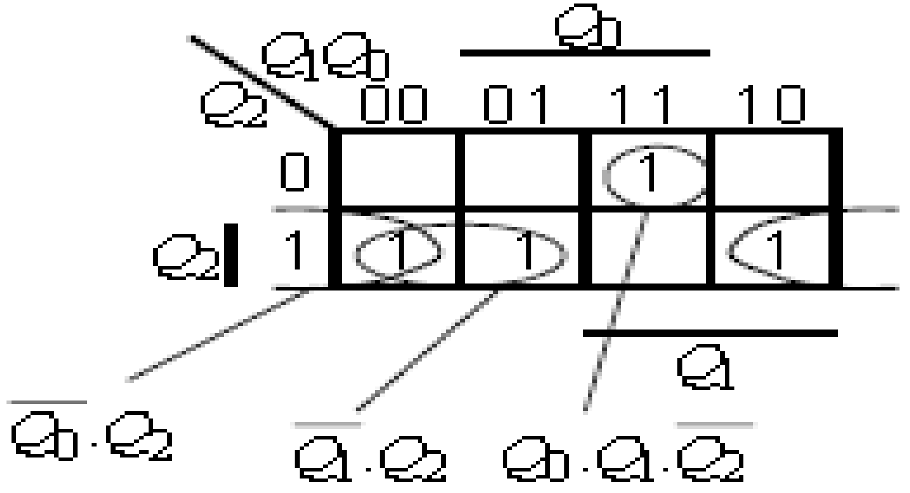
Note: FF$_0$ is toggling

Also, $D_1 = Q_0 \oplus Q_1$

Use a K-map for D_2,


Q_1Q_0
Q_2
Q_0
00
01
11
10
0
1
Q_2
Q_1
1
1
1
1
$\overline{Q_0}\cdot Q_2$
$\overline{Q_1}\cdot Q_2$
$Q_0\cdot Q_1\cdot \overline{Q_2}$

Synchronous Counter Example

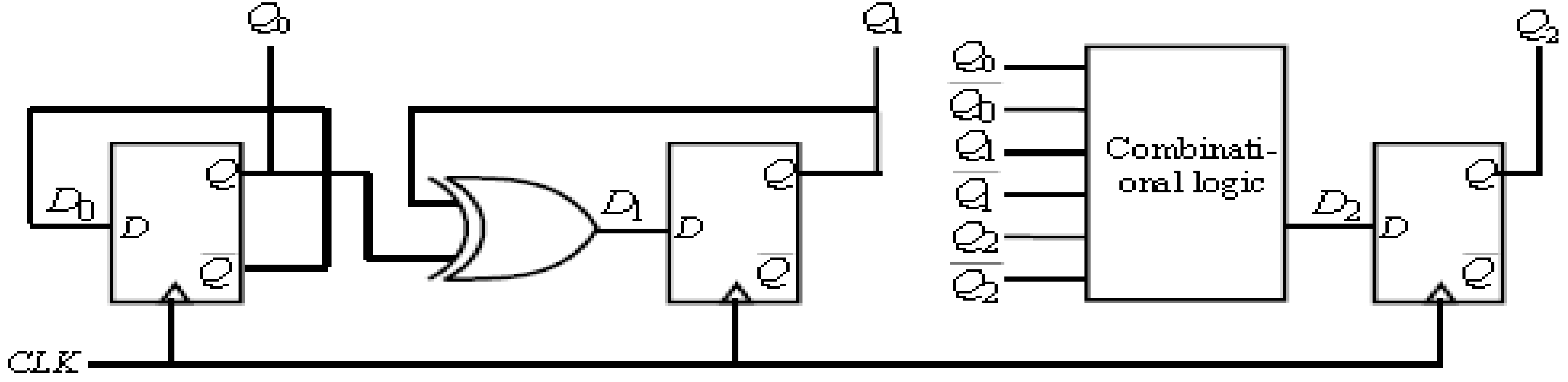


Q_1Q_0
Q_2
Q_0
00 01 11 10
0
1
Q_2
1
1 1
1
1
Q_1
$\overline{Q_0}\cdot Q_2$
$\overline{Q_1}\cdot Q_2$
$Q_0\cdot Q_1\cdot\overline{Q_2}$

So,

$$D_2 = \overline{Q_0}\cdot Q_2 + \overline{Q_1}\cdot Q_2 + Q_0\cdot Q_1\cdot\overline{Q_2}$$

$$D_2 = Q_2\cdot(\overline{Q_0}\cdot + \overline{Q_1}) + Q_0\cdot Q_1\cdot\overline{Q_2}$$



Synchronous Counter

- A similar procedure can be used to design counters having an arbitrary count sequence
  - Write down the state transition table
  - Determine the FF excitation (easy for D-types)
  - Determine the combinational logic necessary to generate the required FF excitation from the current states – **Note:** remember to take into account any unused counts since these can be used as don't care states when determining the combinational logic circuits

Ring Counter

- If the serial output Q_0 of the shift register is connected back to the serial input, then an injected pulse will keep circulating. This circuit is referred to as a ring counter.
- The pulse is injected by entering 00001 in the parallel form after clearing the Flip-Flop's.
- When Clock pulses are applied, this 1 circulates around the circuit.
- The outputs are sequential non-overlapping pulses which are useful for control-state counters, for stepper motor (which rotates in steps) which require sequential pulses to rotate it from one position to the next etc.
- This circuit can also be used for counting the number of pulses.
- The number of pulses counted is read by noting which FF is in state 1.
- Since there is one pulse at the output for each of the N clock pulses, this circuit is referred to as a divide-by-N counter or an N:1 scalar.
- If $\overline{Q_0}$ is connected to the serial output, the resulting circuit is referred to as a twisted ring, johnson, or moebius counter.
- If the clock pulses are applied after clearing the FF's, square waveform is obtained at the Q outputs.
- It is useful for the generation of multiphase clock.

Ring Counter

- Ring counters are constructed by modifying the serial-in, serial-out, shift registers. There are two types of ring counters- basic ring counter & Johnson counter.
- The basic ring counter can be obtained from a serial-in, serial-out, shift register by connecting Q output of the last FF to D input of first FF.
- The ring counter is the decimal counter. It is divide-by-N counter, where N is the number of stages. The keyboard encoder is an application of a ring counter.

–

Twisted Ring Counter (Johnson Counter)

- It is obtained from a serial-in, serial-out shift register by providing a feedback from Q bar of last FF to the D input of first FF, called as twisted ring counter.
- The output Q is connected to the D input of next stage.
- It produces the unique sequence of states.
- Initially all the FF's are reset i.e. the state of counter is 0000.
- After each clock pulse, the level of Q1 is shifted to Q2, the level of Q2 to Q3, Q3 to Q4 & finally Q4 bar to Q1. The sequence is repeated after every 8 clock pulse.
- N Flip-Flop or n-bit Johnson counter can have 2n unique states and can count up to 2n states. So it is mod-2n counter.

Twisted Ring Counter (Johnson Counter)

Advantages:

1. It is more economical than ring counter but less than ripple counter.

Disadvantages:

1. It requires 2 input gates for decoding regardless of number of Flip-Flops.
2. Both the ring counters suffer from problem of lock out i.e. if the counter finds itself in an unused state, it will persist in moving from one unused state to another and will never find its way to a used state.

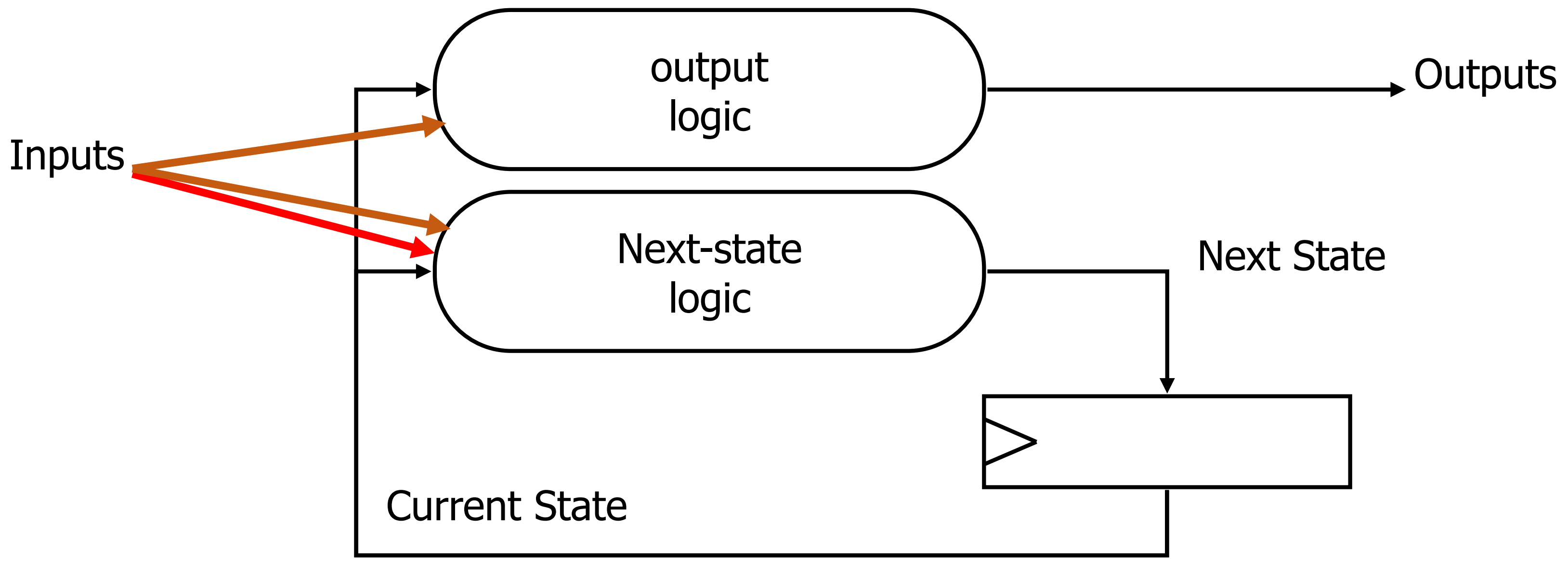
Modulus of the counter (IC 7490)

7490 Asynchronous Counter IC: (Modulus of the counter)

- IC's available are divided into 3 groups A,B,C depending on its features. It consists of four MS-FF.
- The load, set & reset (clear) operations are asynchronous. i.e. independent of clock pulse.
- It consists of four FF's internally connected to provide mod-2 counter & a mod-5 counter. These counters can be used independently or in combination.
- FFA operates as a mod-2 counter whereas the combination of FFB, FFC & FFD form a mod-5 counter.
- The two reset inputs R1 & R2 both are connected to logic 1 level for clearing all FF's.
- The two set inputs S1 & S2 when connected to logic 1 level, are used for setting the counter to 1001.
- E.g. In a 7490 if QA output is connected to B input and the pulses are applied at A input, find the count sequence. Ans. It is a decade counter.

Generalized FSM model: Moore and Mealy

- Combinational logic computes next state and outputs
  - Next state is a function of current state and inputs
  - Outputs are functions of
    - Current state (**Moore** machine)
    - Current state and inputs (**Mealy** machine)

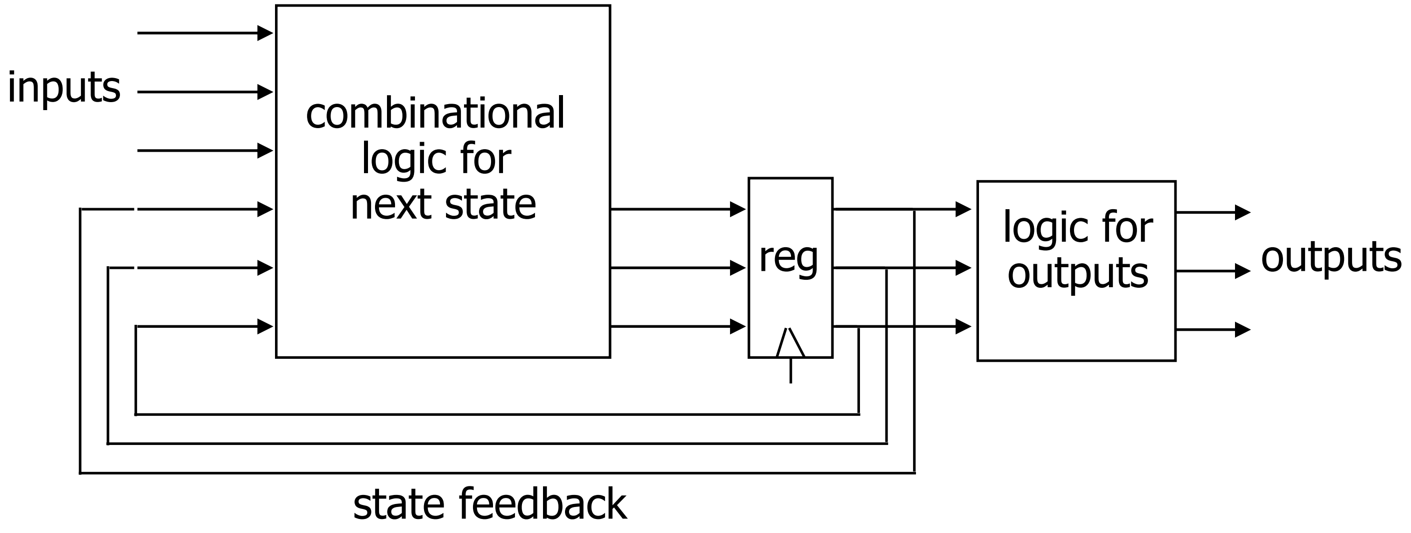


Moore and Mealy Machines

There are two types of finite state machine that can be built from sequential logic circuits:

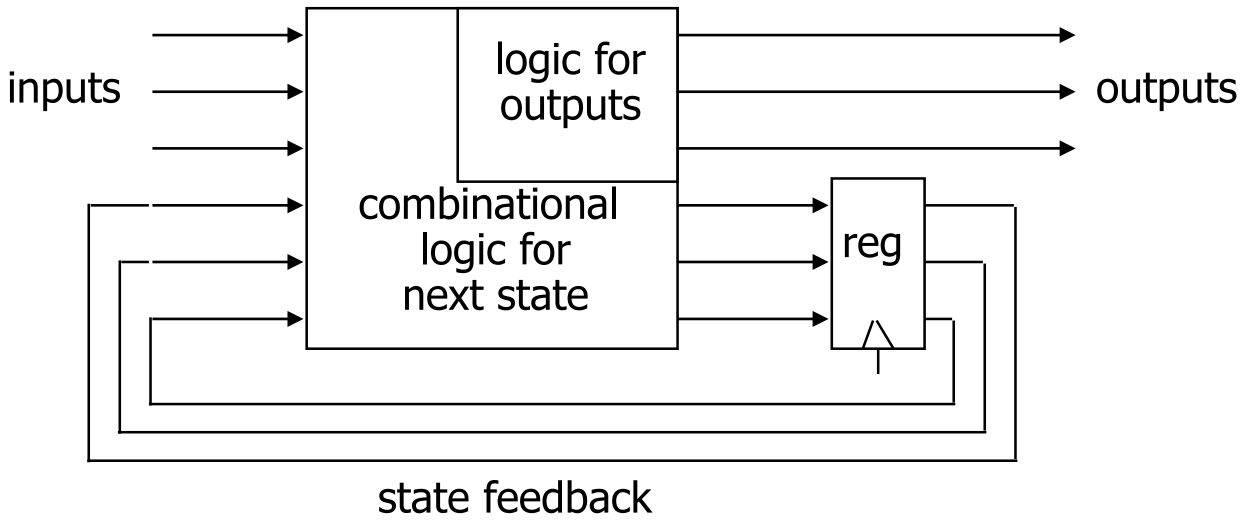
- Moore machine: The output depends only on the internal state. (Since the internal state only changes on a clock edge, the output only changes on a clock edge).
- Mealy Machine: The output depends not only on the internal state, but also on the inputs.
- A clocked sequential system is a kind of Moore machine.

Moore versus Mealy machines



Moore machine
Outputs are a function of current state

Outputs change synchronously with state changes



Mealy machine
Outputs depend on state and on inputs

Input changes can cause immediate output changes (**asynchronous**)

Impacts start of the FSM design procedure

- Counter-design procedure
  1. State diagram
  2. State-transition table
  3. Next-state logic minimization
  4. Implement the design

- FSM-design procedure
  1. State diagram
  2. State-transition table
  3. State minimization
  4. State encoding
  5. Next-state logic minimization
  6. Implement the design

State Diagrams

- Moore machine
  - Each state is labeled by a pair:

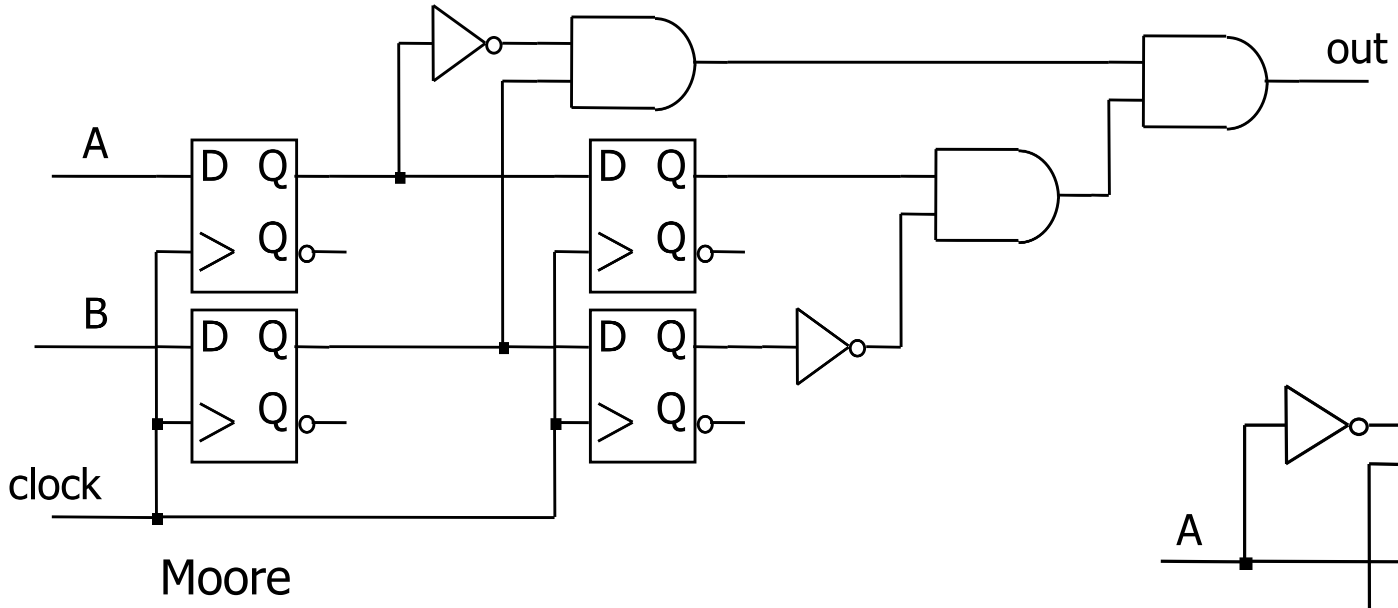
    state-name/output   or   state-name [output]

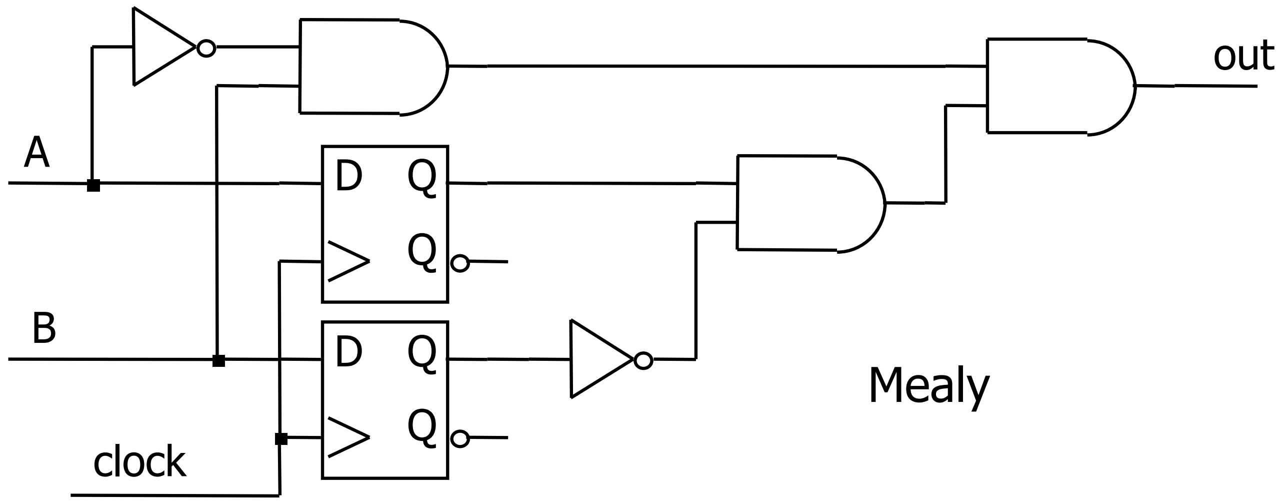
- Mealy machine
  - Each transition arc is labeled by a pair:

    input-condition/output

Example 10 → 01: Moore or Mealy?

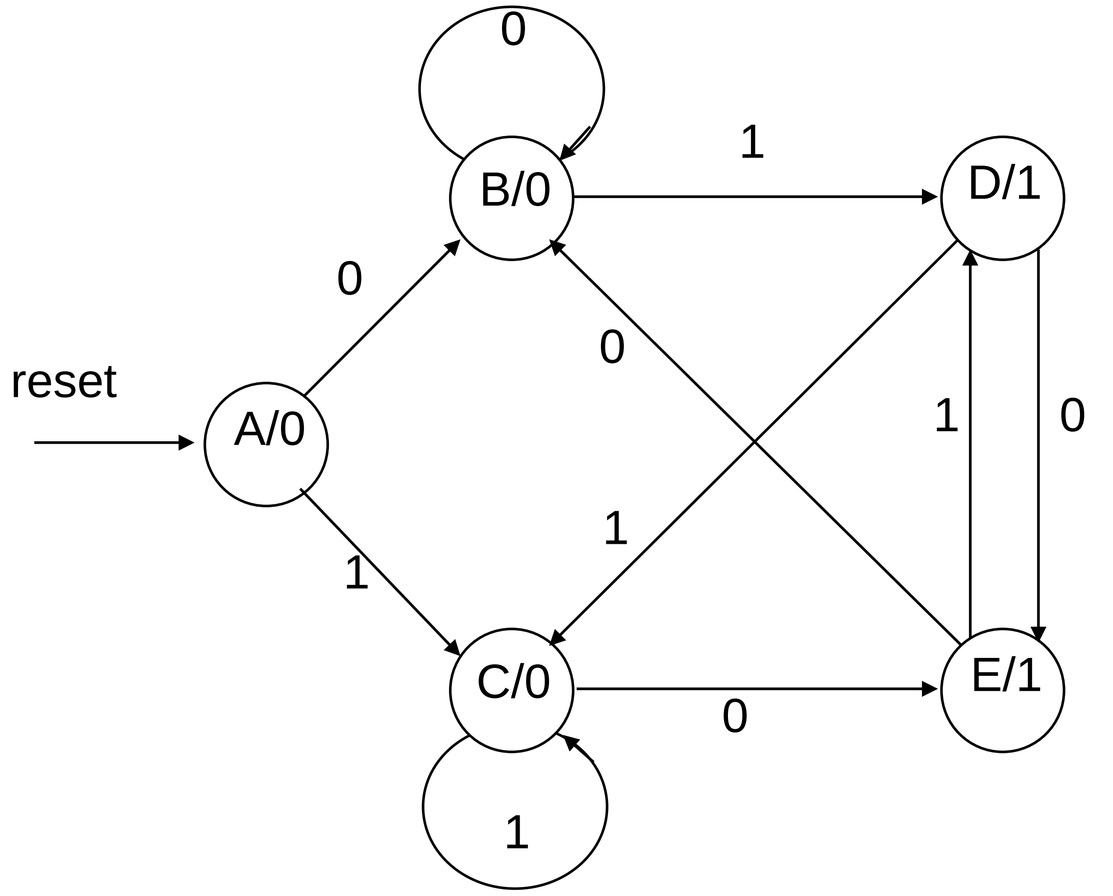
- Circuits recognize AB=10 followed by AB=01
  - What kinds of machines are they?





Example “01 or 10” detector: a Moore machine

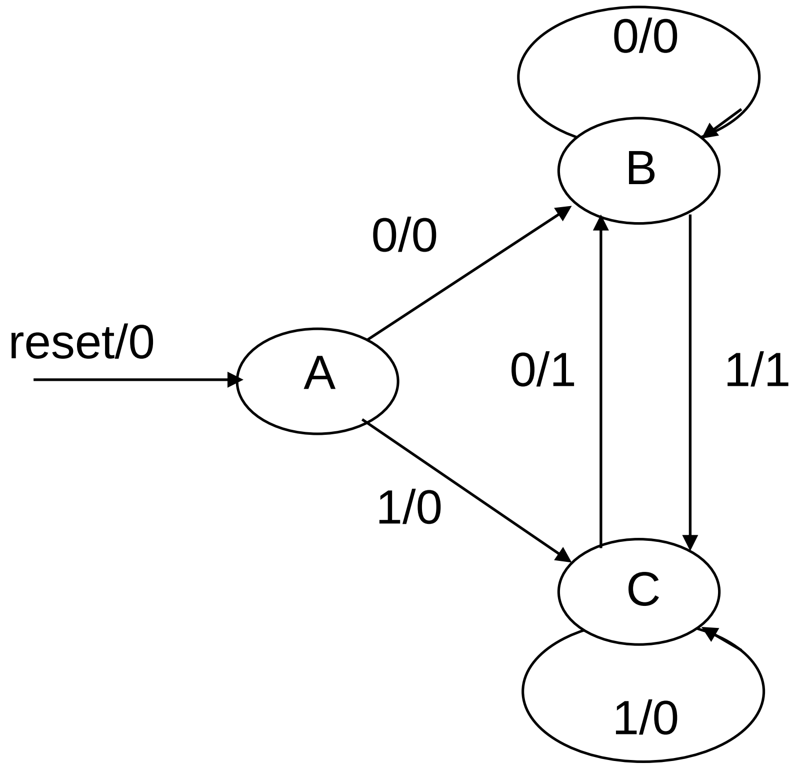
- Output is a function of state only
  - Specify output in the state bubble



reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	0
0	0	C	E	0
0	1	C	C	0
0	0	D	E	1
0	1	D	C	1
0	0	E	B	1
0	1	E	D	1

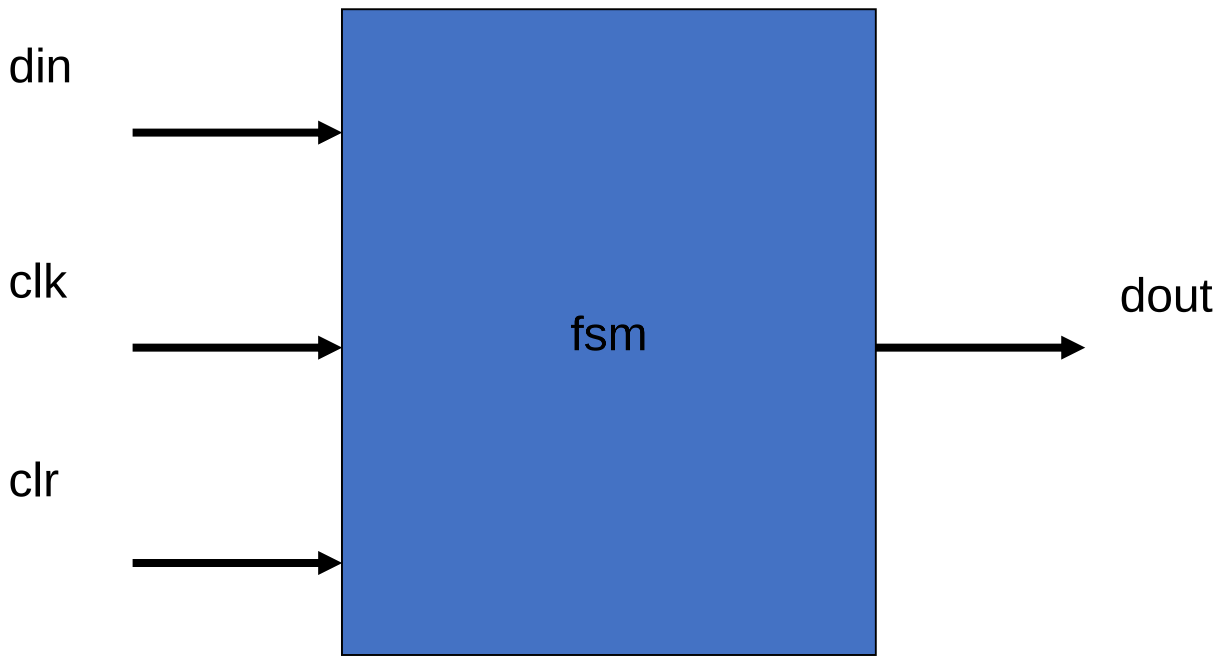
Example “01 or 10” detector: a Mealy machine

- Output is a function of state and inputs
  - Specify outputs on transition arcs



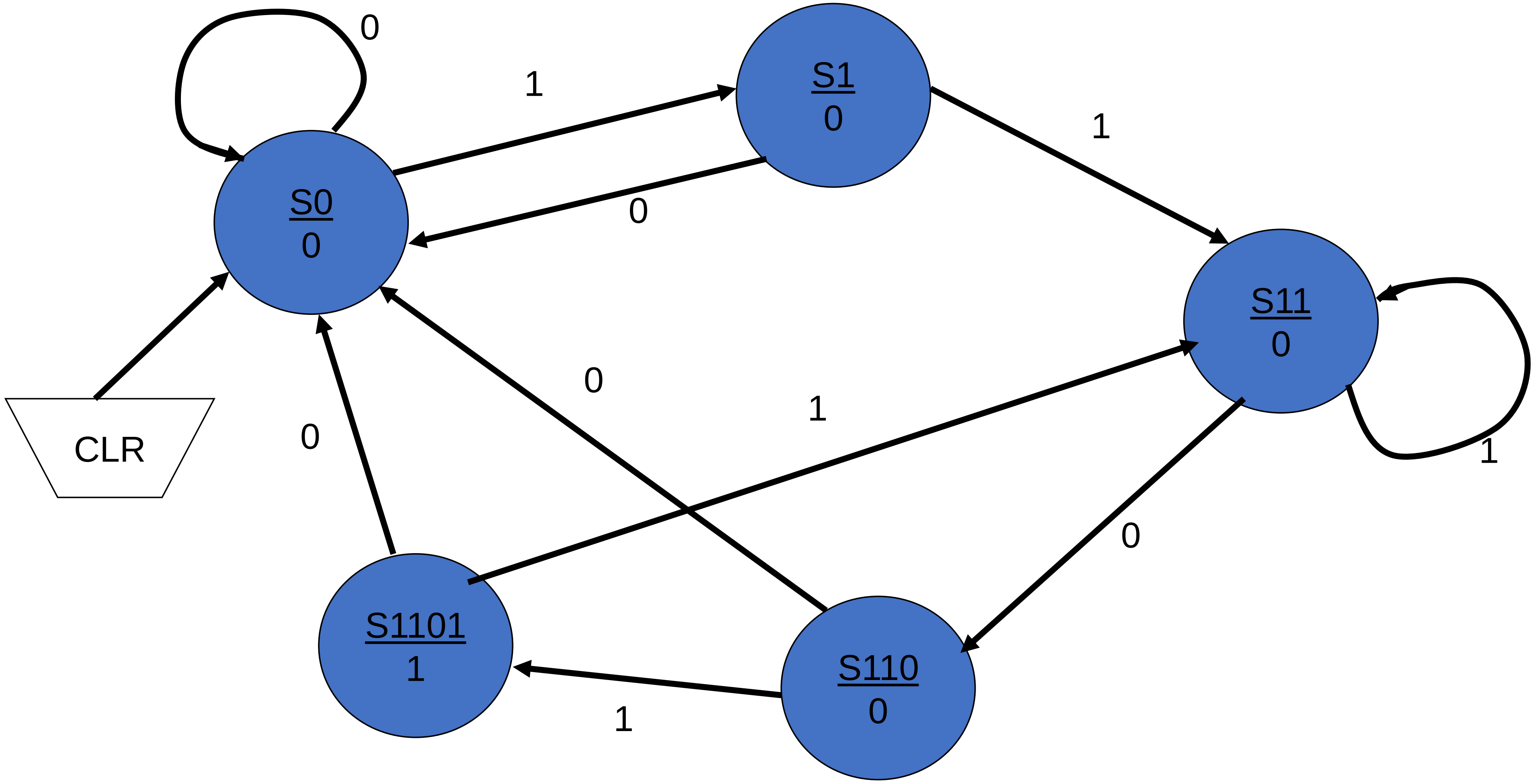
reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

Detect input sequence 1101



din	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0
dout	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0

Creating a State Diagram
Detect input sequence 1101



Mealy and Moore Machine

In general, a sequential machine will have the following:

1. A set S containing a finite number, say *p*, of internal states, so that

   $S=\{S_1, S_2,\ldots\ldots S_p\}$

2. A set X having a finite number, say *n*, of inputs, so that

   $X=\{X_1, X_2,\ldots\ldots X_n\}$

3. A set Z containing a finite number, say *m*, of outputs, so that

   $Z=\{Z_1, Z_2,\ldots\ldots Z_m\}$

4. A characterizing function *f* that uniquely defines the next state S^{t+1} as a function of the present state S^t and the present input X^t , so that $S^{t+1} = f(S^t , X^t)$

5.A) Mealy machine

A characterizing function *g* that uniquely defines the output Z^t as a function of the present input X^t and the present internal state S^t , so that

$$Z^t = g(S^t , X^t)$$

5.B) Moore machine

A characterizing function *g* that uniquely defines the output Z^t as a function of the present internal state S^t , so that

$$Z^t = g(S^t)$$

Mealy and Moore Machine

A sequential machine can therefore formally be defined as follows:

Definition:A sequential machine is a quintuple,

M=(X,Z,S,*f*,*g*), where X, Z and S are the finite and nonempty sets of inputs, outputs, and states respectively.

f is the next-state function, such that

$$S^{t+1} = f(S^t , X^t)$$

and the *g* is the output function such that

$Z^t = g(S^t , X^t)$ for a Mealy machine

$Z^t = g(S^t)$ for a Moore machine

To describe a sequential machine, either a state table or a state diagram is used.

State table

- Table1 is a state table describing an example sequential machine M_1. It can be seen that machine M_1 has a set of four internal states A,B,C and D, a set of two inputs I_1 and I_2 and a set of outputs O_1 ,O_2
- The characterizing functions *f* and *g* are depicted in tabular form, which is the state table.

State table

- State table of a Mealy machine M_1

Present state	Next state, output	
	Input	
	I_1	I_2
A	A,O_1	B,O_2
B	D,O_2	A,O_1
C	B,O_1	D,O_2
D	A,O_1	C,O_1

State table

- For example, for the present state B when the input is I_1, the next state is D and the output is O_2. If the input is I_2, the next state is A and the output is O_1.
- Thus the table shows the next state and the output for each combination of the present state and the input.
- Since the output of the machine M_1 depends on both the present state and the input, it is a Mealy machine.
- Table2 shows the state table of a Moore machine. Here the output is independent of the input and depends only on the present state of the machine.
- Therefore, this table has a separate column defining the outputs, and two input columns defining the next state without having any output associated with it.

State table

- State table of a Moore machine M_2

Present state	Next state		Output
	Input		
	I_1	I_2	
A	B	C	O_1
B	C	D	O_2
C	A	C	O_1
D	A	C	O_2

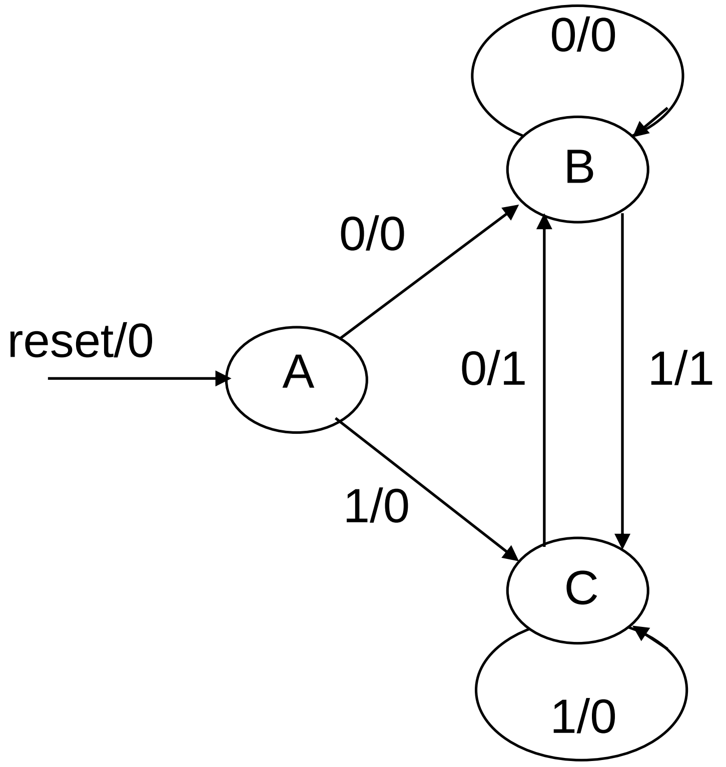
State table

- Another interesting property of of the machines M_1, M_2 which we have depicted in the two state tables is that for all combinations of present state and input, the next state and the output are completely specified. Such machines are therefore called completely specified sequential machines (CSSMs).
- There is another clas of sequential machines, where sometimes the next state or the output or both may remain unspecified. Such machines are known as incompetely specified sequential machines (ISSMs).

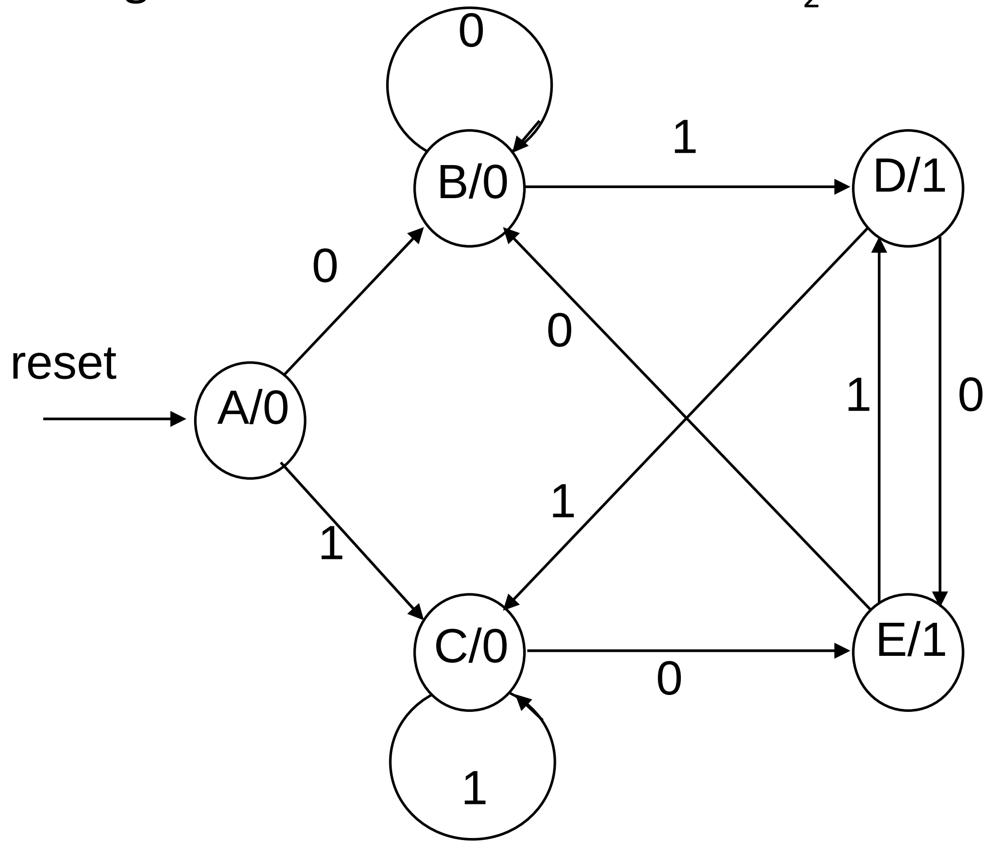
State diagram

- The information contained in the state table can also be shown in a graphical manner with the help of nodes conected by directed graphs. Such diagrams are called state diagrams.
- Following figures show the state diagrams of machines M_1 and M_2 respectively.

State diagram of the Mealy machine M_1



State diagram of the Moore machine M_2

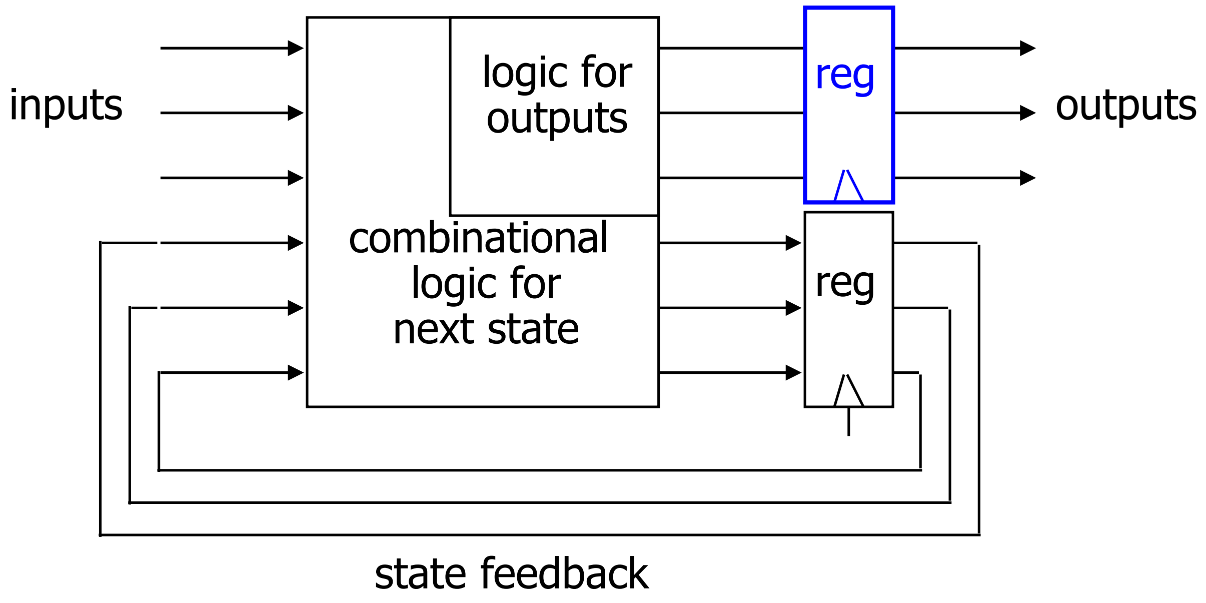


Comparing Moore and Mealy machines

- Moore machines
  - + Safer to use because outputs change at clock edge
  - – May take additional logic to decode state into outputs
- Mealy machines
  - + Typically have fewer states
  - + React faster to inputs — don't wait for clock
  - – Asynchronous outputs can be dangerous

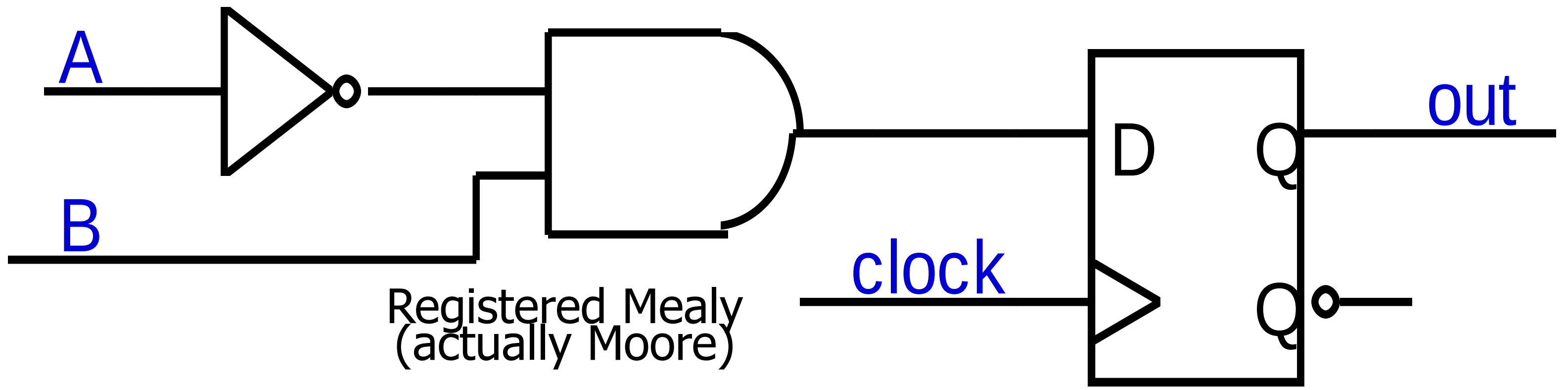
Synchronous (registered) Mealy machine

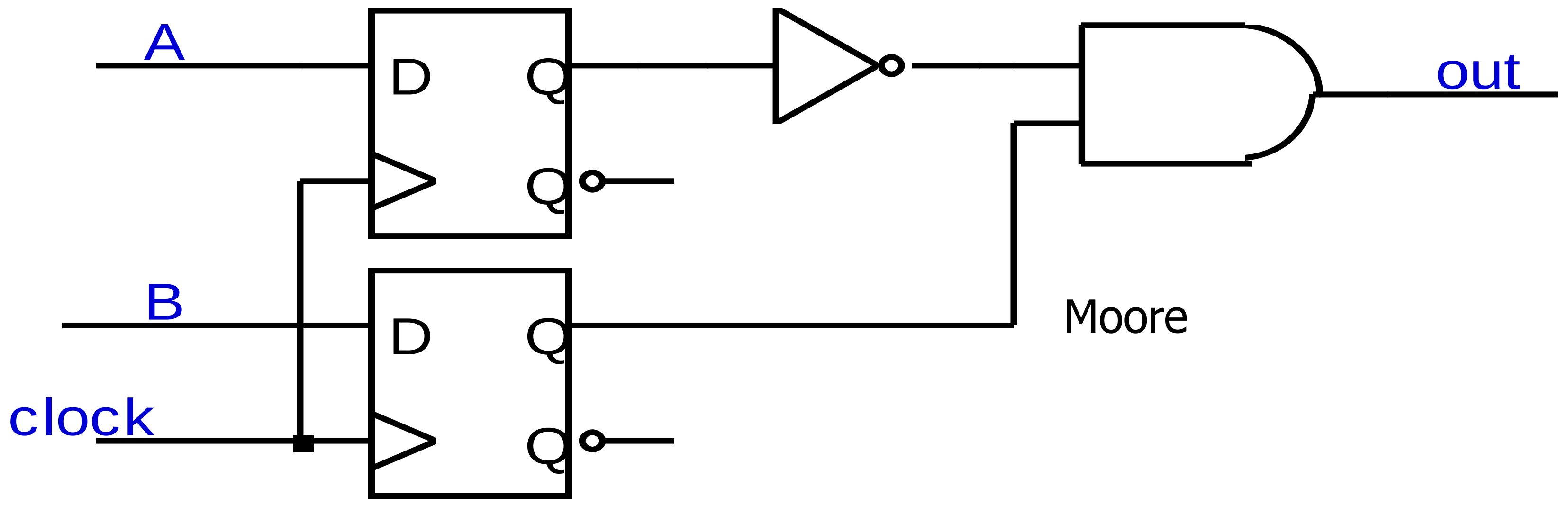
- Registered state and registered outputs
  - No glitches on outputs
  - No race conditions between communicating machines



Example “=01”: Moore or Mealy?

- Recognize AB = 01
  - Mealy or Moore?





Applications of Flip-Flops

Counters

- A clocked sequential circuit that goes through a predetermined sequence of states.
- A commonly used counter is an *n*-bit binary counter. This has *n* FFs and 2*n* states which are passed through in the order 0, 1, 2, ….2*n*-1, 0, 1, .

Uses include:

- Counting
- Producing delays of a particular duration.
- Sequencers for control logic in a processor.
- Divide by *m* counter (a divider), as used in a digital watch.

Memories, e.g.,

- Shift register
- Parallel loading shift register : can be used for parallel to serial conversion in serial data communication
- Serial in, parallel out shift register: can be used for serial to parallel conversion in a serial data communication system.

End
of
Unit-4