ECE 697CE Foundations of Computer Engineering

Lesson 4 & 5

Finite State Machines

Rationale

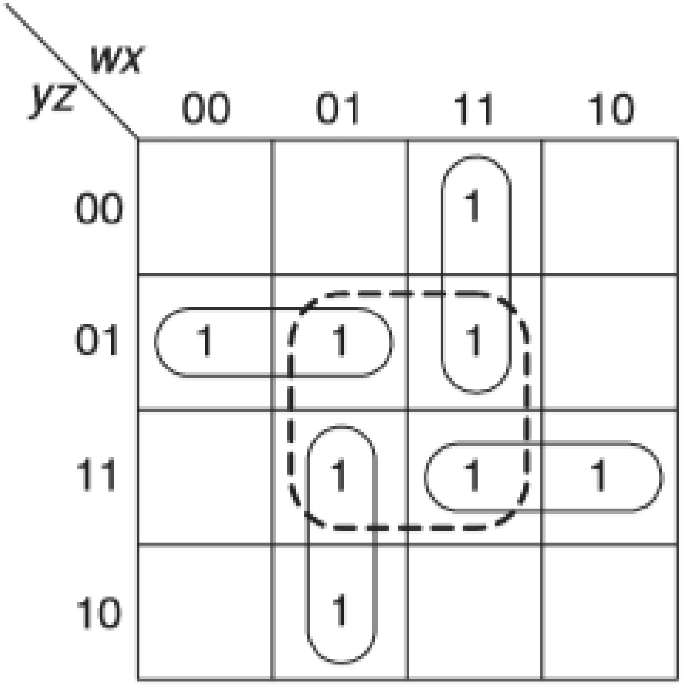
- Sequential circuits have states that store the effect of previously applied inputs
- Number of states for practical implementations is finite or limited
- Moore Machine: output associated to current state only
  - state -> output
- Mealy Machine: output associated to both state and specific input
  - (state, input) -> output

Objectives

- Apply the principles of finite state machine to create sequential circuits.
- Sequential circuits have outputs that are based on its inputs and current state.

Poll: Prior Knowledge

The provided figure is an example of what?



1) Monotonic Function
2) Minimization
3) Fault Detection
4) Don’t Care Combination
5) Karnaugh Map

Prior Knowledge

- State-less circuit = current inputs alone determine the output

- Let’s review from Lesson 2: Boolean Switching Functions
  - Minimization
  - Karnaugh Map
  - Unate & Monotonic Functions
  - Symmetric Functions

Orchestrated Discussion (Hand Raise): Critical Thinking Exercise

- **Discuss two questions about the previous lesson.**

Digital Circuits Classification and Basic building Blocks

- **<u>Combinational circuits:</u> Output = Function of (external inputs)**
- **<u>Sequential circuit:</u> Output = Function of (external inputs + stored information)**
- **Synchronous Vs. Asynchronous Sequential Circuits**
- **Flip-Flops the building blocks of Sequential Circuits**
  - Latches Vs Flip-Flops
  - Flip-Flop types:
    - JK-Type
    - D-Type
    - T-Type

Sequential Circuits and Finite State Machines

- **<u>Sequential circuit:</u>** Output = Function of (external inputs + stored information)
- **<u>Finite-state machine (FSM):</u>** abstract model to describe the synchronous sequential machine and its spatial counterpart, the iterative network
- **<u>Example:</u>** Serial binary adder: block diagram, addition process, state table and state diagram

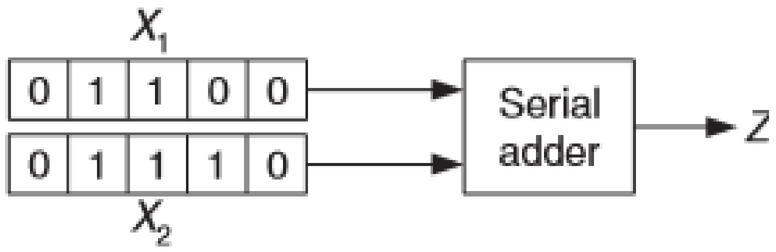


Fig: Block diagram

$$
\begin{array}{rccccl}
 & t_5 & t_4 & t_3 & t_2 & t_1 \\
 & 0 & 1 & 1 & 0 & 0 = X_1 \\
+ & 0 & 1 & 1 & 1 & 0 = X_2 \\
\hline
 & 1 & 1 & 0 & 1 & 0 = Z
\end{array}
$$

Fig: Addition process

Internal States of Machines

Serial adder:

- Output at time t_i is a function of
  - Input values at the time x_1 and x_2
  - Carry generated at t_{i-1}
- Carry in turn depends on input at t_{i-1} and carry at t_{i-2} and so on
- State A means no carry was produced, state B means a carry was produced in the previous add.
- Use <u>internal states</u> to preserve information regarding input values from the time it is set to operating
- State diagram and state tables describe the behavior

	NS, z			
PS	$x_1x_2 = 00$	01	11	10
A	$A, 0$	$A, 1$	$B, 0$	$A, 1$
B	$A, 1$	$B, 0$	$B, 1$	$B, 0$

Fig: State table

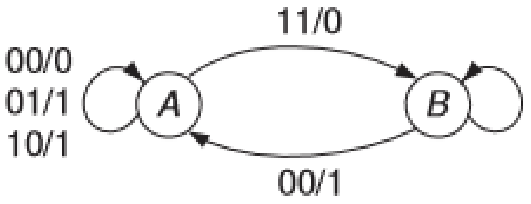


Fig: State diagram

Finite State Machines

<u>Finite State Machines:</u> Machines whose past histories can affect their future behavior in only finite number of ways.

- Serial adder:
  - Response to signal at time t is only a function of input at t and carry at $t - 1$
  - Group input histories to 2 classes:
    - Those resulting in a 1 carry at t
    - Those resulting in a 0 carry at t
- Every FSM contains a finite number of latches used to encode the current state.
- The current state stores information about the previous inputs.

Finite-State Model

Deterministic machines:

- **Next state $S(t+1)$ determined uniquely by**
  - **present state $S(t)$**
  - **present input $x(t)$**

- **State transition function -> δ**

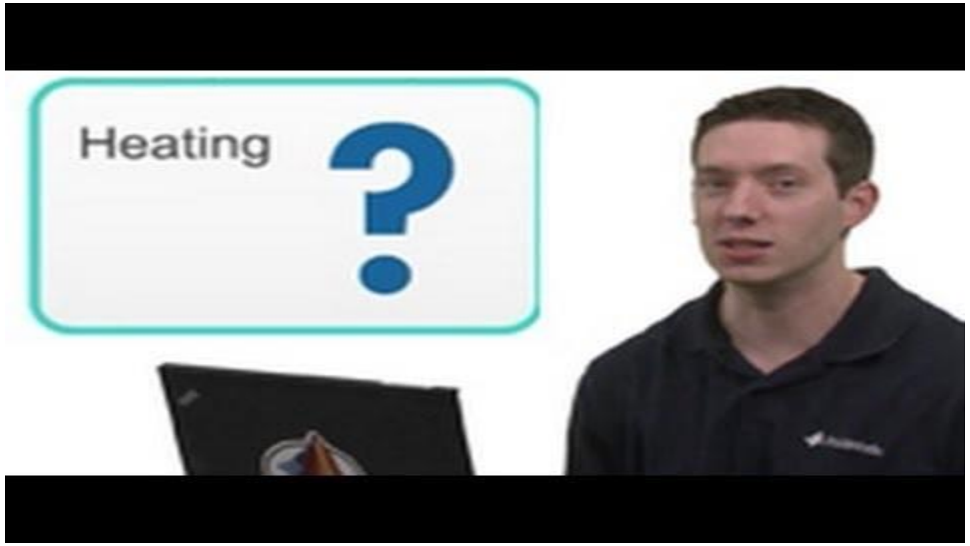
$$S(t+1) \;=\; \delta\ \{S(t), x(t)\}$$

- **Output function -> λ**

<u>Mealy machine</u> : $z(t) \;=\; \lambda\,\{S(t), x(t)\}$

<u>Moore machine</u> : $z(t) \;=\; \lambda\{S(t)\}$

Video: Finite-State Model



A Typical Mealy Model

- Mealy machine: Output depends on the present state and the present inputs
- A Mealy model:
  - 2 D flip flops
  - Input $x(t)$
  - Output $z(t)$

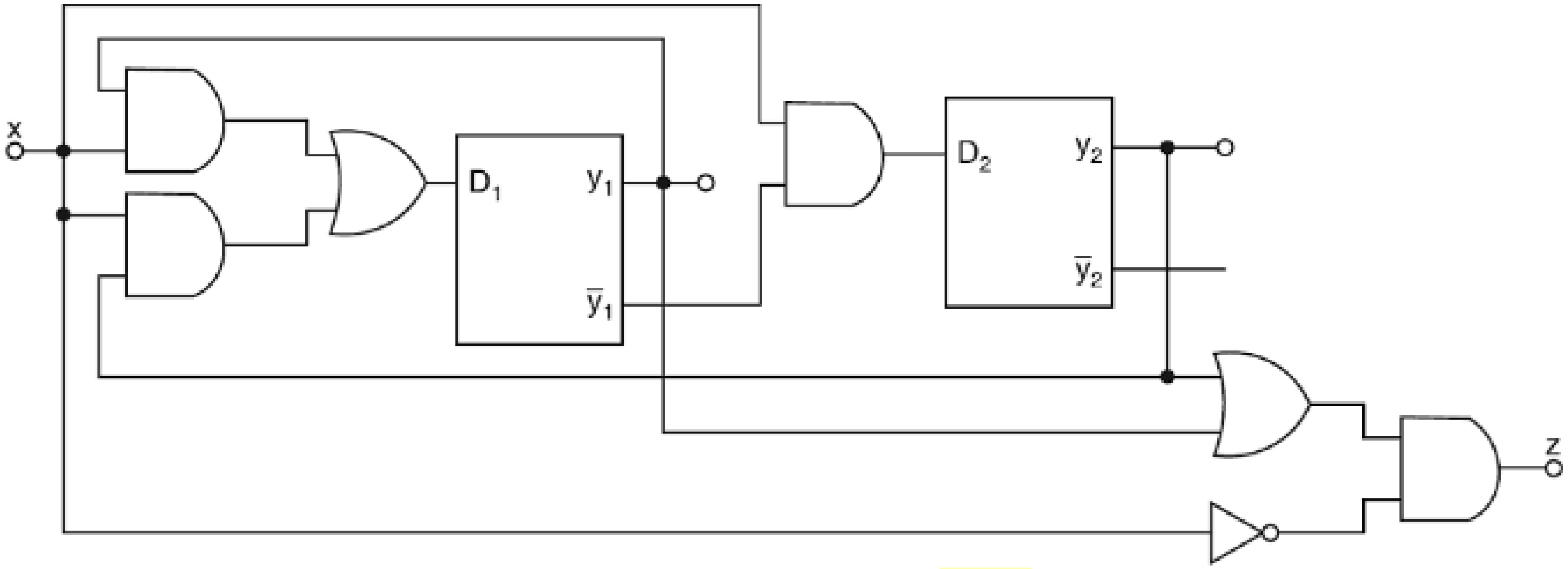


Fig: Logic diagram of a typical Mealy circuit

A Typical Mealy Model

- D input of flip-flop determines the next state value
- Typical state equations
  - $y_1(t+1) = y_1(t)x(t) + y_2(t)x(t)$
  - $y_2(t+1) = \overline{y_1}(t)x(t)$
- Typical output equation
  - $z(t) = \{y_1(t) + y_2(t)\}\,\overline{x}(t)$

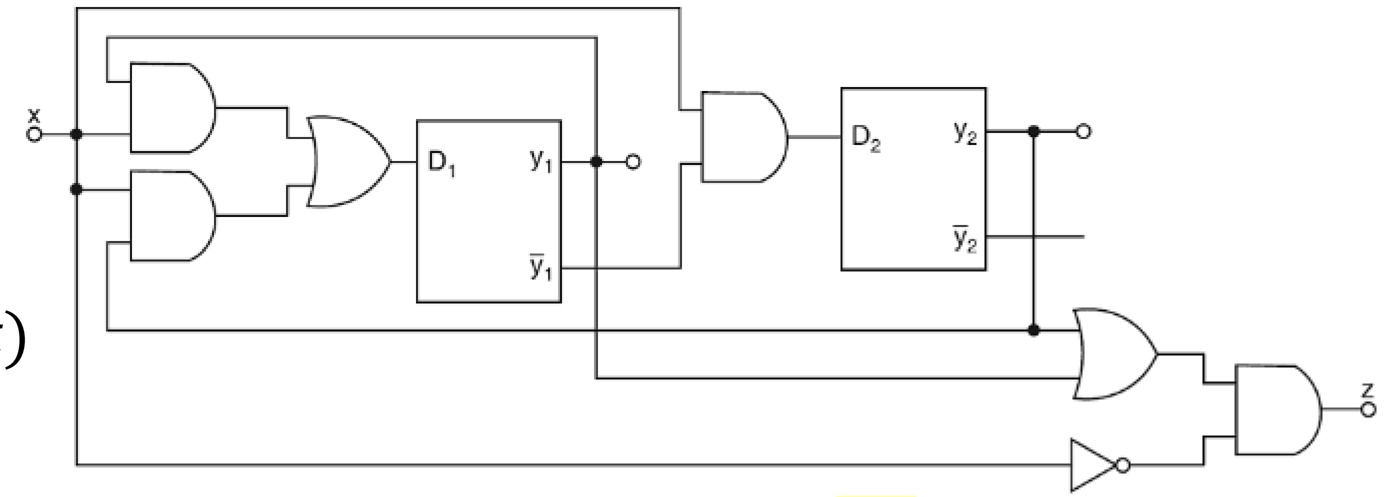


Fig: Logic diagram of a typical Mealy circuit

A Typical Mealy Model

PS		NS				O/P	
		x = 0		x = 1		x = 0	x = 1
y_1	y_2	Y_1	Y_2	Y_1	Y_2	z	z
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Fig: State table

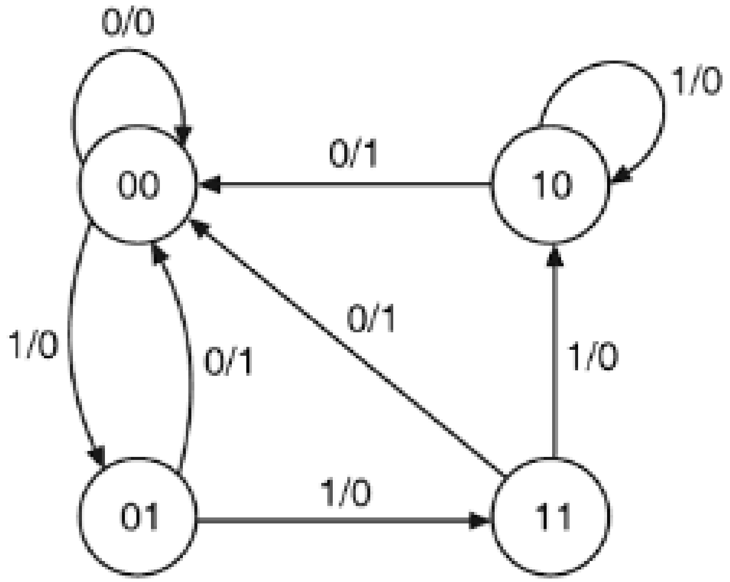


Fig: State diagram

Mealy Circuit Model

- **Block schematic of a Mealy Circuit Model**

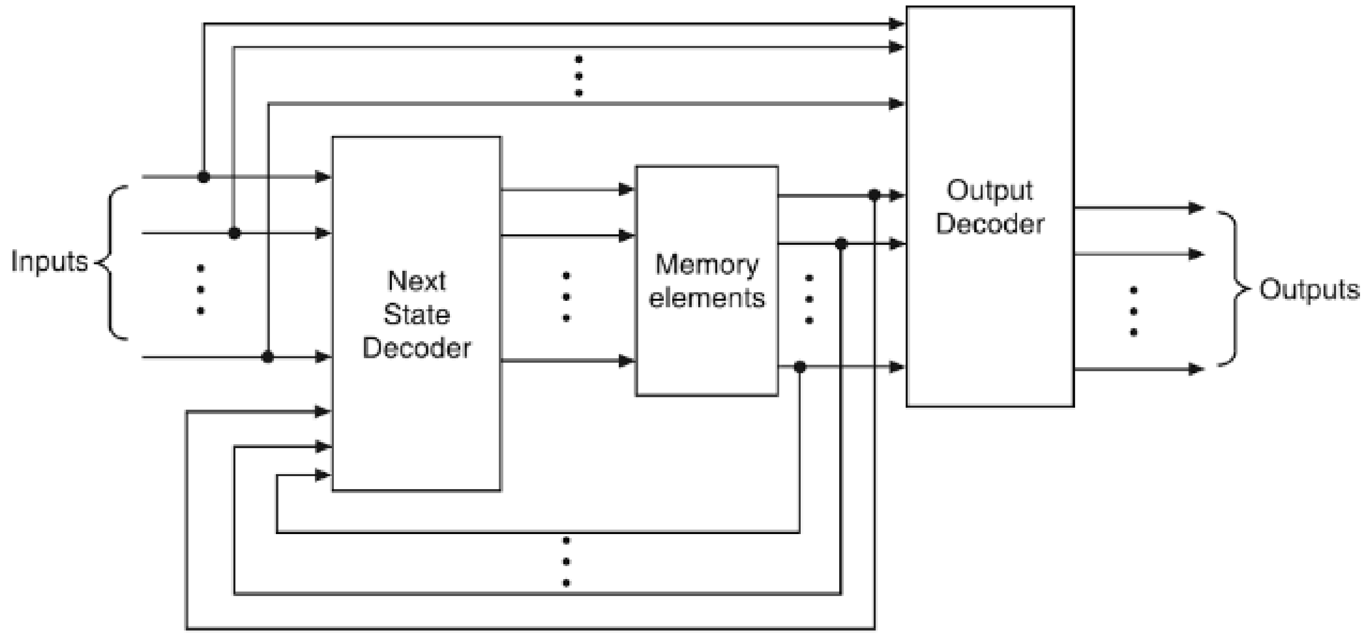


Fig: Mealy circuit model

A Typical Moore Model

- Moore Machine: Output depends only on the present state of the system
- A typical Moore model:
  - 2 T flip-flops
  - Input $x(t)$
  - Output $z(t)$

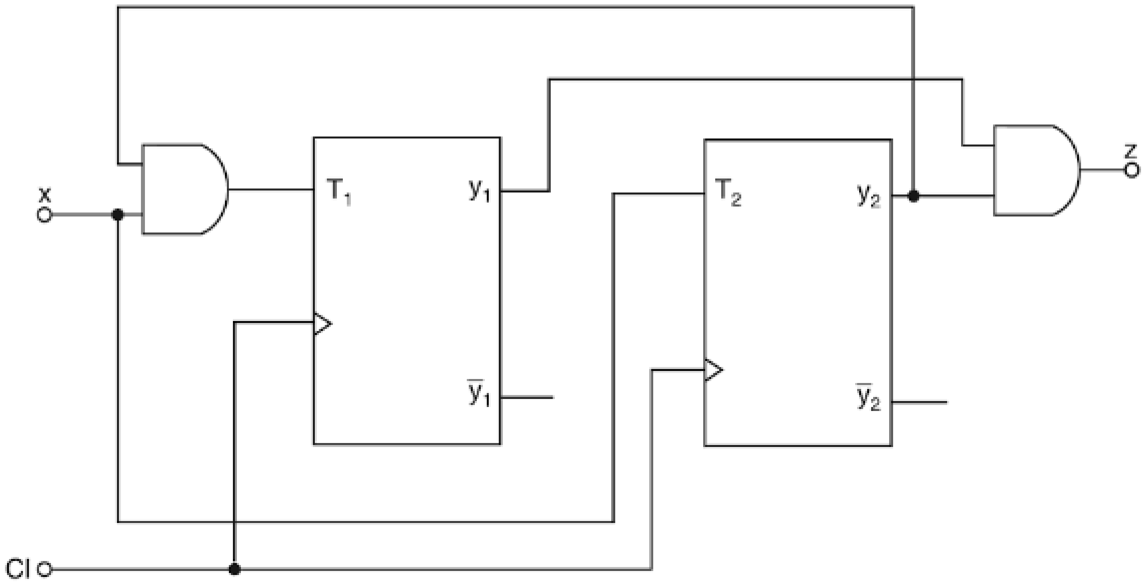


Fig: Logic diagram of a Moore model

A Typical Moore Model

- Input x(t) is used only to determine the inputs of flip-flops
- Its state equations
  - $y_1(t+1) = y_2(t)x(t) \oplus y_1(t)$
  - $y_2(t+1) = x(t) \oplus y_2(t)$
- Its output equations
  - $z(t) = y_1(t)y_2(t)$

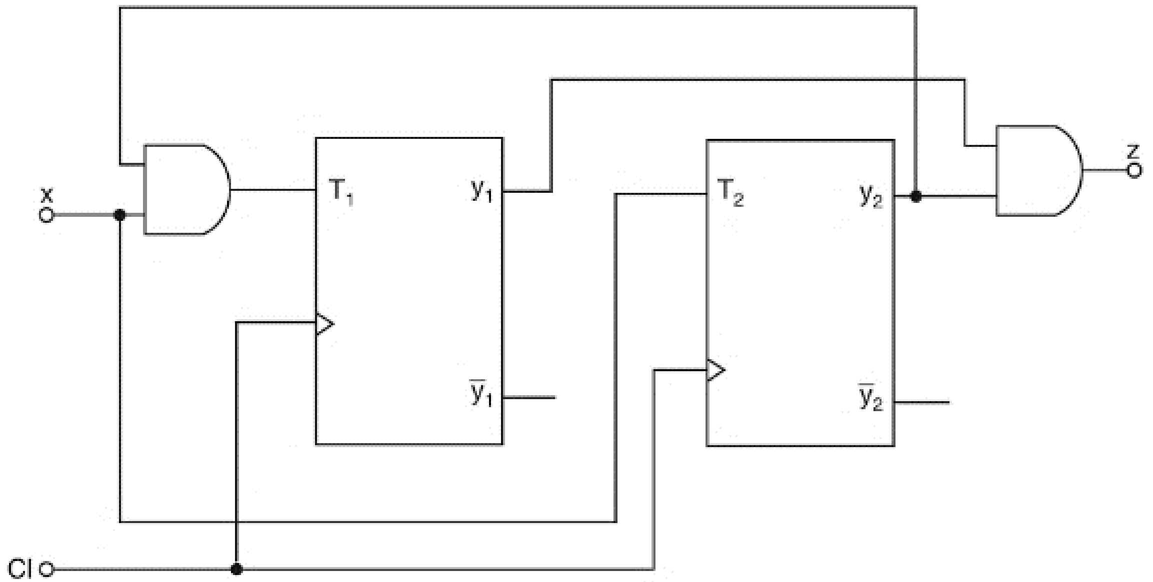


Fig: Logic diagram of a Moore model

A Typical Moore Model

PS		NS				O/P
		x = 0		x = 1		
y_1	y_2	Y_1	Y_2	Y_1	Y_2	z
0	0	0	0	0	1	0
0	1	0	1	1	0	0
1	0	1	0	1	1	0
1	1	1	1	0	0	1

Fig: State table

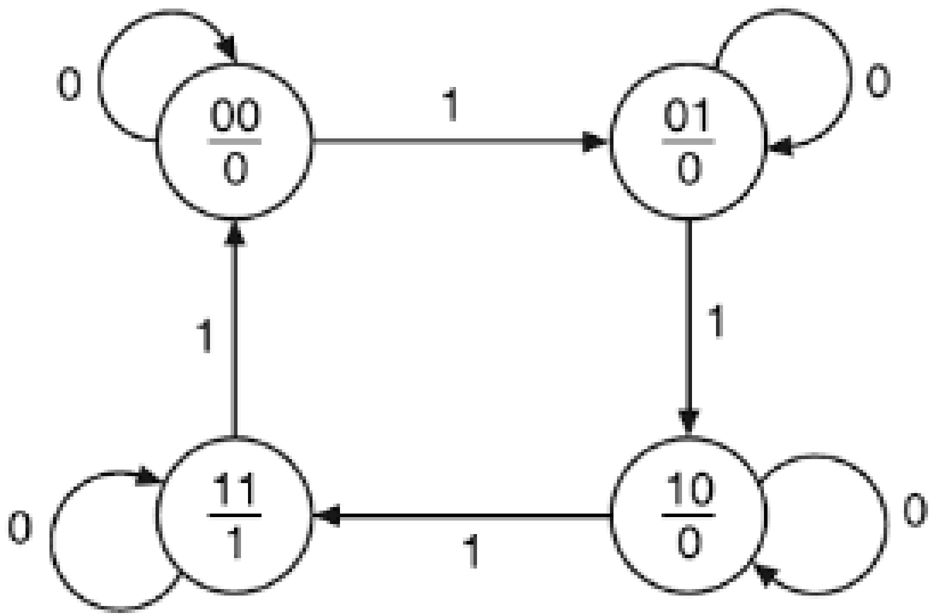


Fig: State diagram

Moore Circuit Model

- **Block schematic of a Moore Model**

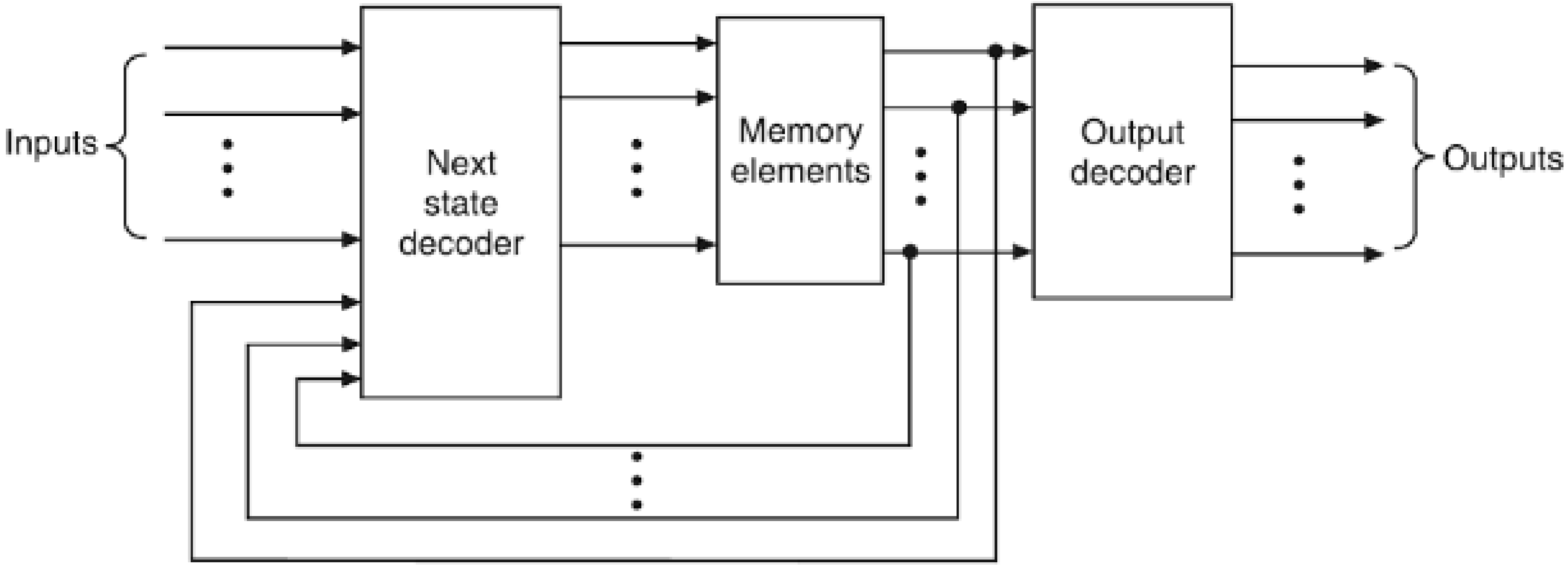


Fig: Moore circuit model

Group Discussion and Report Back (Pen): Mealy and Moore Models

- **Using the provided circuit models, identify at least 3 differences between the Mealy and Moore models.**

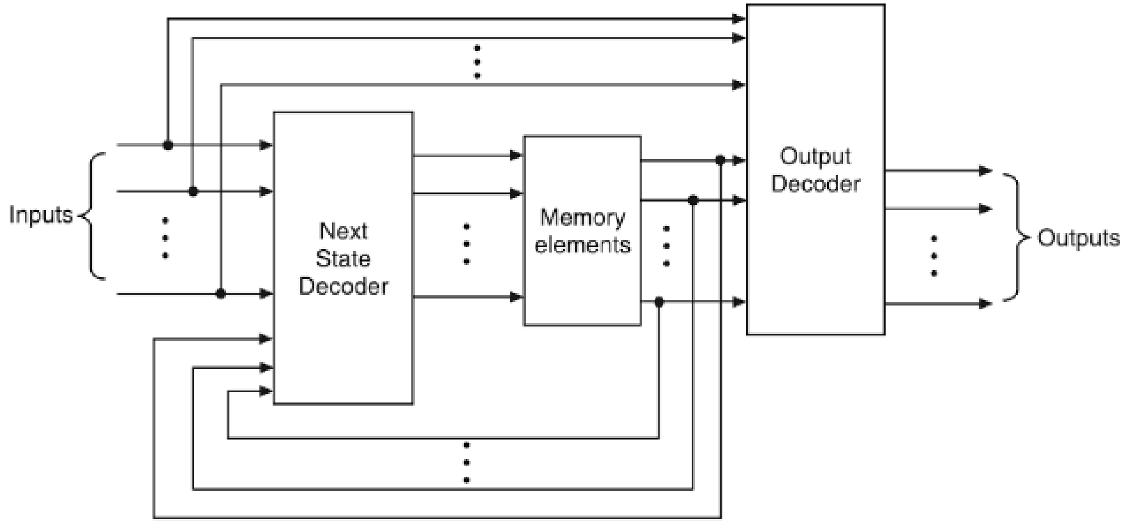


Fig 1: Mealy circuit model

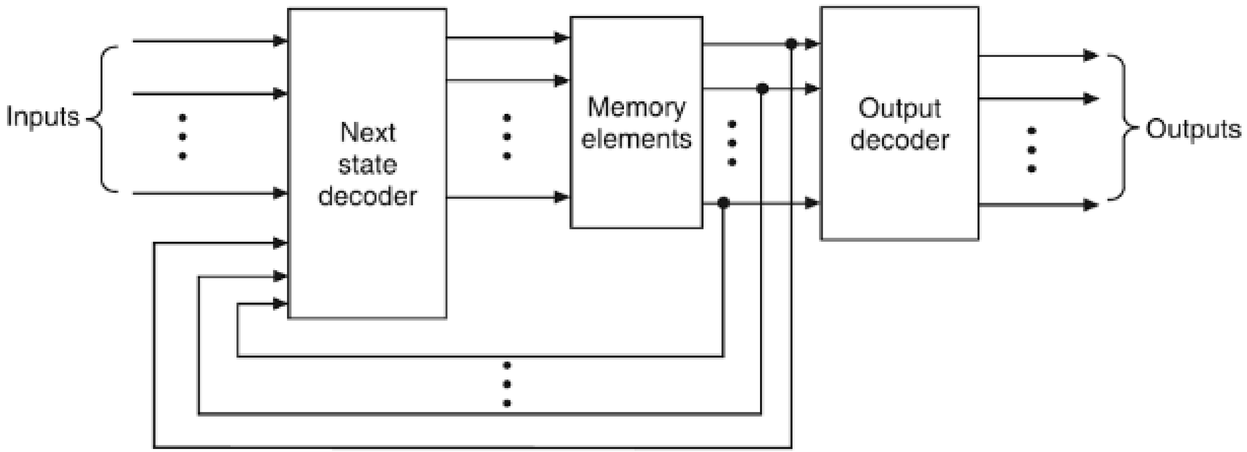


Fig 2: Moore circuit model

Sequence Detector for Moore Model

- Sequence detector for 01 or 10.
- Output 1 when the sequence is seen
- Moore Model: Output is a function of only state

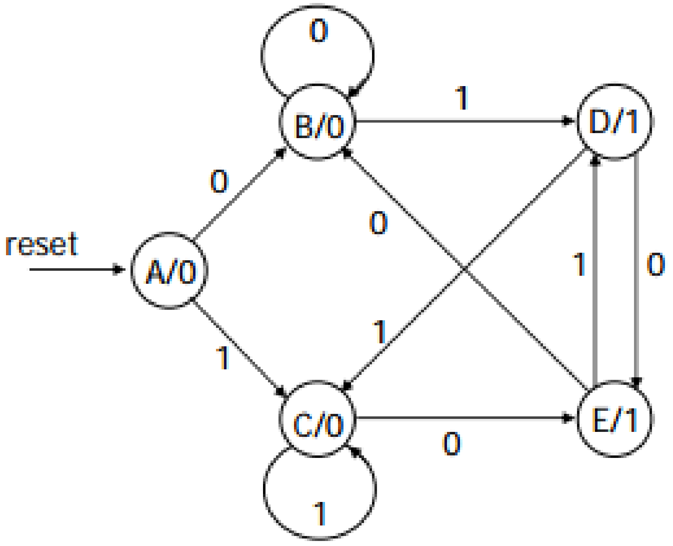


Fig: State diagram

reset	input	current state	next state	output
1	–	–	A	
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

Fig: State table

Sequence Detector for Mealy Model

- Sequence detector for 01 or 10
- Output 1 when the sequence is seen
- Mealy Model: Output is a function of state and inputs

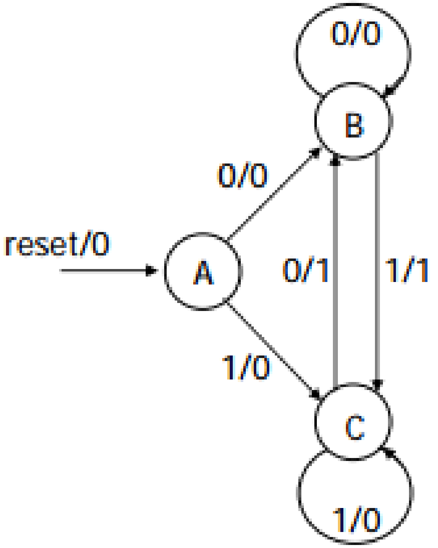


Fig: State diagram

reset	input	current state	next state	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

Fig: State table

Homing Sequence

Homing sequence is X=101, since the final state can be uniquely determined by observing outputs.

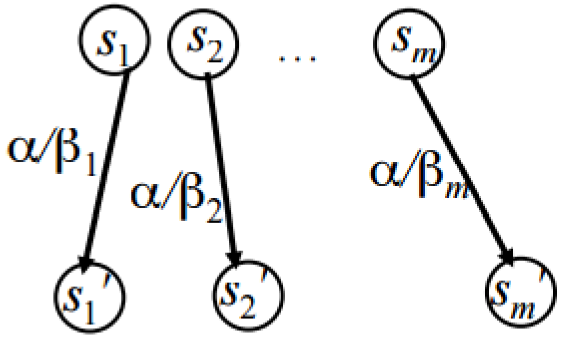
Starting at A, we end-up in state C with outputs 100
Starting at B, we end-up in state A with outputs 100
Starting at C, we end-up in state B with outputs 101
Starting at D, we end-up in state C with outputs 011

If we observe outputs, at the end of 3rd input, we know the current state, even though the initial state was unknown

Present State	Input X=0	Input X=1
A	C,1	D,0
B	D,0	B,1
C	B,0	C,1
D	C,0	A,0

Distinguishing Sequence

- A sequence X is a distinguishing sequence for machine M if the output sequence produced in response to X is distinct for each initial state.
- Distinguishing = separating for nondeterministic machines
- A distinguishing (input) sequence α allows to determine the initial state of the machine under experiment
- After applying α at any state s and observing an output response β the initial state s becomes known



$$out(s_i, \alpha) \cap out(s_j, \alpha) = \varnothing$$

Fig: Distinguishing sequence α

Synchronizing Sequence

- A synchronizing sequence is a sequence of inputs that will force the machine to a specified final state independent of the initial state or output sequence.
- A synchronizing sequence α takes the machine under experiment to a given state after applying α
- After applying α at any state the final state is s’

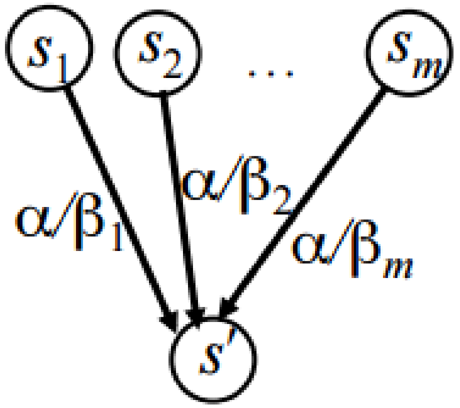


Fig: Synchronizing sequence α

Summary of this Lesson

- Introduction to Finite State Machines (FSMs)
- Moore FSM Machines
- Mealy FSM Machines
- Sequence Detector Machines
- Introduction to Special Sequences

Post-work for Lesson 5

Homework

- **After the Live Lecture, you will complete and submit a homework assignment. Go to the online classroom to view and submit the assignment.**

To Prepare for the Next Lesson

- Read the Required Readings for Lesson 6.
- Complete the Pre-work for Lesson 6.
- Start working on the Project 1.

Go to the online classroom for details.