Chapter 10
Finite State Machine Synthesis
SKEE2263 Digital Systems

Mun'im Zabidi {munim@utm.my}
Ismahani Ismail {ismahani@fke.utm.my}
Izam Kamisian {e-izam@utm.my}

Faculty of Electrical Engineering, Universiti Teknologi Malaysia

February 10, 2018

Table of Contents

FSM Design Process

Step	Description
Conceptualize	Understand the statement of the specification.
Capture the FSM	Translate the problem into state diagram. Determine the number of **states** required, **inputs**, **outputs** and state **transitions**.
Encode the states	Assign a unique binary number to each state.
Create the state table	Create a truth table for the combinational logic for **next state** generation and **output** decoding.
Extract the equations	Express the logic circuits as Boolean equations.
Implement the FSM	Enter and verify the design.

Counters

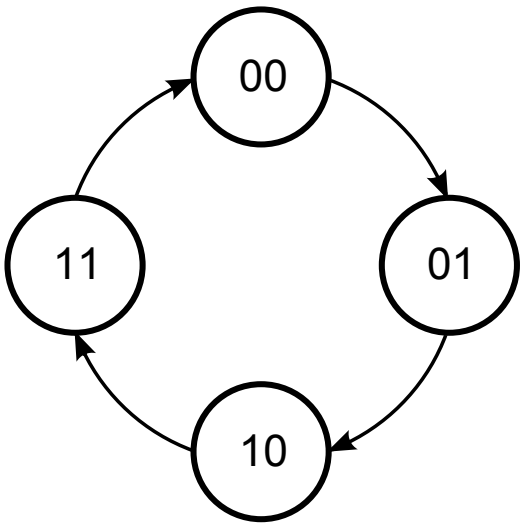
- Definition of counters:
  - Starts from a particular state
  - Goes thru fixed sequence of states
  - Returns to initial state
  - Repeats indefinitely
- Most counters (but not all) are Medvedev state machines
  - Medvedev machine have not output logic.
  - Register outputs = system outputs.
- Single-mode counters:
  - The only input is the clock signal
- Multi-mode counters:
  - Has input to select mode of operation

2-bit Binary Counter

Binary Counters

n state variables $\Rightarrow 2^n$ sequential states

Binary 2-bit counter : straight binary sequence 0→1→2→3→0

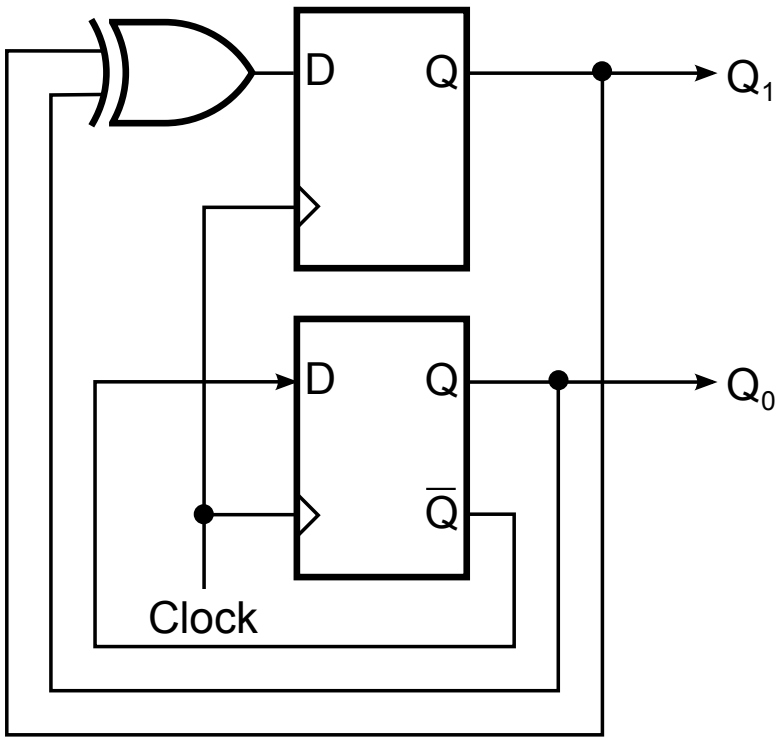


Present State		**Next State**	
Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

2-bit Binary Counter

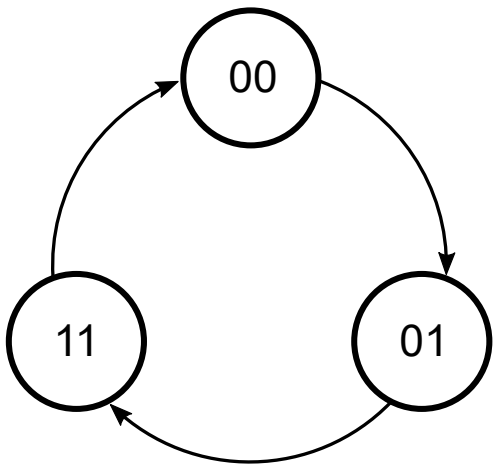
Present State		Next State	
Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

$$Q_1^+ = Q_1 \oplus Q_0$$
$$Q_0^+ = Q_0'$$



Counter with Arbitrary Sequence

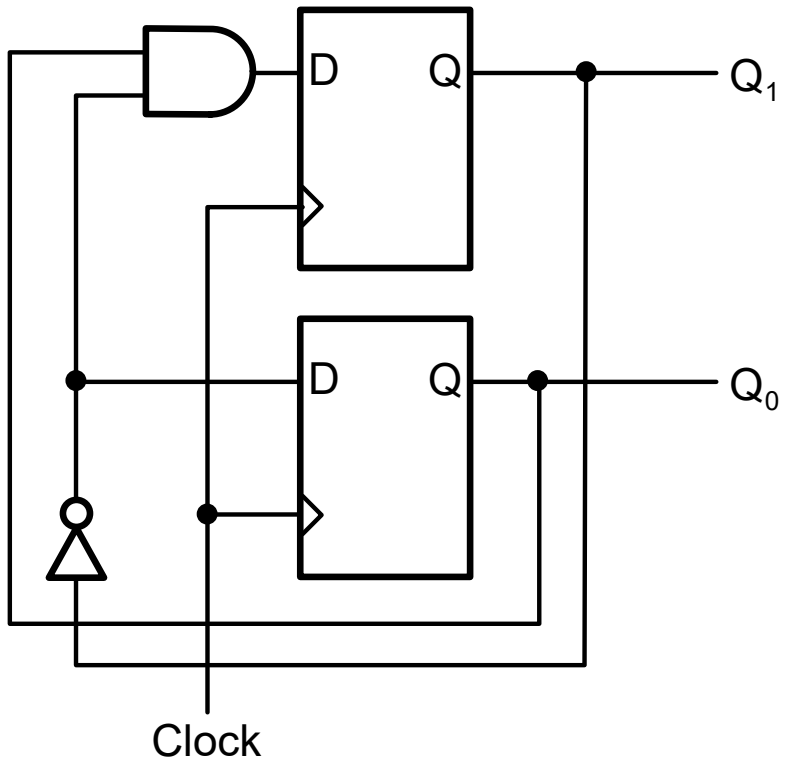
- Counter next states can be non-sequential
- Number of states can be < 2^n
- All valid:
  - 0→3→1→2→0,
  - 0→1→3→2→0,
  - 0→1→3→0



Counter with Arbitrary Sequence

Present State		Next State	
Q_1	Q_0	Q_1^+	Q_0^+
0	0	0	1
0	1	1	1
1	0	×	×
1	1	0	0

$$Q_1^+ = Q_1' Q_0$$
$$Q_0^+ = Q_1'$$



Definition

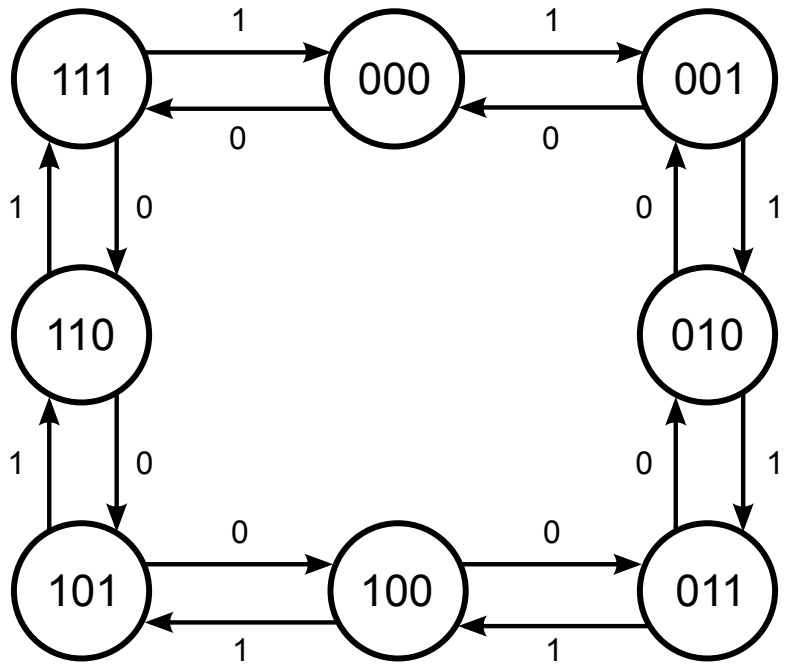
Multi-Mode Counter

A counter whose counting sequence depends not only on the clock but also on some other control signal(s).

Examples:

- Counts up or down
- Counts in binary or Gray sequence
- Increments either +1 or -2
- Counts up or holds count value

3-bit Up/Down Counter



3-bit Up/Down Counter

Present State			Input	Next State		
Q_2	Q_1	Q_0	U	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	1	1	1
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	1
0	1	1	0	0	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	0	1
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0
1	1	1	1	0	0	0

3-bit Up/Down Counter

$$Q_2^+ = Q_2'Q_1'Q_0'U' + Q_2'Q_1Q_0U + Q_2Q_1Q_0' + Q_2Q_0U' + Q_2Q_1'U$$
$$Q_1^+ = Q_1'Q_0'U' + Q_1'Q_0U + Q_1Q_0'U + Q_1Q_0U'$$
$$Q_0^+ = Q_0'$$

Several ways to realize the design:

- Logic gates: complex wiring
- Mux
- Decoder
- ROM
- Behavioral Verilog: easiest

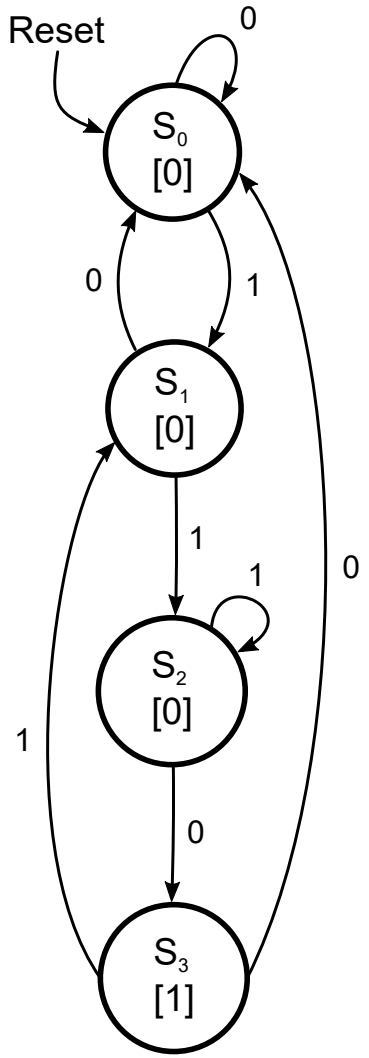
How to Design a Sequence Detector

- Start with the expected sequence first.
- Assert the output at the last state.
- Add exit arrows to cover all possible transitions.
  - Each state must have two exit arrows.

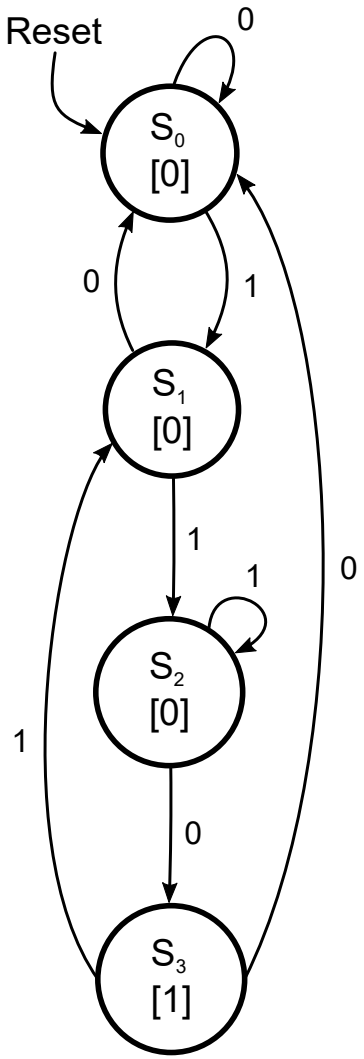
110 Sequence Detector: Moore

```
IN  :  1 1 0 0 0 1 1 0 1 1 1 1 0
OUT :  0 0 1 0 0 0 0 1 0 0 0 0 1
```

State	Meaning
S_0	No matching bits yet
S_1	1 (first bit) found
S_2	1 (second bit) found
S_3	0 (third bit) found, assert output



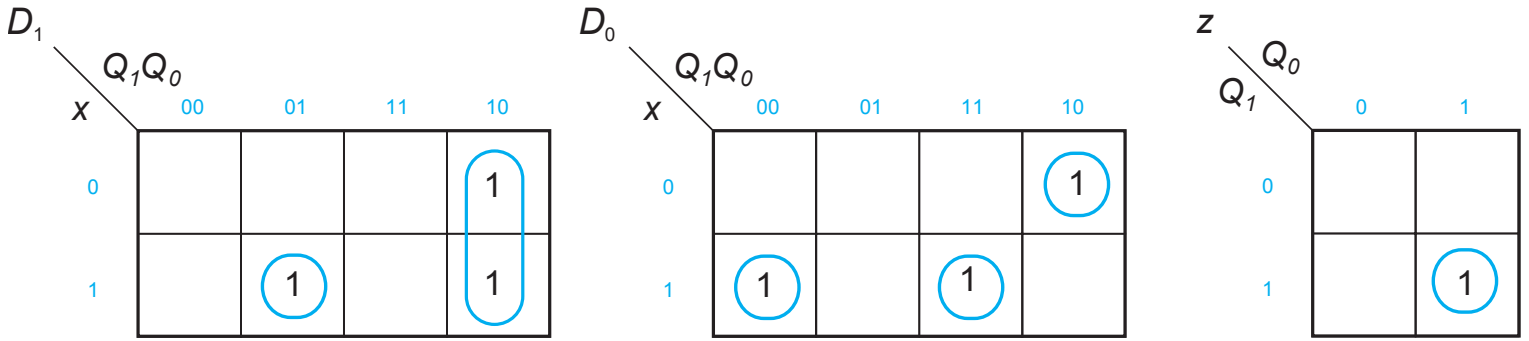
110 Sequence Detector: Moore



State Encoding	
Symbolic	Encoded
S0	00
S1	01
S2	10
S3	11

Present State		Input	Next State		Output
Q_1	Q_0	x	Q_1^+	Q_0^+	z
0	0	0	0	0	0
0	0	1	0	1	
0	1	0	0	0	0
0	1	1	1	0	
1	0	0	1	1	0
1	0	1	1	0	
1	1	0	0	0	1
1	1	1	0	1	

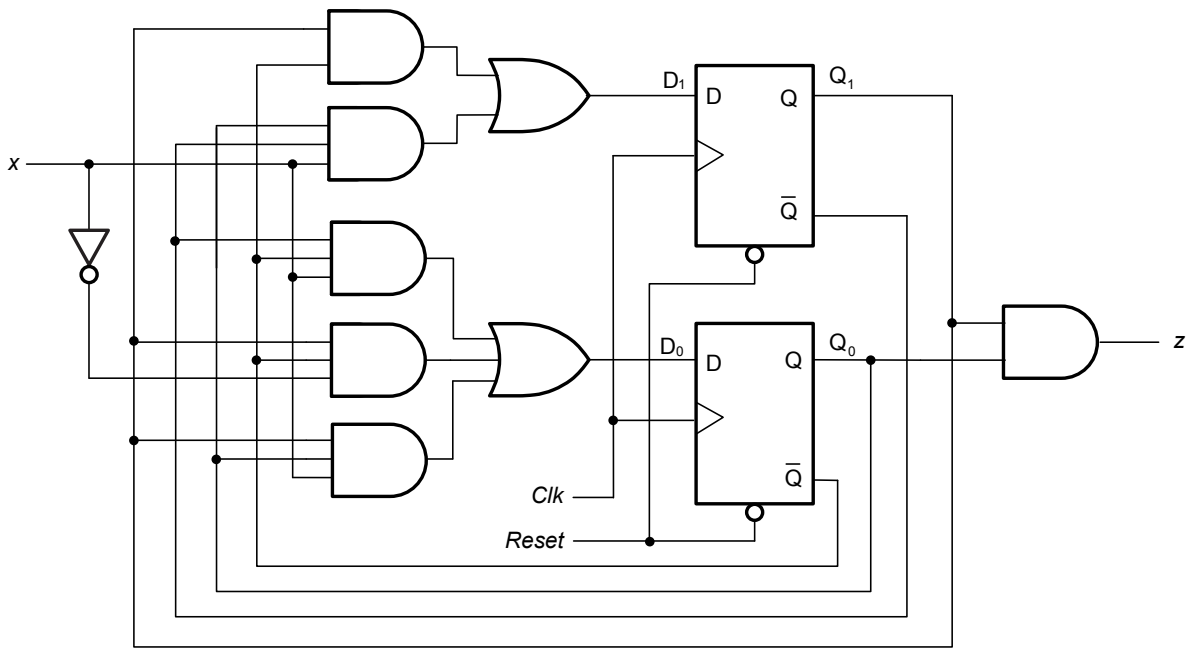
110 Sequence Detector: Moore



$$D_1 = Q_1 Q_0' + Q_1' Q_0 x$$
$$D_0 = Q_1' Q_0' x + Q_1 Q_0' x' + Q_1 Q_0 x$$
$$z = Q_1 Q_0$$

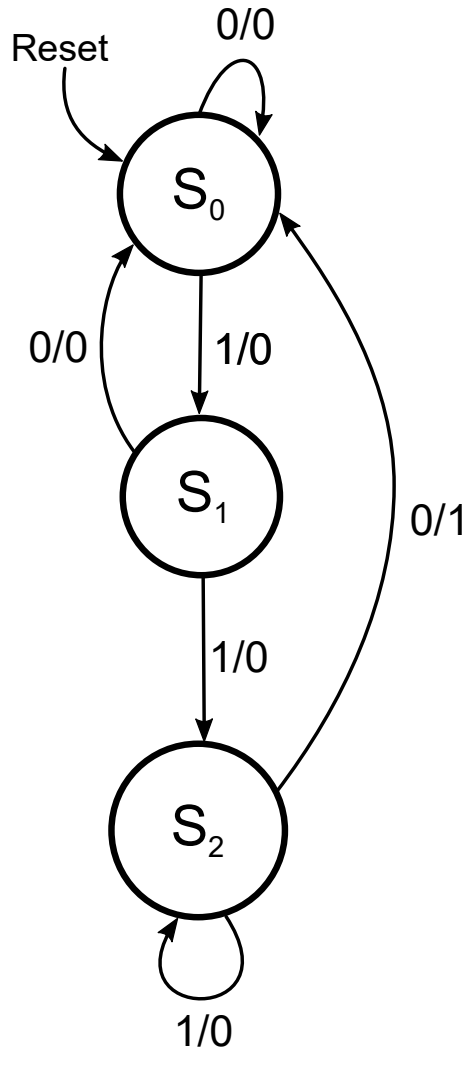
110 Sequence Detector: Moore

$$D_1 = Q_1 Q_0' + Q_1' Q_0 x$$
$$D_0 = Q_1' Q_0' x + Q_1 Q_0' x' + Q_1 Q_0 x$$
$$z = Q_1 Q_0$$

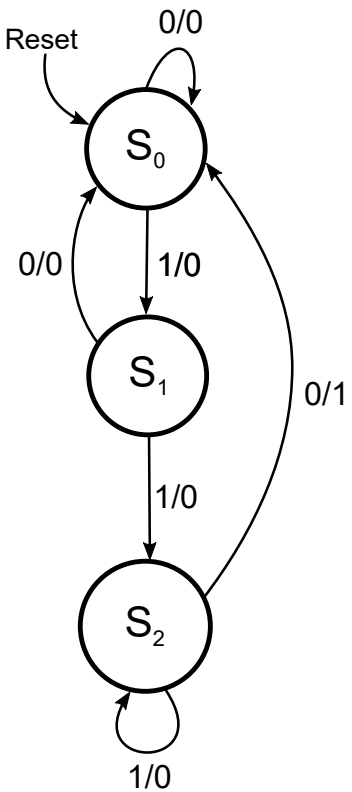


110 Sequence Detector: Mealy

```
IN   :  1 1 0 0 0 1 1 0 1 1 1 1 0
OUT  :  0 0 1 0 0 0 0 1 0 0 0 0 1
```



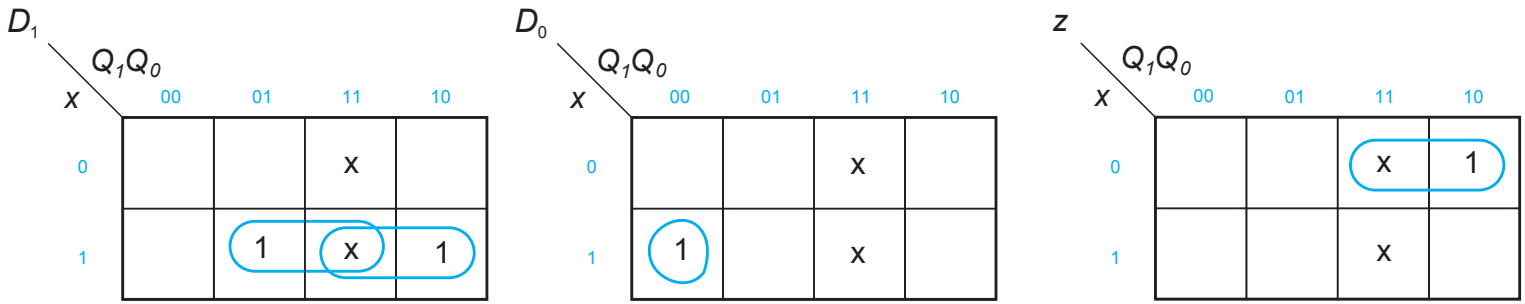
110 Sequence Detector: Mealy



State Encoding	
Symbolic	Encoded
S0	00
S1	01
S2	10
S3	11

Present State		Input	Next State		Output
Q_1	Q_0	x	Q_1^+	Q_0^+	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	X	X	X
1	1	1	X	X	X

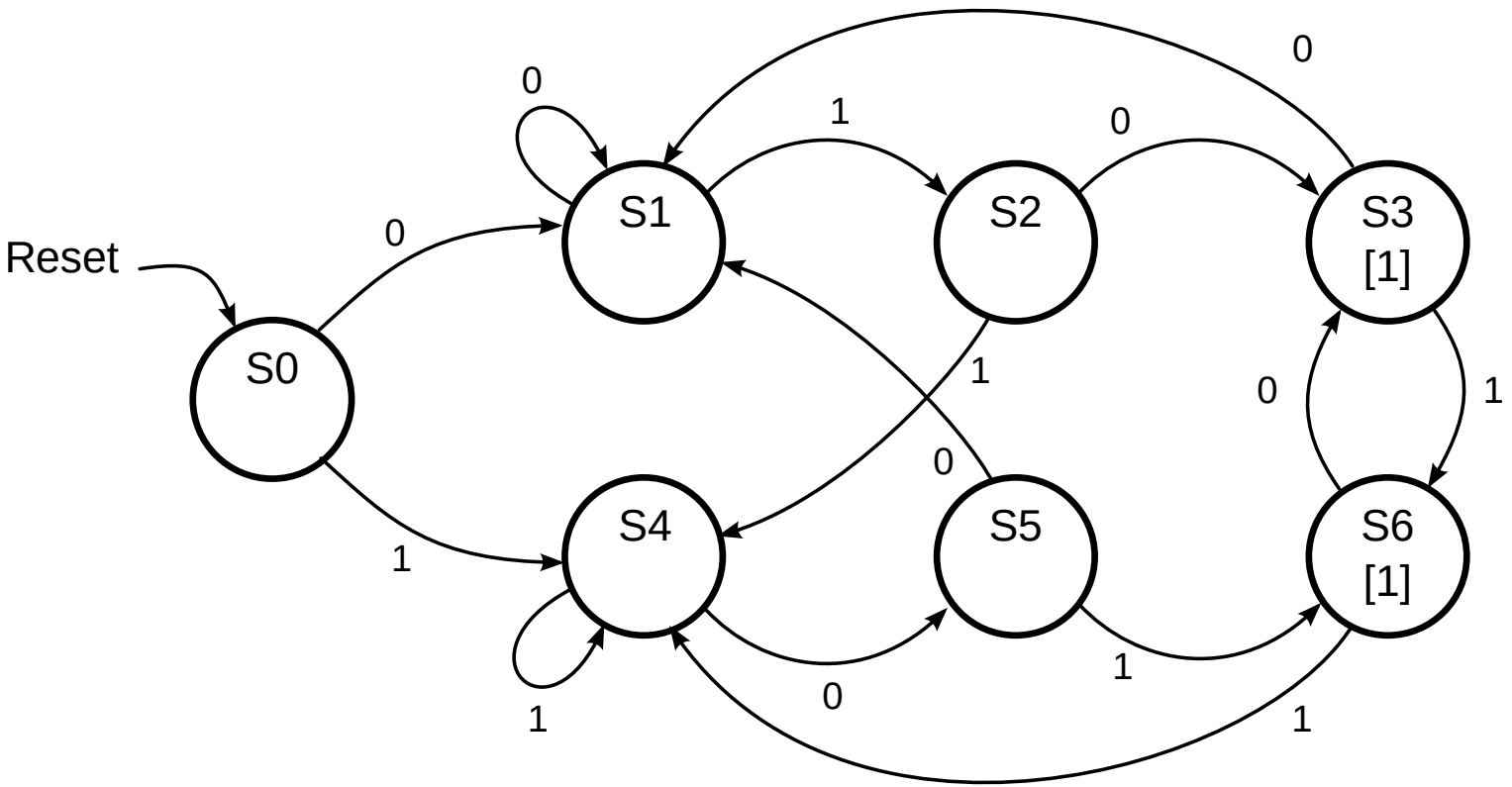
110 Sequence Detector: Mealy



$$D_1 = Q_1 x + Q_0 x$$
$$D_0 = Q_1' Q_0' x$$
$$z = Q_1 x'$$

010 or 101 Detector

x	:	0	1	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	1	1	0
z	:	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0



Alternating Input Detector

010 or 101 detector

Present State			Input	Next State			Output
Q_2	Q_1	Q_0	x	Q_2^+	Q_1^+	Q_0^+	z
0	0	0	0 1	0 1	0 0	1 0	0
0	0	1	0 1	0 0	0 1	1 0	0
0	1	0	0 1	0 1	1 0	1 0	0
0	1	1	0 1	0 1	0 1	1 0	1
1	0	0	0 1	1 1	0 0	1 0	0
1	0	1	0 1	0 1	0 1	1 0	0
1	1	0	0 1	0 1	1 0	1 0	1
1	1	1	X	X	X	X	X