UNIVERSITY OF VICTORIA

CENG 241

DIGITAL DESIGN I

Lab 6
Finite state machines: Mealy and Moore circuits

Instructor:
Dr. Amirali BANIASADI

Teaching Assistant:
Grace HUI

Yves SÉNÉCHAL V00213837
Tyler STEPHEN V00812021
A01 - B03

July 13, 2015

University of Victoria

1 Introduction

This lab will explore the difference between Mealy and Moore state machines. Both types of state machine will be constructed to detect a specific input sequence. The entire design process will be demonstrated for both machines.

2 Discussion

Suppose we want to detect the sequence “1101” within an input stream. Overlapping sequences will be detect (i.e. the last 1 of the first sequence can be the first 1 of the next sequence). The input and output will follow this pattern:

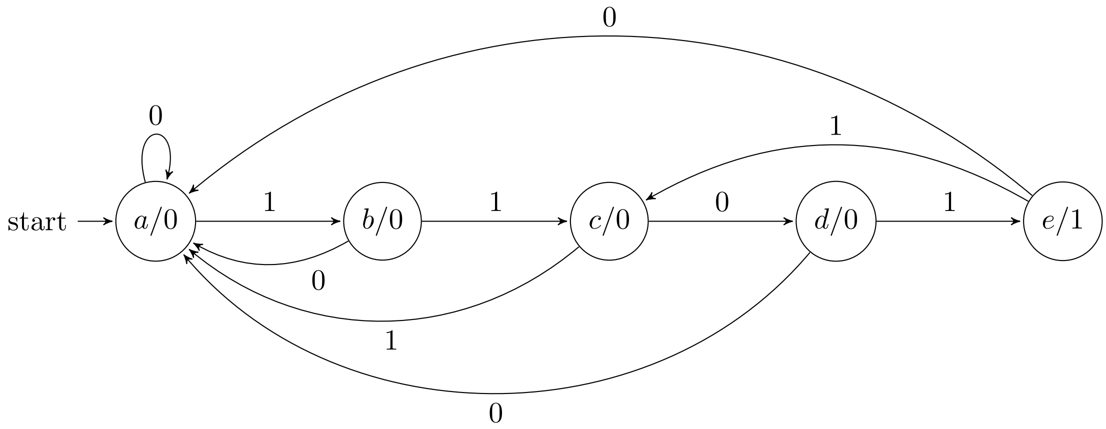
Input	1001 1011 0100 1101
Output	0000 0010 0100 0001

It is possible to design a sequential circuit to accomplish this task. First, the system’s response to input will be codified into a state diagram. Next, this state diagram will be translated into state tables. From there, Karnaugh maps and Boolean expressions will inform the physical design of the circuit.

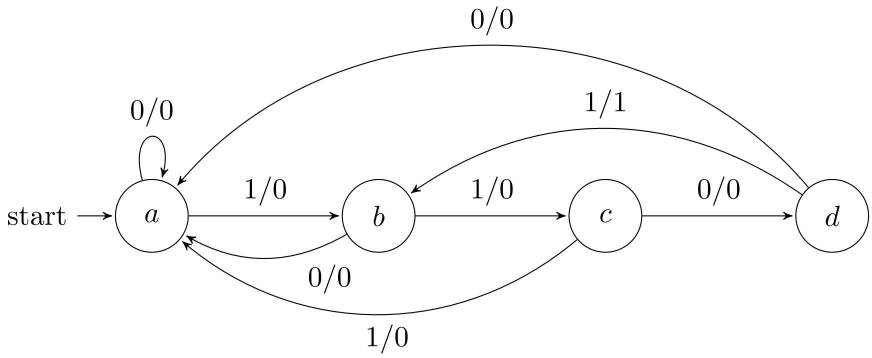
2.1 State diagrams

There are two types of state machine to choose from: the Moore machine and the Mealy machine. The Moore machine can determine its output based entirely on the current state. The Mealy machine needs to know the input in addition to the current state to determine its output. Since Mealy machines link the output to the input for a state, they will always require the same or fewer states than the equivalent Moore machine.

Figure 1 shows both state diagrams for the sequence detector. Overlapping sequence detection is controlled by having the final 1 input link to the “interior” of the state diagram rather than the initial state.



(a) Moore machine



(b) Mealy machine

Figure 1: State machines to detect sequence “1101” with overlap

2.2 Transition tables

To create the state transition tables, first enumerate the states. The number of flip-flops required will be equal to $\lceil \log_2(\text{number of states}) \rceil$. The next state is determined by following the state diagram. For the output, the Mealy transition table requires the input in addition to the current state. D flip-flops were used to construct the circuits in this lab. Consequently, the results of the next state table can be used directly as inputs for the flip-flops. The next state table will require an accompanying excitation table for T or JK flip-flops.

State	S_2	S_1	S_0
a	0	0	0
b	0	0	1
c	0	1	0
d	0	1	1
e	1	0	0
-	1	0	1
-	1	1	0
-	1	1	1

(a) State enumeration

S_2	S_1	S_0	X	S_2^+	S_1^+	S_0^+
0	0	0	0	0	0	0
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	1	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	-	-	-
1	0	1	1	-	-	-
1	1	0	0	-	-	-
1	1	0	1	-	-	-
1	1	1	0	-	-	-
1	1	1	1	-	-	-

(b) Next state

S_2	S_1	S_0	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	-
1	1	0	-
1	1	1	-

(c) Output

Figure 2: Transition tables for the Moore machine

State	S_0	S_1
a	0	0
b	0	1
c	1	0
d	1	1

(a) State enumeration

S_1	S_0	X	S_1^+	S_0^+
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

(b) Next state

S_1	S_0	X	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(c) Output

Figure 3: Transition tables for the Mealy machine

2.3 Karnaugh maps

Karnaugh maps are generated from the transition tables. The Moore and Mealy methods are identical from this point in the design going forward.

$S_2S_1 \backslash S_0X$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

(a) S_2^+

$S_2S_1 \backslash S_0X$	00	01	11	10
00	0	0	1	0
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

(b) S_1^+

$S_2S_1 \backslash S_0X$	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

(c) S_0^+

$S_2 \backslash S_1S_0$	00	01	11	10
0	0	0	0	0
1	1	X	X	X

(d) Z

Figure 4: Karnaugh maps for the Moore machine

The optimal boolean functions for the Moore machine are

$$
\begin{aligned}
S_2^+ &= S_1S_0X \\
S_1^+ &= S_1S_0'X' + S_1'S_0X + S_2X \\
S_0^+ &= S_1S_0'X' + S_1'S_0'X \\
Z &= S_2
\end{aligned}
$$

S_1 \ S_0X	00	01	11	10
0	0	0	1	0
1	1	0	0	0

(a) S_1^+

S_1 \ S_0X	00	01	11	10
0	0	1	0	0
1	1	0	1	0

(b) S_0^+

S_1 \ S_0X	00	01	11	10
0	0	0	0	0
1	0	0	1	0

(c) Z

Figure 5: Karnaugh maps for the Mealy machine

The optimal boolean functions for the Mealy machine are

$$
\begin{aligned}
S_1^+ &= S_1 S_0' X' + S_1' S_0 X \\
S_0^+ &= S_1 S_0' X' + S_1' S_0' X + S_1 S_0 X \\
Z &= S_1 S_0 X
\end{aligned}
$$

3 Simulations

We chose to construct the Mealy machine because it required one fewer flip-flop. The Boolean expression was implemented in the schematic in Figure 6.

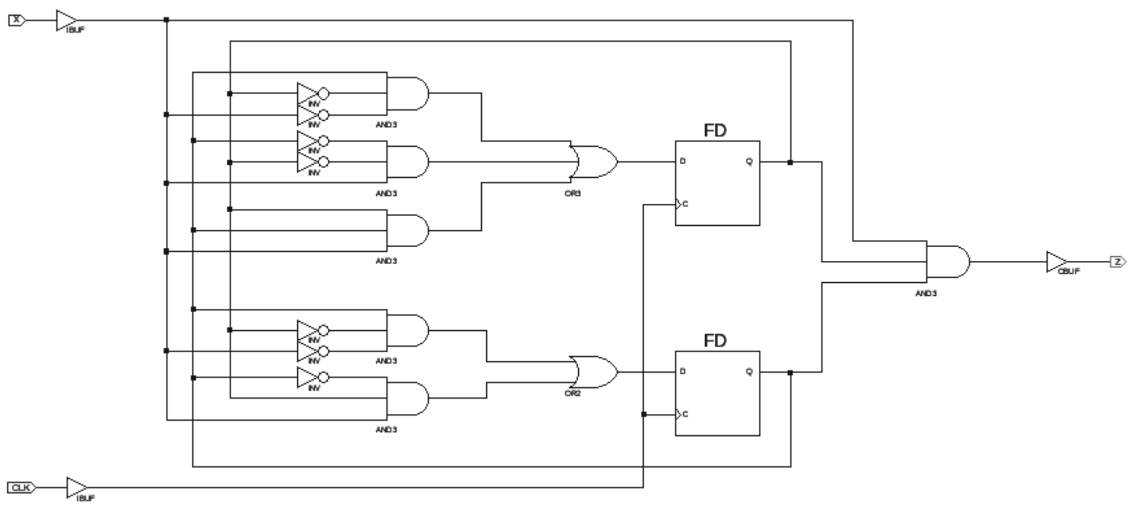


Figure 6: Mealy machine schematic

The input stream from the beginning of Section 2 was used as a simulation input source. Figure 7 shows that the schematic gives the correct output in a Xilinx simulation.

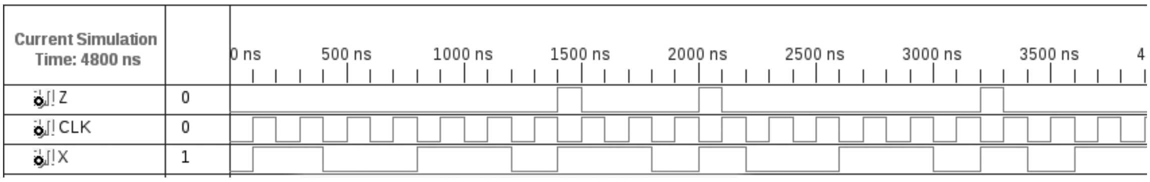


Figure 7: The Mealy machine correctly identifies the sequence "1101" from the input stream

The schematics were used to construct a Mealy machine using physical ICs. It was also successful in identifying the sequence from the input stream.

4 Conclusion

Sequential circuits can be created from Moore and Mealy machines. We were successful in creating a sequence detecting machine using the Mealy method and confirmed the results with computer and physical simulations.