Lab 02 – Mealy to Moore Machine Conversion

Name: Ruby Jean B. Solomon
Course: Bachelor of Science in Computer Science (BSCS)
University: University of Southern Mindanao

1. Introduction

In this laboratory activity, I learned how to convert a **Mealy Machine** into its equivalent **Moore Machine**. A Mealy machine's output depends on both the current state and the input symbol, while a Moore machine's output depends only on the current state. The main goal of this lab is to show how both machines can produce the same outputs even if they work a bit differently.

2. Mealy Machine Representation

The Mealy machine used in this experiment has five states: A, B, C, D, and E. Each state changes based on the input (0 or 1) and produces a specific output symbol.

Transition Table

Present State	Input = 0	Input = 1
A	A / A	A / B
B	B / C	A / D
C	C / D	C / B
D	D / B	B / C
E	E / D	C / E

Mealy Machine Diagram

Below is the state diagram of the given Mealy machine:

Mealy Machine Diagram

3. Conversion Process (Mealy → Moore)

To convert the Mealy machine into a Moore machine, I followed these steps:

1. Identified all unique output symbols that appear on transitions going **into** each state.
2. Created new Moore states that combine the state and its output (for example, A_A means State A with output A).
3. Assigned the proper output to each Moore state.
4. Added a new start state (S0) that doesn't have an output but connects to the right Moore state after reading the first input.
5. Defined transitions between Moore states based on the Mealy machine's transitions.

This process basically adds more states, but it makes the machine's behavior easier to trace since outputs only change when the state changes.

4. Moore Machine Representation

After the conversion, the Moore machine ended up with the following states:

- A_A (State A with output A)
- A_B (State A with output B)
- B_C (State B with output C)
- C_D (State C with output D)
- C_B (State C with output B)
- D_B (State D with output B)
- E_D (State E with output D)
- C_E (State C with output E)
- S0 (Start state with no output)

Moore Machine Diagram

Here's what the equivalent Moore machine looks like:

Moore Machine Diagram

5. Python Implementation

I used Python to automate the conversion and to simulate both machines. The code below shows the Mealy machine definition and the logic that converts it into a Moore machine.

```
mealy = {
    'A': {'0':('A','A'), '1':('A','B')},
    'B': {'0':('B','C'), '1':('A','D')},
    'C': {'0':('C','D'), '1':('C','B')},
    'D': {'0':('D','B'), '1':('B','C')},
    'E': {'0':('E','D'), '1':('C','E')},
}

initial = 'A'
inputs = ["00110", "11001", "1010110", "101111"]

# Conversion and simulation logic implemented here
# (Full code executed in Python to generate outputs)
```

6. Simulation Results

After running the program, the following outputs were produced by the Moore machine for each input sequence:

Input Sequence	Output Sequence
00110	AABBA
11001	BBAAB
1010110	BABABBA
101111	BABBBB

The outputs show that the converted Moore machine behaves exactly like the original Mealy machine.

7. Conclusion

This lab helped me understand the relationship between Mealy and Moore machines. Although the Moore version usually has more states, it's easier to follow because the output is fixed for each state. The experiment also showed how both machines can produce the same outputs for the same inputs, which proves that they are functionally equivalent.

End of Lab 02 Report