ShanghaiTech University
School of Information Science and Technology
Student name: Yourong Cao
Student ID: 2022533062

CS110: Computer Architecture
Spring, 2024

Assignment 4: Digital circuit

<span style="color:red">**Attention:**</span> **Recommend using LaTeXto complete your work. You can use any tool, such as Logisim, Visio, Draw.io, PowerPoint, etc., to create diagrams. However, handwritten or hand-drawn content is not acceptable.**

1 Combinational logic

Analyze the circuit shown in Fig. 1 and answer the following questions:

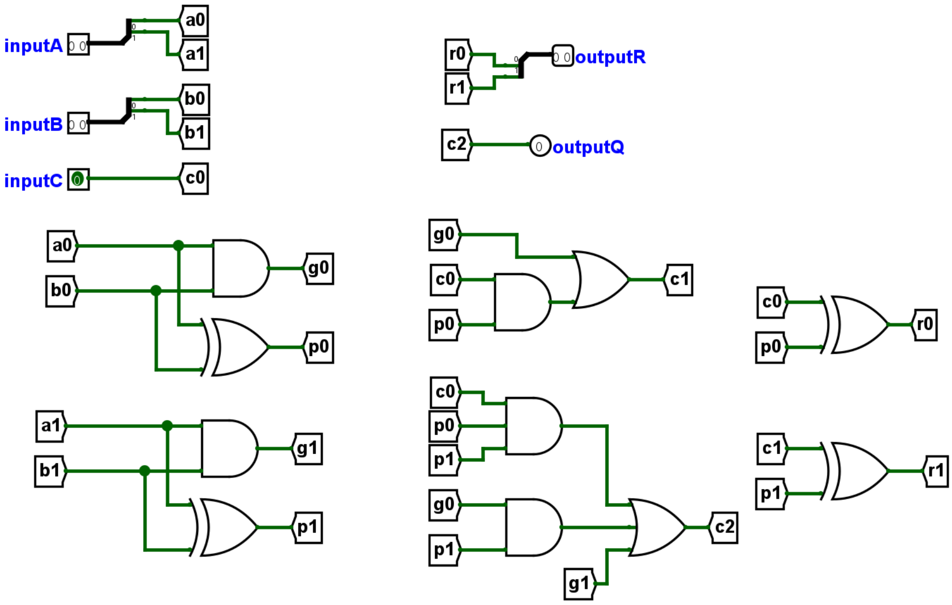


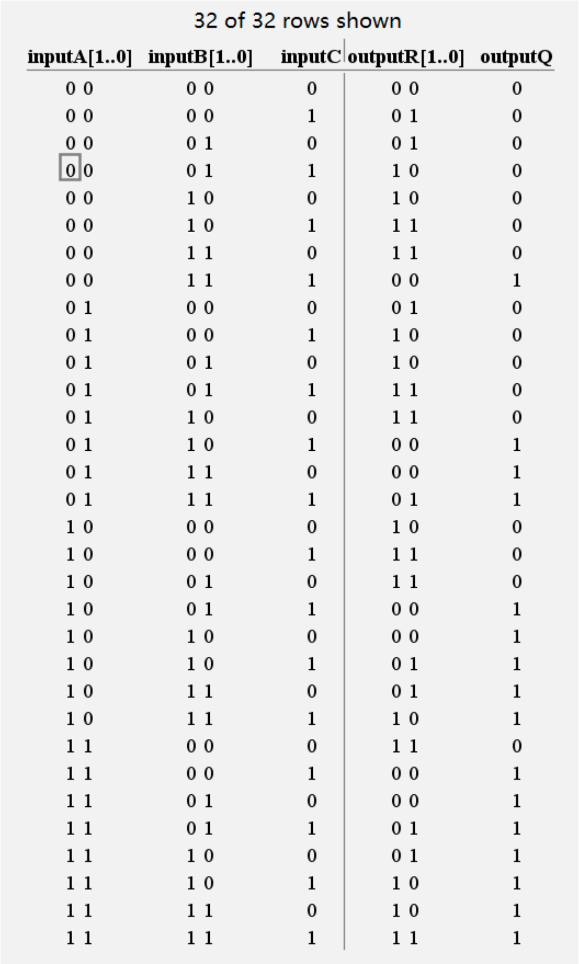
Figure 1: A 2-bit arithmetic circuit

(a) Draw the truth table of this circuit.[**10 pt**]

(b) Which kind of arithmetic operation (addition, subtraction, multiplication, division, shift, or comparison) is performed by this circuit? What are the advantages and disadvantages of the circuit in Fig. 1 compared to the corresponding arithmetic circuit mentioned in Digital circuits I?[**10 pt**]

(b) Assume that all 2-input logic gates have 1 ns delay, all 3-input logic gates have 2 ns delay, and other delays are not considered. Calculate the max delay of this circuit.[**10 pt**]

Answer to Question 1

(a) 

32 of 32 rows shown

inputA[1..0]	inputB[1..0]	inputC	outputR[1..0]	outputQ
0 0	0 0	0	0 0	0
0 0	0 0	1	0 1	0
0 0	0 1	0	0 1	0
0 0	0 1	1	1 0	0
0 0	1 0	0	1 0	0
0 0	1 0	1	1 1	0
0 0	1 1	0	1 1	0
0 0	1 1	1	0 0	1
0 1	0 0	0	0 1	0
0 1	0 0	1	1 0	0
0 1	0 1	0	1 0	0
0 1	0 1	1	1 1	0
0 1	1 0	0	1 1	0
0 1	1 0	1	0 0	1
0 1	1 1	0	0 0	1
0 1	1 1	1	0 1	1
1 0	0 0	0	1 0	0
1 0	0 0	1	1 1	0
1 0	0 1	0	1 1	0
1 0	0 1	1	0 0	1
1 0	1 0	0	0 0	1
1 0	1 0	1	0 1	1
1 0	1 1	0	0 1	1
1 0	1 1	1	1 0	1
1 1	0 0	0	1 1	0
1 1	0 0	1	0 0	1
1 1	0 1	0	0 0	1
1 1	0 1	1	0 1	1
1 1	1 0	0	0 1	1
1 1	1 0	1	1 0	1
1 1	1 1	0	1 0	1
1 1	1 1	1	1 1	1

(b) addition.

advantages:

1.such circuit can handle more inputs than arithmetic circuit mentioned in Digital circuits I (two bits than one bit) making it more flexible than a the arithmetic circuit mentioned in Digital circuits I in some cases.

2.For some applications, the use of a larger logic cell can reduce the number of components in a circuit because it enables more functions to be performed, which may lead to simpler design and wiring.(and Such circuit is larger than arithmetic circuit mentioned in Digital circuits I)

3.This circuit may integrate more logic functions such as and, or, and not to fulfill more needs.

4.This circuit computes two bits in parallel, so the total time it takes is reduced.

disadvantages:

1.Although such circuit is able to handle more inputs, the corresponding logic expressions may be more complex, which may make the design and debugging process more difficult.

2.Since the circuit may require more gates to delay signal propagation, such circuit may have higher latency than simpler logic cells.

3.Larger logic cells typically require more logic gates and resources, which can increase the cost and power consumption of the circuit.

(c) 5ns.(one of such is $a0- > 2 - xor- > 3 - and- > 3 - or- > c2$)

2 SDS

Draw a counter that counts from 0 to 5 using three D flip-flops (each flip-flops represents one output bit) and some 2-input logic gates (AND, OR, NOT). Please use the method taught in class to build a Moore FSM that implements the circular counter. Complete the state transition logic and output logic. **[35 pt]**

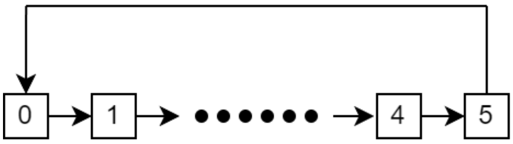
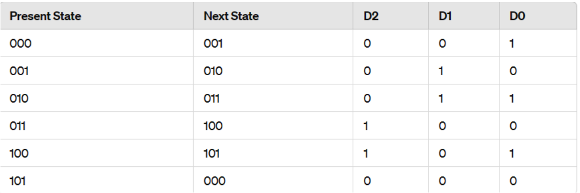


Figure 2: The counter cycles through the process of counting from 0 to 5.

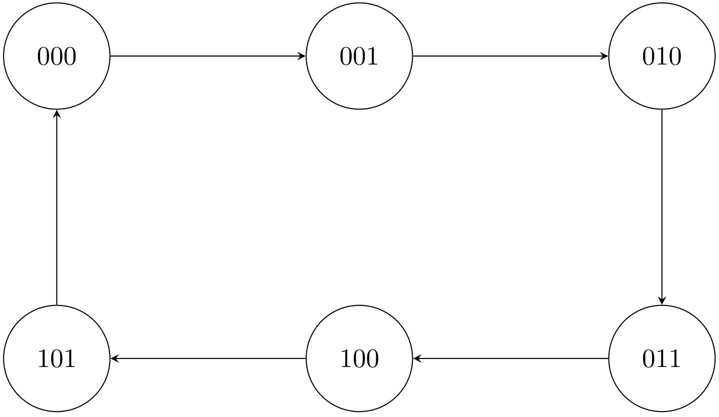
Answer to Question 2

we use 3-bit binary representation of the state number to represent the present state and next state. D2 represents the msb of the state bits and D0 represents the lsb of the state bits.(i.e. output)
The truth table and current-state to next-state table is:

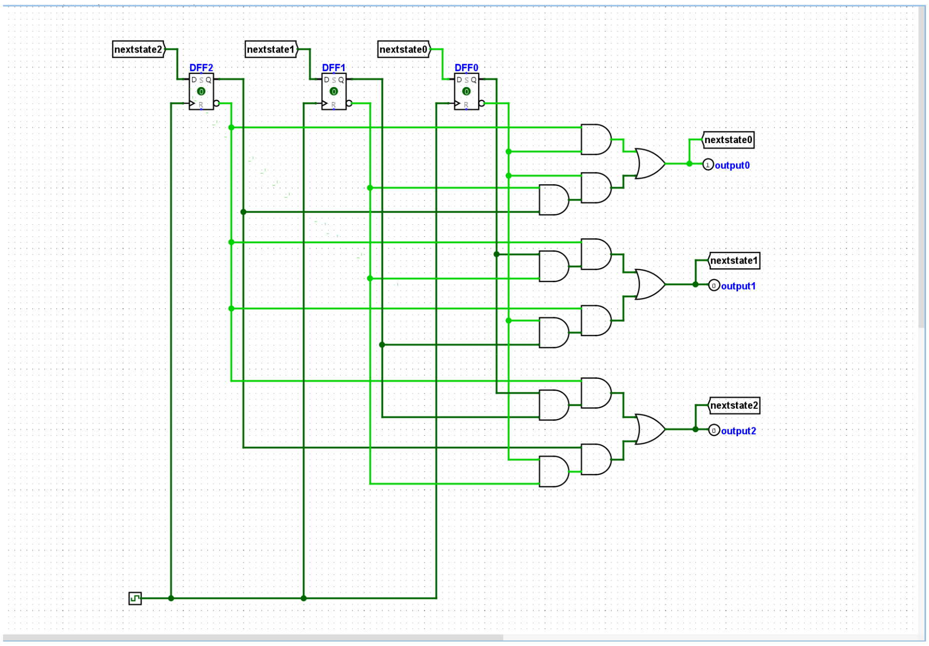


Present State	Next State	D2	D1	D0
000	001	0	0	1
001	010	0	1	0
010	011	0	1	1
011	100	1	0	0
100	101	1	0	1
101	000	0	0	0

So we can get the state transition diagram:



So we can get state transition logic that:
D0 $= \bar{D}2\bar{D}1\bar{D}0 + \bar{D}2D1\bar{D}0 + D2\bar{D}1\bar{D}0$
D1 $= \bar{D}2\bar{D}1D0 + \bar{D}2D1\bar{D}0$
D2 $= \bar{D}2D1D0 + D2\bar{D}1\bar{D}0$
So we can draw the logic circuit that:
the reset state is:



among it, the output0 is the lsb of the next-state and the output2 is the msb of the next-state the DFF0 is the lsb of the current-state and the DFF2 is the msb of the current state.

3 Finite state machine

The function of a vending machine which sells bottles of soda is described below:

- Each bottle costs $1.50.
- The machine only accepts $0.50 and $1 coins. If a customer inserts enough coins, the machine will dispense a bottle of soda (FSM will output "1", otherwise "0") and returns change if needed , e.g., the output of DISPENSE states may be "1 $0.5", other states' output may be "0 $0".
- The process happens one coin at a time, and there is no simultaneous insertion of multiple coins or shipping of multiple bottles. After each transaction, the vending machine enters the IDLE state.
- We don't need to account for a scenario where a customer inserts coins but decides not to make a purchase.

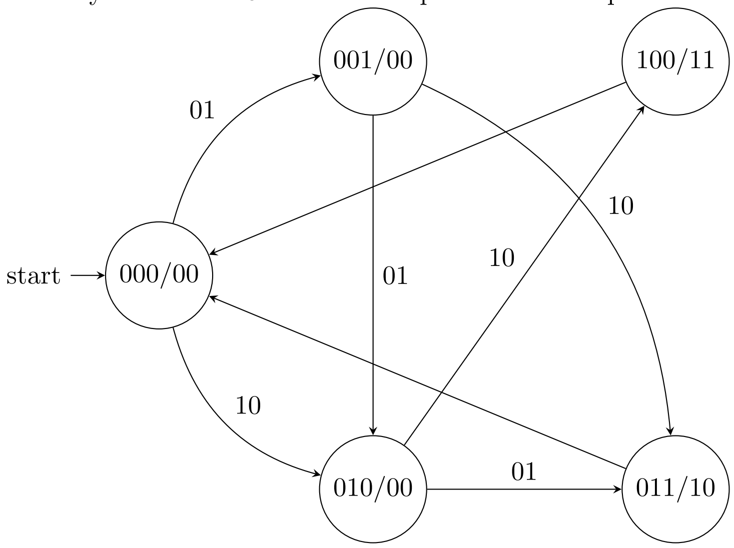
(a) Draw the FSM (Moore machine) for this vending machine.[**15 pt**]

(b) Draw the FSM (Mealy machine) for this vending machine.[**10 pt**]

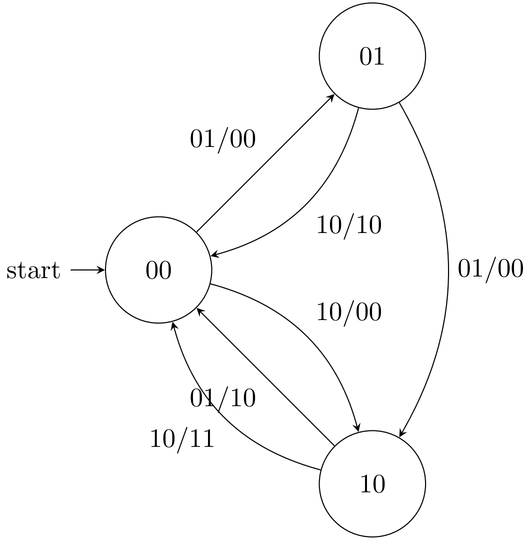
(c) Could Moore machines and Mealy machines be converted into each other to implement the same function? Compare their difference.[**10 pt**]

Answer to Question 3

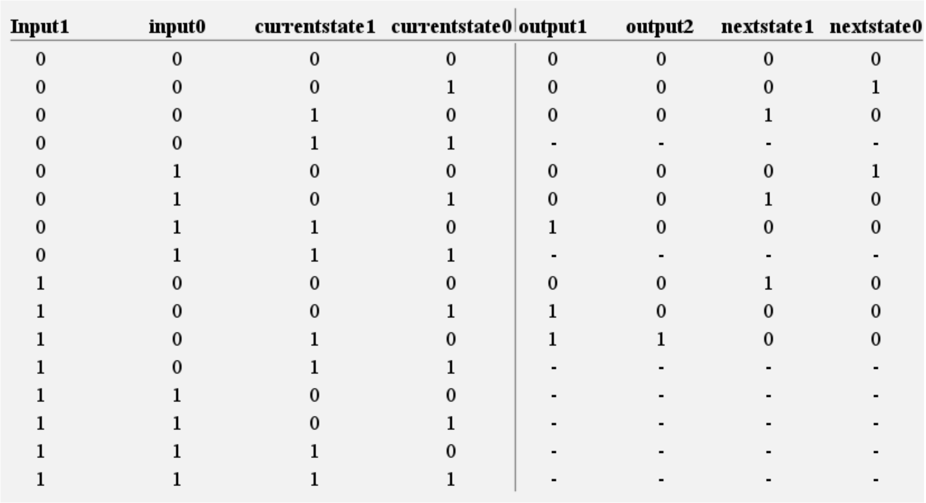
(a) We introduce eight bit:input1 and input0 represents the coin inserted by customers(00 == 0*0.5$ 01 == 1*0.5$ 10 == 2*0.5$), output1 represents whether the machine will dispense a bottle of soda and output2 represents whether the machine will return change(it will only be 0$ or 0.5$ i.e. the output2 will only be 0 or 1). Moreover, of course, we have currentstate1, currentstate0 and nextstate1, nextstate0 to represents the coins which have been inserted into the machine.
for Moore machine, we introduce currentstate2 and nextstate2, which are the msb of state bits and currentstate0 and nextstate0 are the lsb of the state bits.output1 is the msb of output state bits and output2 is the lsb of the output state bits.
set 000/00 and 00 to be IDLE
2-bit in binary above the line is the input1 and input0, 3-bit in binary is the currentstate and 2-bit in binary behind the 3-bit is the output1 and the ouput2.



(b) for mealy machine, we only use two bits to represnet the currentstate, the 2-bit in binary above the line is the input1 and input0, the other 2-bit in binary hehind the 2-bit is output1 and output2. The 2-bit in binary in the circle is the currentstate1 and the currentstate0.



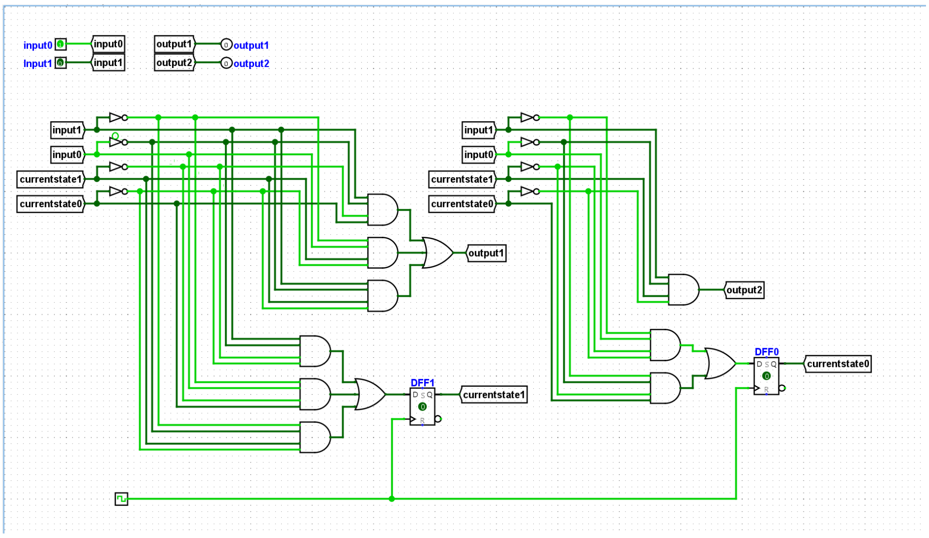
So, we have the truth table:



Input1	input0	currentstate1	currentstate0	output1	output2	nextstate1	nextstate0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	-	-	-	-
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	0
0	1	1	1	-	-	-	-
1	0	0	0	0	0	1	0
1	0	0	1	1	0	0	0
1	0	1	0	1	1	0	0
1	0	1	1	-	-	-	-
1	1	0	0	-	-	-	-
1	1	0	1	-	-	-	-
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

Among it, ”-”represents such state(and the corresponding currentstate does not exist too) does not exist.(we can not change the currentstate input, so using ”-” to represents such situation does not exist)

And according to the truth table, we can draw the Mealy machine by logisim:



(c) Yes,they can be converted into each other to implement the same function.

Difference:

In a Moore machine, the outputs depend only on the current state of the machine.Outputs are associated with states, meaning each state has a defined output value.The output of a Moore machine changes only when the state changes.

In a Moore machine, the outputs depend only on the current state of the machine.Outputs are associated with state transitions, meaning the output can change as soon as an input is received and a state transition occurs.The output of a Mealy machine can change with each clock cycle or input event, not just when the state changes.

1.Moore machines have deterministic output behavior, as the output is solely determined by the current state while Mealy machines have output behavior that can change more frequently, as outputs depend on both the current state and the input.

2.In Moore machines, the state transition logic defines the next state only, without considering the input while In Mealy machines, the state transition logic defines both the next state and the output based on the input and the current state.

3.Moore machines can be more synchronized as their outputs are updated only when the state changes, typically synchronized to a clock signal while Mealy machines can have outputs that update asynchronously with respect to the state changes, as they can be influenced by the inputs directly.