ShanghaiTech University
School of Information Science and Technology
Student name: Sihan Zhuang
Student ID: 2022533091

CS110: Computer Architecture
Spring, 2024

Assignment 4: Digital circuit

Attention: 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:

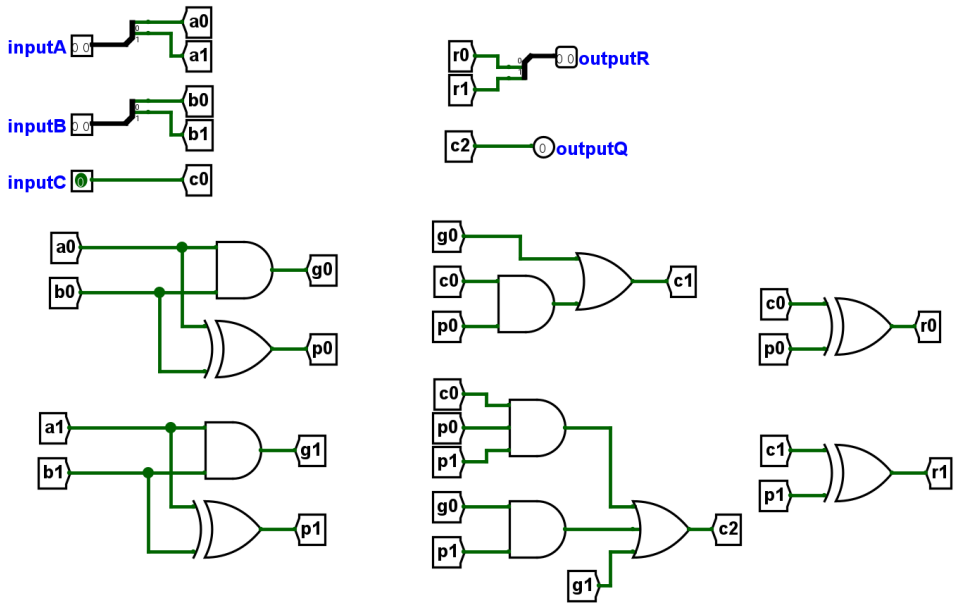


Figure 1: A 2-bit arithmetic circuit

(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)

A	B	C	$a1$	$a0$	$b1$	$b0$	$c0$	$ouputQ$	$outputR$
00	00	0	0	0	0	0	0	0	00
00	00	1	0	0	0	0	1	0	01
00	01	0	0	0	0	1	0	0	01
00	01	1	0	0	0	1	1	0	10
00	10	0	0	0	1	0	0	0	10
00	10	1	0	0	1	0	1	0	11
00	11	0	0	0	1	1	0	0	11
00	11	1	0	0	1	1	1	1	00
01	00	0	0	1	0	0	0	0	01
01	00	1	0	1	0	0	1	0	10
01	01	0	0	1	0	1	0	0	10
01	01	1	0	1	0	1	1	0	11
01	10	0	0	1	1	0	0	0	11
01	10	1	0	1	1	0	1	1	00
01	11	0	0	1	1	1	0	1	00
01	11	1	0	1	1	1	1	1	01
10	00	0	1	0	0	0	0	0	10
10	00	1	1	0	0	0	1	0	11
10	01	0	1	0	0	1	0	0	11
10	01	1	1	0	0	1	1	1	00
10	10	0	1	0	1	0	0	1	00
10	10	1	1	0	1	0	1	1	01
10	11	0	1	0	1	1	0	1	01
10	11	1	1	0	1	1	1	1	10
11	00	0	1	1	0	0	0	0	11
11	00	1	1	1	0	0	1	1	00
11	01	0	1	1	0	1	0	1	00
11	01	1	1	1	0	1	1	1	01
11	10	0	1	1	1	0	0	1	01
11	10	1	1	1	1	0	1	1	10
11	11	0	1	1	1	1	0	1	10
11	11	1	1	1	1	1	1	1	11

(b) (1) addition

(2) Advantages: This circuit $Carry-Lookahead\ adder$ is much faster by going through fewer gate in sequence. It can determine the carry in to the high-order bits in advance, and the computation of each bit can be executed in parallel, while the origin circuit $Ripple\ Carry$ must compute each bit one by one since it uses the result C_{out} of last bit as current bit's input C_{in}.

(3) Disadvantages: Design of carry in will get complicated as the number of bit increases, which makes hardware cost of this circuit become expensive. In $Ripple\ Carry$, cost grows linearly with the number of bits.

(c) Critical path in this circuit is from input to $c2$.
Step 1: $p0$ and $p1$ can be computed in parallel passing a 2-input XOR gate. 1 ns delay.
Step 2: From $p0$ and $p1$ to $c2$, it goes through one 3-input AND gate and one 3-input OR gate. 2 * 2 ns delay.

$$Total = 1ns + 4ns = 5ns$$

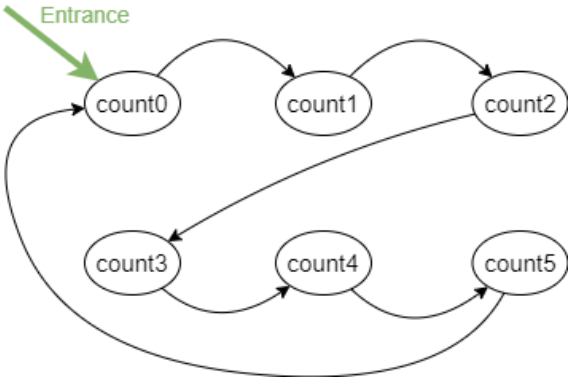
2 SDS

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

Step 1: Draw finite state machine of the desired function

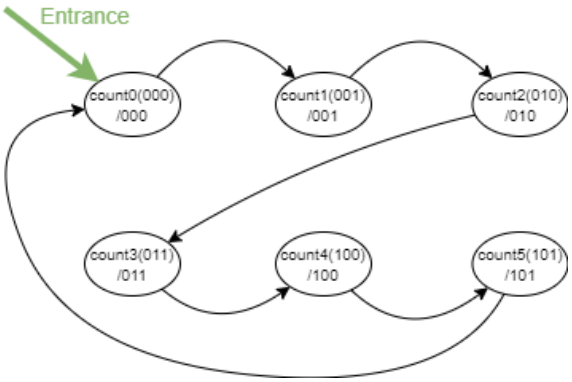


Step 2: Define/assign binary numbers to represent the states, the inputs and the outputs
State:

$$count0 = 000, count1 = 001, count2 = 010,$$
$$count3 = 011, count4 = 100, count5 = 101$$

Output:

$$Output0 = 000, Output1 = 001, Output2 = 010,$$
$$Output3 = 011, Output4 = 100, Output5 = 101$$

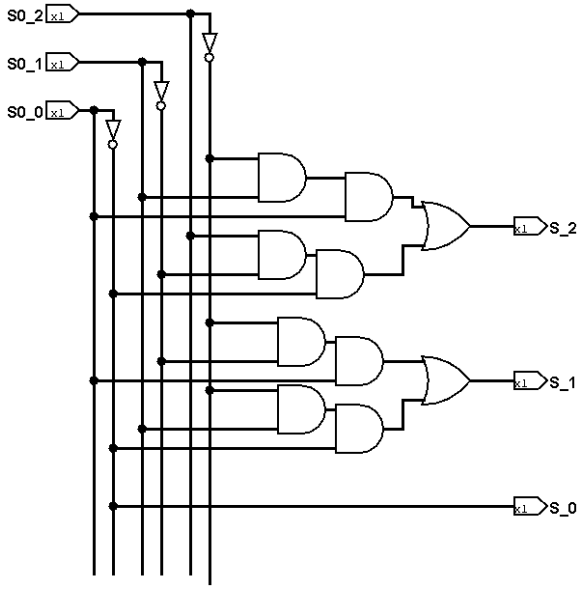


Step 3: Write down the truth table (enumerate input/previous state (and current state) and their corresponding current state (and output))

$S_{k-1}[2]$	$S_{k-1}[1]$	$S_{k-1}[0]$	$S_k[2]$	$S_k[1]$	$S_k[0]$	$Ouput[2]$	$Ouput[1]$	$Ouput[0]$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	0	0	0	0	0	0

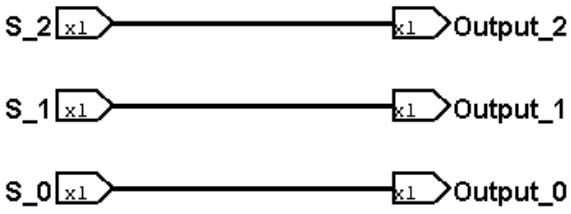
Step 4: Use template and decide the combinational block for state transition and output logic
State transition logic:

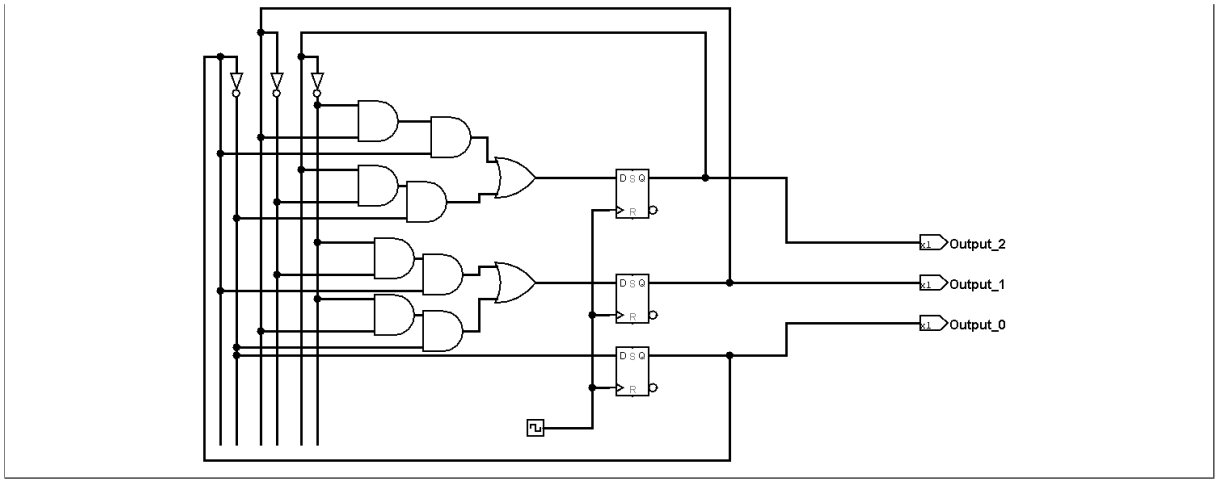
$$S_k[2] = \overline{S_{k-1}[2]}S_{k-1}[1]S_{k-1}[0] + S_{k-1}[2]\overline{S_{k-1}[1]S_{k-1}[0]}$$
$$S_k[1] = \overline{S_{k-1}[2]S_{k-1}[1]S_{k-1}[0]} + \overline{S_{k-1}[2]}S_{k-1}[1]\overline{S_{k-1}[0]}$$
$$S_k[0] = \overline{S_{k-1}[0]}$$



Output logic:

$$Output[2] = S_k[2]$$
$$Output[1] = S_k[1]$$
$$Output[0] = S_k[0]$$




D S Q
R
D S Q
R
D S Q
R
Output_2
Output_1
Output_0

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) **State:**

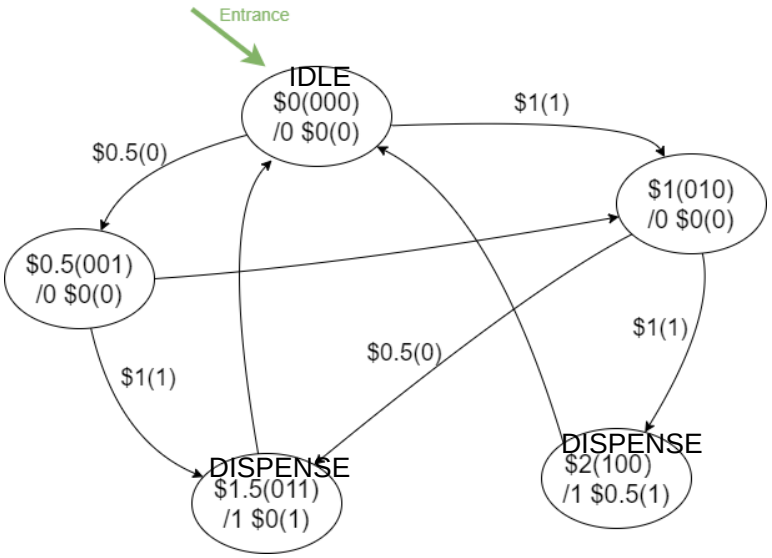
$$IDLE\ state = 000, \$0.5 = 001, \$1 = 010,$$
$$\$1.5 = 011, \$2 = 100$$

Input:

$$\$0.5 = 0, \$1 = 1$$

Output:

$$0\ \$0 = 0$$
$$1\ \$0, 1\ \$0.5 = 1$$



(b) **State:**

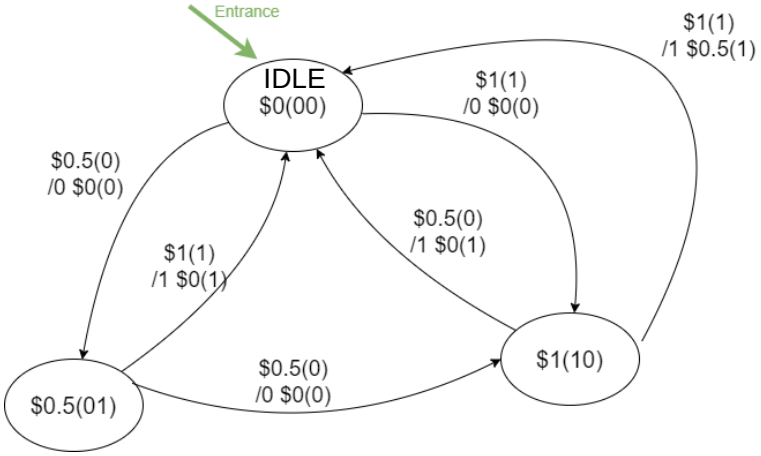
$$IDLE\ state = 00, \$0.5 = 01, \$1 = 10$$

Input:

$$\$0.5 = 0, \$1 = 1$$

Output:

$$0\ \$0 = 0$$
$$1\ \$0, 1\ \$0.5 = 1$$



(c) Yes.

Moore	Mealy
Output depends solely on the current state	Output depends on both the current state and input
Output controlled by the clock somewhat, one clock cycle later to react	Output uncontrolled by the clock, changing with input
More states are required	Less number of states