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

CS110: Computer Architecture
Spring, 2026

Assignment 4: Digital circuit

Attention: Make sure use LaTeX to complete your work and submit a pdf file. 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

The circuit shown in Figure. 1 is a three-input priority circuit. Answer the following questions.

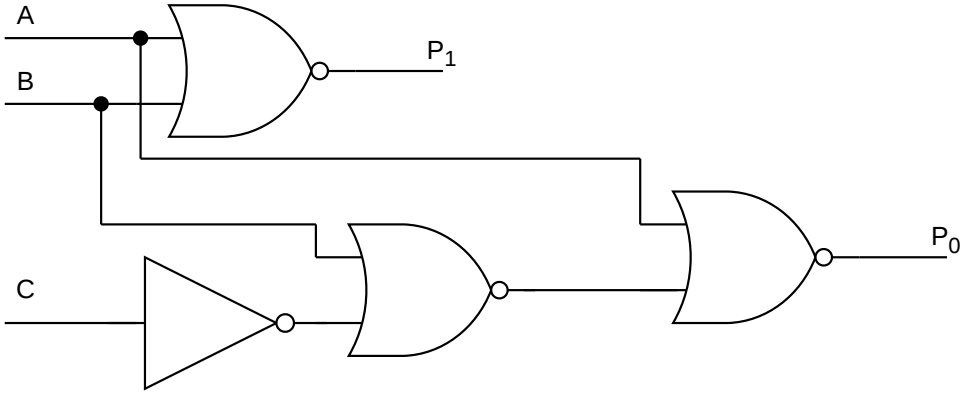


Figure 1: A 3-input priority circuit

(a) Draw the truth table of this circuit in the following table.**[7 pt]**

(b) Write the simplified logic expressions for $P0$ and $P1$ in a Sum of Product format. **[8 pt]**

(c) Based on the circuit in Figure. 1, a fourth input signal D is introduced, and the outputs are modified to Y_2, Y_1, and Y_0. The priority hierarchy is defined as $A > B > C > D$. The behavior of the system is specified as follows:

When $A = B = C = D = 0$, the outputs are $Y_2 = 1, Y_1 = 0, Y_0 = 0$.

When at least one of $\{A, B, C\}$ is non-zero, regardless of the value of D, $Y_2 = 0$, and the outputs (Y_1, Y_0) match the behavior of the truth table for (P_1, P_0) in part (a).

When $A = B = C = 0$ and $D = 1$, the outputs are $Y_2 = 0, Y_1 = 1, Y_0 = 1$.

Assuming the circuit in Figure 1 is treated as a primitive component (sub-module), derive the combinational logic expressions for Y_2, Y_1, and Y_0 in terms of D, P_1, and P_0 in a Sum of Product form. **[9 pt]**

(d) If modifications are permitted to the internal logic of the circuit in Figure. 1, write the Boolean expressions for the outputs Y_2, Y_1, and Y_0 in terms of the primary inputs A, B, C, and D **[8 pt]**

Answer to Question 1

Your answer here.
(a) Do not modify the given values in the truth table.

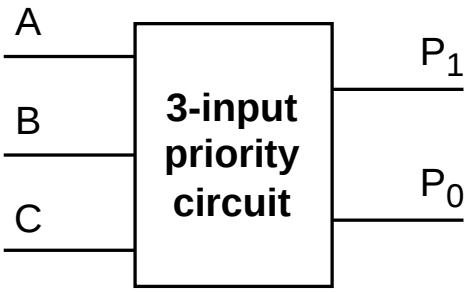


Figure 2: A 3-input priority component diagram

A	B	C	P1	P0
0	0	0	1	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Each row 1 pt, Lose the 1 pt for any error in a row.

(b)

$$P_1 = \overline{A}\,\overline{B}$$
$$P_0 = \overline{A}\,B + \overline{A}\,\overline{C}$$

Each 4 pt.

(c)

$$Y_2 = P_1 P_0 \overline{D}$$
$$Y_1 = P_1 + \overline{P_1}\,\overline{P_0} D$$
$$Y_0 = P_0 + \overline{P_1}\,\overline{P_0} D$$

or

$$Y_2 = P_1 P_0 \overline{D}$$
$$Y_1 = P_1 D + P_1 \overline{P_0}$$
$$Y_0 = P_0 D + P_0 \overline{P_1}$$

3 pt for each of Y_2,Y_1,Y_0.

(d)

$$Y_2 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$
$$Y_1 = \overline{A}\,\overline{B}\,C + \overline{A}\,\overline{B}\,D$$
$$Y_0 = \overline{A}\,B + \overline{A}\,\overline{C}\,D$$

2 pt for Y_2, 3 pt for each of Y_1,Y_0.

2 SDS

In the following circuit, NOT gates have a delay of 1ns, AND gates have a delay of 3ns, NAND gates have a delay of 5ns, OR gates have a delay of 3ns, NOR gates have a delay of 2ns. The registers have a clk-to-q delay of 2ns and setup time of 2ns. Assume the inputs come from registers. All the delays refer to propagation delay.

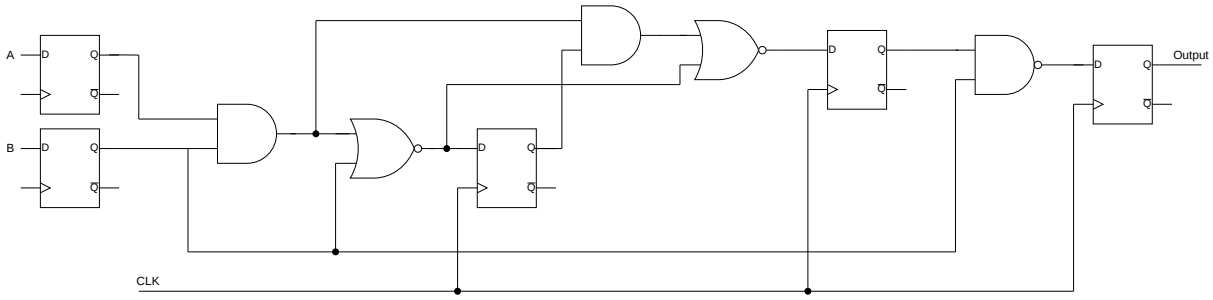


Figure 3: Circuit Diagram

What is the minimum acceptable clock cycle time for this circuit? What clock frequency does it correspond to? (please include enough explanation) **[18 pt]**

Answer to Question 2

$T \geq t_{clk-to-q} + t_{setup} + t_{logic}$

For the first D-flipflop, $T \geq 2 + 2 + 3 + 2 = 9ns$
For the second D-flipflop, $T \geq 2 + 2 + 3 + 3 + 2 = 12ns$
For the third D-flipflop, $T \geq 2 + 2 + 5 = 9ns$

So the minimum acceptable clock cycle is 12ns, The corresponding frequency is 83.3MHz.

12 pt for minimum clock cycle. 6 pt for frequency.

3 Finite state machine

In this part, you are required to implement a sequence detector. When the detector identifies the specific bit pattern "1011", it outputs a 1. The system processes a bit sequence by taking one bit as input per clock period, moving from left to right.

(Hint: Sequences can overlap, so any input '1' can serve as the start of a sequence; please carefully consider the relationships between states.)

(a) Draw the FSM (Moore machine) for this detector in five states represented by a 3-bit signal CS_2, CS_1, CS_0: 000 (start), 001 ('1' detected), 010 ('10' detected), 100 ('101' detected), 101 ('1011' detected).When drawing the state diagram, use the format $CS_2CS_1CS_0$/output for each state. **[10 pt]**

(b) Draw the FSM (Mealy machine) for this detector in no more than four states represented by $\{S_0, S_1, S_2, S_3\}$.When drawing the state diagram, use the format "*input/output*" for each state transition.**[10 pt]**

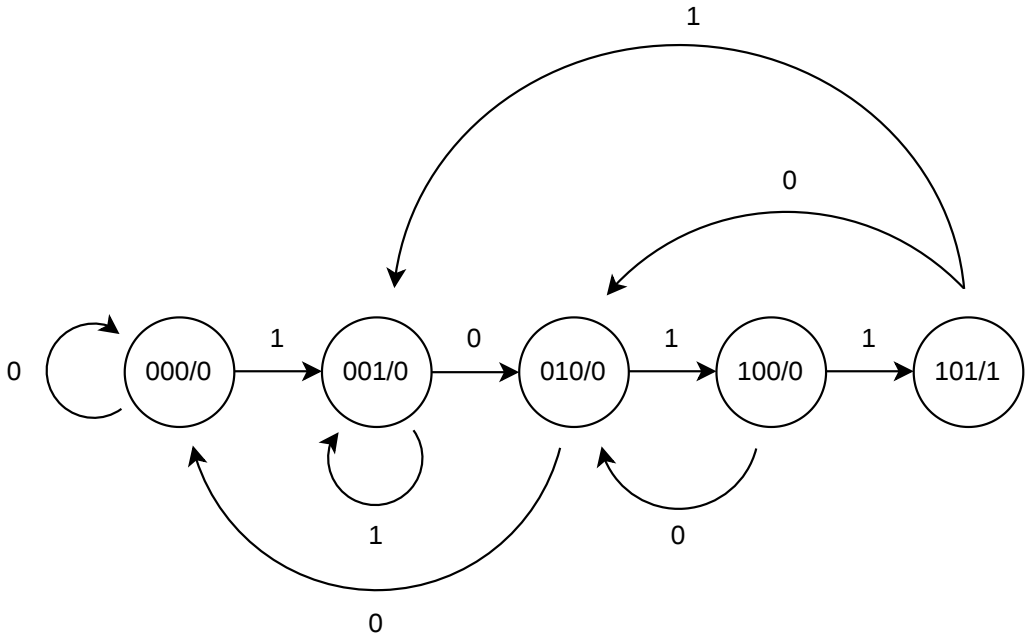
(c) Fill the truth table for the next-state and output logic based on the Moore FSM. Then Write the logic expressions for $NS2$,$NS1$,$NS0$ and output. **[20 pt]**

(d) Draw the circuit diagram for the update of CS1 using only AND gate, inverter and a D-flipflop. **[10 pt]**

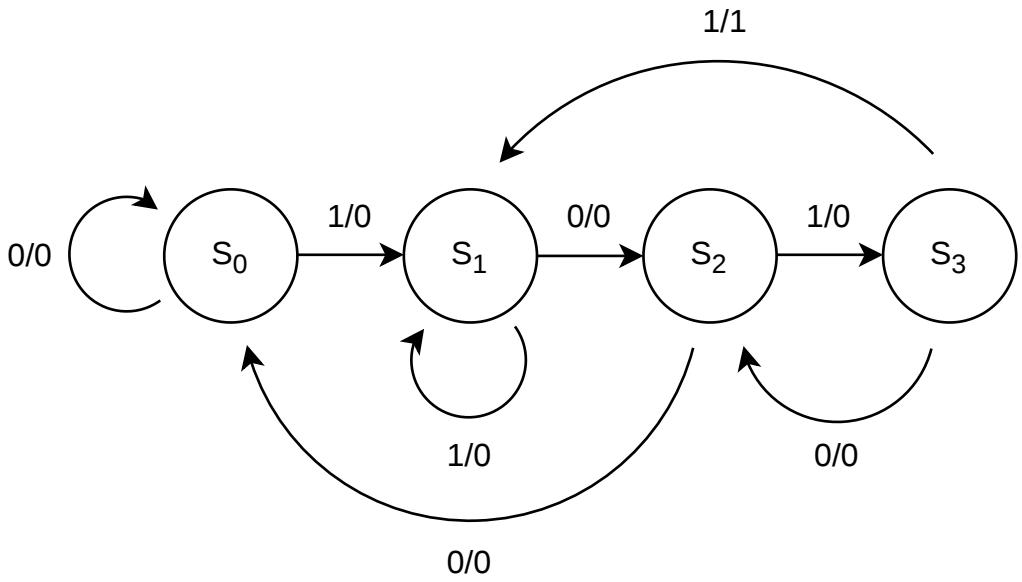
(Hint: The diagram for NS1 actually updates the signal CS1)

Answer to Question 3

Your answer here.
(a)



10 pt for totally correct, else 0 pt. (b)



10 pt for totally correct, else 0 pt.

(c) Do not modify the given values in the truth table.

CS[2]	CS[1]	CS[0]	input	NS[2]	NS[1]	NS[0]	output
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	1	0	0
1	0	0	1	1	0	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	0	1	1
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

10 pt for the table Each row 1 pt, Lose the 1 pt for any error in a row.

$$NS_2 = (CS_1\overline{CS_2} + CS_2\overline{CS_1}) \cdot \overline{CS_0}In$$

$$NS_1 = (CS_0 + CS_2) \cdot \overline{CS_1}\,\overline{In}$$

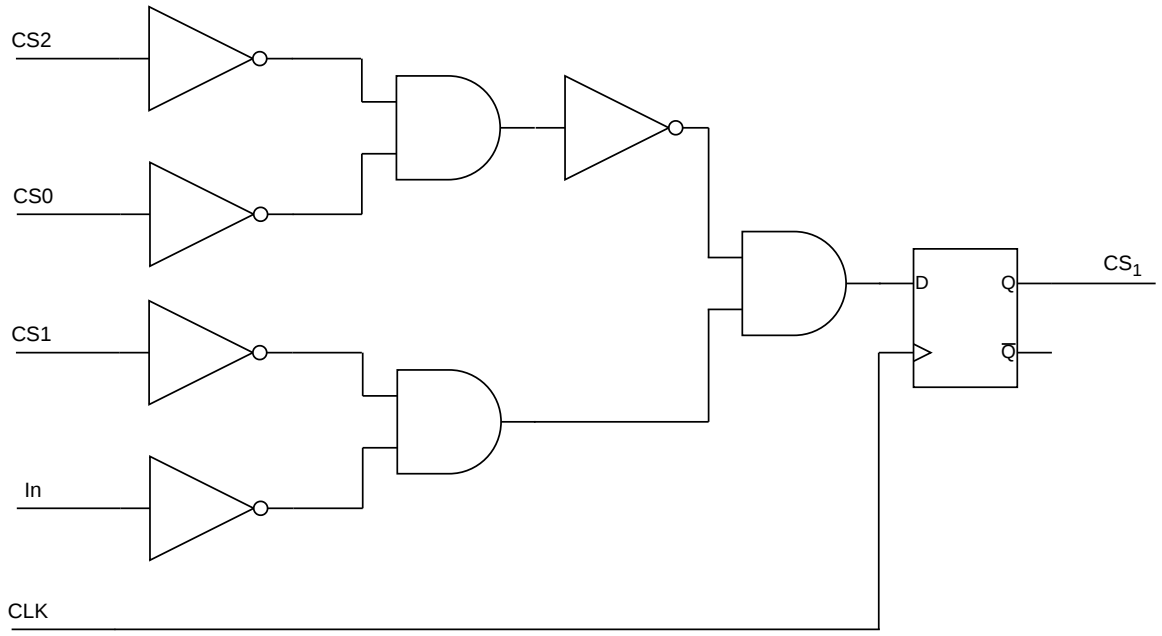
$$NS_0 = \overline{CS_1}\, In$$

$$Output = CS_2\overline{CS_1}CS_0$$

10 pt for the equation 3 pt for each of $NS2$,$NS1$,$NS0$, 1 pt for Output.

(d)

$$NS_1 = (CS_0 + CS_2) \cdot \overline{CS_1}\, \overline{In} = \overline{(\overline{CS_1} \cdot \overline{CS_1})} \cdot \overline{CS_1}\, \overline{In}$$



10 pt for totally correct, else 0 pt.

4 Error Summary and Suggestions

(1). Avoid Overline Connection: When using overlines to represent the NOT operation for multiple adjacent variables, ensure there is a clear, intentional gap between the lines. If the overlines are connected, it will be interpreted as a single NAND-like operation (NOT of the product) rather than the product of two independent NOT variables. (Example: Use $\overline{A} \cdot \overline{B}$ or $\overline{A}\,\overline{B}$ instead of $\overline{AB}$, as the latter implies $\overline{A \cdot B}$.)

(2). Hidden Simplifications: Many students expressed P_0 as $\overline{A}B + \overline{A}\,\overline{B}\,\overline{C}$ in Question 1(b). While technically correct, this expression can be further simplified using the absorption law variant ($\overline{A}B + \overline{A}\,\overline{C}$). Such simplifications can be subtle and require careful attention.

Always check if the question explicitly requests a "simplified" logic expression. If it does, you must provide the most reduced form to receive full credit. Even when not explicitly required, completing obvious simplification steps is highly recommended.

Aside from Question 1(b), we have verified all answers for questions that did not specify a "simplified" format. However, keep in mind that overly complex or redundant expressions are not ideal for scoring well in an exam setting.

(3). In Question 2 regarding Synchronous Digital Systems (SDS), the majority of errors stemmed from the incorrect identification of logic gates within the circuit diagram. It is highly recommended to familiarize yourself with the standard symbols for different gates (such as AND, NAND, OR, and NOR) to avoid such fundamental mistakes in future assignments and exams.

Additionally, a small number of errors resulted from calculating the minimum propagation delay rather than the maximum delay (the critical path). It is important to remember that the minimum acceptable clock cycle time is determined by the maximum delay across the logic paths, which in turn defines the maximum (worst-case) operating frequency.

(4).FSM Diagrams and Clarity: In Question 3 regarding Finite State Machines (FSM), some students provided only text descriptions instead of drawing both required diagrams as specified in the prompt. For those who did include diagrams, please ensure that state transition arrows do not overlap or cross unnecessarily, and that the transition labels (inputs/outputs) are clearly associated with their respective lines. Diagrams with significant ambiguity in their logic flow or labeling resulted in a 2-point deduction.

Regarding the truth table, please note that this question evaluates the Moore-type FSM. In a Moore machine, the output is strictly dependent on the current state. This refers to the output generated within the current clock cycle based on the current state, rather than the output corresponding to the next state.

When drawing circuit diagrams, please strictly adhere to the specific logic gates permitted in the prompt. For instance, this assignment explicitly required the use of only AND gates and Inverters to evaluate your ability to transform Boolean expressions. Using unauthorized gate types (such as OR or NAND gates) will be considered incorrect.