

Assignment 4: Digital circuit

Attention: Recommend using \LaTeX to 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

The circuit shown in Figure. 1 is a 1-bit comparator. Answer the following questions.

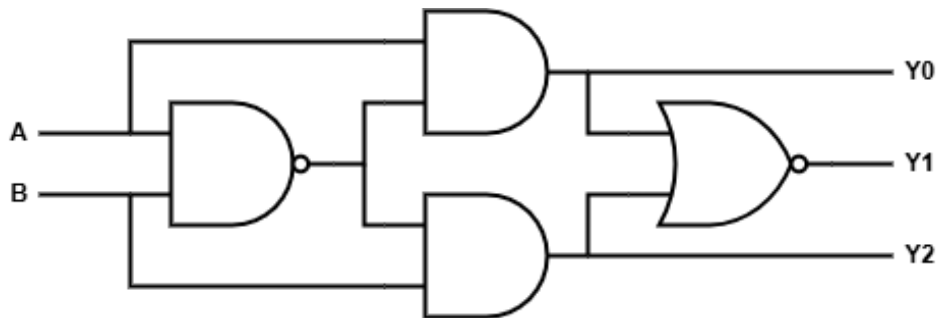


Figure 1: A 1-bit comparator circuit

- Write the un-simplified logic expressions for $Y0$, $Y1$ and $Y2$. [6 pt]
- Draw the truth table of this circuit in the following table. [6 pt]
- Write the sum of minterm for $Y0$, $Y1$ and $Y2$. [6 pt]
- What comparison do the outputs $Y0$, $Y1$ and $Y2$ represent respectively? e.g: $Y0 = 1$ represents $A = B$, $A < B$ or $A > B$ (one of the three cases). [6 pt]
- Draw the circuit of an unsigned 2-bit comparator using this 1-bit comparator and the following logic gates: 2-input AND, 2-input OR, and 1-input NOT. The 2-bit comparator has two 2-bit inputs $A1A0$ and $B1B0$, three outputs $Y0$, $Y1$ and $Y2$ with the same function as the 1-bit comparator. You can use the 1-bit comparator as a basic logic block as shown in Figure. 2. [10 pt]

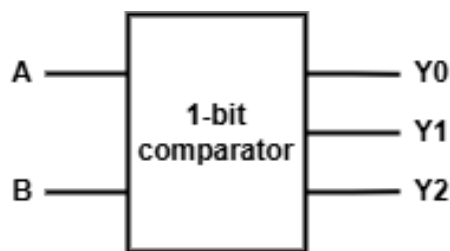


Figure 2: A 1-bit comparator diagram

Answer to Question 1

(a)

$$Y0 = A \cdot \overline{AB}$$

$$Y1 = \overline{A \cdot \overline{AB} + B \cdot \overline{AB}}$$

$$Y2 = B \cdot \overline{AB}$$

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

A	B	Y2	Y1	Y0
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

(c)

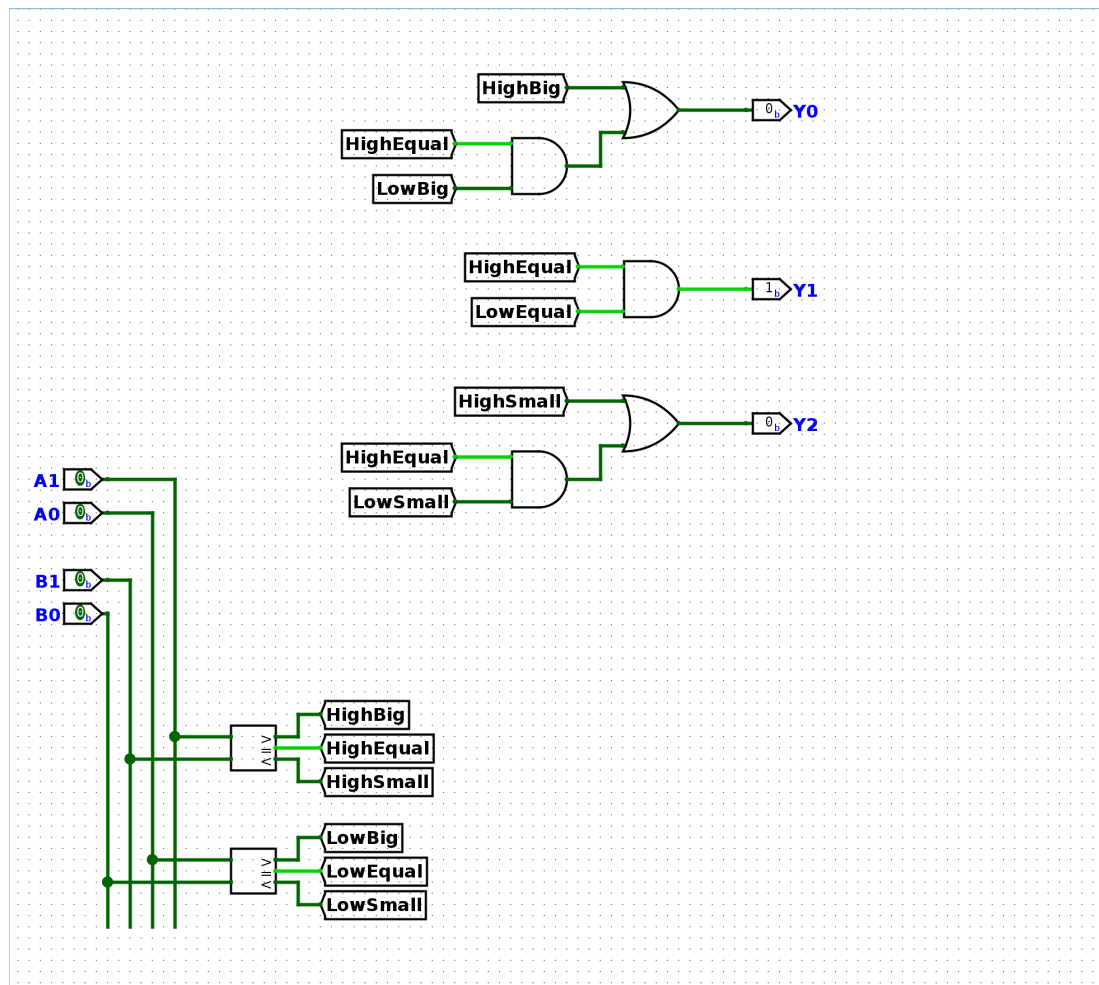
$$Y0 = A \cdot \overline{B}$$

$$Y1 = AB + \overline{A} \cdot \overline{B}$$

$$Y2 = B \cdot \overline{A}$$

(d) $Y0 = 1$ represents $A > B$, $Y1 = 1$ represents $A = B$, and $Y2 = 1$ represents $A < B$

(e)



2 SDS

In the following circuit, NOT gates have a delay of 1ns, AND gates have a delay of 4ns, NAND gates have a delay of 3ns, OR gates have a delay of 4ns, NOR gates have a delay of 3ns. 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.

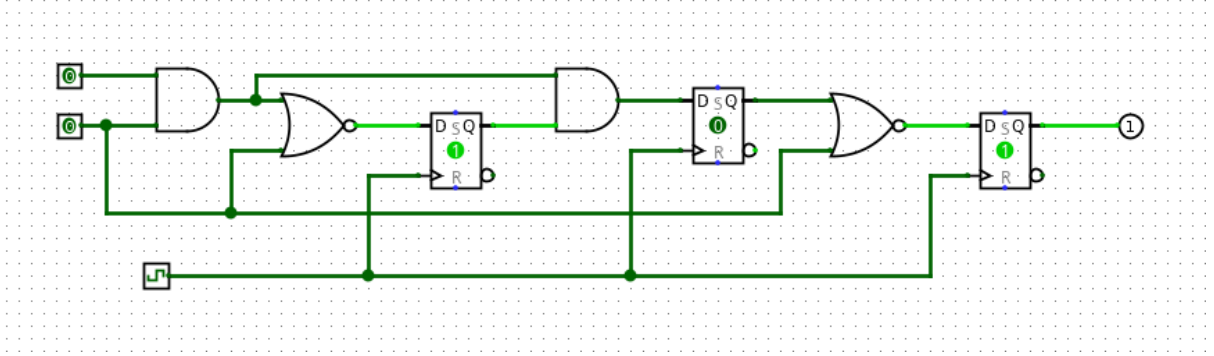


Figure 3: Circuit Diagram

What is the minimum acceptable clock cycle time for this circuit? What clock frequency does it correspond to? (please include enough explanation) [16 pt]

Answer to Question 2

t_1 is the time from input to the first register. t_2 is the time from the first register to the second register. t_3 is the time from the second register to the third register. t_4 is the time from the third register to the output. The minimum clock cycle time is the maximum of these four times.

$$t_1 = t_{clk-to-q} + t_{AND} + t_{NOR} + t_{setup} = 11ns$$

$$t_2 = t_{clk-to-q} + 2 * t_{AND} + t_{setup} = 12ns$$

$$t_3 = t_{clk-to-q} + t_{NOR} + t_{setup} = 7ns$$

$$t_4 = t_{clk-to-q} = 2ns$$

$$t_{min} = \max\{t_1, t_2, t_3, t_4\} = 12ns$$

$$f_{clock} = \frac{1}{t_{min}} = \frac{1}{12ns} = 83.33MHz$$

3 Finite state machine

In this part, you need to implement a detector. When receiving two or more successive '0's or '1's, it outputs 1. For a bit sequence, it inputs one bit a period from left to right. e.g: input='11101001', output='01100010'.

(a) Draw the FSM (Moore machine) for this detector in five states: {start}, {10}(discrete '0'), {01}(discrete '1'), {00}(successive '0's), {11}(successive '1's).

e.g: input='011001', state={start}→{10}→{01}→{11}→{10}→{00}→{01} [10 pt]

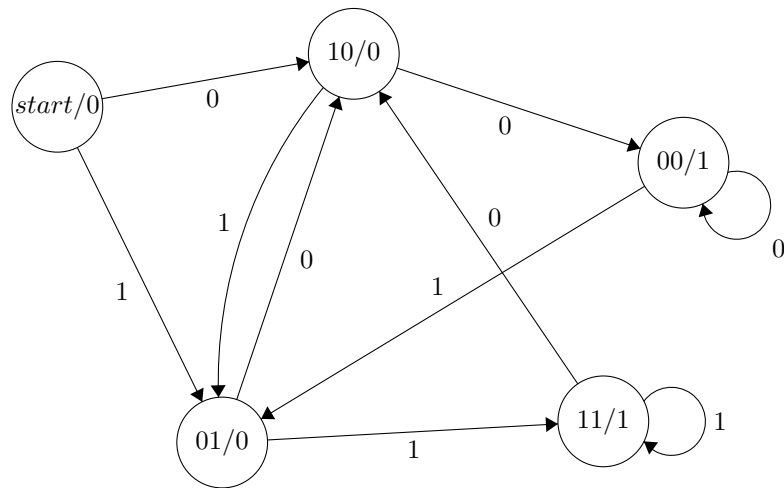
(b) Draw the FSM (Mealy machine) for this detector in no more than three states.[10 pt]

(c) Assign '000' to represent state {start}, '110' to represent {10}, '101' to represent {01}, '100' to represent {00}, '111' to represent {11}. We use 'CS' to represent current state and 'NS' for next state. Fill the truth table for the next-state and output logic based on the Moore FSM. [15 pt]

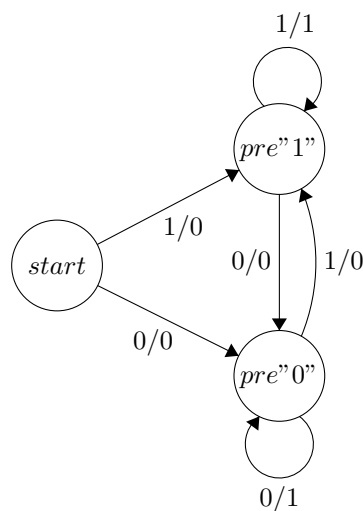
(d) Draw the circuit diagram for NS and output. [15 pt]

Answer to Question 3

(a)



(b)



(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	1	1	0	0
0	0	0	1	1	0	1	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	1

(d)

