
ECE 327 Solution to Final

2020t1 (Winter)

	Total	Approx.	
	Marks	Time	Page
Q1 Honesty Declaration	1	1	1
Q2 Ten Years from Now...	1	1	2
Q3 VHDL Coding	20	60	3
Q4 RTL Simulation	12	15	4
Q5 DFD Analysis	12	10	8
Q6 Retiming	10	20	9
Q7 Real-Time Performance	12	15	13
Q8 Performance Analysis	12	15	14
Q9 Latch Analysis	12	15	16
Q10 Elmore Delay	10	20	19
<hr/>			
Totals	102	172	

Q1 (1 Mark) Honesty Declaration*(estimated time: 1 minutes)*

In the space below, write a sentence stating that you promise that you will not discuss the exam with anyone until the exam period is over.

Q2 (1 Mark) Ten Years from Now...*(estimated time: 1 minutes)*

Ten years from now, what, if anything other than the lack of TimBits, will you remember about this course?

Q3 (20 Marks) VHDL Coding*(estimated time: 60 minutes)*

Write a synthesizable VHDL program named `max2` that satisfies the following specification:

1. The input signals shall be:

- `clk, reset i_valid : std_logic`
- `i_data : unsigned(7 downto 0)`

2. The outputs shall be:

- `o_done : std_logic`
- `o_max1, o_max2 : unsigned(7 downto 0)`

3. The system receives a sequence of 8 input values and outputs the two highest values from the sequence.

4. Each parcel of input data is denoted by a single clock cycle of `i_valid='1'`.

5. The environment guarantees that there are at least 2 bubbles between each `i_valid='1'`.

6. Within 3 clock cycles after the 8th input value in a sequence has been received (`i_valid='1'`), the system shall set `o_done='1'`, `o_max1` to the maximum value in the sequence, and `o_max2` to the second highest value in the sequence. That is, if the 8th `i_valid='1'` happens at time t , then `o_done='1'` must happen between t and $t+3$ inclusive.

7. When `o_done='1'`, the system shall hold `o_done`, `o_max1`, and `o_max2` constant until `reset='1'` or `i_valid='1'`.

8. The system shall allow multiple sequences to be sent consecutively without a `reset='1'` between the end of one sequence and the beginning of the next sequence.

Marking: The importance of various characteristics of the design in decreasing order of importance

- Synthesizability
- Correct functionality
- Elegance of design and cleanliness of code

Q4 (12 Marks) RTL Simulation*(estimated time: 15 minutes)*

Rewrite (decompose and sort) the code below to prepare it for RTL simulation. If the code is not compatible with RTL simulation, then explain why.

NOTES:

1. Do not perform any logical or arithmetic optimizations.
2. Ignore the elaboration error in the multiplication operations caused by the VHDL "feature" that the length of the output of a multiplication is sum of the lengths of the inputs.

Original program

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity sim is
end entity;

architecture main of sim is

    signal clk, a, b : std_logic;
    signal e, f, g, q, r, s, t : unsigned( 7 downto 0 );

begin
```

```
process begin
  clk <= '0';
  wait for 5 ns;
  clk <= '1';
  wait for 5 ns;
end process;

process begin
  a <= '0';
  b <= '1';
  wait for 7 ns;
  a <= '1';
  g <= to_unsigned( 3, 8 );
  wait for 10 ns;
  a <= '0';
  b <= '0';
  g <= to_unsigned( 5, 8 );
  wait;
end process;

process (a, b, e, f, g, q) begin
  if a = '1' then
    q <= e + f;
    if b = '1' then
      r <= q + g;
      t <= e + g;
    else
      r <= q - g;
      t <= e - g;
    end if;
  else
    q <= e * f;
    r <= q * g;
    t <= e * g;
  end if;
end process;

process (q, r) begin
  s <= q + r;
end process;

process begin
  wait until rising_edge( clk );
  e <= f;
  f <= g;
end process;

end architecture;
```

Answer:**1. Timed processes**

```
process begin
  clk <= '0';
  wait for 5 ns;
  clk <= '1';
  wait for 5 ns;
end process;

process begin
  a <= '0';
  b <= '1';
  wait for 7 ns;
  a <= '1';
  g <= to_unsigned( 3, 8 );
  wait for 10 ns;
  a <= '0';
  b <= '0';
  g <= to_unsigned( 5, 8 );
  wait;
end process;
```

2. Clocked processes

```
process begin
  wait until rising_edge( clk );
  e <= f;
  f <= g;
end process;
```

3. Decompose and sort combinational processes

```
process (a, e, f ) begin
  if a = '1' then
    q <= e + f;
  else
    q <= e * f;
  end if;
end process;

process (a, b, g, q ) begin
  if a = '1' then
    if b = '1' then
      r <= q + g;
    else
      r <= q - g;
    end if;
  else
    r <= q * g;
  end if;
end process;

process (q, r) begin
  s <= q + r;
end process;

process (a, b, e, g) begin
  if a = '1' then
    if b = '1' then
      t <= e + g;
    else
      t <= e - g;
    end if;
  else
    t <= e * g;
  end if;
end process;
```

4. Notes:

- The process driving t must come after a , b , e , and g . The order of t with respect to q , r , and s does not matter.

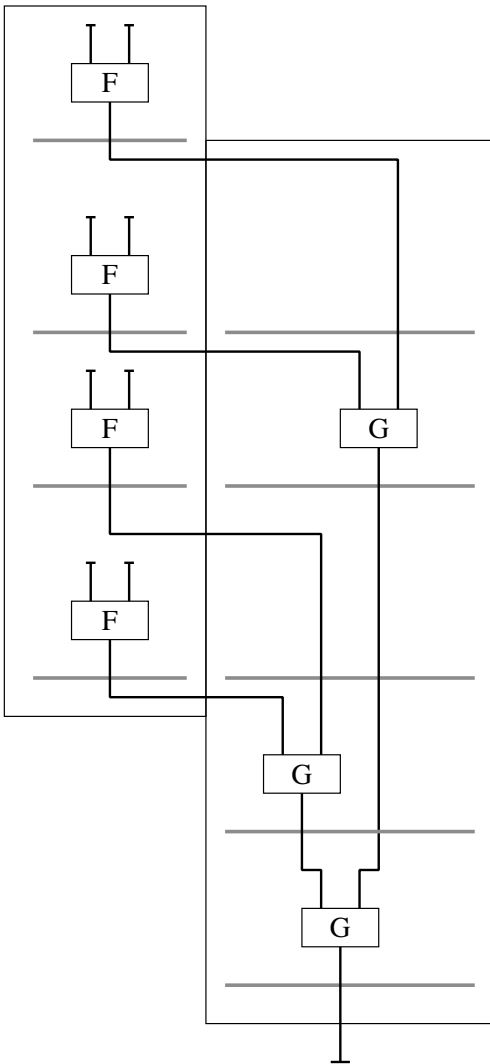
Marking:

- +3 marks Categorize processes according to timed, clocked, combinational.
- +3 marks Decompose combinational processes.
- +3 marks Sort processes by category: timed, clocked, comb.
- +3 marks Sort combinational processes by dependencies.
- 2 marks Decompose clocked or timed processes.
- 2 marks Each additional mistake

Q5 (12 Marks) DFD Analysis

(estimated time: 10 minutes)

Analyze the dataflow diagram by answer the questions below:



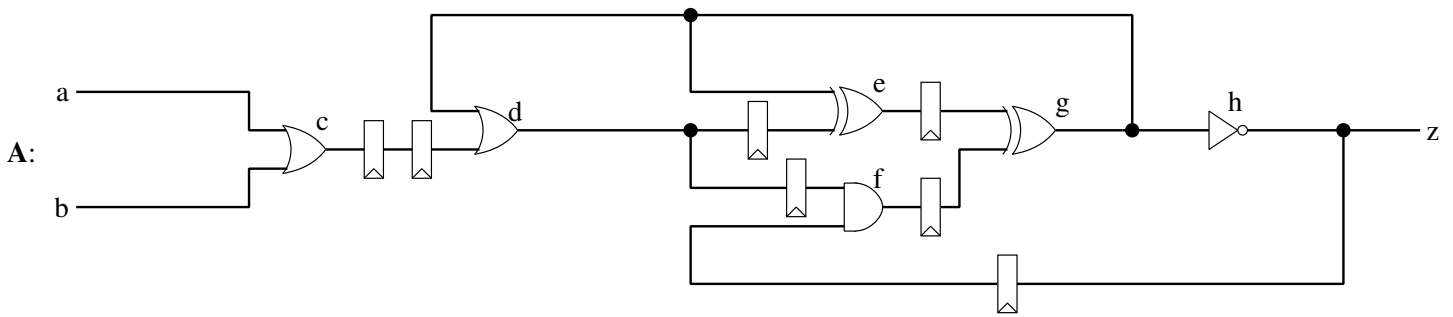
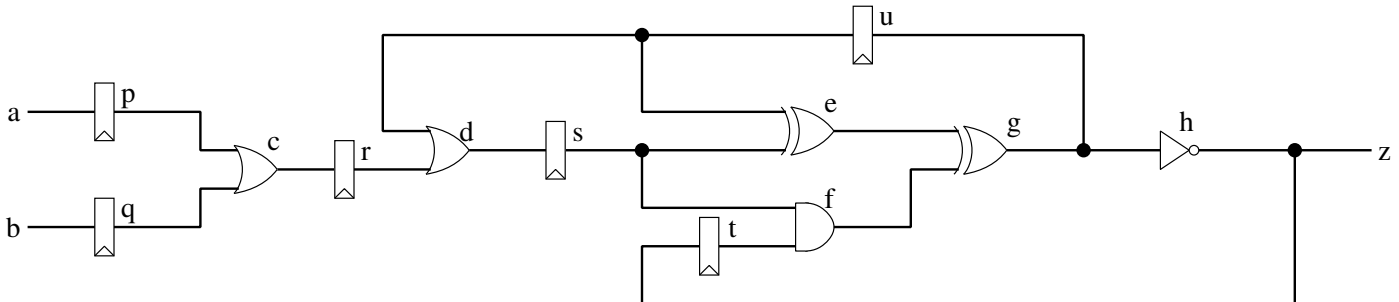
Throughput: 1/4
 Latency: 6
 Number of registers: 3
 Minimum latency: 3

Q6 (10 Marks) Retiming

(estimated time: 20 minutes)

Which **one** of the circuits (A–D) is a possible retiming of the original circuit.

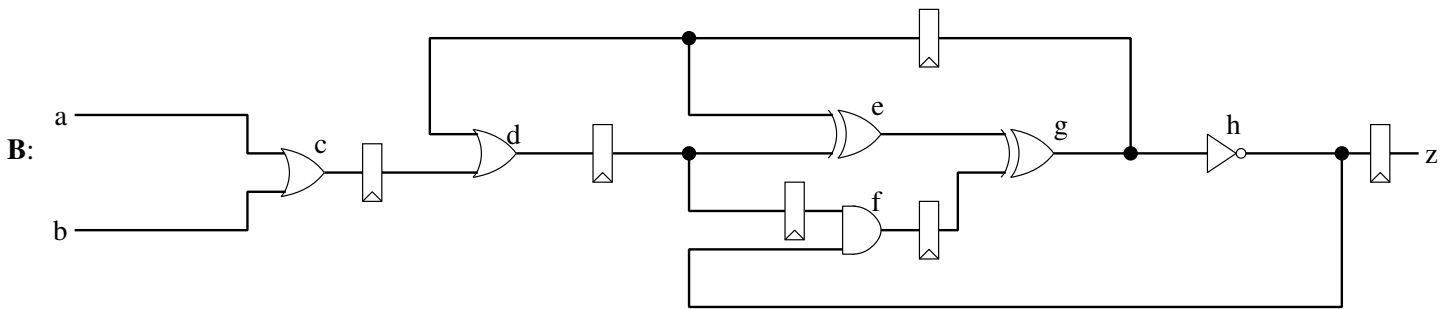
Original circuit:



Answer:

No: Any of the differences below are sufficient to see that the circuit is not a retiming of the original.

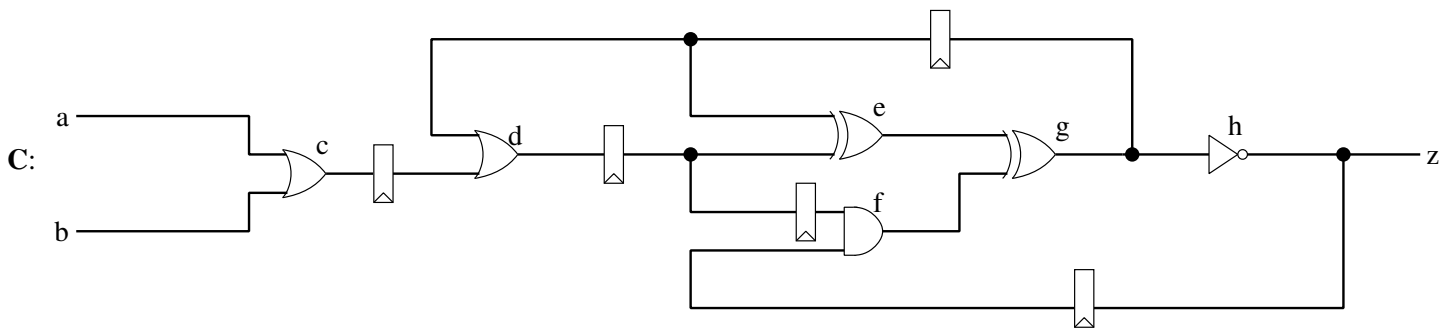
	Original	A
a, c, d, e, g, h, z	3 flops	4 flops
a, c, d, f, g, h, z	3 flops	4 flops
loop f, g, h	1 flop	2 flops
loop d, e, g	2 flops	1 flop



Answer:

No: Any of the differences below are sufficient to see that the circuit is not a retiming of the original.

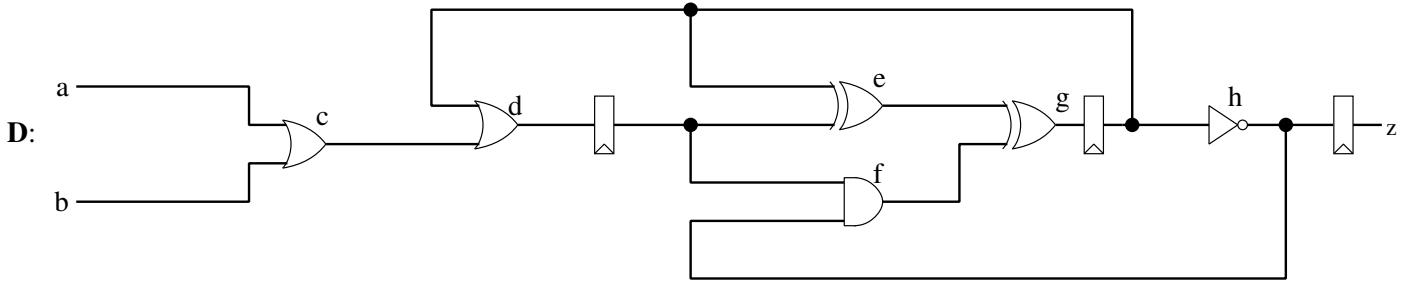
	Original	A
a, c, d, e, g, h, z	3 flops	2 flops
loop d, f, g	2 flops	3 flops



Answer:

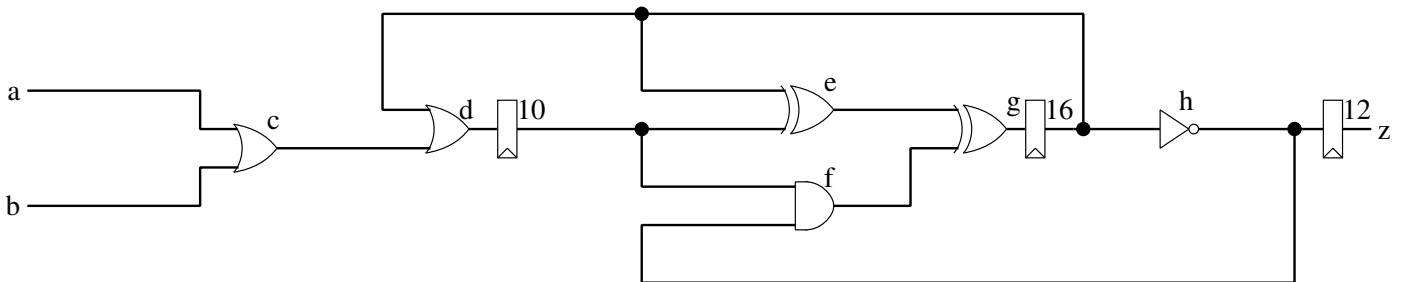
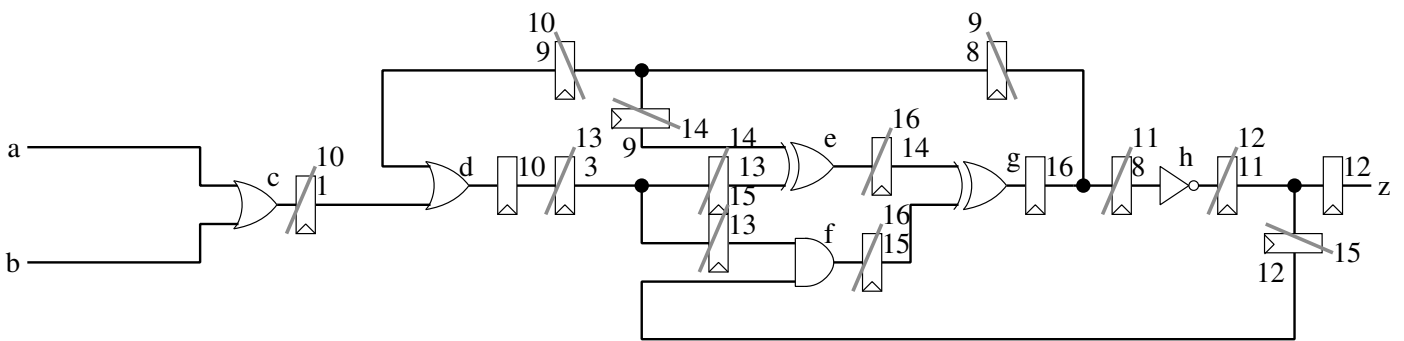
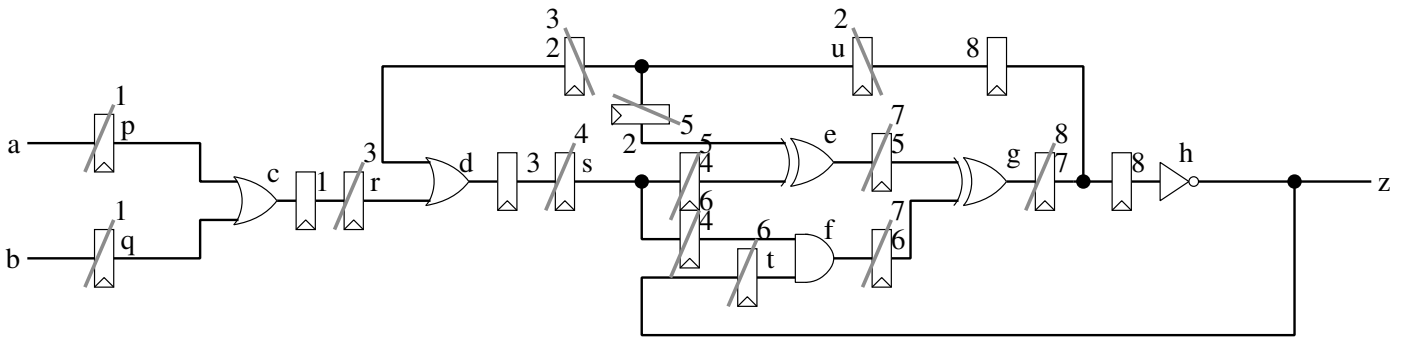
No: Any of the differences below are sufficient to see that the circuit is not a retiming of the original.

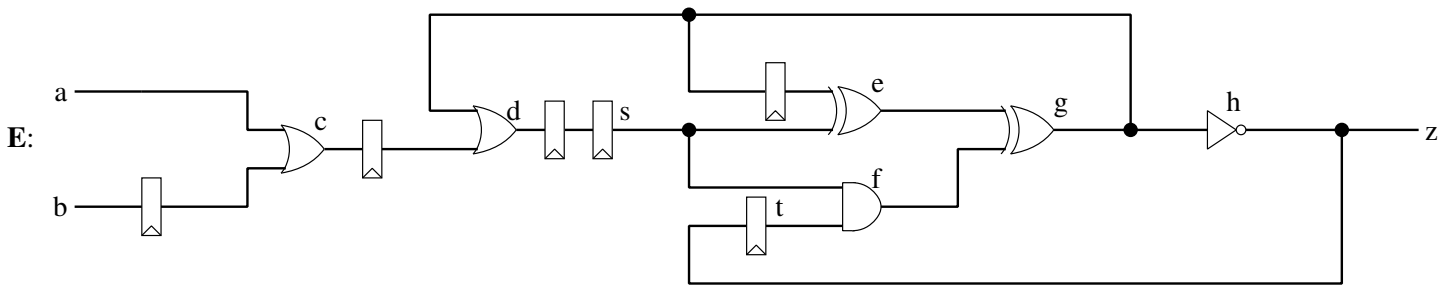
	Original	A
a, c, d, e, f, h, z	3 flops	5 flops
loop d, f, g	2 flops	4 flops



Answer:

Yes: The sequence of retiming operations is shown below





Answer:

No: There is a flop between b and c, but no flop between a and c.

Marking:

10 marks D

6 marks A, B, C

2 marks E

Q7 (12 Marks) Real-Time Performance

(estimated time: 15 minutes)

The ECE-327 course notes state that for real time systems (systems like anti-lock brakes that must respond within a certain amount of time), latency is often more important than throughput.

Briefly explain the probable justification for this statement.

Answer:

Latency measures the time from when the inputs are available until the output is produced. Real time systems need to react within a certain amount of time, which matches the definition of latency.

In contrast, throughput measures the rate at which data enters or exits the system. So, a system with a high throughput could have a very long latency, which means that the system would be slow to respond to an input.

Marking:

+3 marks *definition of latency*

+3 marks *definition of throughput*

+3 marks *real-time needs quick response to input*

+3 marks *real time requirements matches definition of latency*

Q8 (12 Marks) Performance Analysis*(estimated time: 15 minutes)*

After more than a decade of success developing professional-grade high-performance Waterluvian filters, your company without a clever name has developed a low-cost consumer-grade Waterluvian filter. The first version of the consumer-grade Waterluvian filter (Cv1) has 25% lower performance and 5% higher area than your professional model (Pro). You are now working on version 2 of the consumer filter (Cv2). If Cv2 has the same area as Cv1, how much will you have to increase the performance from Cv1 to Cv2 for the optimality of Pro to be 10% more than the optimality of Cv2?

NOTES:

1. Optimality is measured as performance/area.
2. To earn part marks, explain your method for solving the problem and show the key equations that you use.

Answer:

$$\begin{aligned}
 P_{cv1} &= (1 - 0.25) \bullet P_{pro} \\
 A_{cv1} &= 1.05 \bullet A_{pro} \\
 A_{cv2} &= A_{cv1} \\
 O_{pro} &= 1.10 \bullet O_{cv2} \\
 O &= P/A
 \end{aligned}$$

Convert optimality ratio to performance ratio

$$\begin{aligned}
 O_{pro} &= 1.10 \bullet O_{cv2} \\
 \frac{P_{pro}}{A_{pro}} &= 1.10 \bullet \frac{P_{cv2}}{A_{cv2}}
 \end{aligned}$$

We want to solve for P_{cv2}/P_{cv1}

$$\frac{P_{cv2}}{P_{cv1}} = \dots$$

Only one equation mentions P_{cv2}

$$\frac{P_{pro}}{A_{pro}} = 1.10 \cdot \frac{P_{cv2}}{A_{cv2}}$$

$$P_{cv2} = A_{cv2} \cdot \frac{1}{1.10} \cdot \frac{P_{pro}}{A_{pro}}$$

$$\frac{P_{cv2}}{P_{cv1}} = A_{cv2} \cdot \frac{1}{1.10} \cdot \frac{P_{pro}}{A_{pro}} \cdot \frac{1}{P_{cv1}}$$

Only one equation mentions P_{pro}

$$P_{cv1} = (1 - 0.25) \cdot P_{pro}$$

$$P_{pro} = \frac{P_{cv1}}{0.75}$$

Only one equation mentions A_{pro}

$$A_{cv1} = 1.05 \cdot A_{pro}$$

$$A_{pro} = \frac{A_{cv1}}{1.05}$$

$$\frac{P_{cv2}}{P_{cv1}} = A_{cv2} \cdot \frac{1}{1.10} \cdot \frac{P_{cv1}}{0.75} \cdot \frac{1.05}{A_{cv1}} \cdot \frac{1}{P_{cv1}}$$

Simplify and use $A_{cv1} = A_{cv2}$

$$\frac{P_{cv2}}{P_{cv1}} = \frac{1}{1.10} \cdot \frac{1}{0.75} \cdot 1.05$$

$$= 1.27$$

A 27% performance increase from CV1 to CV2 is needed.

Marking:

- +4 marks Initial equations
- +4 marks Goal is to solve for P_{cv2}/P_{cv1}
- +4 marks Strategy and algebra
- 1 mark Other mistakes (each mistake)

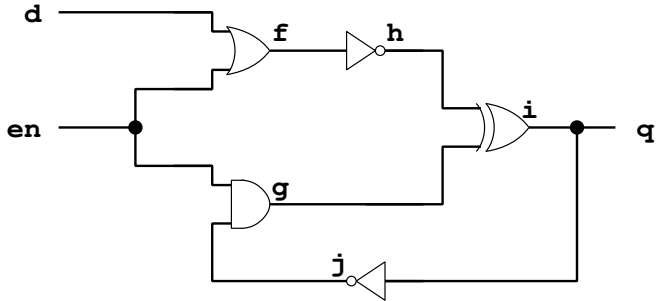
Q9 (12 Marks) Latch Analysis

(estimated time: 15 minutes)

Does the circuit below behave like a latch?

If yes, answer whether the latch is active high or active low; and calculate the clock-to-Q, setup, and hold times.

If no, choose locations to add inverters to make the circuit act as a latch. (For example, "add 2 inverters between i and q"). The goal is to add the minimum possible number of inverters.

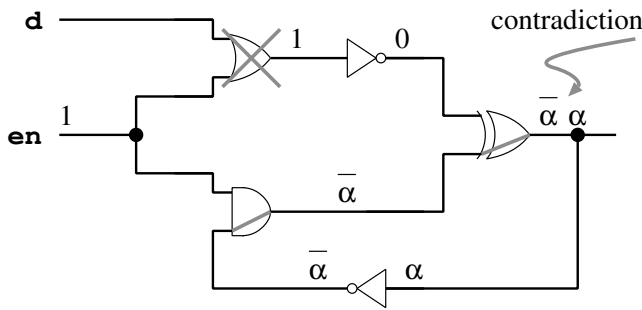


Answer:

If said that latch is incorrect

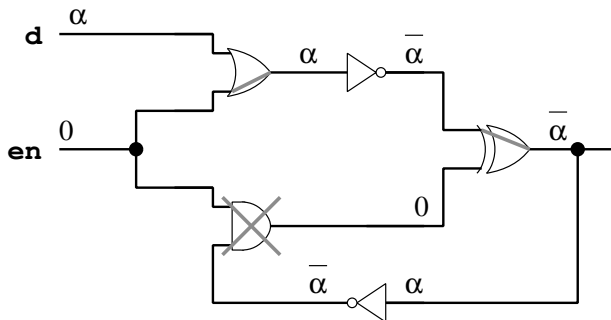
The circuit is not a latch, because:

- Test with $en='1'$



The store loop is enabled, but has an odd number of inversions.

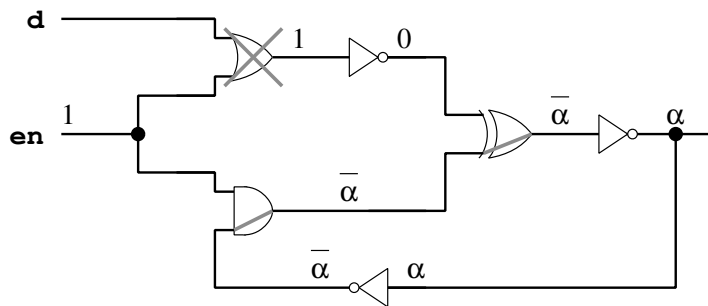
- Test with $en='0'$



The load-path is enabled, but has an odd number of inversions

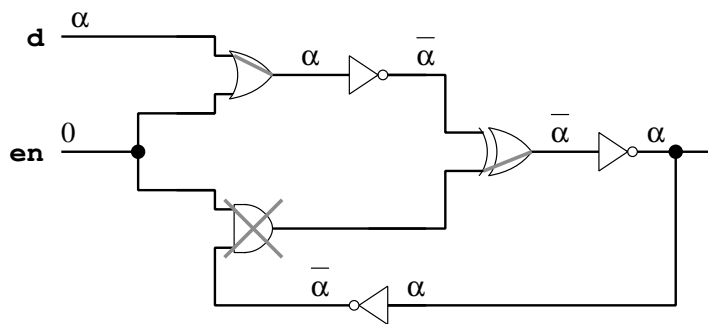
Add a latch after *i*. (Other correct answers are listed below)

Store-mode:



Even number of inversions.

Load-mode:



Even number of inversions.

The path $en \rightarrow \text{store-enable} \rightarrow \text{join}$ is shorter than the path $en \rightarrow \text{load-enable} \rightarrow \text{join}$. Therefore, the latch is correct.

List of correct answers:

- Answer 1: Add 1 inverter between *i* and wire fork
- Answer 2: Add 1 inverter between *f* and *h* (or between *h* and *i*)
- Answer 3:
 - Add 1 inverter between *d* and *f*
 - Add 1 inverter between wire fork after *i* and *j*, or between *j* and *g*.
- There might be additional correct answers

Incorrect answers:

- (9 marks) Add 1 inverter between *g* and *i*: $en \rightarrow \text{load-enable} \rightarrow \text{join}$ is not faster than $en \rightarrow \text{store-enable} \rightarrow \text{join}$.
- (6 marks) Add 1 inverter between *en* and wire fork: doesn't affect inversions on load-path or store-loop.

Marking:

+3 marks *Load-path and store-loop are mutually exclusive*

+3 marks *Load-path has even number of inversions*

+3 marks *Store-loop has even number of inversions*

+3 marks *The path $en \rightarrow store\text{-}enable \rightarrow join$ is shorter than the path $en \rightarrow load\text{-}enable \rightarrow join$*

Answer:

If said that the latch is correct:

Active Lo 2 marks

Clock to Q 3 2 marks






Setup 4 2 marks

Hold 0 2 marks

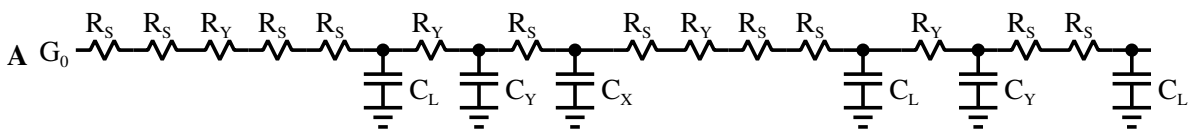
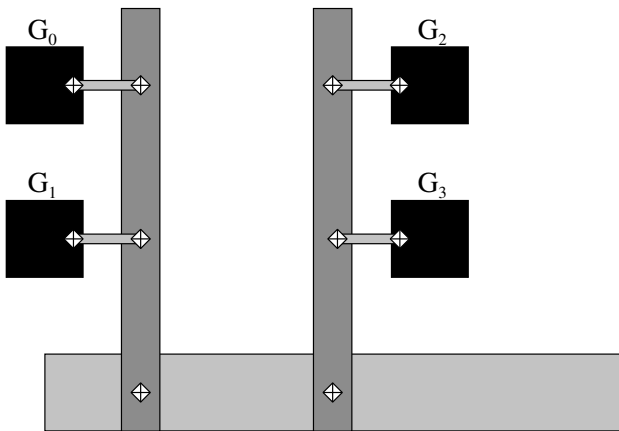
Q10 (10 Marks) Elmore Delay

(estimated time: 20 minutes)

Which one of the RC-networks (A-E) is the best model for the layout below?

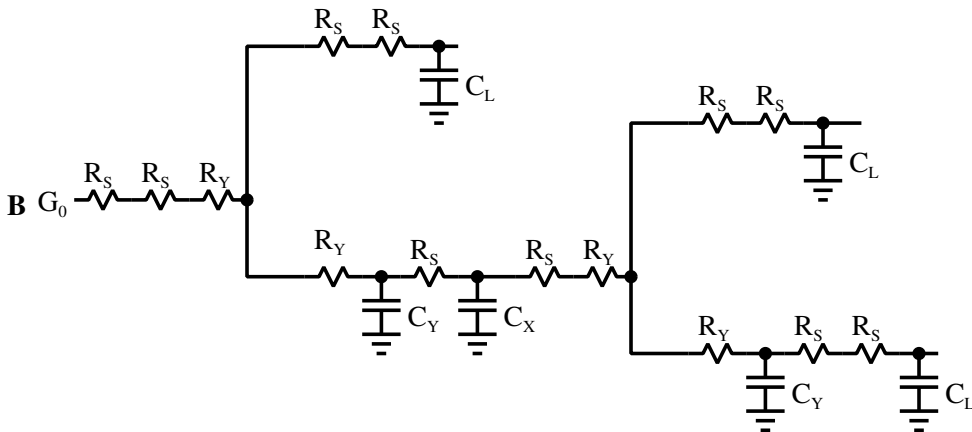
Symbol	Description	Capacitance	Resistance
	Interconnect level 3	C_X	0
	Interconnect level 2	C_Y	proportional to distance
	Interconnect level 1	0	0
	Gate	C_L	0
	Switchbox	0	R_S

The resistances and capacitances for the physical layout are:



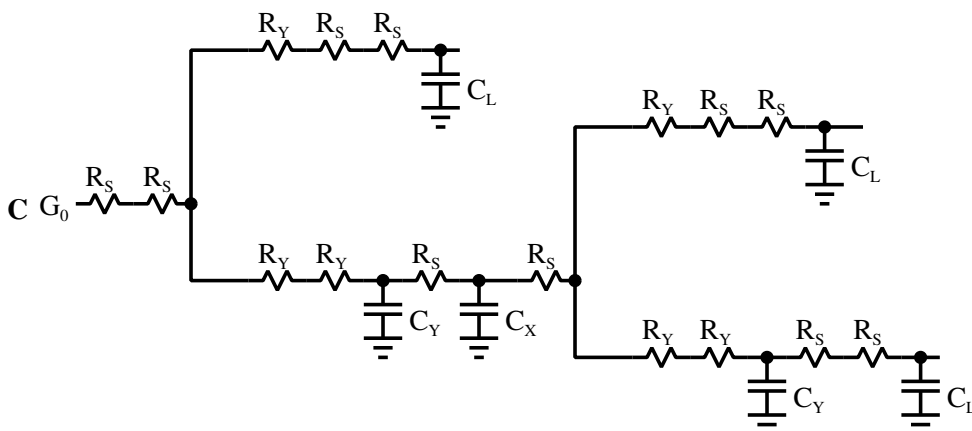
Answer:

Incorrect, because the R_S resistances for G_1 and G_3 are upstream from all of the remaining circuitry. These R_S resistances should have only their local C_L as a downstream capacitance.

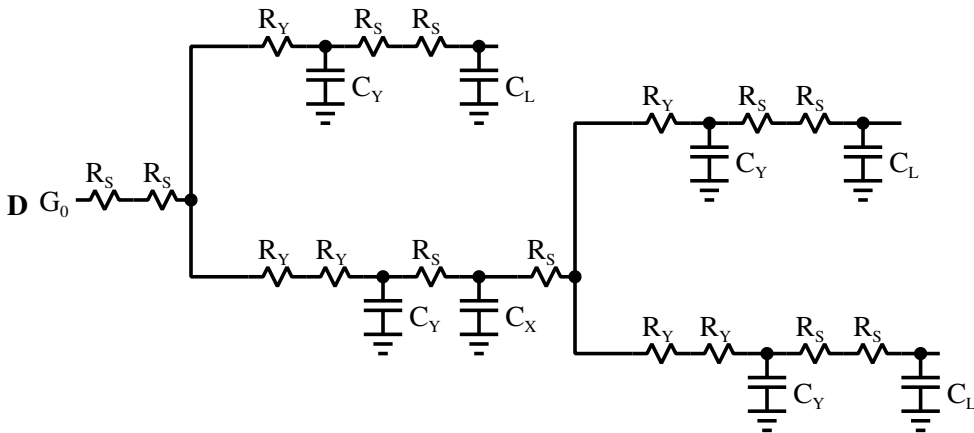
**Answer:**

Correct:

- *Each R_S has only its local R_L as a downstream capacitance.*
- *The R_Y resistances have one C_Y downstream capacitance for their "Y" wire.*

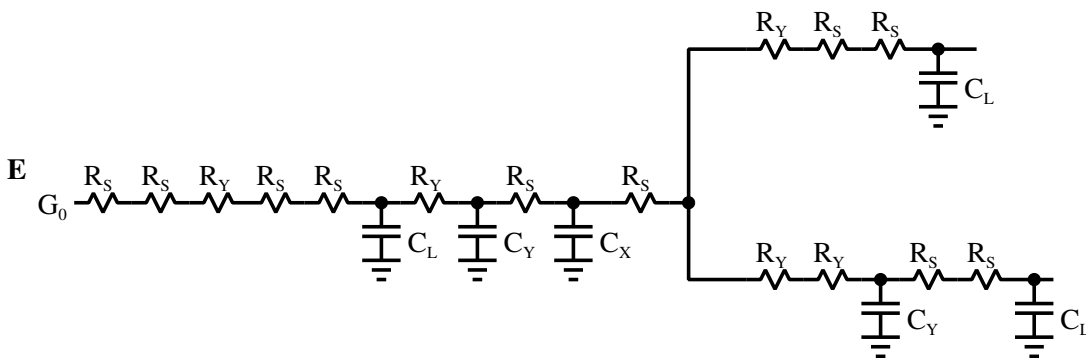
**Answer:**

Incorrect. There are two R_Y resistances for each "Y" wire.



Answer:

Incorrect. There are two R_Y resistances and two C_Y capacitances for each “Y” wire.



Answer:

Incorrect, because the R_S resistances for G_1 is upstream from all of the remaining circuitry. This R_S resistance should have only its local C_L as a downstream capacitance.

Marking:

10 marks *B*

8 marks *C*

6 marks *D*

3 marks *E*

0 marks *A*