Please print in pen:
Waterloo Student ID Number:

WatIAM/Quest Login Userid:

UNIVERSITY OF WATERLOO

Examination
Midterm
Spring 2025
ECE 327

Times: Wednesday 2025-06-18 at 13:30 to 15:00 (1:30 to 3PM)
Duration: 1 hour 30 minutes (90 minutes)
Exam ID: 6239829
Sections: ECE 327 LEC 001,002
Instructors: Andrew Boutros

Special Materials

Candidates may bring only the listed aids.

- Calculator - Non-Programmable
- Study Notes - Double-Sided 8.5x11 (One handwritten sheet)

Marking Scheme (For Examiner Use Only)

Question	Marks Given	Out Of	Marker's Initials
1		20	
2		20	
3		10	
4		10	
5		14	
6		20	
7		6	
Total		100	

Exam instructions:

- This exam is closed book. You are only allowed a letter-size double-sided handwritten cheat sheet.
- Use of a non-programmable calculator is permitted.
- No other electronic aids are permitted (e.g., laptops, tablets, phones).
- Turn off all communication devices prior to the start of the examination. During the examination, the use of any communication devices is prohibited. Taking a communication device to a washroom break is not allowed and will be interpreted as an academic offense.
- Place all knapsacks, backpacks, and bags at the front of the examination room or beneath your table.
- The exam consists of 7 questions with multiple parts.
- The exam duration is 90 minutes and there are a total of 100 marks. Some marks are easier to earn than others. Read the paper carefully and use your time wisely.
- Verify that your name and student ID number is on the cover page of your examination.
- Write your answers directly on the question sheets in the space provided. You may use either pen or pencil to answer questions. Your exam paper will be scanned and graded electronically. Answers that are too light to read or smudged will be assigned a grade of zero.
- If you need more space to complete an answer, you may be writing too much. However, if you need extra space, use the extra blank scratch pages at the end of the exam. Clearly label the question and indicate that you have done so in the main solution space of the question.
- Do not take apart the exam paper sheets.
- After reading and understanding the instructions, sign your name in the space provided below.

Signature

Question 1: SystemVerilog Basics [20 marks]

Read the following SystemVerilog code then answer the questions on the following pages.

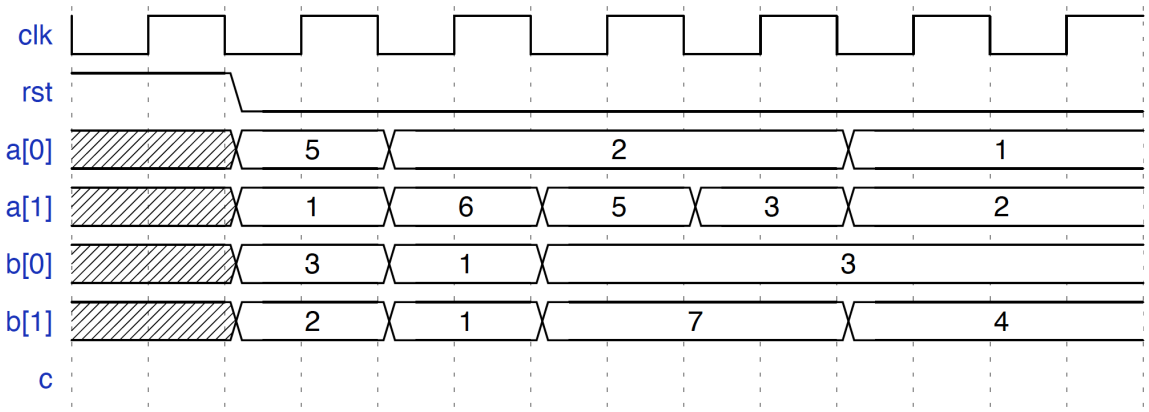
```
module my_circuit # (
      parameter K = 4,
      parameter N = 8,
      parameter M = 8,
      parameter L = 8
)(
      input clk,
      input rst,
      input [N-1:0] a [0:K-1],
      input [N-1:0] b [0:K-1],
      output logic [M-1:0] c
);


logic [L-1:0] tmp1 [0:K-1];
logic [M-1:0] tmp2;


integer i, j;


always_comb begin
      for (i = 0; i < K; i = i + 1) begin
            tmp1[i] = a[i] * b[i];
      end
end


always_comb begin
      tmp2 = 0;
      for (j = 0; j < K; j = j + 1) begin
            tmp2 = tmp2 + tmp1[j];
      end
end


always_ff @ (posedge clk) begin
      if (rst) c <= 'd0;
      else c <= tmp2;
end

endmodule
```

(a) Inside the frame below, draw a diagram of the circuit generated when elaborating this SystemVerilog code with the parameter K set to 4. **[6 marks]**

(b) For an instance of this module with parameters K = 4 and N = 8, what are the minimum values of parameters M and L for this circuit to be functionally correct? **[2 marks]**

M = _____ L = _____

(c) Complete the waveform below based on your understanding of the functionality of the circuit described above when parameter K is set to 2 and N is set to 3. **[4 marks]**



(d) Complete the body of the testbench code below that would generate the waveform shown in part (c) of the question. Assume that the clock period is 2 ns. **[8 marks]**

```
`timescale 1ns/1ps

module my_circuit_tb();

localparam CLK_PERIOD = 2;

logic clk, rst;
logic [2:0] a [0:1];
logic [2:0] b [0:1];
logic [7:0] c;

my_circuit # (
      .K(2), .N(3)
) dut (
      .clk(clk), .rst(rst), .a(a), .b(b), .c(c)
);









// Continue on the next page
```

```
endmodule
```

Question 2: FSM Design [20 marks]

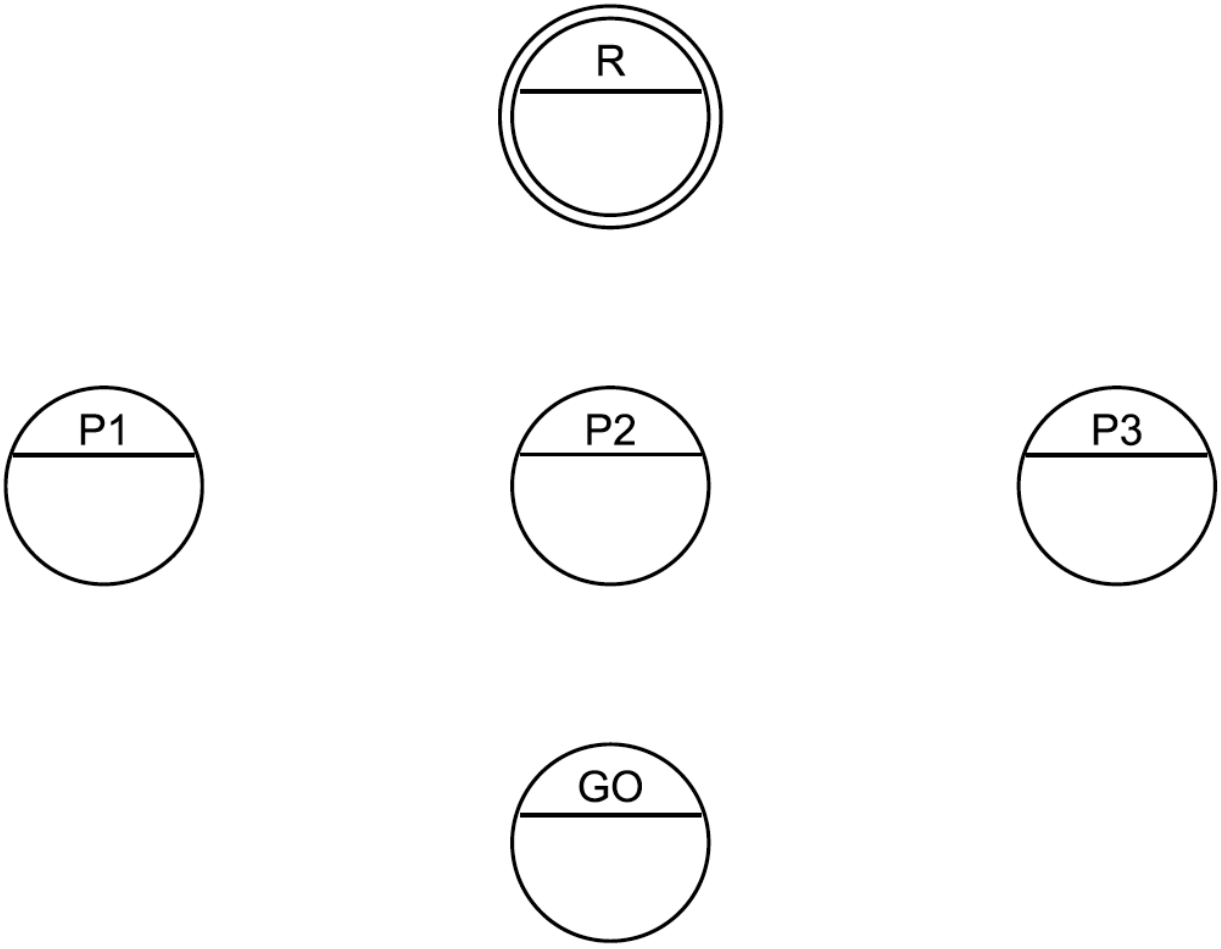
ECE 327 students are organizing a game night, and the main event is the "Battle of the Bits" game! In this game, three players take turn in a round robin fashion (player 1, player 2, player 3, player 1, …) to answer hardware design questions. For their turn, each player chooses a category of questions: Easy, Medium, or Hard. An easy question is worth 1 point, a medium question is worth 2 points, and a hard question is worth 3 points. If a player answers a question correctly, the question points are added to their score. If they answer incorrectly, the question points are deducted from their score. However, a player cannot score less than 0 if they keep answering incorrectly. The game ends when a player scores 30 points or more.

You are asked to implement a finite-state machine (FSM) that keeps track of the scores of the 3 players in this game and declares a winner when the game ends. This FSM takes as inputs a 2-bit value "`q_status`" and a 3-bit one-hot value "`difficulty`". It outputs three 6-bit values (`score1, score2, score3`) that represent the current scores of all players, and a 2-bit value "`winner`" that specifies the ID of the winning player. The tables below list the meanings of different values of the `q_status`, `difficulty`, and `winner` signals.

Inputs				**Outputs**	
`q_status`		**`difficulty`**		**`winner`**	
Value	**Meaning**	**Value**	**Meaning**	**Value**	**Meaning**
2'b00	Waiting for answer	3'b001	Easy	2'b00	No winner yet
2'b01	Correct answer	3'b010	Medium	2'b01	Player 1 wins
2'b10	Incorrect answer	3'b100	Hard	2'b10	Player 2 wins
2'b11	Never occurs	Other combinations never occur		2'b11	Player 3 wins

The FSM consists of 5 states: R, P1, P2, P3, and GO that represent the initial reset, player 1 turn, player 2 turn, player 3 turn, and gameover states, respectively.

(a) Complete the FSM diagram on the following page by specifying the appropriate transitions, transition conditions, state actions, and transition actions needed to implement the required functionality. If transition conditions or actions are too long to write on the diagram, you can label conditions (e.g., C1, C2) and label actions (e.g., A1, A2) on the diagram and define them in the white space on the same page. **[13 marks]**



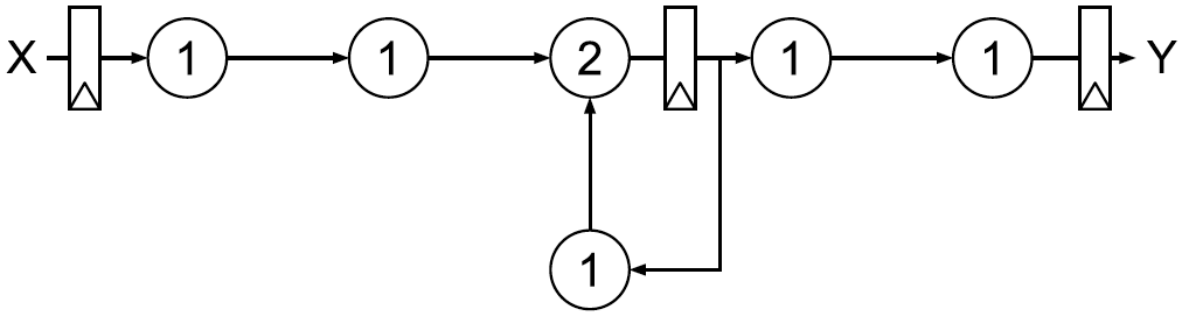
(b) Complete the SystemVerilog code below to implement the FSM you designed in part (a) of the question. **[7 marks]**

```
module battle_of_the_bits_fsm (
      input clk,
      input rst,
      input [1:0] q_status,
      input [2:0] difficulty,
      output logic [5:0] score1,
      output logic [5:0] score2,
      output logic [5:0] score3,
      output logic [1:0] winner
);

// Declarations start here









// Declarations end here


always_ff @ (posedge clk) begin

















end
// Continue on next page
```

```
always_comb begin: state_decoder









end

always_comb begin: output_decoder









end
endmodule
```

Question 3: Pipelining [10 marks]

The following diagram shows a digital circuit with an input X and output Y. The circles represent different combinational operators and the numbers in them are the propagation delays of these operators in nanoseconds. Assume all wire delays are zero.

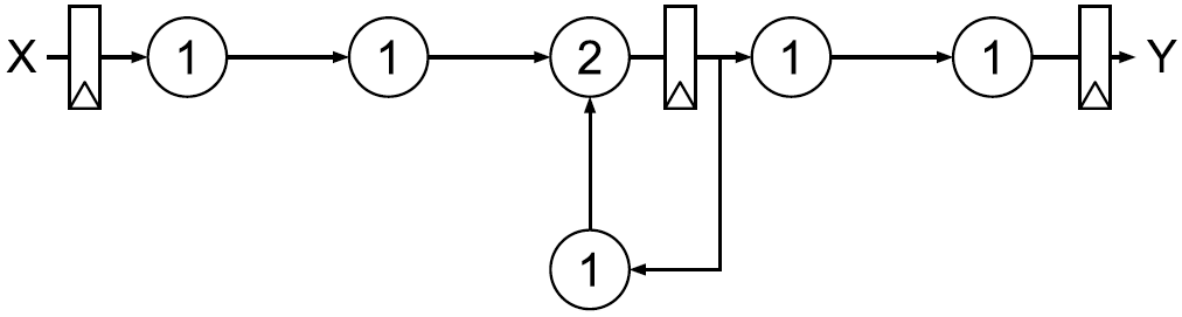


(a) How long is the critical path delay of this circuit? And what is its maximum safe operating frequency? **[4 marks]**

The critical path delay is _____ nanoseconds.

The maximum safe operating frequency is _____ MHz.

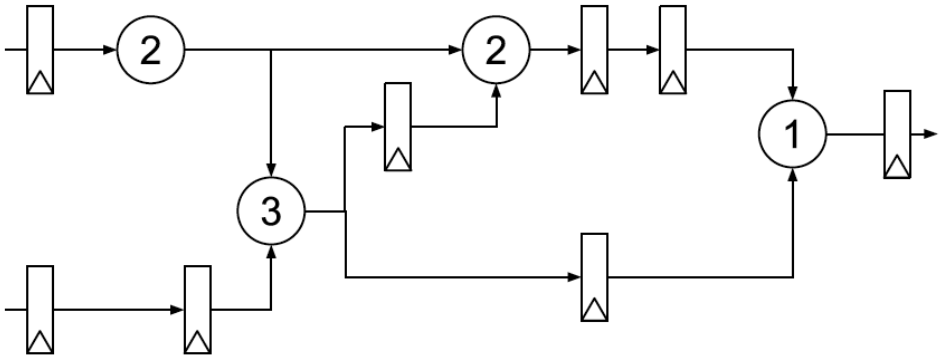
(b) Assume you have only a single stream of input data X. If you would like to maximize the frequency of this circuit by adding as few pipeline registers as possible while maintaining correct functionality, how many registers do you need and where should they be added? Indicate that by drawing the register(s) on the diagram below. What is the maximum safe operating frequency in that case? **[6 marks]**



The maximum safe operating frequency after pipelining is _____ MHz.

Question 4: Throughput & Latency [10 marks]

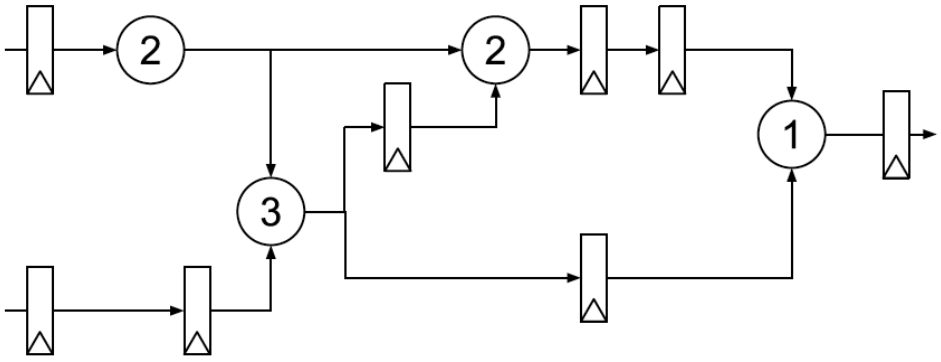
The following diagram shows a digital circuit. The circles represent different combinational operators and the numbers in them are their propagation delays in nanoseconds. Assume all wire delays are zero and each operator counts as a single operation.



(a) Fill in the following blanks: **[5 marks]**

1. The critical path delay of this circuit is _____ nanoseconds.
2. Assuming that the clock period is set to be equal to the critical path delay, the latency of this circuit is _____ nanoseconds.
3. The throughput of the circuit is limited to _____ operations per second to guarantee correct functionality.

(b) Add the necessary pipeline registers to maximize the throughput of the circuit by drawing them on the diagram below. Assume that you cannot pipeline the internals of the combinational operators. What is the throughput of the circuit in this case? **[5 marks]**



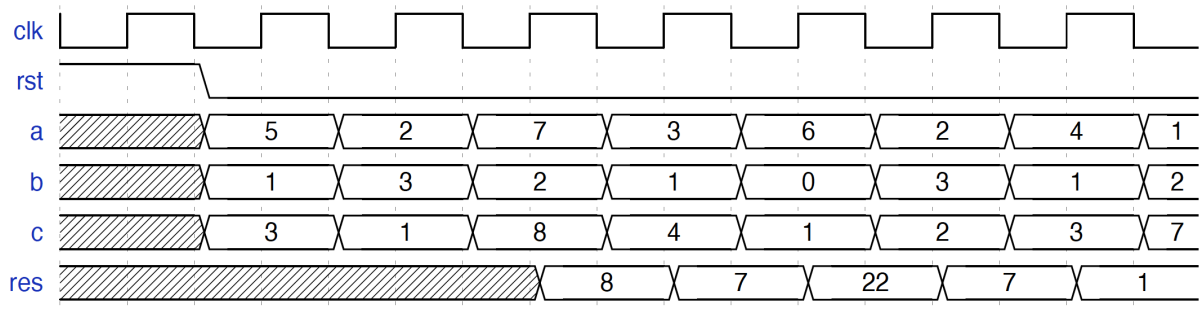
The throughput of the circuit after balancing the pipeline is _____ operations per second.

Question 5: Retiming [14 marks]

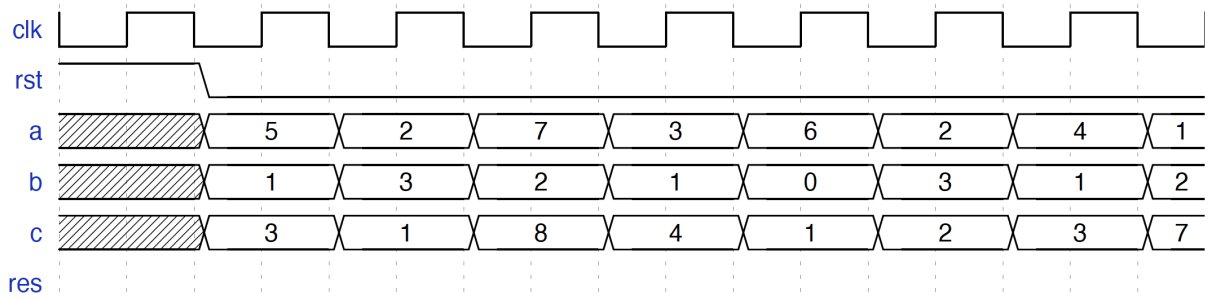
Consider an arbitrary feed-forward circuit with no feedback loops, three registered inputs (a, b, c) and one registered output (res). There is a single pipeline stage in an arbitrary location in the circuit between the input and output registers. When the circuit is retimed, its maximum safe operating frequency is improved from 100 MHz to 150 MHz.

(a) The waveform given below shows the operation of the circuit before retiming. Complete the waveform to show the operation of the circuit after retiming. **[4 marks]**

Before Retiming



After Retiming



(b) For the metrics below, write one of the following symbols between brackets: "↑" or "↓" or "–" or "?", such that "↑" means this metric increased, "↓" means this metric decreased, "–" means this metric stayed the same, and "?" means cannot tell if this metric increased, decreased, or stayed the same from the given information. **[10 marks]**

a. Throughput in operations per second ( )
b. Latency in nanoseconds ( )
c. Number of pipeline stages in the circuit ( )
d. Number of flip-flops used in the circuit ( )
e. Critical path delay ( )

Question 6: Latency-insensitive Design [20 marks]

For each one of the following statements, specify whether it is True or False by writing "T" for True or "F" for False in the table below.

1. As semiconductor process technology advances, resistance of a fixed length of wire decreases and its capacitance increases resulting in slower wire propagation delay.

2. Latency-sensitive design means that modules are expecting inputs to arrive at specific cycles. This will always result in incorrect functionality and therefore should not be used when designing any digital system.

3. Latency-insensitive protocols assume that the internal logic of a module is purely combinational.

4. Relay stations in latency-insensitive design are used to improve throughput without affecting functionality.

5. Latency-insensitive designs are guaranteed to be deadlock-free.

6. Latency-insensitive design makes it easier to reuse IPs in many different systems implemented using different process technologies or FPGA devices.

7. A latency-insensitive protocol allows data to be dropped if it is not consumed within a specific number of clock cycles.

8. Adding relay stations to latency-insensitive channels in a design can break functionality if not properly balanced across all the channels providing inputs to a certain module.

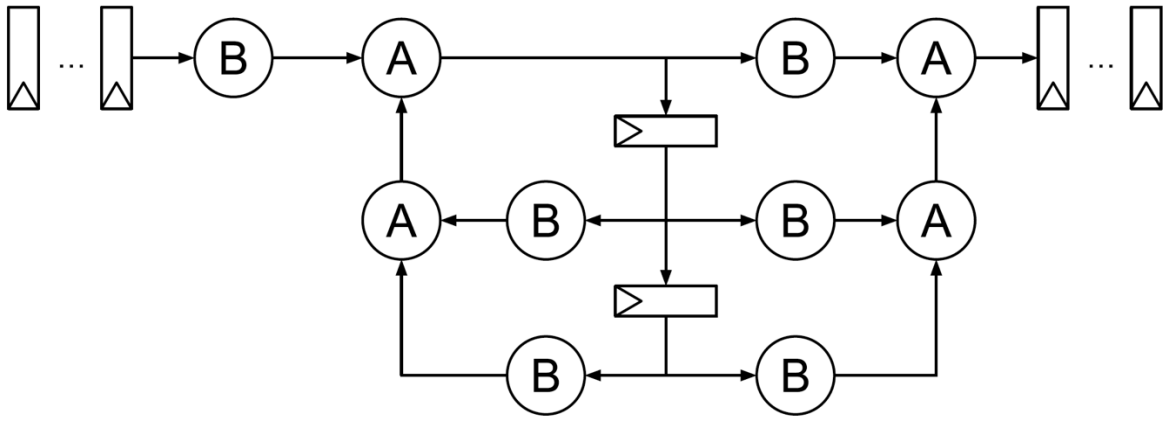
9. The throughput of a latency-insensitive design is the average throughput of its modules.

10. Using a latency-insensitive design style eliminates the need for timing analysis during the physical design process.

1	2	3	4	5	6	7	8	9	10

Question 7: More Retiming [6 marks]

In the circuit shown below, A and B are two combinational operators with delay 2 ns and 4 ns, respectively. Assume all wire delays are zero.



(a) What is the sequence of operations on the critical path of this circuit? And how long is the critical path delay? **[3 marks]**

Sequence of operations on the critical path:

Critical path delay is _____ ns.

(b) Assuming there is an unlimited number of registers available at the input and output ports of the circuit, retime the circuit to maximize its frequency while maintaining correct functionality. Assume that you cannot pipeline the internals of an operator. After retiming, what is the sequence of operations on the critical path? And how long is the critical path delay? **[3 marks]**

Sequence of operations on the critical path after retiming:

Critical path delay is _____ ns.

(Extra Blank Scratch Page)

(Extra Blank Scratch Page)

(Extra Blank Scratch Page)