CSE 370, Autumn, 2007, Exam 2

Please do not turn the page until instructed to do so.

Rules:

- Please remove everything from your desk area except one sheet of notes and whatever pens/pencils you want to use.
- Please stop working promptly at 11:20.
- If you rip the pages apart, please staple them back together when you are done.


Advice:

- The exam should have 9 pages; check before you start.
- Read questions carefully before you start writing.
- Write down partial solutions for partial credit.
- There are 100 points on the exam distributed unevenly; try to distribute your effort roughly according to point value.
- The questions are not necessarily ordered according to difficulty. Skip around to find parts that are easy for you.
- If you have questions, ask.
- The last two exercises are "challenge exercises". They do not count towards your normal class score at all. If you complete them well, it could have a small effect when assigning final grades at the end of the quarter. Do not work on them unless you are 100% sure you are done with the rest of the exam.

Grading Summary	
1:	/ 20
2:	/ 20
3:	/ 15
4:	/ 10
5:	/ 15
6:	/ 20
Total:	/ 100

1. Draw waves and point out the hazards (20 points)

In the following circuit, the gates are marked with the number of time units of delay they have. Draw the values of X, Y, Z and F as a function of time. This circuit has a dynamic hazard. Point out in the wave for F the glitch caused by this hazard. There is also a static hazard that results in a glitch in one of the internal wires. Point out the glitch and <u>briefly</u> describe the kind of change that could be made to remove the hazard. Assume that A, B and C had their initial "time 0" values for a long time before time 0. Also assume that B and ¬B change at the same time.
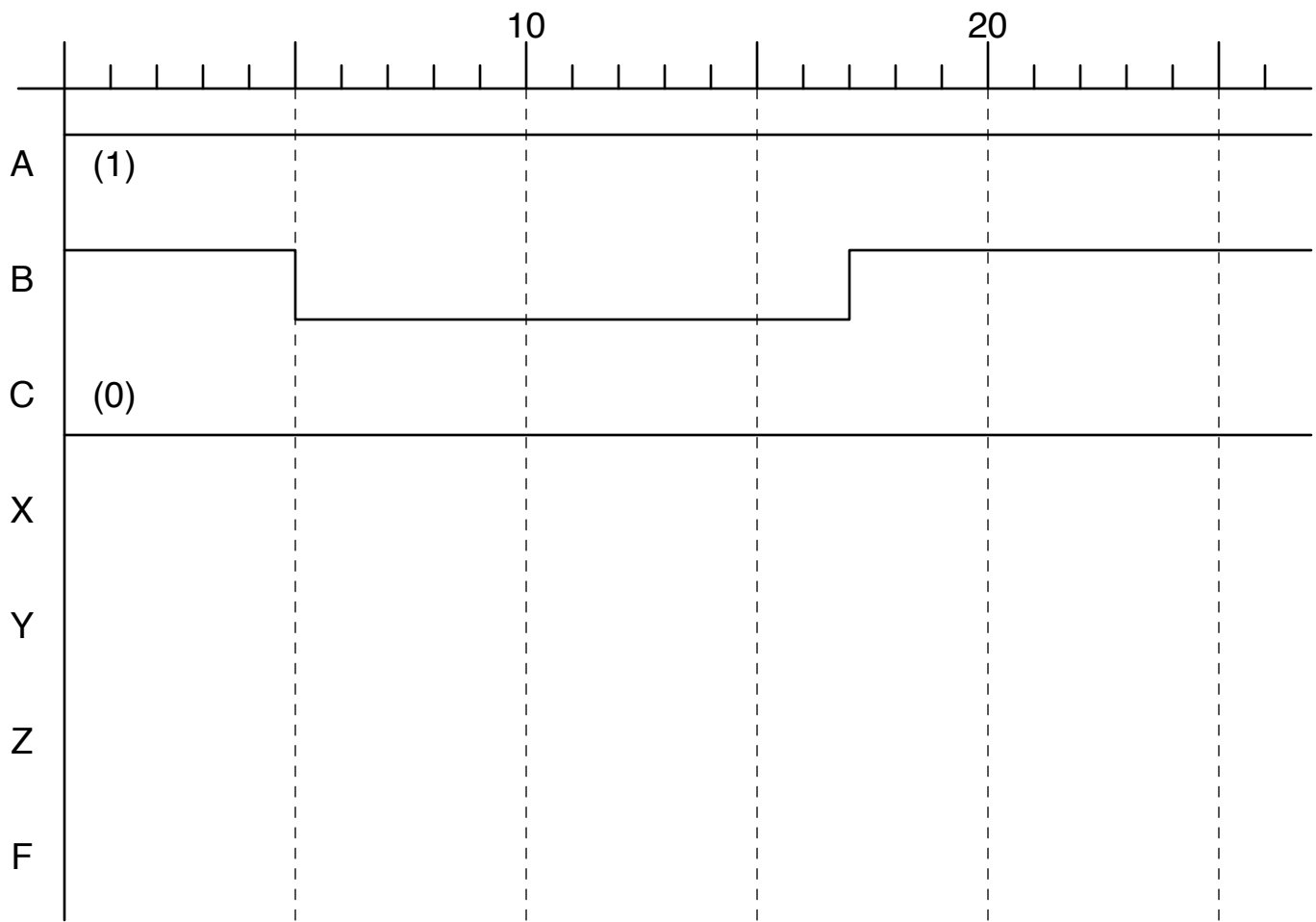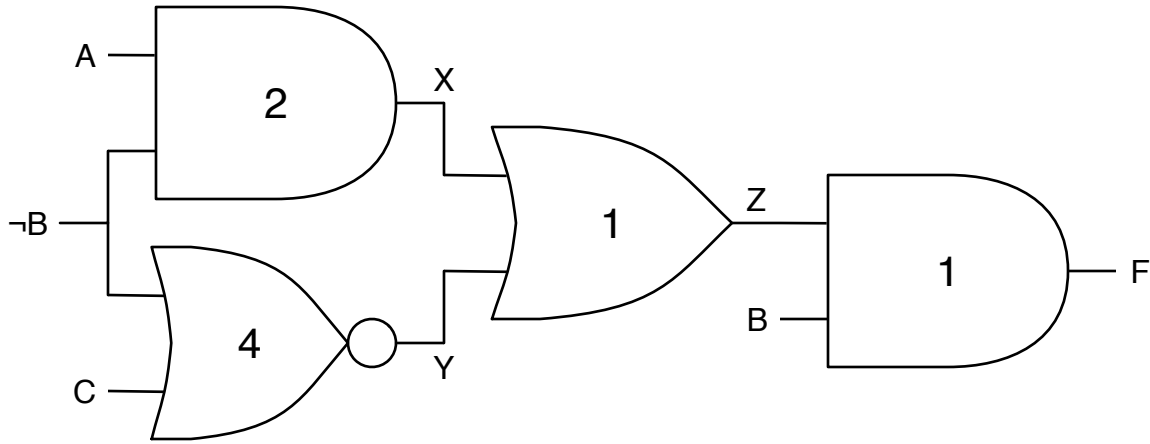
2. Multi-level circuit design (20 points)

Design a circuit for the following function that uses as few total gate inputs as possible. You may only use inverters, AND, NAND, OR and NOR gates. Hint: There are lots of paths to take, in optimizing circuits, but it may be useful to remember that Benjamin is a fan of Mr. DeMorgan.

CD \ AB	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	0	1
10	0	0	1	0

3. Design an absolute difference unit (15 points)

Design a circuit that calculates the absolute value of the difference between two 8-bit 2's complement numbers. You can assume the two inputs are within a small enough range that subtracting either one from the other does not result in overflow. A more formal definition of the function you are to implement: if A < B, then B - A, else A - B.
You may use the following components:

Name	Function	Size	Delay
8-bit 2's complement subtracter	O = X - Y	50	16
8-bit 2's complement inverter	O = -X	25	12
8-bit 2's complement less than	O = X < Y	50	16
8-bit 2-1 multiplexer	O = Z ? X : Y	24	2
Individual 2-input gate		1	1

Make your circuit as small and fast as you can. (There are at least two equally "correct" answers.) If you want to draw an 8-bit connection as a single line, but access individual wires as well, clearly label which bits you are accessing.

4. Find the critical path (10 points)

In the circuit diagram below, the boxes represent flip-flops and the clouds represent some combinational gates. The labels on the clouds define what their delay is in Generic Time Units (GTU). Highlight the critical path.
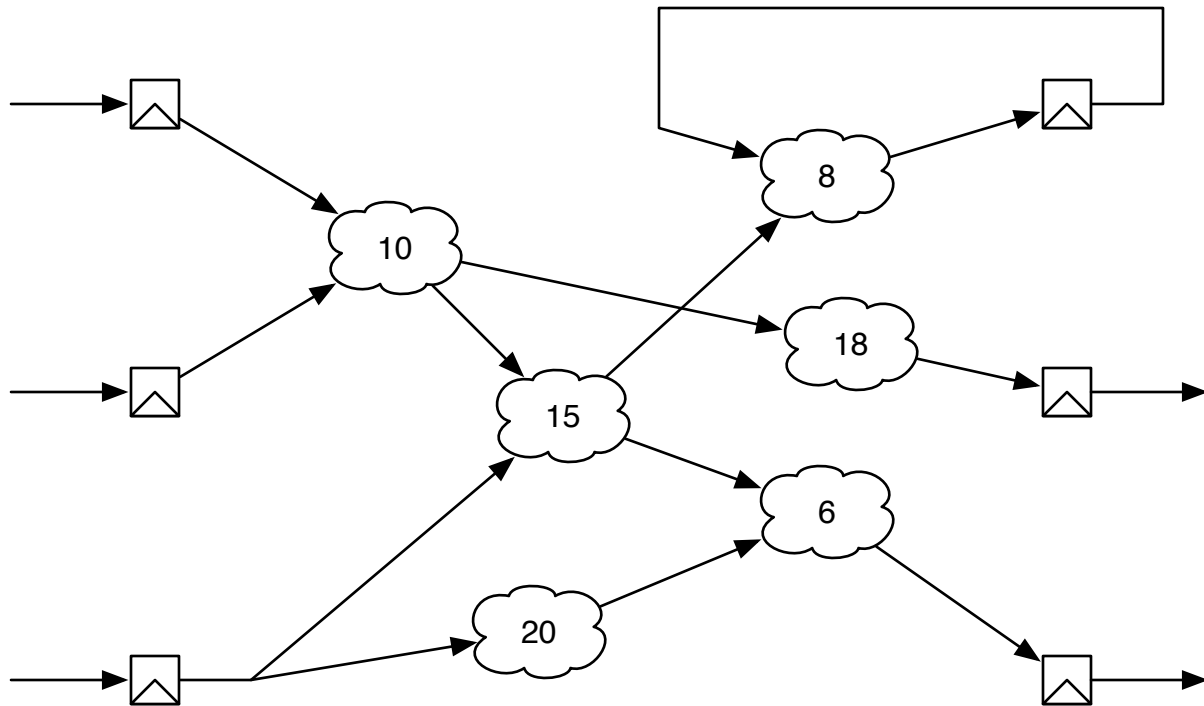
Assuming the flip-flops are all connected to the same periodic clock signal, and have a setup time of 4 GTU, a hold time of 1 GTU, and a propagation delay of 2 GTU, what is the smallest clock period that will allow this circuit to function properly?

5. More Verilog dos and don'ts (20 points)

The following Verilog code is intended to implement the circuit on the right. It has 5 substantial errors in it. Point out the errors and <u>briefly</u> describe them.
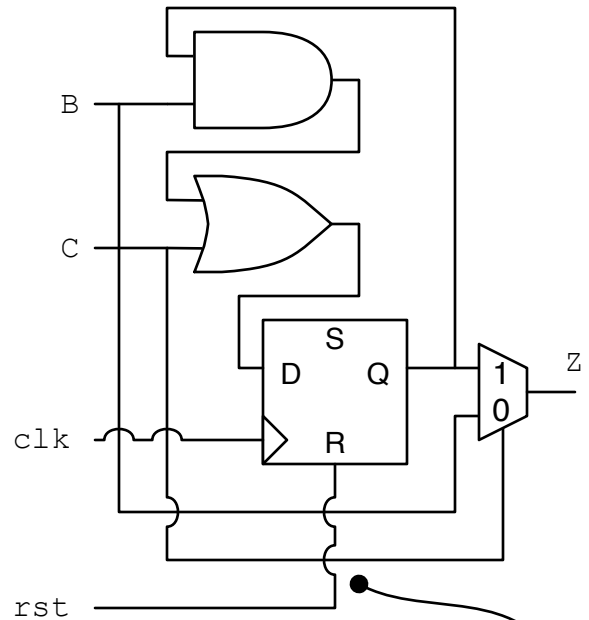
```
module bad(clk, rst, B, C, Z);
   input clk, rst, B, C;
   output Z;
   reg Q, nextQ, Z;

   always @ (posedge clk or B) begin
     if (rst)
       Q <= 1;
     else
       Q <= nextQ;
     Z = C ? Q : B;
   end

   always @ (B or C)
   begin
     if (C)
       nextQ <= 1;
     else begin
       if (B)
         nextQ <= Q;
       else
         nextQ <= 0;
     end
   end
endmodule
```

B

C

clk

rst

S

D   Q

R

1
0

Z

In the version of this exam that was actually taken, this logic was incorrect. That was an exam writing error. If the mistake was pointed out, full credit was given.

(These "hops" were not in the original version of the exam)

6. Design a state machine Moore and Mealy-style (20 points)

Draw two state machines. Both take a single bit input and produce a single bit output. Both output a 1 when the last two inputs were 0, and then 1. One of the machines should be Moore-style, and one should be Mealy-style. For both, use as few states as you can.