

Name: _____

Perm #: _____

Lab Section TA: _____

ECE 152A-Winter 2005

Prof. Volkan Rodoplu

FINAL EXAMINATION

INSTRUCTIONS:

1. READ THIS PAGE THOROUGHLY WHEN YOU RECEIVE IT, BUT DO NOT START TURNING TO THE OTHER PAGES UNTIL YOU ARE INSTRUCTED TO DO SO.
2. WHENEVER INDICATED, YOU MUST WRITE YOUR ANSWERS ON THE ANSWER LINES PROVIDED IN CERTAIN PROBLEMS. NO PARTIAL CREDIT WILL BE GIVEN ON THESE PROBLEMS.
3. On other problems, PARTIAL CREDIT will be given only to true statements that make progress towards the correct answer. Partial credit may be given to correct reasoning in developing structures such as K-maps, truth tables, state diagrams and state tables in which the variables are clearly labeled. NO partial credit will be given for incorrect statements or statements to which no truth value can be assigned (such as a bunch of numbers or algebraic expressions). NO partial credit will be given for statements that use symbols that the problem statement or you have not defined. NO credit will be given for any work that is not clearly labeled with the part and problem number to which this work provides an answer.
4. All the exam rules in the course syllabus apply to this exam.
5. You may remove the staple from the exam pages, if is more convenient. We will provide a stapler at the end of the exam.
6. There are 20 pages in this exam (printed on two sides). When you are instructed to start, first check that you have all the pages.
7. There is a total of 100 points on this exam.
8. Write the solution of each problem in the space provided for that problem.

PROBLEM # 1 ELEMENTARY PROBLEMS [8 points]

In this problem, write down your answers in the Answer Box provided at the bottom of the page.

TRUE/FALSE QUESTIONS:

A1. The difference between a Moore and a Mealy machine is that the next state in a Moore machine is a function of only the current output whereas in a Mealy machine, the next state is a function of both the current input and the current output.

A2. The fundamental difference between a positive edge-triggered flip-flop and a positive level-sensitive latch is that the flip-flop samples its data input at the rising clock edge whereas the latch is transparent to its input while the clock is high.

A3. The fundamental difference between Verilog and C++ is that Verilog is a concurrent programming language whereas C++ is a sequential programming language.

B1. For a negative level-sensitive D latch, the data input to the latch must be stable:

- a) a D-to-Q delay after the rising edge of the clock
- b) a set-up time before the falling edge of the clock
- c) a hold time after the falling edge of the clock
- d) a set-up time before the rising edge of the clock
- e) None of the above

Answer Box:

A1 _____ (Write "TRUE" or "FALSE") (2 points)

A2 _____ (Write "TRUE" or "FALSE") (2 points)

A3 _____ (Write "TRUE" or "FALSE") (2 points)

B1 _____ (Write "a", "b", "c", "d", or "e") (2 points)

PROBLEM # 2 FINITE STATE MACHINE DESIGN [20 points]

Design a finite state machine (FSM) that has a 1-bit data input X , a 1-bit synchronous control input called “Reset” and a 1-bit output Y , and operates as follows: The machine observes an incoming bitstream, **one bit at a time** (the observed bit is denoted by X). If Reset is asserted, the machine resets to the initial state (i.e. a state in which no bits have been observed thus far). If Reset is not asserted, the machine sets its output Y to 1 if the number of 1’s observed in the bitstream thus far is a multiple of 3, and otherwise, the machine sets Y to 0.

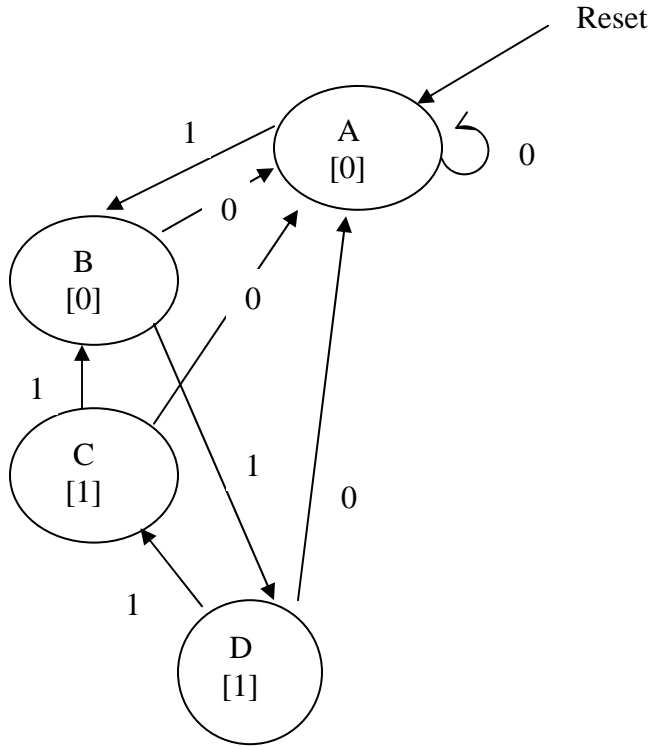
(a) **(10 points)** Draw the state diagram for a MOORE machine implementation of this FSM. Explain clearly how each state is defined. Define your state variables, and label all of the transition arcs. (Aim for the minimum number of states. Designs that use an unnecessarily large number of states will be penalized.)

(b) **(10 points)** Draw the state diagram for a MEALY machine implementation of this FSM. Explain clearly how each state is defined. Define your state variables, and label all of the transition arcs. (Aim for the minimum number of states. Designs that use an unnecessarily large number of states will be penalized.)

PROBLEM # 3 FSMs and VERILOG

[12 points]

Figure 1 shows the state diagram of a finite state machine that has a 1-bit data input X and a 1-bit output Y. Implement this machine as a module using behavioral Verilog, based directly on this state diagram description. (Note: If Reset is asserted, the machine transitions to state A regardless of the value of its data input X.)



- Figure 1 -

Start your work here:

PROBLEM # 4 VERILOG

[15 points]

Write down Verilog modules to solve the following problems. (Note that sections (a) and (b) below are independent problems.)

(a) **(5 points)** Implement in **behavioral Verilog**, a **positive** edge-triggered T flip-flop with **asynchronous** Reset.

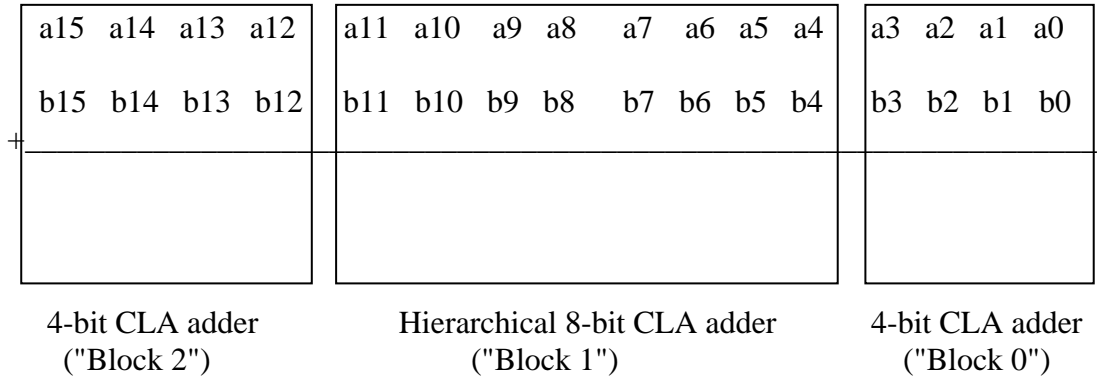
(b) **(10 points)** Implement in Verilog, a comparator circuit that compares two 32-bit numbers A and B, both represented in 2's complement notation. The machine has a 2-bit output Z that is set to [0 0] if $A = B$, is set to [1 0] if $A > B$, and is set to [0 1] if $A < B$.

PROBLEM # 5

ADDERS

[25 points]

Your friend Ace proposes a new adder design that uses “variable-size blocks”. Your job in this problem is to evaluate analytically the delay performance of this new adder, using a 16-bit adder as an example. Figure 3 shows the conceptual diagram of this adder.



- Figure 3 -

In this design, the 16-bit adder is implemented in three consecutive blocks, of sizes 4, 8, and 4 as shown in the figure. **The blocks themselves are connected to each other using ripple carry: That is, the carry-out of Block 0 is fed into the carry-in bit of Block 1, and the carry-out of Block 1 is fed into the carry-in bit of Block 2.**

Each of the blocks 2 and 0 is implemented as a 4-bit carry lookahead adder (CLA). Block 1 is implemented as a hierarchical 8-bit CLA, made up of two 4-bit CLA's, and a carry lookahead unit connecting the 4-bit CLA's. (This internal structure of the hierarchical 8-bit adder is not shown in Figure 3.)

Throughout the delay analysis in this problem, ASSUME THAT THE DELAY OF EACH GATE IS 1, REGARDLESS OF THE FAN-IN OR FAN-OUT OF THAT GATE.

(a) (20 points) Evaluate the worst-case delay of this 16-bit adder. Carefully draw the blocks, their inputs and outputs as a separate figure below. Carefully define all of your variables (e.g. the propagates and generates) and indicate the scope of each variable in your equations) and show these variables on the blocks in your figure. Show when each of the variables in your figure becomes available and how the worst-case delay of the adder is calculated.

Finally, write down your answer for the worst-case delay of this 16-bit adder below:

Answer: _____

(b) (**5 points**) What is the critical path of this 16-bit adder? Give two inputs that achieve the critical delay of the adder. You must fully justify your answer.

PROBLEM # 6**TIMING****[20 points]**

NO PARTIAL CREDIT will be given on this problem. Your answer must be written clearly on the lines indicated. We will check both your answer and your work. If your answer is incorrect, you will get no credit. If your answer is correct, but your derivation is wrong, you will get no credit. Use the following data in this problem:

There are two different types of D flip-flops used in this problem. They use different technologies and have the following features, noted on their data sheets.

D flip-flop --- TYPE F1:

Set-up time: 2.3 ns
Hold time: 1.1 ns
CLK-to-Q propagation delay: 3.0 ns

D flip-flop ---- TYPE F2:

Set-up time: 3.1 ns
Hold time: 0.3 ns
CLK-to-Q propagation delay: 1.5 ns

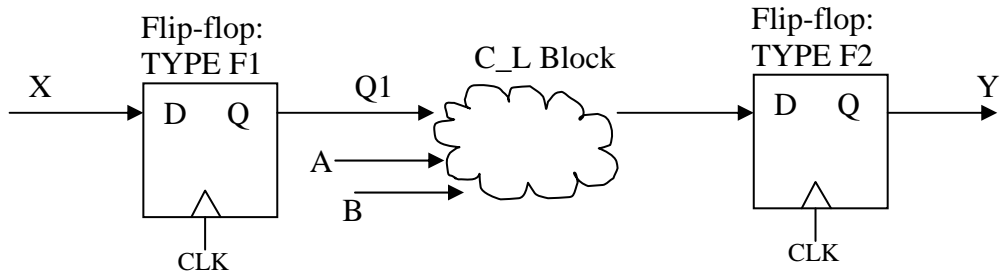
Combinational Logic elements' delays:

OR gate: 1.4 ns
NAND gate: 1.8 ns
Inverter: 0.6 ns

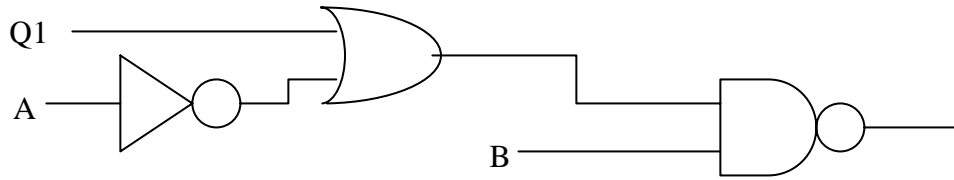
In this problem, you will apply in a new setting the sequential circuit timing concepts that you have learned in class. In computer architecture, the data enters a computational pipeline from the left, is operated on in a number of stages of flip-flops, and exits the pipeline on the right.

Figure 4 will help us analyze the key characteristics of two consecutive stages of a pipeline. It shows two D flip-flops, and some combinational logic (denoted by "C_L") in between. The main idea is as follows: Just like in a shift register architecture, the data (which comes from the left) must move **only 1 stage at a time** to the right, at each clock cycle. (Each flip-flop shown here, is a stage of the pipeline.) If the data does not make it on time to the next stage, the next stage samples the wrong value.

Note that the first D flip-flop is of TYPE F1, and the second D flip-flop of TYPE F2. The circuitry inside the combinational logic block is shown separately. There are two 1-bit data inputs A and B, that are also input into the combinational logic block. The CLK shown in Figure 4 refers to the same global clock. (Assume that the clock skew is zero.)



Inside the C_L Block, the circuitry is as follows:



- Figure 4 -

(a) (5 points) Compute the minimum clock period such that this fragment of the pipeline will operate correctly.

Answer: _____

(b) **(3 points)** Compute the minimum duration for which both of the inputs A and B have to remain stable after the rising edge of the clock:

Answer: _____

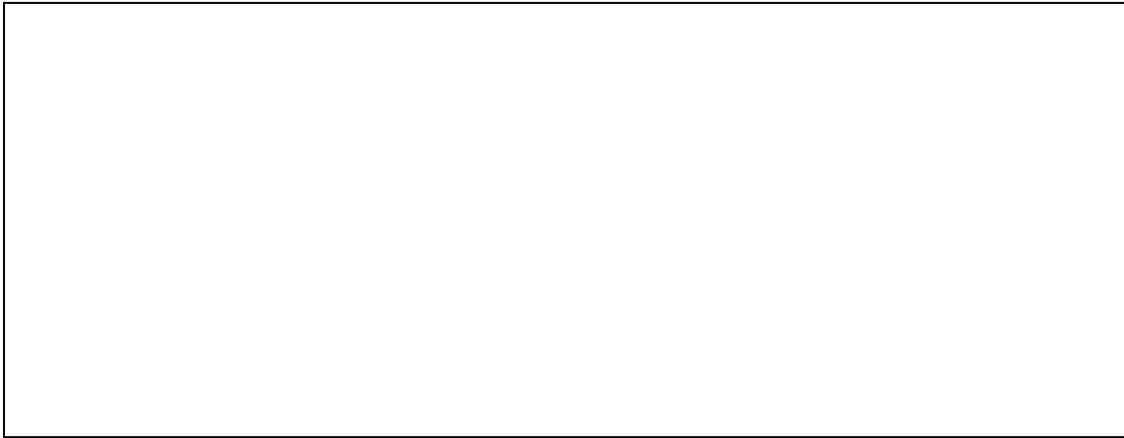
(c) **(3 points)** Compute the minimum duration for which the input A has to remain stable before the rising edge of the clock:

Answer: _____

(c) Your friend BitTwiddle advises you to change the design: She wants to add an extra pipeline stage on the right (using a flip-flop of type F1) and move some of the combinational logic block C_L from this stage to the next. The aim is to distribute the combinational logic between stages, and increase the maximum clock frequency at which this circuit can be run.

(3 points) Show this new design in the answer box below.

Answer Box:



(3 points) What is the minimum clock period of this new design?

Answer: _____

(3 points) What is the **disadvantage** of the new design that uses a higher clock frequency?

Answer Box:

(Do your additional work below:)

