University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS150 J. Wawrzynek
Spring 2009 4/1/09

Midterm Exam

Name: _____

ID number: _____

This is a *closed-book, closed-note* exam. No calculators or any other electronic devices, please.

Read all the questions **before** you begin. Each question is marked with its number of points (one point per expected minute of time). Although you might not need it, you have until 9pm.

You can tear off the spare pages at the end of the booklet and/or use the backs of the pages to work out your answers. Neatly copy your answer to the places allocated for them.

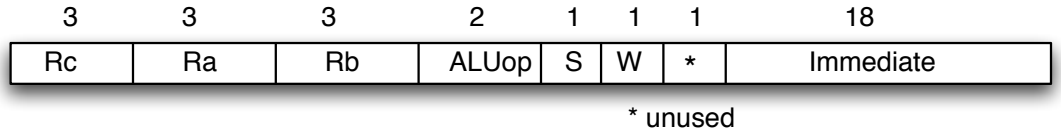
Neatness counts. We will deduct points if we need to work hard to understand your answer. **Simplicity also counts.** In the design problems, correct simpler designs with fewer components will be awarded a higher score than more complex designs with more components.

Put your name and SID on each page.

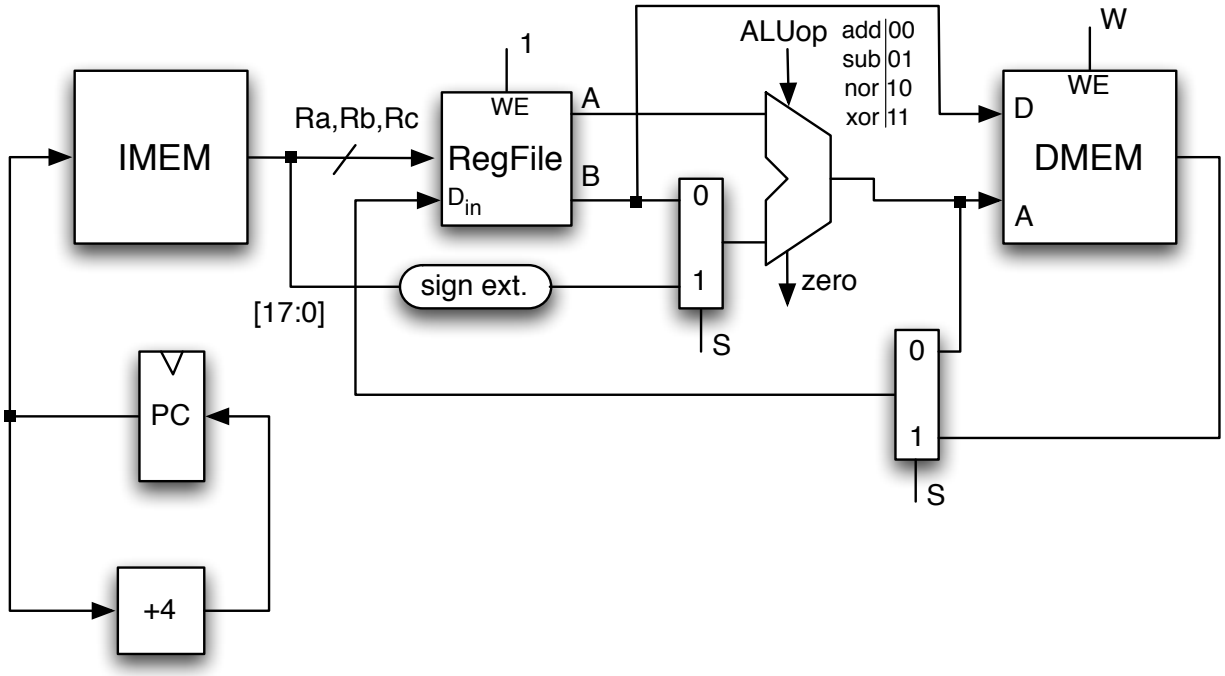
problem	maximum	score
1	15pts	
2	12pts	
3	12pts	
4	12pts	
5	12pts	
6	20pts	
7	12pts	
8	25pts	
Total	120pts	

1. Processor Implementation [15pts].

Below is shown the implementation of a single-cycle processor. It has operation similar to a single-cycle MIPS, but with a different instruction set and instruction encoding. This processor has no control unit. Instead, the control signals are encoded directly in the instruction. All instructions are encoded with the same instruction format, shown below.



The register file conforms to the MIPS register file specification. However, note that in this processor, a register write occurs on every cycle.



(a) For the assembly language instructions listed below, write a valid corresponding machine language representation (instruction encoding), in hex:

```
nor  $2, $1, $0      _____
sw   $4, 3($5)       _____
lw   $6, 5($7)       _____
```

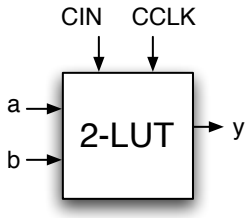
(b) Now consider adding a conditional branch instruction, similar to the MIPS `beq` instruction. Neatly modify the datapath drawing above to provide support for `beq`.

(c) Write a valid machine language represention for the `beq` instruction below (There are two instructions after the `beq` and before the instruction at "`skip`"):

```
        beq $1, $2, skip    _____
        ...
        ...
skip:   ...
```

2. LUT Implementation [12pts].

Consider the design of a 2-input FPGA lookup table (LUT). An abstracted view of the 2-LUT to the right. The ports, a, b, and y, are the data inputs and output, respectively. The input CIN stands for "configuration input", and CCLK stands for "configuration clock". The configuration clock is used at FPGA initialization time to shift in the LUT contents, one bit per clock cycle, over the CIN port.



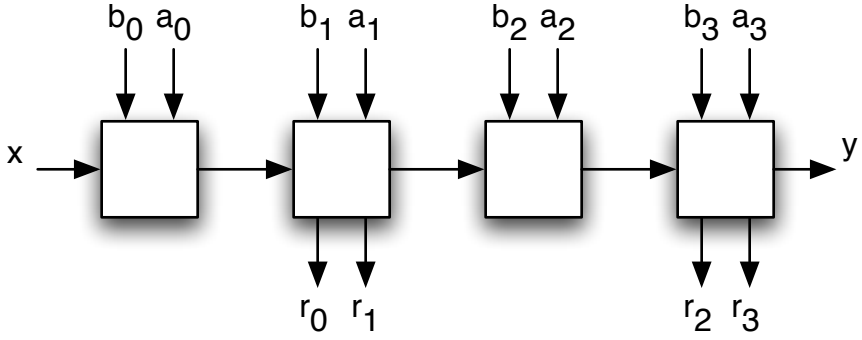
(a) In the space provided, draw a circuit diagram for the internal implementation of the 2-LUT, including configuration loading. Use only the following circuit components: Inverter, 2-input and, 2-input or, Flip-flop. Remember to label all inputs and outputs.

(b) Imagine now that the 2-LUT is programmed to implement the following function: a' · b. Using your circuit diagram above, label the appropriate nodes in your circuit diagram with the correct configuration values.

3. LUT Mapping [12pts].

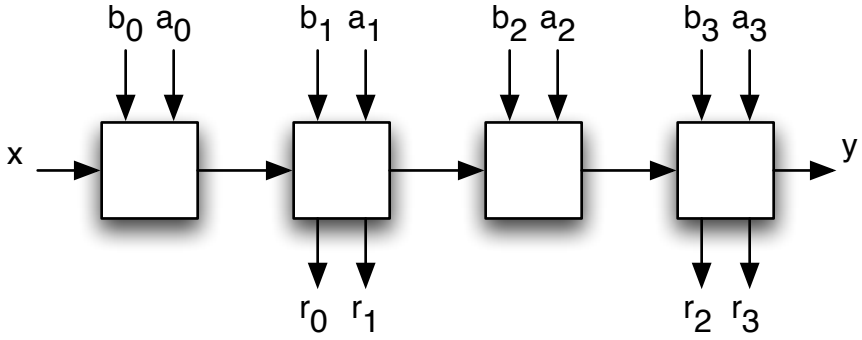
The circuit shown in the diagram below must be implemented on an FPGA chip by mapping it to a collection of LUTs. Assume that the blocks in the diagram are indivisible units. You go to your parts catalog and find that different FPGAs are available with different size LUTs, from 2-input to 8-input— all with a single output. You also find that an n-input LUT has a cost of $2^n + n$, and has delay of n.

(a) Using only one size LUT, which size (which value of n) would you choose to minimize the total implementation *cost*? Draw on the diagram to indicate how you would map the circuit to LUTs. Write down the total cost.



Total Cost =

(b) Which LUT size would you choose to minimize the total *delay* thought the circuit? Once again, draw on the diagram to indicate how you would map the circuit to LUTs. Write down the total delay.

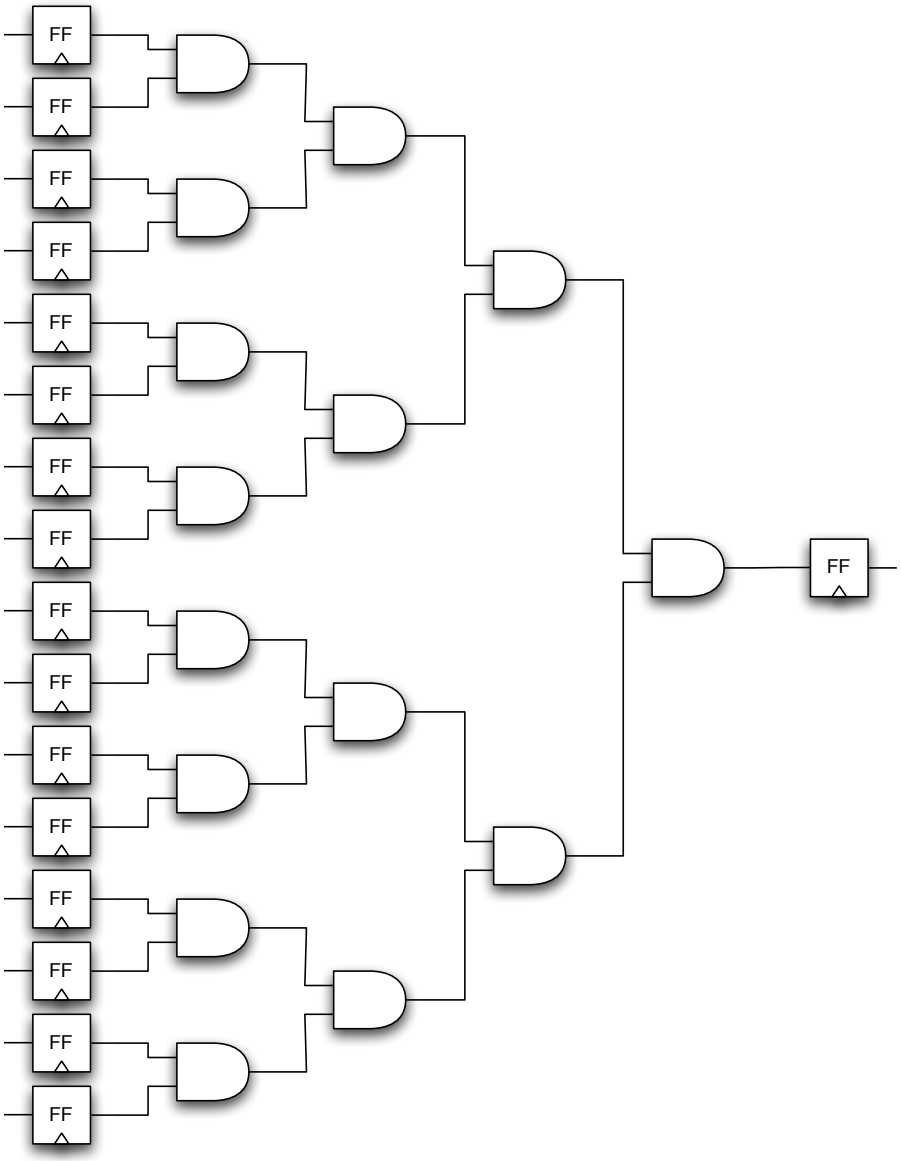


Total Delay =

4. Pipelining [12pts].

   Consider the circuit shown below. Your challenge is to add flip-flops to the circuit to minimize the *product* of the clock period and the total cost of flip-flops. Assume that there is no clock skew and that the delay through an and-gate (τ_{and}) is 1, the flip-flop setup time (τ_{setup}) is 1, and the flip-flop clock to q time ($\tau_{clk \to q}$) is 1. Count each flip-flop as 1 unit of cost. Therefore the flip-flop count as shown below is 17.

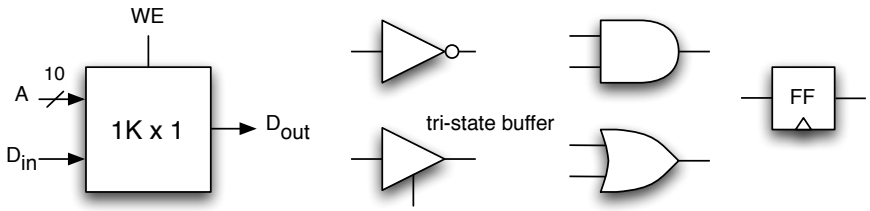
   Indicate on the drawing where you would add flip-flops. Write down the values of your chosen clock period and total flip-flop cost (including the 17 original flip-flops).



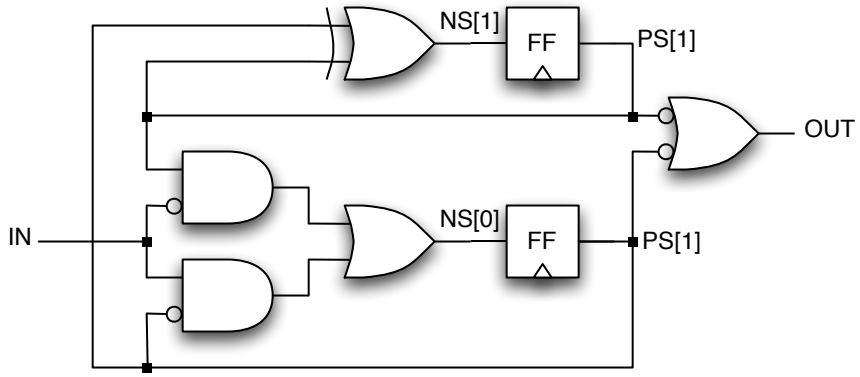
Clock Period = **Flip-flop Cost =**

5. Memory Cascade [12pts].

   Your are given a single-ported memory block with 1K 1-bit entries. Using multiple instances of this block along with simple elements listed below, draw a circuit diagram for the implementation of a single-ported 3K by 2-bit memory. You may assume that no addresses will ever be generated outside of the range of this memory. Label all inputs and outputs. *Remember, simplicity and neatness counts.*



6. Verilog Circuit Implementation [20pts].



(a) In the space below write a Verilog description of the circuit shown above, using continuous assignment for the NS and OUT signals.

module foo

endmodule

(b) Draw a state transition diagram describing the behavior of the circuit. Within each state bubble indicate the bit encoding for that state. Remember to label the arcs with input values and state with output values.

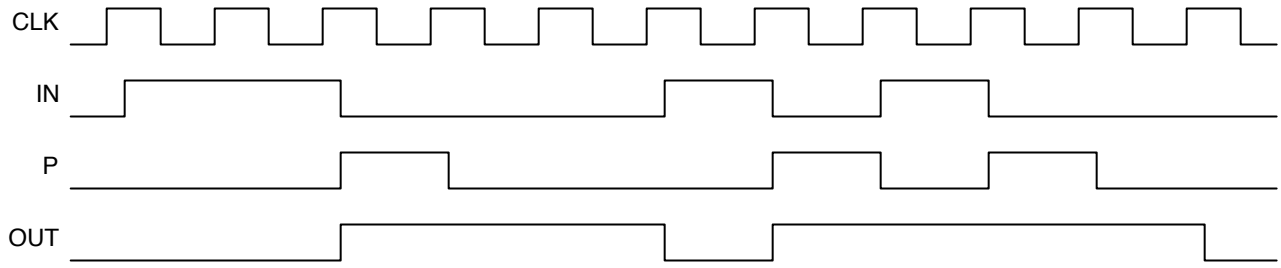
(c) Write a Verilog description for the function of the circuit based on the state transition diagram from part b (use a `case` statement).

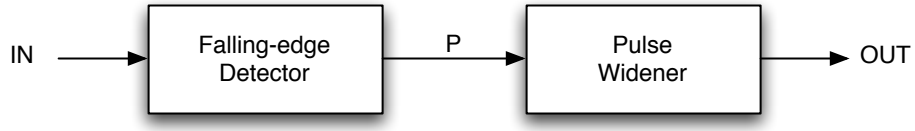
module FSM

endmodule

7. Clocked Circuits [12pts].

Your task is to design a simple synchronous circuit that outputs a pulse of width 3 clock cycles on each falling edge of the input signal. An example input and output signal is shown below. To make things easier, we will split the design into two parts. The "Falling-edge Detector" outputs a pulse in response to a failing edge on the input, as shown below. The "Pulse Widener" stretches out the pulse for 3 cycles.





(a) Draw your circuit for the Falling-edge Detector below. Use only the following circuit components: Inverter, 2-input and-gate, 2-input or-gate, flip-flop. Remember, simpler designs are worth more points.

(b) Draw your circuit for the Pulse Widener below.

8. Short Answers [25pts].

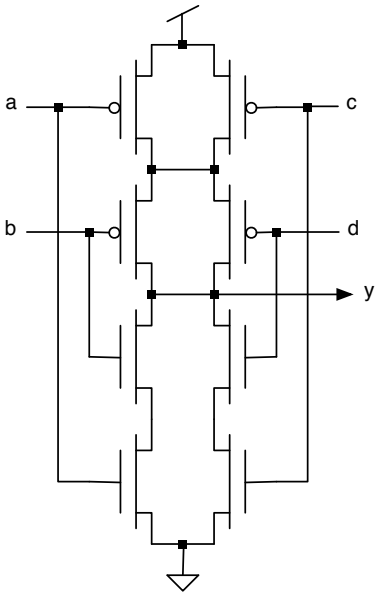
(a) [1pt] (T/F) Increased performance in digital systems always comes at the expense of more cost.

(b) [1pt] (T/F) Reducing clock speed is an effective means for lowering energy consumption per operation.

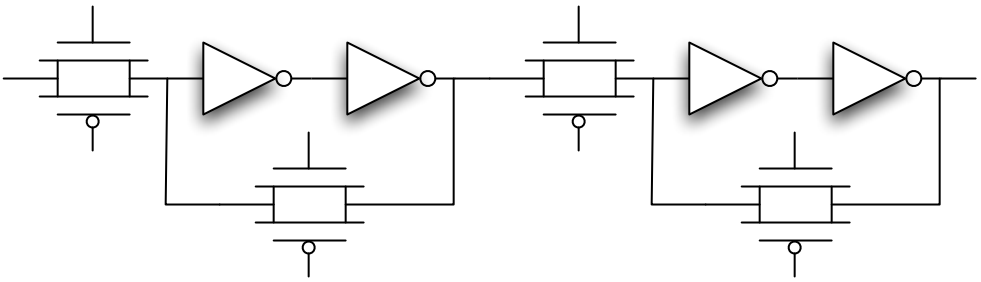
(c) [1pt] List the primary reason that clock routing on FPGAs uses a special purpose tree of wires rather than the general programmable interconnect.

(d) [3pts] A CMOS NAND gate with two inputs, a and b, drives an output capacitance C and is connected to a supply voltage V. Its a input is connected to a 100MHz square wave and its b input is connected to a 200MHz square wave. Write a formula for the dynamic power consumption of this gate.

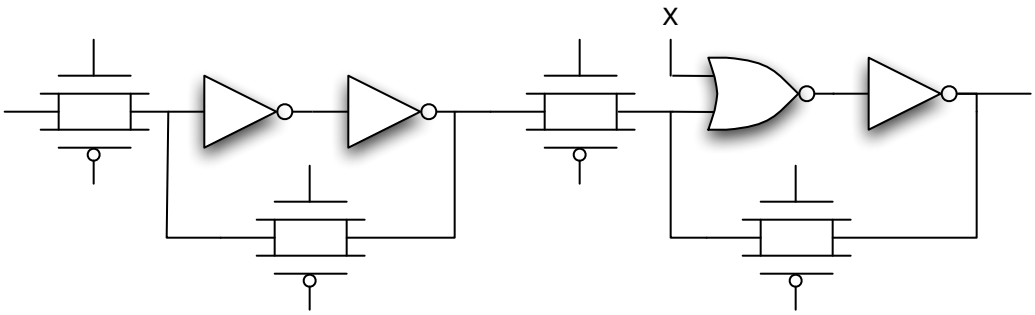
(e) [3pts] Write a boolean expression that represents the function of this gate.



(f) [2pts] Label the unconnected signals in the following **negative** edge-triggered CMOS flip-flop.



(g) [3pts] The CMOS flip-flop shown below has a mystery input. Circle its function from the list:



asynchronous clear **asynchronous preset** **synchronous reset** **synchronous set**

(h) [2pt] What is the most efficient MUX configuration (number of inputs with 1 output) using a single LUT in a 6-LUT architecture?

In a 4-LUT architecture?

(i) [2pt] Using multiple instances of simple dual-port memories, why is it more expensive to add write ports than to add read ports? Be brief.

(j) [2pt] Give two examples of architectural features of an FPGA that are responsible for its relative high power consumption compared to custom ICs.

(k) [3pt] For each of the following answer either "FPGA" or "ASIC". Assume you are a company that designs and manufacturers digital video recorder boxes.

i. Which one would give you better "time to market"?

ii. Which one would give you less engineering costs?

iii. Which one would give you less per unit cost?

(l) [2pt] Put your name and SID on each page.

Spare page. *Will not be graded.*

Spare page. *Will not be graded.*

Spare page. *Will not be graded.*

Spare page. *Will not be graded.*

Spare page. *Will not be graded.*