

**CS/ECE 552: INTRODUCTION TO COMPUTER ARCHITECTURE
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON**

Prof. Mark D. Hill
TA Brandon Schwartz

Midterm Examination I
In Class
Wednesday, October 11, 2000
Weight: 25%

1:15 minutes.

CLOSED BOOK, etc., but one cheat sheet allowed (two-sided 8.5x11 page).

The exam is two-sided and has **EIGHT** pages, including two blank pages at the end.

Plan your time carefully, since some problems are longer than others.

NAME: _____

ID# _____

Problem Number	Maximum Points	Actual Points
1	3	
2	3	
3	3	
4	5	
5	4	
6	7	
Total	25	

Problem 1 (3 points)

Program A runs for 1,000,000 instructions and executes a branch every 8 instructions (i.e., averages 7 sequential instructions and one branch every 8 instructions). Program B runs for 3,000,000 instructions and branches every 5 instructions. Consider a workload where A and B are run equally often.

Write an expression for the average number instructions per branch.

Problem 2 (3 points)

Consider a workload with 10% stores, 20% loads, 25% branches, and 45% other integer instructions. Consider two machines BASE and NEW.

On BASE, let stores take *1 cycle*, loads *2 cycles*, branches 3 cycles, and others 1 cycle.

On NEW, let stores take *2 cycles*, loads *1 cycle*, branches 3 cycles, and others 1 cycle.

In this case, the speedup of NEW with respect to BASE is S . Write an expression for S .

Problem 3 (3 points)

Consider a server computer using a 1 GHz Intel PentiumIII processor versus another server computer using a 667 MHz Compaq Alpha 21264 processor. Assume that benchmark results show that the Alpha-based systems is faster than the PentiumIII-based system running SPEC95 integer benchmarks.

In at most one-half page, propose at least two different explanations for why the system with the slower clock might be faster.

Problem 4 (5 points)

(a) (2 points) Consider carry-lookahead addition. Fill in the values in the table for all eight g_i 's and p_i 's.

	Bit 3	Bit 2	Bit 1	Bit 0
Input A	0	1	1	0
Input B	1	1	0	1
Carry-generate g				
Carry-Propagate p				

(b) (1 point) Would the block of four bits in part (a) generate a carry? Would it propagate a carry? Why?

(c) (2 points) Consider a *carry save adder* (CSA) with three four bits inputs A, B, and C, and two 5-bit outputs D and E. Recall that a CSA maintains the invariant that $A+B+C = D+E$.

Fill in the nine remaining bit values for D and E. Assume E is fed from carry outputs.

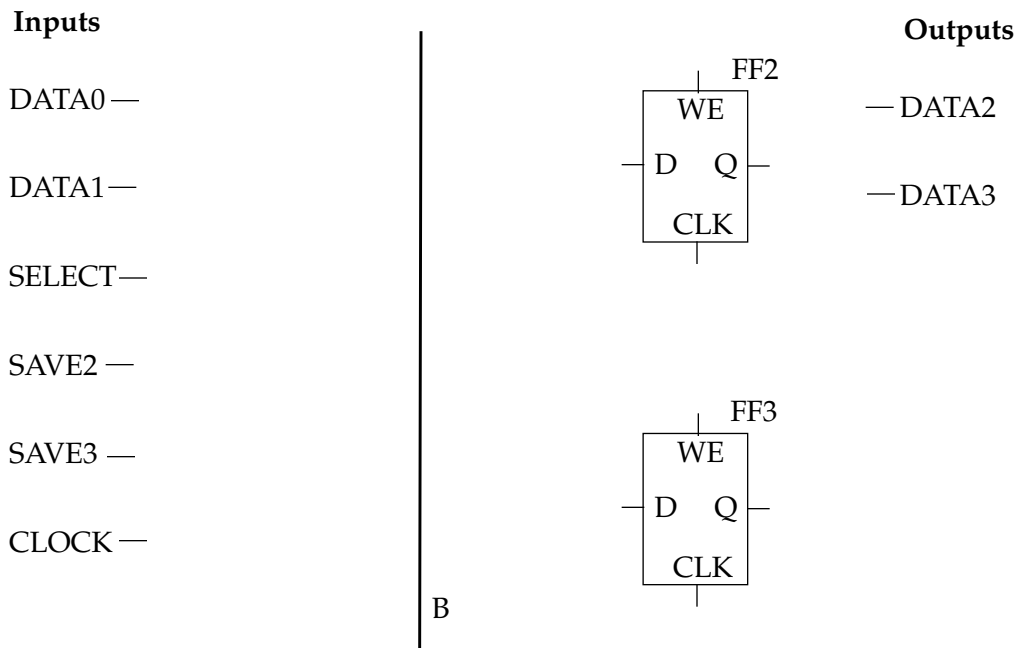
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input A	none	0	1	1	0
Input B	none	1	1	0	0
Input C	none	1	1	0	0
Output D					
Output E					0

Problem 5 (4 points)

You are to implement a circuit with the input and outputs listed below using the clocking methodology discussed in class. The flop-flops provided load a new input value from D if WE (write enable) is high when CLK transitions from high to low. Q is the value of the flip-flop.

By adding wires, gates (AND, OR, NOT, etc.), and tri-state buffers (*but no multiplexors*), implement a circuit that:

- (a) Provides the output of flip-flop FF2 to DATA2,
- (b) Provides the output of flip-flop FF3 to DATA3,
- (c) Loads a flip-flop FF2 with the value on bus B if SAVE2 is 1 (when CLOCK transitions from high to low),
- (d) Loads a flip-flop FF3 with the value on bus B if SAVE3 is 1 (when CLOCK transitions from high to low),
- (e) Writes DATA0 on bus B if SELECT is 0 and the value of bus B is needed by FF2 or FF3,
- (f) Writes DATA1 on bus B if SELECT is 1 and the value of bus B is needed by FF2 or FF3, and
- (g) Saves power by *not writing* a value onto bus B when no value is needed by FF2 or FF3.



Problem 6 (7 points)

- (a) (5 points) Using n -bit-wide 4-to-1 multiplexors and gates (AND, OR, NOT, etc.), implement a combinational circuit that accepts inputs $\text{DATAIN}\langle 15:0 \rangle$ and $\text{SHIFTAMOUNT}\langle 3:0 \rangle$ and outputs $\text{DATAOUT}\langle 15:0 \rangle$ whose value is $\text{DATAIN}\langle 15:0 \rangle$ *logically right shifted* by $\text{SHIFTAMOUNT}\langle 3:0 \rangle$.

- (b) (2 points) Discuss how your circuit in part (a) would change if a third input SHIFTTYPE was added with the requirement that (i) an *logical right shift* should be performed if SHIFTTYPE is 1 and (ii) an *arithmetic right shift* should be performed if SHIFTTYPE is 0.

Scratch Sheet 1 (in case you need additional space for some of your answers)

Scratch Sheet 2 (in case you need additional space for some of your answers)