Name _____

CS/COE 0447 Example Problems for Exam 2
Fall 2009

1. This time, consider a non-leaf function `lisa`. This function has no arugments or return value. The return address is on the stack at offset 12 and the activation record is 12 bytes. Give a sequence of three MIPS instructions that cause a function return.

2. Suppose the stack pointer (`$sp`) has the value 0xFF0000020 and the activation record has two halfword fields. Give a single instruction that will load the second field in the activation record into register `$t0`.

For the next questions, consider the program code below (with line numbers):

```
                # assume $sp = 0xFFFF0020
0               li    $s0,1
1               li    $s1,2
2               jal   _bart
3               j     quit
4    _bart:     addi     $sp,$sp,-8
5               sw       $s0,0($sp)
6               sw       $ra,4($sp)
7               jal      _homer
8               lw       $ra,4($sp)
9               lw       $s0,0($sp)
10              addi     $sp,$sp,8
11              jr       $ra
12   _homer:    add      $sp,$sp,-4
13              sw       $s1,0($sp)
14              addi     $s1,$0,10
15              move     $v0,$s1
16              lw       $s1,0($sp)
17              addi     $sp,$sp,4
18              jr       $ra
19   quit:      ....
```

3. Give the value of `$sp` on each line from the code:

Line 5: `$sp`'s value is _____

Line 8: `$sp`'s value is _____

Line 13: `$sp`'s value is _____

Line 19: `$sp`'s value is _____

Name _____

4. Assume memory is all 0s. Fill in the table below to show the memory contents (as words) after the code above executes (i.e., when line 19 is reached):

Address	**Value at this Address**
0xFFFF0028	
0xFFFF0024	
0xFFFF0020	
0xFFFF001C	
0xFFFF0018	
0xFFFF0014	
0xFFFF0010	
0xFFFF000C	

5. Show the steps to multiply the 4-bit numbers 3 and 5 with the "fast shift-add multipler". Use the table below. List the multiplicand and product in binary. In the field "step", write "ADD" when the multiplicand is added. Write "SHIFT" to indicate when the product is shifted. In the iteration "Start" write the initial values for the mutiplicand and product. You may not need all steps (rows) in the table.

Iter.	**Multiplicand (M)**	**Product**	**Step**
Start			
1			
2			
3			
4			
5			
6			
7			

Name _____

6. Show the steps to multiply an 6-bit number 17 and 3 with Booth's algorithm. Use the table below. List the multiplicand and product in binary. In the field "step", write "ADD", "SUB", or "NO OP" to indicate which operation is done on each iteration.

Iter.	Multiplicand (M)	Product	Step
Start			
1			
2			
3			
4			
5			
6			
7			

7. Suppose we want to do the computation S = A + B. A and B are positive 2's complement 8-bit binary numbers. Give a boolean expression that indicates whether there was an overflow when these numbers are added. To represent a certain bit *i* in *A*, *B* or *S*, use A_i, B_i or S_i. E.g., bit position 3 in *A* is A_3. Assume the bits are numbered 0 to 7 (right to left).

8. Give the negation in one's complement binary representation (5 bits) for the decimal numbers:

5d Negation (in one's complement binary) _____

10d Negation (in one's complement binary) _____

-15d Negation (in one's complement binary) _____

9. Give the negation in two's complement binary representation (5 bits) for the decimal numbers:

11d Negation (in two's complement binary) _____

15d Negation (in two's complement binary) _____

-13d Negation (in two's complement binary) _____

Name _____

10. Give Booth's encoding for the 8-bit numbers:

-19d Booth's encoding _____

27d Booth's encoding _____

62d Booth's encoding _____

11. Using 1-bit adders, draw the circuit for a 4-bit ripple-carry addition unit.

12. Using 1-bit adders and 1-bit inverters (i.e., the *not* of a bit), draw a circuit for a 4-bit ripple-carry subtract unit.

13. Consider restoring division with hardware design #3 (the design with a 32-bit divisor and a 64-bit remainder register that holds the remainder and quotient). Assume the quotient and divisor are 5 bit unsigned numbers. Fill in the table below for 17 / 3. For each step, indicate what shift, subtraction, and addition operations are done in the "Step Notes" column.

Iteration	Divisor (D)	Remainder (R)	Step Notes
Init			
1			
2			
3			
4			
5			
6			
Done			

14. Now, consider non-restoring division with hardware design #3 (the design with a 32-bit divisor and a 64-bit remainder register that holds the remainder and quotient). Assume the quotient and divisor are 5 bit unsigned numbers. Fill in the table below for 17 / 3. For each step, indicate when shift, addition and/or subtraction operations are done in the "Step Notes" column.

Iteration	Divisor (D)	Remainder (R)	Step Notes
Init			
1			
2			
3			
4			
5			
6			
Done			

15. Floating point numbers represent a "richer" set of values than integer numbers. Nevertheless, processors support integer numbers and programs frequently use them. What primary advantage does integer numbers and operations offer over floating point numbers and operations?

Name _____

16. Using IEEE 754 representation for single precision floating point, give the 32-bit binary encoding for the numbers below. Show the sign, exponent, and significand.

Number: -2.40625 Float: _____

Number: 11.2265625 Float: _____

Number: -0.00244140625 Float: _____

17. Suppose we have the following numbers encoded as IEEE 754 single precision floats. is the decimal value (i.e., base 10) for each number:

sign	**exp**	**significand**
0	10000001	011010...0
1 bit	*8 bits*	*23 bits*

sign	**exp**	**significand**
1	01111011	10...0
1 bit	*8 bits*	*23 bits*

18. When the bias is 127, give the binary encoding for the number 39d.

19. When the bias is 1023, give the binary encoding for the number -39d.

20. Biased representation for the exponent offers a significant advantage over other representations, like two's complement. What is the advantage?

21. Consider the sum of products boolean equation: A'BC + ABC + A'B'C. Give the truth table representation for this boolean equation.

22. For the boolean equation and truth table from question 21, give its Karnaugh map.

23. Based on the Karnaugh map from question 22, can the boolean equation be minimized? If so, give its minimized form. Otherwise, simply list the original equation to answer this question.

24. For your answer from question 23, draw the circuit that implements the boolean equation.

25. Give the minimized boolean equation for the Karnaugh map below.

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

Name _____

26. Give the minimized boolean equation for the Karnaugh map below.

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

27. Suppose we want to construct a 4:1 multiplexor. This multiplexor selects as an output one of its four inputs. How many rows are in the full truth table for this operation?

28. Using AND, OR, and NOT gates, draw the circuit for the boolean equation from question 26.

29. Using only NAND gates, draw a minimal circuit for the boolean equation A'B' + CD. Hint: Use straightforward substitutions of ANDs, ORs and NOTs with the NAND equivalents (see lecture slides). Observe whether any of the operations after substitution can be eliminated.

30. Prove that your circuit from question 29 is equivalent to A'B' + CD. Hint: Write a boolean equation for the circuit and transform it, step-by-step, into A'B' + CD with Boolean algebra.

31. Using only NOR gates, draw a minimal circuit for the boolean equation A'B + CD. (Can you prove this circuit is equivalent to A'B' + CD?)

32. Suppose we want to implement a 1-bit ALU that can perform the logical operations OR, AND, XOR and NOT. This 1-bit ALU takes as input two 1-bit numbers: A and B. It produces as an output a single 1-bit number: C. If the ALU does a unary NOT operation, it ignores B. To build this ALU, you can use OR, AND, XOR and NOT gates. You also need a multiplexor to select among the four operations. Answer the following questions:

a) How many control signals are needed to select the operation to do with the ALU?

b) How many inputs does the ALU's multiplexor have?

c) Draw the ALU circuit (using symbols for a mux, AND, OR, NOT and XOR).

33. Larger muxes can be constructed from smaller ones. For example, a 4:1 mux can be constructed with 2:1 muxes. A 4:1 mux selects among four inputs for its one output. Draw a circuit diagram for a 4:1 mux, built with 2:1 muxes. (Hint: You need three 2:1 muxes.)

34. Now, let's consider a 3:1 mux built with 2:1 muxes. Draw a diagram that shows the circuit. (Hint: You need two 2:1 muxes.)

35. Suppose a logic gate (OR, AND, NOT) takes 1 ns to compute a value. The time it takes a gate to compute a value is a "delay". If multiple gates are put together in succession, the total time for the combinational circuit will be the sum of the gate delays. For example, a circuit that has an AND gate followed by an OR gate has a delay of 2ns (1ns for each gate). As another example, a circuit that has an AND gate that operates in *parallel* with an OR gate has a delay of 1ns

(since the OR and AND are done in parallel). If each gate takes 1ns, compute the total delay for a 16:1 mux built with 2:1 muxes. Hint: Compute the time for a single 2:1 mux. Next, determine how many muxes are connected in succession. The "longest path" through this circuit gives the total delay.