Computer Architecture — Due on Friday, February 27, 2015

Assignment 2

CS3350B — *University of Western Ontario*

Submission instructions. With each assignment, you are required to hand in an Assignment Submission Form, on which you certify that the material you've handed in is exclusively your own work. Put this form inside a 9-by-12-inch envelope along with your assignment, bearing your name and the course number CS3350b, which will be used to return your assignment. Hand in your envelope (contained with your completed assignment and the Assignment Submission Form) in class, or drop it in the CS3350b locker (locker #308, on the third floor of the Middlesex College building) by the midnight on the due date.

- If these are `MIPS` programs, you can either print them out or hand-write them.
- If these are LOGISM circuits, you can either print them out or draw them by hand.

PROBLEM 1. [15 points] Consider a 2-way-set-associative cache with a 32-bit address, including 4-bit data block offset and 6-bit index. Assume that a word consists of four bytes. Starting from power on, the following word-addressed cache references are recorded.

$$11, 26, 21, 52, 23, 30, 47, 6, 35, 44, 54, 5$$

Use LRU (least recently used) replacement.

1.1 List the final state of the cache, with each valid entry represented as a record of <index, tag, data block(s)>.

1.2 What is the hit ratio?

		Data block (4 words)			
Index	Tag	3	2	1	0
000000	0	Memory(11)	Memory(10)	Memory(9)	Memory(8)
000000	0	Memory(7)	Memory(6)	Memory(5)$^{\text{hit}}$	Memory(4)
000001	0	Memory(31)	Memory(30)	Memory(29)	Memory(28)
000001	0	Memory(23)$^{\text{hit}}$	Memory(22)	Memory(21)	Memory(20)
000010	0	Memory(47)	Memory(46)	Memory(45)	Memory(44)$^{\text{hit}}$
000010	0	Memory(35)	Memory(34)	Memory(33)	Memory(32)
000011	0	Memory(55)	Memory(54)$^{\text{hit}}$	Memory(53)	Memory(52)
000011	0				
...	-	-	-	-	-

The hit ratio is 4/12 = 33.3%.

PROBLEM 2. [15 points] We would like to expand the `MIPS` register file to 128 registers (rather than usual 32) and expand the instruction set by letting 'op' field be 20 bits. Assume that a word consists of eight bytes.

2.1 Assume that the 'op' field determines the function field (say, the number of bits of 'funct' and 'op' are same). How this would this affect the size of each of the bit fields in the R-type instructions? Label all the fields with their name and bit length and explain the consequence.

op	rs	rt	rd	shamt	funct
20 bits	7 bits	7 bits	7 bits	3 bits	20 bits

The consequence of this is that the shift amount field is increased to 10 bits, thus allowing to do maximum number of $2^3 - 1 = 7$*-bit shifts per instruction. Therefore, if one wants to shift more than 7 bits, one must use more than one instruction.*

2.2 How this would this affect the size of each of the bit fields in the I-type instructions? Label all the fields with their name and bit length and explain the consequence.

op	rs	rt	rd	constant or address
20 bits	7 bits	7 bits	7 bits	23 bits

The consequence of this is that the range of address will be within $\pm 2^{22}$*, expanded by a factor of* $2^7 = 128$*.*

PROBLEM 3. [15 points] Consider the following MIPS loop:

```
LOOP: slt $t2, $0, $t1
      beq $t2, $0, DONE
      srl $t1, $t1, 1
      addi $s1, $s1, 1
      j LOOP
DONE:
```

3.1 Assume that the register `$t1` is initialized to the value 10. What is the value in register `$s1` assuming `$s1` is initially zero?

4.

3.2 For the above loop, write the equivalent C code routine. Assume that the registers `$s1, $t1` and `$t2` are integers `A, i` and `j`, respectively.

```
while (0 < i) {
      i >>= 1;
      A++;
}
```

Compute the number of bits in binary to represent a decimal number.

3.3 Assume that the register `$t1` is initialized to the value `N`. How many MIPS instructions are executed?

$5 * \lceil \log_2(N) \rceil$

PROBLEM 4. [15 points] Consider the following C code function to compute the Fibonacci number:

```
int fib (int n) {
     if (n < 2)
          return 1;
     else
          return fib(n-1) + fib(n-2);
}
```

Then, starting from n = 0, the sequence will look like

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots$$

4.1 Implement the above C code in MIPS assembly. Try to use a minimum number of MIPS instructions as possible.

4.2 Write a complete MIPS code and run it in QtSpim to compute fib(20). Show the final values of registers you used.

See 'fib.asm'.

PROBLEM 5. [20 points] Assume that we have a machine, which is a byte-addressable machine and a word consists of four bytes. The table below shows 32-bit values of an array stored in the memory.

Address	Data
24	2
28	4
32	7
36	3
40	8
44	6
48	5
52	1

5.1 Write C code to find the maximum value among the above data. Assume that the data shown represents the C variable called A, which is an array type of int.

```
int i, max = 0;
int A[] = {2, 4, 7, 3, 8, 6, 5, 1};
for (i = 0; i < 8; ++i) {
     if (A[i] > max)
          max = A[i];
}
```

5.2 Write MIPS code to find the maximum value among the above data. Use a minimum number of MIPS instructions as possible. Assume the base address of A is stored in register $s4.

5.3 Write a complete MIPS code and run it in QtSpim. Show the final values of registers you used.

See 'max.asm'.

PROBLEM 6. [20 points] Assume that X consists of 3 bits: x2 x1 x0 and that y is represented by 1 bit. We expect that y is true if and only if X contains only two 1s.

6.1 Show the true table with each entry represented by <x2 x1 x0 y>.

See in Table 1.

x2	x1	x0	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table 1: The true table.

6.2 Write the logic function of y.

$y = \overline{x2} \cdot x1 \cdot x0 + x2 \cdot \overline{x1} \cdot x0 + x2 \cdot x1 \cdot \overline{x0}$ *or* $y = (\overline{x2} \cdot x1 + x2 \cdot \overline{x1}) \cdot x0 + \cdot x2 \cdot x1 \cdot \overline{x0}$

6.3 Using Logism, draw the gate diagram using AND, OR and NOT gates only.

6.4 Using Logism, draw the gate diagram using two AND gates, one XOR, NOT and OR gate, respectively. (Or use less than this amount of gates to implement y.)

See 'gates.circ'.