CSE340: Computer Architecture

Chapter 2 Class Practice Problems

1. Write the equivalent MIPS code for the following code:
   `z = w + x - y` where z is in $s0 and w, x, y are in $s1, $s2 and $s3 respectively.

2. What is ‘word’ in MIPS?

3. What is the size of each slot in our main memory?

4. Why is the index multiplied by 4 when calculating the address for lw and sw?

5. **Write the MIPS code for the following code:**

   ```
   int x = 5;

   A[5] = x + B[6] - A[0] - 15;

   int z = A[6];
   ```

   The base address of A is at $s5, and B is at $s6. Use $s1 for x and $s2 for z.

6. Can we subtract an integer from a value stored in a register? Give an example.

7. Write the MIPS code to move a value from $s3 to $t0.

8. In the case of signed numbers, why can’t we represent (−(−(2^(n−1))))?

9. Consider the following MIPS code: sub $17, $9, $10; what type of instruction is this? Write the machine code for this instruction. Consider, the identifying value for the instruction is 37. Write the final answer in hexadecimal.

10.

Suppose the data `1234ABCD` is stored in the memory in the following way-

0x4AB00100	CD
0x4AB00101	AB
0x4AB00102	34
0x4AB00103	12

a. Is this a Big-Endian system or Little-Endian System? Why?
b. What do you understand by 'The Memory is Word Aligned in MIPS"?

11. What is the smallest value we can subtract using a MIPS immediate instruction?

12. Suppose the newly created BRACS Architecture System supports a word size of 32-bit and each memory cell in the architecture is of size 8-bit. In this architecture, what will be the location of an array element Arr[6] if the base address of the array is 0x11224814?

13. In the I-type instruction, the rt 5 bits can hold both source and destination register numbers. For which instruction does it hold the source register number? For which instruction does it hold the destination register number? Give an example of each.

14. Show the encoding of the following MIPS instructions by specifying the instruction type (R/I/J). You may consider any number in the range of 6 bits as the identifying values for each instruction. However, please mention the values you are considering.

```
i) addi $17, $11, -5

ii) add $16, $17, $18

iii) lw $t3, 32($s7)

iv) sw $13, 8($19)

v) and $t1, $t3, $t5

vi) beq $s2, $s3, 0xC93E
```

vii) `srl $9, $21, 5`

viii) `j 1024`

15. Calculate the size of the data memory for a 64-bit MIPS architecture.

16. What is conditional and unconditional branching?

17. Apart from left shifting by 2-bit, what other operation is required to complete jump address calculation?

18. Consider that the PC has the value (in hex) 0xB0001F79. If the offset value (in decimal) is 476, then calculate the conditional and unconditional target address.

19. Let's assume instruction J 1020 is in memory address 0x00223380. Now calculate the Jump target address and write it in hexadecimal.

20. **Write the MIPS code for the following code:**

```
Arr[5] = 48x + 29y;
```

The base address of Arr is in $s5, and x and y are in $s1 and $s2 respectively.

21. What is the difference between the instruction memory and the data memory in MIPS? Explain with an example.

22. How can a processor differentiate between an AND instruction and an OR instruction?

23. Suppose you have a switchboard that has 8 switches for turning the lights on and off. The current state of the switches can be represented by the binary number: 10110011 where 1 means the corresponding switch is on and 0 means it is off. Now using MIPS instructions, you wish to automate the process of turning on and off. How can you turn off the 3rd and 4th switch from the left and keep everything else as is using MIPS instructions?

24. Construct the equivalent MIPS code of the following code given x is in $17 and y is in $18.

```
if x == y {
     x = y * 128
}
else{
     x = y / 32
}
```

25. **Construct** the equivalent MIPS code of the following code.

```
for(c=7;c>0;c--)
{
     if (A[c] != c)
        B[c] = A[B[c+1]]
}
```

Hints: Consider base addresses of array A and B are in register $10 and $11. Also consider *c* is in register $8. **Also, you can use any register from $1 to $31 for intermediate calculations.**

26. **Translate** the following code to its equivalent MIPS code, given i is in $s3.

```
add(int A[], int B[]) {
   for (int i =0; i < 10; i = i + 1) {
      A[i] = A[i] +B[i];
   }
   return A[]
}
```

27. Write the MIPS code that can store the hex value 0xAA112233 into register $t0.