Faculty of Computers and Artificial Intelligence

CS222: Computer Architecture

Assignment no 09:
Chapter 6: Architecture

Note: You can check the exercises after the Chapter. In our assignment, we are using the 2nd Edition of "Digital Design and Computer Architecture" By David and Sarah Harris.

Exercise 6.11 **Convert** the following MIPS assembly code into machine language. Write the instructions in hexadecimal.

```
addi $s0, $0, 73
sw $t1, −7($t2)
sub $t1, $s7, $s2
```

Exercise 6.14 **Convert** the following program from machine language into MIPS assembly language. The numbers on the left are the instruction addresses in memory, and the numbers on the right give the instruction at that address. Then reverse engineer a high-level program that would compile into this assembly language routine and write it. Explain in words what the program does. $a0 is the input, and it initially contains a positive number, n. $v0 is the output.

```
0x00400000    0x20080000
0x00400004    0x20090001
0x00400008    0x0089502A
0x0040000C    0x15400003
0x00400010    0x01094020
0x00400014    0x21290002
0x00400018    0x08100002
0x0040001C    0x01001020
0x00400020    0x03E00008
```

Exercise 6.17 (a) **Implement** the following high-level code segments using the slt instruction. Assume the integer variables g and h are in registers $s0 and $s1, respectively.

```
if (g > h)
g = g + h;
else
g = g − h;
```

Exercise 6.25 (a,d,e) **Convert** the following beq, j, and jal assembly instructions into machine code. Instruction addresses are given to the left of each instruction.

(a)
```
0x00401000 beq $t0, $s1, Loop
0x00401004 . . .
0x00401008 . . .
0x0040100C Loop: . . .
```

(d)
```
0x00403000 jal func
… …
0x0041147C func: . . .
```

(e)
```
0x00403004 back: . . .
… …
0x0040400C j back
```