Mario Palacios
CpE 142 – Section 01
Assignement 03

Consider the following C-code fragment:

for (i=0; i<64; i++)

z[i] = a[i] + b[i] - c[i];

Assume that: a, b, c, and z are arrays of 32-bit words. The machine code for this fragment is stored in memory starting from location 00000100 Hex. Arrays a, b, c, and z are stored in memory beginning with memory locations 300hex, 400hex, 500hex, and 600hex, respectively.

a) Convert this code to MIPS. You must provide the numeric offset for each branch or jump instruction that you use. (50 points)

```
100   andi $t0, $t0, 0          # initialize i and make it = 0
104   addi $s1, $zero, 300      # Set array A to base 300hex
108   addi $s2, $zero, 400      # Set array B to base 400hex
10C   addi $s3, $zero, 500      # Set array C to base 500hex
110   addi $s4, $zero, 600      # Set array Z to base 600hex
114   addi $t1, $zero, 64       # set s4 to be 64, the loop constant
118   sll $t2, $t0, 2           # shift left by 2
11C   add $t3, $s0, $t2         # Increment array A to next index
120   add $t4, $s1, $t2         # Increment array B to next index
124   add $t5, $s2, $t2         # Increment array C to next index
128   add $t6, $s3, $t2         # Increment array Z to next index
12C   lw $t7, 0($t3)            # load a[i]
130   lw $t8, 0($t4)            # load b[i]
134   add $t7, $t7, $t8         # a[i] + b[i]
138   lw $t8, 0($t5)            # load c[i]
13C   sub $t7, $t7, $t8         # result in $t7 minus c[i]
140   sw $t7, 0($t6)            # store result in z[i]
144   addi $t0, $t0, 1          # increment by one
148   beq $t0, $t1, 1           # on equal go to, PC = 150h, skips 1 inst.
14C   j 0x46                    # jump to 118h (0x46*4 = 118h)
150   …….                       # Exit
```

Mario Palacios
CpE 142 – Section 01
Assignement 03

b) Assume that the CPIs for MIPS instructions on a multi-cycle system M1 are as listed below:

- lw: 5 cycles [3 instructions]
- R-type, addi, sw, and other immediate Arithmetic instruction: 4 Cycles [13 instructions]
- beq, bne, and j: 3 cycles [2 instructions]
- other instructions: 4 cycles [2 instructions]

How many clock cycles does it take to execute the C-code fragment on system M1? Justify your answer (20 points)

Outside Loop + Iterations of Loop * (Inside Loop) = Total Clock Cycles

=> 4cycles*(5) + 64 * [5cycles*(3) + 4cycles*(9) + 3cycles*(2)] = 3668 Clock Cycles

c) Assume that the clock rate for M1 is 2.0 GHz. What is the CPU execution time for the code on M1? (10 points)

Clock Cycles / Clock Rate = CPU Time
=> 3668 Clock Cycles / 2.0GHz = 1834ns

d) Compute CPU execution time of the code for a single-cycle system with clock rate 2.5 Ghz (20 points)

Total Clock Cycles = 5 + 64 [3 + 9 + 2] = 901 Clock Cycles

CPU Time = 901 Clock Cycles / 2.5GHz = 3600ns