The University of Alabama in Huntsville
ECE Department
CPE 431 01
Test 1 Solution
Fall 2015

1.   (1 point) A **_server_** is a computer used for running larger programs for multiple users, often simultaneously, and typically accessed only via a network.

2.   (1 point) An **_acronym_** is a word constructed by taking the initial letters of a string of words.

3.   (1 point) The **_opcode_** is the field that denotes the operation and format of an instruction.

4.   (1 point) Hiding details from the higher level is an example of the great idea of **_abstraction_**.

5.   (1 point) **_Speculation_** is an approach whereby the compiler or processor guesses the outcome of an instruction to remove it as a dependence in executing other instructions.

6.   Extra Credit (1 point) If at first you don't succeed, **_redefine success_**.

7.   (15 points Write down the hexadecimal representation of the decimal number -25.1 assuming the IEEE 754 double precision format.

```
/2       0    1              x2    0.1
/2       1    1              x2    0.2      0
/2       3    0              x2    0.4      0
/2       6    0              x2    0.8      0
/2      12    1              x2    1.6      1
/2      25                   x2    1.2      1
                            X2    0.4      0
                            X2    0.8      0
                            X2    1.6      1
                            X2    1.2      1
```

$11001.000110011001100110011… = 1.1001\ 0001\ 1001\ 1001\ … \times 2^4$
$S = 1, EXP + BIAS = 4 + 1023 = 1027 = 1000\ 0000\ 011$
Fraction = 1001 0001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
1100 0000 0011 1001 0001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
= 0xC039 1999 9999 9999

8.   (15 points) For the MIPS assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

| Statement	Comment
sll $t0, $s0, 2	**$t0 ← f * 4**
add $t0, $s6, $t0	**$t0 ← &A[f]**
sll $t1, $s1, 2	**$t1 ← g * 4**
add $t1, $s7, $t1	**$t1 ← &B[g]**
lw $t3, 0($t0)	**$t3 ← A[f]**
addi $t2, $t0, 8	**$t2 ← &A[f + 2]**
lw $t2, 0($t2)	**$t2← A[f + 2]**
sub $t0, $t3, $t2	**$t0 ← A[f] − A[f+2]**
sw $t0, 0($t1)	**B[g] ← A[f] − A[f+2]]**

9.   (10 points) Provide the type and hexadecimal representation of the following instruction:
      `addu    $s3, $s1, $a3`

      type = Rtype , opcode = 0, rd = 19. rs = 17, rt = 7, shamt = 0, funct = 33
      000000 10001 00111 10011 00000 100001
      0x 0227 9821

10.   (15 points) When processor designers consider a possible improvement to the processor
      datapath, the decision usually depends on the cost/performance trade-off. In the following
      three problems, assume that we are starting with the datapath shown.

Element	I-Mem	Add	Mux	ALU	Regs	D-Mem	Control	ALU Control	Sign Extend	Shift Left 2
Latency (ps)	300 ps	100	40	120	150	300	100	85	15	30

   a.   (10 points) What is the clock cycle time for this datapath? **1100 ps**
   b.   (5 points) How would it change if the control time was 300 ps rather than 100 ps? **1295 ps**

Original Wire Delay
Original Element Delay
Wire & Element with Changed Control
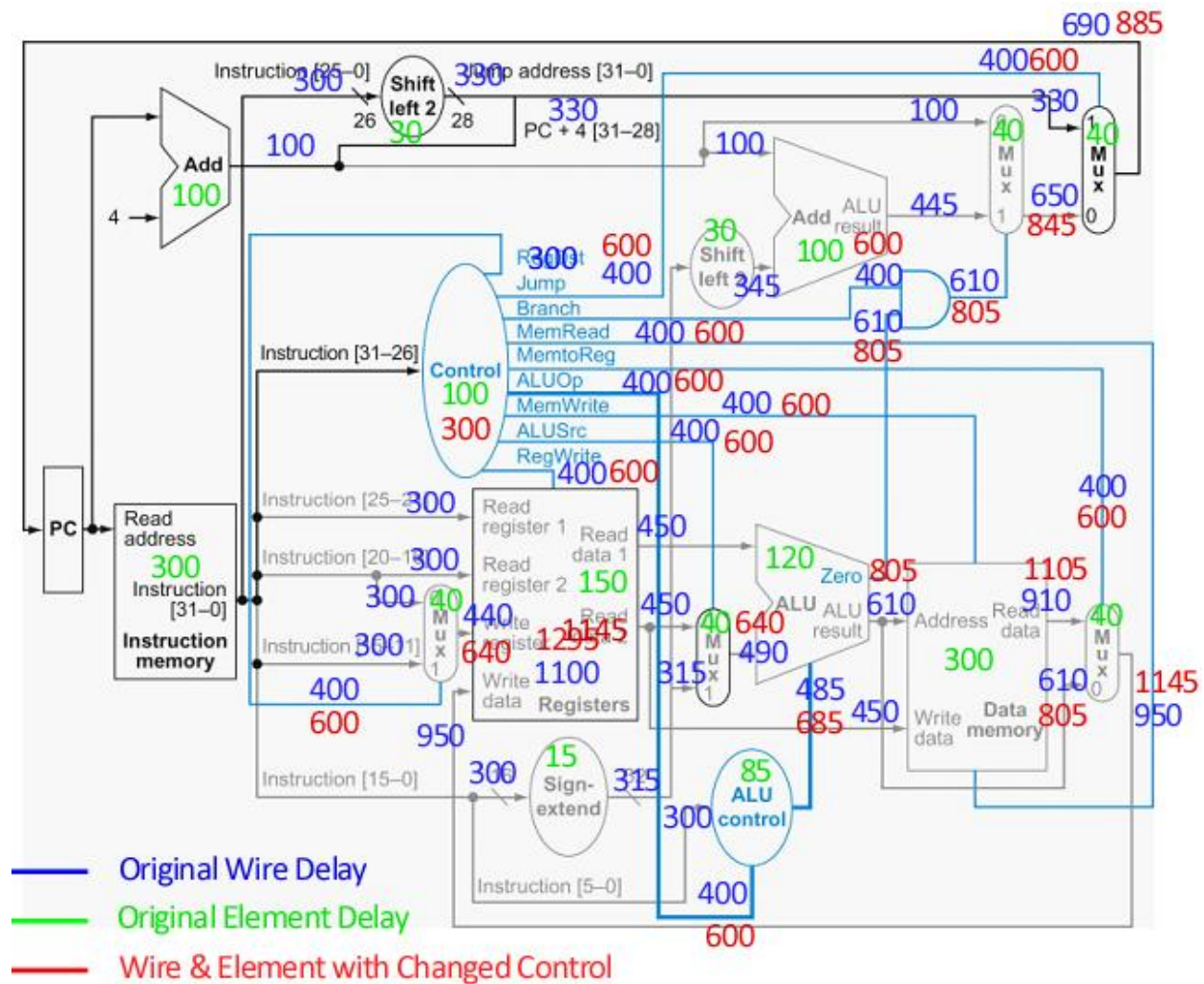
11. (20 points) Consider three different processors P1, P2, and P3 executing the same instruction set.

         P1 has a 2.8 GHz clock rate and a CPI of 1.5.

         P2 has a 2.5 GHz clock rate and a CPI of 1.2.

         P3 has a 4.0 GHz clock rate and has a CPI of 2.5.

a. (12 points) Which processor has the highest performance?

Since the processors are executing the same instruction set, $IC_{P1} = IC_{P2} = IC_{P3} = IC$

$$\frac{PP_a}{PP_b} = \frac{ET_b}{ET_a} = \frac{\dfrac{IC_b CPI_b}{CR_b}}{\dfrac{IC_a CPI_a}{CR_a}} = \frac{CPI_b CR_a}{CR_b CPI_a}$$

$$\frac{PP_1}{PP_2} = \frac{CPI_2 CR_1}{CR_2 CPI_1} = \frac{1.2 \times 2.8}{2.5 \times 1.5} = 0.896$$

$$\frac{PP_1}{PP_3} = \frac{CPI_3 CR_1}{CR_3 CPI_1} = \frac{2.5 \times 2.8}{4.0 \times 1.5} = 1.17$$

$$\frac{PP_2}{PP_3} = \frac{CPI_3 CR_2}{CR_3 CPI_2} = \frac{2.5 \times 2.5}{4.0 \times 1.2} = 1.3$$

Since PP2/PP1 > 1 and PP2/PP3 > 1, P2 has the highest performance.

b. (4 points) If P2 executes this program in 10 seconds, find the number of cycles and the number of instructions for P2.

ET = 10 s, ET = CC/CR, CC = ET x CR = 10s x 2.5 E9 cycles/s = 2.5 E10 cycles

CC = IC x CPI, IC = CC/CPI = 2.5 E10 cycles/(1.2 cycles/instruction) = 2.08 E10 instructions

c. (4 points) We are trying to reduce the execution time for P2 by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction for P2?

ET_{new} = 0.7 x 10 s = 7 s, CPI_{new} = 1.2 x 1.2 = 1.44, IC = 2.08 E10 instructions

ET = (IC x CPI)/CR

CR = (IC x CPI)/ET = (2.08 E10 instructions x 1.44 cycles/instruction)/7 s = 4.28 E9 cycles/s

12. (20 points) Consider the following code executing on a MIPS five stage pipeline that has full forwarding and in which branches are resolved in the MEM stage. Neglecting pipeline fill, how many cycles does it take to execute this code, given the branch taken/not taken information given in the comments?

```
(1)           add   $t1, $t1, $t0
(2)           lw    $t2, 0($t1)
(3)label1:    beq   $t2, $zero, label2 # not taken once, then taken
(4)           lw    $t3, 0($t2)
(5)           sw    $t3, 0($t1)
(6)           beq   $t3, $zero, label1 # taken
(7)           add   $t1, $t3, $t1
(8)label2:    sw    $t1, 0($t2)
```

Dependences

(1)-(2) forward
(2)-(3) stall one, then forward
(4)-(5) stall one, then forward
(4)-(6) forward
(7)-(8) never encountered, if branch not taken, forward
(6)-(3) flush 3
(3)-(8) flush 3

Cycle	IF	ID	EX	MEM	WB
1	add $t1				
2	lw $t2	add $t1			
3	beq $t2	lw $t2	add $t1		
4	lw $t3	beq $t2	lw $t2	add $t1	
5	lw $t3	beq $t2	bubble	lw $t2	add $t1
6	sw $t3	lw $t3	beq $t2	bubble	lw $t2
7	beq $t3	sw $t3	lw $t3	beq $t2	bubble
8	beq $t3	sw $t3	bubble	lw $t3	beq $t2
9	add $t1	beq $t3	sw $t3	bubble	lw $t3
10	sw $t1	add $t1	beq $t3	sw $t3	bubble
11	after	sw $t1	add $t1	beq $t3	sw $t3
12	beq $t2	bubble	bubble	bubble	beq $t3
13	lw $t3	beq $t2	bubble	bubble	bubble
14	sw $t3	lw $t3	beq $t2	bubble	bubble
15	beq $t3	sw $t3	lw $t3	beq $t2	bubble
16	sw $t1	bubble	bubble	bubble	beq $t2
17		sw $t1	bubble	bubble	bubble
18			sw $t1	bubble	bubble
19				sw $t1	bubble
20					sw $t1

Neglecting pipeline fill, it takes 16 cycles to execute this code.