The University of Alabama in Huntsville
ECE Department
CPE 431 01
Test 1 Solution
Fall 2014

1. (1 point) A _**supercomputer**_ belongs to the class of computers with the highest performance and cost.

2. (1 point) An _**operating system**_ is the supervising program that manages the resources of a computer for the benefit of the programs that run on the computer.

3. (1 point) _**Throughput**_ is the number of tasks completed per unit time.

4. (1 point) A _**benchmark**_ is a program selected for use in comparing computer performance.

5. (1 point) A _**pixel**_ is the smallest individual picture element.

6. (15 points) In a von Neumann architecture, groups of bits have no intrinsic meanings by themselves. What a bit pattern represents depends entirely on how it is used. What decimal number does the bit pattern 0xFDCB A987 represent if it is a floating point number? Use the IEEE 754 standard.

   0xFDCB A987 = 1111 1101 1100 1011 1010 1001 1000 0111
   S = 1, the number is negative
   E = 1111 1011
   F = 100 1011 1010 1001 1000 0111

   $(-1)^S \times (1.F) \times 2^{E-Bias} = (-1)^1 \times (1.100\ 1011\ 1010\ 1001\ 1000\ 0111) \times 2^{251-127} = -(CBA987)/2^{23} \times 2^{124} = -3.3839 \times 10^{37}$

7. (15 points) Translate the following C code to MIPS. Assume that the variables `f`, `g`, `h`, `i`, and `j` are given and 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. Assume that the elements of the arrays A and B are 4-byte words:

   ```
   B[j] = A[i] + B[8]
   ```

   ```
   lw    $t0, 32($s7)
   sll   $t1, $s3, 0x2
   add   $t1, $t1, $s6
   lw    $t1, 0($t1)
   add   $t0, $t0, $t1
   sll   $t1, $s4, 0x2
   add   $t1, $t1, $s7
   sw    $t0, 0($t1)
   ```

8. (10 points) Provide the type and hexadecimal representation of the following instruction:
`jr $s3`

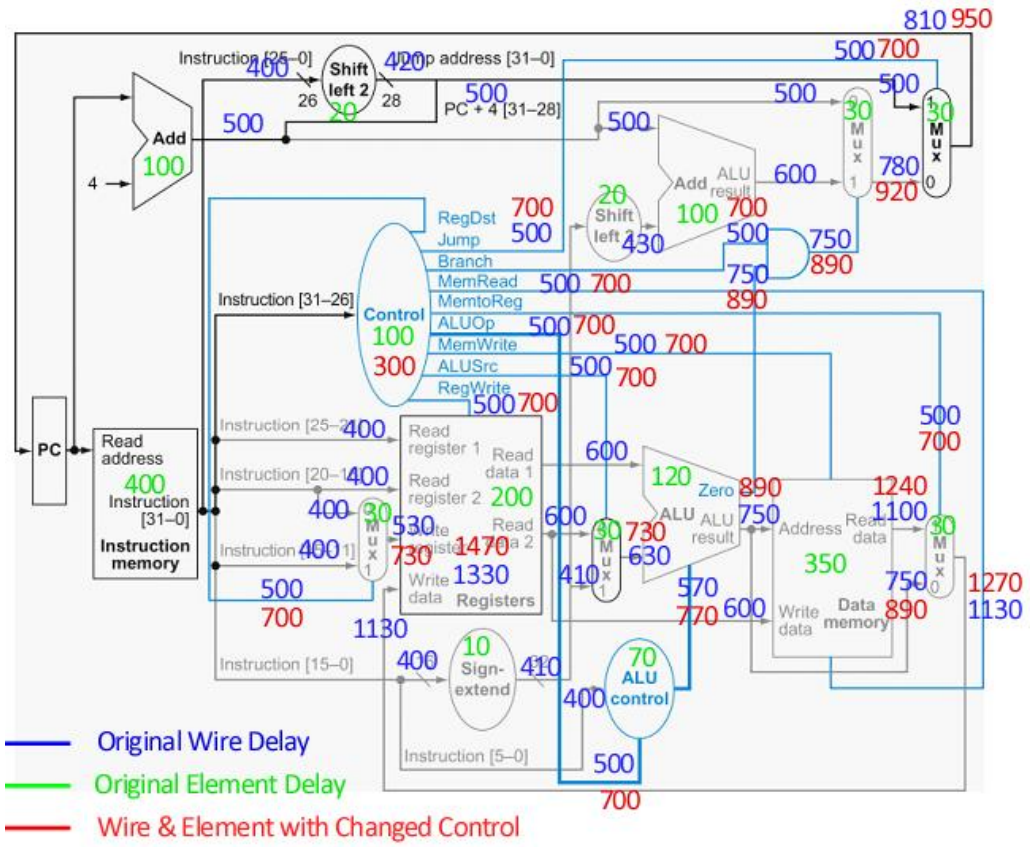
Type is R-type
opcode = 000000
rs = 10011
rt = 00000
rd = 00000
shamt = 00000
funct = 001000

Instruction word: 0000 0010 0110 0000 0000 0000 0000 1000
= 0x0260 0008

9. (20 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)	400 ps	100	30	120	200	350	100	70	10	20

a. (15 points) What is the clock cycle time for this datapath? **1330 ps**
b. (5 points)How would it change if the control time was 300 ps rather than 100 ps? **1470 ps**



10. (20 points) The results of the SPEC CPU2006 bxip2 benchmark running on an AMD Barcelona has an instruction count of 3.256E12, an execution time of 750 s, and a reference time of 9650 s. Suppose that we are developing a new version of the AMD Barcelona processor with a 3.5 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 680 s and the new SPECratios is 13.7.

a. (10 points) Find the new CPI.

$$ET = IC \times CPI \times CT = \frac{IC \times CPI}{CR}$$

$$CPI = \frac{ET \times CR}{IC} = \frac{680s \times 3.5 \times 10^9 \, cycles/s}{0.85 * 3.256 \times 10^{12} \, instructions} = 0.86 \frac{cycles}{instruction}$$

b. (10 points) Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.

$$CR = \frac{IC \times CPI}{ET} = \frac{0.85 * 3.256 \times 10^{12} \, instructions \times 0.86 cycles/instruction}{0.9 * 680s} = 3.889 \times 10^9 \frac{cycles}{second}$$

11. (15 points) Consider the following code executing on a MIPS five stage pipeline that has full forwarding and in which branches are resolved in the ID 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?

```
         lw    $t2, 0($t1)
label1:  beq   $t2, $zero, label2    # not taken once, then taken
         lw    $t3, 0($t2)
         beq   $t3, $zero, label1    #taken
         add   $t1, $t3, $t1
label2:  sw    $t1, 0($t2)
```

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

So, it takes 12 cycles to execute the code once the pipeline is full.