The University of Alabama in Huntsville
ECE Department
CPE 431 01
Test 1 Solution
Fall 2013

1. (1 point) An __**embedded**_ computer is one that is used for running one predetermined application or collection of software

2. (1 point) A _**datapath**_ is the component of the processor that performs arithmetic operations.

3. (1 point) _**Silicon**_ is a natural element that is a semiconductor.

4. (1 point) The _**stack**_ is a data structure for spilling registers organized as a last-in-first-out queue.

5. (1 point) The _**global pointer**_ is the register that is reserved to point to the static area.

6. (10 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 0xF359 0AC9 represent if it is a floating point number? Use the IEEE 754 standard.

   0xF359 0AC9 = 1111_0011_0101_1001_0000_1010_1100_1001

   Sign = 1, number is negative
   Exponent = 11100110 = 230
   Fraction = 101_1001_0000_1010_1100_1001

   $(-2)^{Sign} \times 2^{Exponent - Bias} \times 1.Fraction = (-1)^1 \times 2^{230-127} \times 1.101\_1001\_0000\_1010\_1100\_1001)$
   $= -1 \times 2^{103} \times 0xD90AC9/2^{22}$
   $= -1.7196 \times 10^{31}$

7. (15 points) Compilers can have a profound impact on the performance of an application on a given processor. For the same program, two different compilers are used. The table shows the execution time of the two different compiled programs. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.

Compiler A		Compiler B	
# Instructions	**Execution time**	**# Instructions**	**Execution Time**
1.00 E+09	1.3 s	1.50E+09	1.8 s

   A new compiler is developed that uses only 600 million instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using Compiler A or B on the original processor?

   CT = 1 ns
   ET = IC x CPI x CT
   CPI = ET/(IC x CT)
   $CPI_A = 1.3 \text{ s} / (1.00 \times 10^9 \times 1 \times 10^{-9} \text{ s/cycle}) = 1.3$ cycles/instruction
   $CPI_B = 1.8 \text{ s} / (1.50 \times 10^9 \times 1 \times 10^{-9} \text{ s/cycle}) = 1.2$ cycles/instruction

$ET_{new\_compiler}$ = 600 x 10^6 instructions x 1.1 cycles/instruction x 1 ns = 600 ms
Speedup of new_compiler compared to compiler A = $ET_A/ET_{new\_compiler}$ = 1.3/0.66 = 1.97
Speedup of new_compiler compared to compiler B = $ET_B/ET_{new\_compiler}$ = 1.8/0.66 = 2.73

8. (10 points) If the current value of the PC is 0x0FFF FFFC, can you use a single jump instruction to get to the PC address 0x0FBE 4000?

For PC = 0x0FFF FFFC, PC + 4 = 0x1000 0000. A jump address is formed by taking 26 bits from the jump instructions, shifting it left by 2 and concatenating it with the 4 most significant bits of PC + 4. For this PC + 4, the range is 0x1000 0000 to 0x1FFF FFFC. Though, 0x0FBE 4000 is on a word boundary, it does not fall in the range for a jump.

9. (15 points) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
150 ps	220 ps	300 ps	280 ps	140 ps

(a) (8 points) What is the total latency of an LW instruction in a pipelined and non-pipelined processor?
(b) (7 points) If we can split one stage of the pipelined datapath into two new stages, each with half of the latency of the original stage, which stage would you split and what is the new latency for an LW instruction

(a) Pipelined – Clock cycle time = 300 ps, latency = 5 stages x 300 ps/stage = 1500 ps
Non-pipelined – Clock cycle time = 150 ps + 220 ps + 300 ps + 280 ps + 140 ps = 1090 ps, latency = 1090 ps

(b) Split the EX stage into two stages of 150 ps each. Clock cycle time = 280 ps, latency is now 6 stages x 280 ps/stage = 1680 ps.

10. (15 points) Find the shortest sequence of MIPS instructions that perform the same operation as `rpt $t2, loop`, where the meaning is

```
if (R[rs] > 0)
{
    R[rs] = R[rs] – 1
    PC = PC + 4 + BranchAddr
}
```

```
loop   slt   $t0, $zero, $t2
       beq   $t0, $zero, skip
       addi  $t2, $t2, -1
       beq   $t2, $t2, loop
skip
```

11. **(**15 points**)** Consider two different implementations, P1, and P2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. P1 has a clock rate of 4 GHz, and P2 has a clock rate of 6 GHz. The average number of cycles for each instruction class for P1 and P2 are listed in the following table.

	CPI Class A	CPI Class B	CPI Class C
P1	1	2	3
P2	2	3	5

If the number of instructions executed in a certain program is divided among the three instruction classes so that the number of cycles for each class is the same on processor P1, how much faster is P2 than P1?

For P1, CCA = CCB = CCC. The least common multiple of 1, 2, and 3 is 6. So, in each 18 cycles of execution, there will be 6 class A instructions, 3 class B instructions, and 2 class C instructions, giving weights of 6/11, 3/11, and 2/11.

$$ET_{P1} = \frac{IC * \left(\frac{6}{11} * 1 + \frac{3}{11} * 2 + \frac{2}{11} * 3 \right)}{4 \times 10^9} = \frac{\frac{18IC}{11}}{4 \times 10^9} = \frac{18IC}{44 \times 10^9}$$

$$ET_{P2} = \frac{IC * \left(\frac{6}{11} * 2 + \frac{3}{11} * 3 + \frac{2}{11} * 5 \right)}{6 \times 10^9} = \frac{\frac{31IC}{11}}{6 \times 10^9} = \frac{31IC}{66 \times 10^9}$$

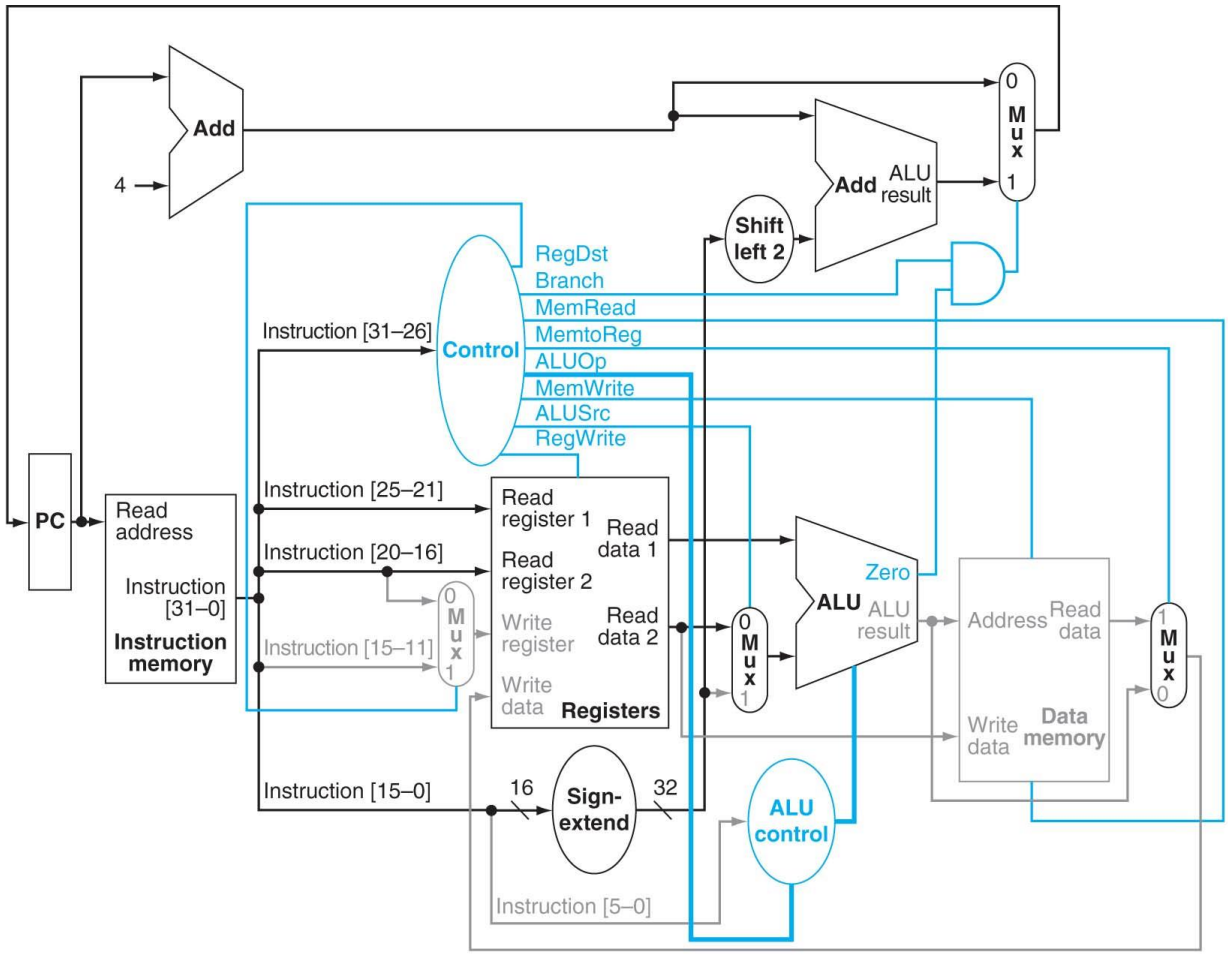
$$\frac{P_{P2}}{P_{P1}} = \frac{ET_{P1}}{ET_{P2}} = \frac{\frac{18IC}{44 \times 10^9}}{\frac{31IC}{66 \times 10^9}} = \frac{18IC}{44 \times 10^9} \frac{66 \times 10^9}{31IC} = 0.871$$

P2 is 0.71 times as fast as P1.

12. (15 points) Add a variant of the `lw` instruction which sums the contents of two registers to obtain the address of the data and which uses the R format to the single-cycle datapath shown in the figure below. Add any necessary datapaths and control signals and show the necessary additions to the table of control signals given.

```
lwr   $t1, $a2, $s6
$t1 ← MEM[$a2+$s6]
```

36	6	22	9	0	32



No additions need to be made to the datapath.

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALU Op1	ALU Op0	
R-format	1	0	0	1	0	0	0	1	0	
lw	0	1	1	1	1	0	0	0	0	
sw	d	1	d	0	0	1	0	0	0	
beq	d	0	d	0	0	0	1	0	1	
lwr	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	

d – don't care