



## School of Electrical Engineering and Computer Science

**CptS260: Introduction to Computer Architecture - Fall 2015**

**Final Exam, Duration: 120 Minutes, December 14, 2015**

**NAME:** Solution

**ID:** Final 2015 Fall

Notes:

- This test includes 8 questions each of which may have multiple parts (a, b, ... etc).
- You may bring a double-side letter-size cheat-sheet and a calculator to the test. No other resources are allowed! In particular, NO textbook, lecture notes, internet access, smartphone usage, ...etc are allowed!
- Make sure to write your name and WSU ID down on all pages.
- You may use both sides of each page if needed!
- Show your work for each question. Even if you don't know exact answer to a question, show your work to get partial credit.
- MIPS reference data is provided on the last page of this test.
- Please make sure to complete the course evaluation before the deadline of 12/18/2015.

Question	1	2	3	4	5	6	7	8
Total Points	10	9	12	16	10	15	15	13
Points Earned								

**Total Points Earned:** .....

1. **Concepts (10 points).** For each sentence below, indicate if it is a *True* or *False* statement.

- a. A CPU with a faster clock frequency always has higher performance than one with a slower clock. *False*
- b. The 'beq' instruction always modifies the program counter register regardless of whether or not the two registers under comparison hold equal contents. *True*
- c. Compared to an un-pipelined processor, a pipelined architecture usually has a longer clock cycle. *False*
- d. Structural hazards in a pipeline are due to data dependency of the instructions. *False*
- e. If we increase the number of stages in a pipeline, throughput of our system will definitely increase. *False*
- f. Flash memory is as fast as SRAM (Static Random Access Memory) *False*
- g. Temporal locality refers to the idea that data in memory which is close in address to memory data which has recently been accessed is likely to be accessed soon. ~~True~~ *True*
- h. Fully associative cache memory has a higher hit time compare to a direct mapped cache memory of the same size. *True*
- i. For primary (L1) cache in multilevel cache systems, focus is on low miss rate. *False*
- j. High associativity in a cache reduces compulsory misses. *False*

2. Computer Performance (9 points). Assume that we have a processor with 10 Kbytes of D-Cache (Data Cache Memory) and 10 Kbytes of I-Cache (Instruction Cache Memory). The amount of miss rate for both of these cache memories is 10%. Furthermore, hit time is 1ns (one nano-second) for both caches.
- (3 points) Assume that the main memory access time is 50ns. What is the average memory access time (AMAT) for each of D-Cache and I-Cache?
  - (3 points) Assume that 45% of the instructions are memory accesses. Assuming that the base/ideal CPI is 1, calculate the actual CPI for this processor. You may assume that the clock cycle time is equal to the hit time of the cache.
  - (3 points) If we add an L2 cache memory with 2% miss rate and 5ns access time, what would be the new actual CPI?

a)  $AMAT = \text{hit\_time} + \text{miss rate} \times \text{mispenalty}$

$$= 1\text{ ns} + 10\% \times 50\text{ ns} = \boxed{6\text{ ns}}$$

\* It's the same for both I-cache, D-cache

b)  $CPI = CPI_{\text{ideal}} + \frac{\text{stalls}_{\text{Icache}}}{\text{Icache}} + \frac{\text{stalls}_{\text{Dcache}}}{\text{Dcache}}$

$$\text{Icache} : 0.10 \times \frac{50\text{ ns}}{1\text{ ns}} = 5$$

$$\text{D-cache} : 0.1 \times 0.45 \times \frac{50\text{ ns}}{1\text{ ns}} = 2.25$$

$$\Rightarrow CPI = 1 + 5 + 2.25 = \boxed{8.25}$$

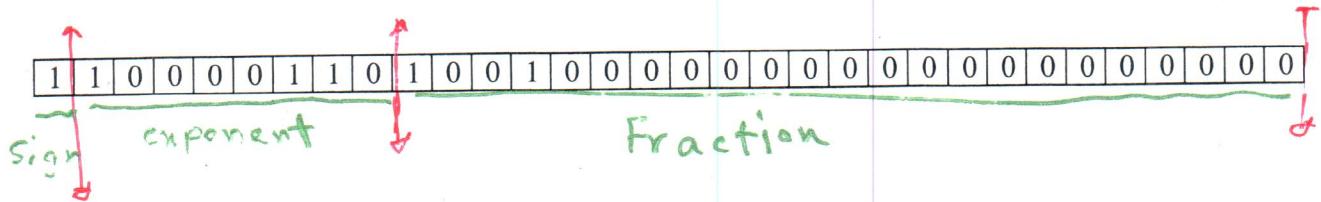
c) New mispenalty :  $5\text{ ns} + \frac{2}{100} \times 50\text{ ns} = 6\text{ ns}$

$$\text{Icache} : 0.1 \times \frac{6\text{ ns}}{1\text{ ns}} = 0.6$$

$$\text{D-cache} : 0.1 \times 0.45 \times \frac{6\text{ ns}}{1\text{ ns}} = 0.27$$

$$CPI = 1 + 0.6 + 0.27 = \boxed{1.87}$$

3. **Number Representation (12 points).** Assume the following 32-bit binary represents a floating point number using single precision IEEE 754 standard.
- (5 points) Specify fraction, exponent, and sign bit of this number in binary.
  - (7 points) Find the equivalent decimal number. Show your work!



$$\text{sign} \quad \text{exponent } -127 \\ (-1) \times 1.\text{Fraction} \times 2^{\text{exponent}-127}$$

$$\Rightarrow -1 \times 1.10010... \times 2^{134-127} = -(11001000) = -200d$$

4. **MIPS Assembly Programming (16 Points)**. Trace the following assembly program. Assume that the registers' initial values and memory content are as shown in the tables. The program starts from location 10000d in the instruction memory.
- (8 points) Show the final state of processor including registers' content/data, and memory contents.
  - (4 points) What is the general functionality of this program?
  - (4 points) What is the final value of PC at the end? What is the value of address loop1?

Address	Initial Data	Final Data
100d	1d	17d
104d	2d	28d
108d	3d	41d
112d	4d	56d
116d	5d	73d
...	...	...
200d	8d	8d
204d	12d	12d
208d	16d	16d
212d	20d	20d
216d	24d	24d

Register	Initial Data	Final Data
\$t0	0	120d
\$t1	0	220d
\$t2	0	5d
\$t3	0	5d
\$s0	0	5d
\$s1	0	48d
\$s2	0	73d

b) get two arrays with five elements:

$$x_i = x_i^2 + 2y_i \text{ for } i=1:5$$

$$C) PC_0 = 10000d$$

$$PC_{final} = PC_0 + 4 \times \text{Number of Ins}$$

$$\Rightarrow PC_{final} = 10000 + 4 \times 15$$

$$PC_{final} = 10060d$$

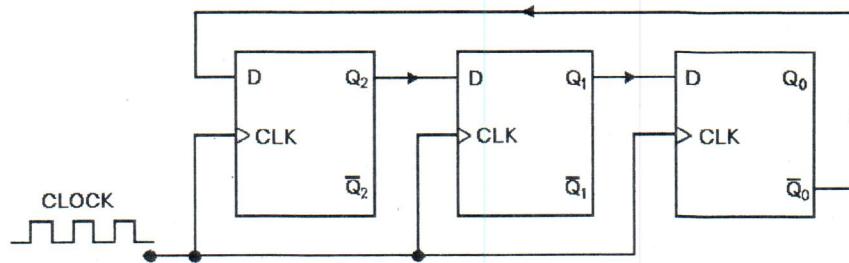
```

1      li $t3, 5d
2      li $t2, 0d
3      la $t0, 100d
4      la $t1, 200d
5      loop1: lw $s0, 0($t0)
6      lw $s1, 0($t1)
7      mult $s0, $s0
8      mflo $s2
9      sll $s1, $s1, 1d
10     add $s2, $s2, $s1
11     sw $s2, 0($t0)
12     addi $t0, $t0, 4
13     addi $t1, $t1, 4
14     addi $t2, $t2, 1
15     bne $t2, $t3, loop1
16     exit:

```

5. **Basics of Digital Design (10 points).** The following figure shows a sequential circuit with three D Flip Flops with a common input clock. Note that the inverted output of the last flip-flop ( $\bar{Q}_0$ ) is connected to the input of the first flip-flop (D input of the far left flip flop). Assume that the flip flops are initialized at '0'. That is,  $Q_2 = Q_1 = Q_0 = 0$  during Cycle 0. This means the output of this circuit is initially  $Q_2Q_1Q_0 = '000'$ .

- (5 points) Compute the value of each output signal ( $Q_2$ ,  $Q_1$ , and  $Q_0$ ) for 10 cycles using the table below.
- (5 points) Convert the 3-bit binary  $Q_2Q_1Q_0$  to its equivalent decimal in the table. How many unique output states (i.e.,  $Q_2Q_1Q_0$ ) does this circuit produce?



Clock Cycle	$Q_2$	$Q_1$	$Q_0$	Decimal ( $Q = Q_2Q_1Q_0$ )
0	0	0	0	0
1	1	0	0	4
2	1	1	0	6
3	1	1	1	7
4	0	1	1	3
5	0	0	1	1
6	0	0	0	0
7	1	0	0	4
8	1	1	0	6
9	1	1	1	7
10	0	1	1	3

These are 6 distinct states: 0, 4, 6, 7, 3, 1

6. **Simple MIPS Architecture (15 points).** The set of instructions of a specific architecture is called ISA (Instruction Set Architecture). MIPS has its own ISA. You know that the *LW* instruction will get a constant value and a register content, and the summation of these two numbers will be used as the address of the memory location from which a word is read. Now assume that you are asked to add a new instruction to this processor with the following specifications:

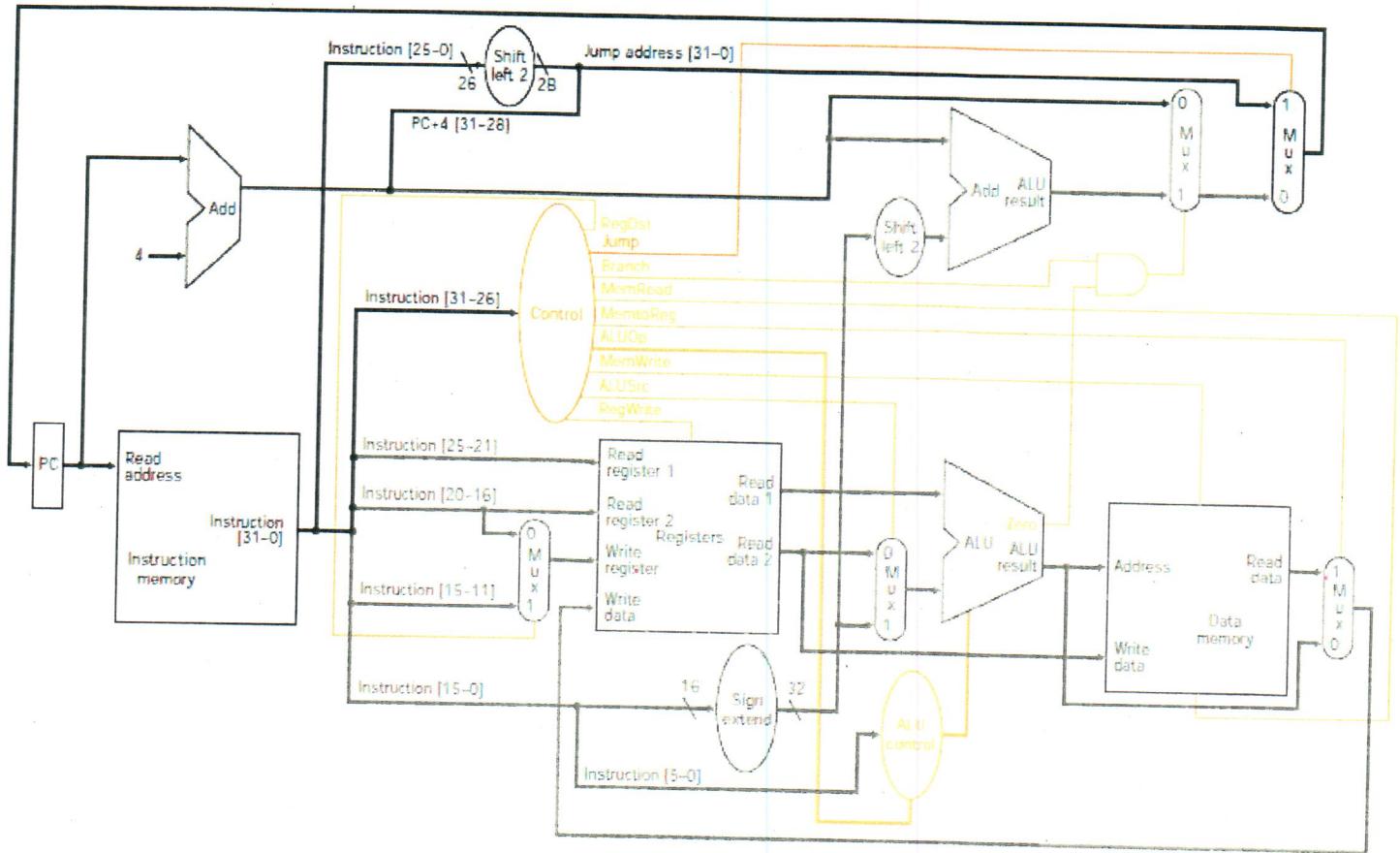
**Instruction: *LWI Rt, Rd(Rs)***

**Interpretation:  $Rt = \text{MEM} [ \text{Reg}[Rd] + \text{Reg}[Rs] ]$**

Difference between this instruction (i.e., *LWI*) and regular *LW* is that the address is based on the data of two registers instead of the constant value and one register. Answer the following questions assuming that we want to implement *LWI*. MIPS single cycle processor is shown on the next page.

- a. (5 points) Which existing blocks will be needed for *LWI*?
- b. (5 points) Which new function or block you need to add (if any) for this instruction to the single cycle processor. Explain your answer.
- c. (5 points) What changes need to be made in control unit for this instruction (if any)? Explain your answer.

- a) The same set of blocks as *LW* is needed, except we don't need the sign-extend block, as we don't have a constant value for address calculation
- b) No new block is needed, we only need to allocate an instruction code for *LWI*
- c) Control unit should be redesigned to generate correct control signals for *LWI*



MIPS Single cycle processor

#### Instruction Format for R-Type, I-Type, and Jump instructions

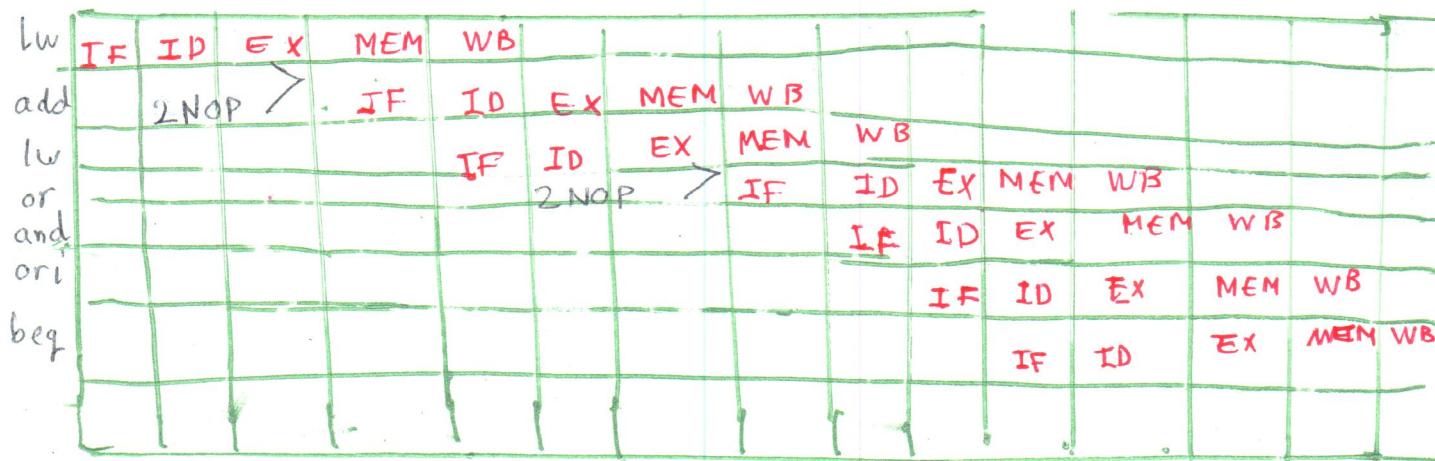
R	opcode	rs	rt	rd	shamt	funct	
	31 26 25	21 20	16 15	11 10	6 5	0	
I	opcode	rs	rt		immediate		0
	31 26 25	21 20	16 15				0
J	opcode			address			0
	31 26 25						0

7. Pipelining (15 points). Consider the following assembly program running on a 5-stage pipelined MIPS.

- (7 points) How many stalls are needed assuming that the pipeline does not have a forwarding unit? The remaining elements of the pipeline are exactly the same as the 5-stage pipeline discussed in the class. Explain your answer.
- (8 points) How many stalls are needed assuming that the forwarding unit is also implemented on the pipelined architecture? Reorder the program code or use other registers to minimize the number of stalls assuming the 5-stage pipeline discussed in the class.

	Lable1:	\$2, 0(\$1)	\$1, \$2, \$3	\$3, 0(\$4)	\$4, \$3, \$5	\$6, \$6, \$7	\$7, \$7, 1d	\$2, \$3, Label1	dependencies (need Nops)
lw									
add									
lw									
or									
and									
ori									
beq									

a)



without forwarding, we need to add 4 Nop's

b) between (lw, add), we need 1 Nop's

between (lw, or), we need 1 Nop's

Label1 : lw \$2, 0(\$1)

and \$6, \$6, \$7

add \$1, \$2, \$3

lw \$3, 0(\$4)

ori \$7, \$7, 1

or \$4, \$3, \$5

beq \$2, \$3, Label1

⇒ no stall is needed!

8. **Cache Memory (13 points).** Consider a 2-way associative cache memory with a total capacity of 32 words. The block size is 2 words and the processor has an 8-bits address bus. The following table is intended to show the cache content (in columns labeled as 'Way-1' and 'Way-2') as well as the line/set address (in column labeled as 'Index').

Index	Way-1			Way-2		
	Valid	Tag	Data	Valid	Tag	Data
000	1	0010	MEM[32]			
			MEM[33]			
001						
010						
011						
100						
101	1	0010	MEM[42]			
			MEM[43]			
110	1	0000 0010	MEM[12] MEM[44]	1	0001 0000	MEM[28] MEM[29]
			MEM[13] MEM[45]			
111	1	0000	MEM[14] MEM[15]			

- a. (3 points) Assume that the cache contain data for two memory references with addresses 13 and 43. Update the above table for these addresses. (Hint: when you reference a memory location, you will need to move a block that contains this memory address).

$$13 = \underbrace{000}_{\text{tag}} \underbrace{01101}_{\text{Index}}$$

$$43 = \underbrace{0010}_{\text{tag}} \underbrace{1011}_{\text{Index}}$$

- b. (8 points) Now assume that the CPU accesses the following memory references (i.e., numbers from left to right represent memory addresses)

12, 29, 33, 45, 12, 32, 44, 14, 13, 15

Update the table from part (a) with index, valid bit, tag, and data fields in appropriate rows/places to show the cache content. Also, fill out the following table and mention if a reference to each location/address is a cache hit or cache miss. Assume an LRU (Least Recently Used) replacement policy for this cache.

Word Address	Binary Address	Tag	Index	Hit/Miss
12	00001100	0000	110	hit
29	00011101	0001	110	miss
33	00100001	0010	000	miss
45	00101101	0010	110	miss
12	00001100	0000	110	miss
32	00100000	0010	000	hit
44	00101100	0010	110	hit
14	00001110	0000	111	miss
13	00001101	0000	110	hit
15	00001111	0000	111	hit

- c. (2 points) What is the amount of miss rate (in percentage) for the memory accesses in part (b)?

$$\text{miss rate} = \frac{5}{10} = 50\%$$

