ข้อสอบปลายภาคการศึกษาวิชา Computer Architecture เทอมต้น (มิ.ย.- ก.ย.) พ.ศ. 2554

ข้อสอบมี 6 ข้อใหญ่ แต่ละข้อใหญ่มีคะแนนเท่ากัน (ความยากง่ายอาจจะไม่เท่ากัน) เวลาทำทั้งหมด 120 นาที ไม่จำกัดเอกสาร หรือ อุปกรณ์การคำนวณ

หมายเหตุ

* ใส่หน่วยให้ถูกต้อง (รวมทั้ง ไม่ใส่หน่วย กับ ค่าที่ไม่มีหน่วยด้วย)
* บางข้ออาจจะง่ายๆมาก บางข้ออาจจะซับซ้อนบ้าง ขึ้นกับธรรมชาติของเนื้อหา

1. Data representation. (2 ข้อย่อย: 4 คำถาม)

1.1 ถ้า register $s1 เก็บค่า 0x**FFFFFFFC** แล้ว:

ก. ค่านี้เท่ากับค่าเท่าไรในเลขฐานสอง _____

ข. ถ้าค่านี้คือ `signed integer (2's complement)` จะมีค่าเท่ากับ เท่าไรในฐานสิบ _____

1.2 ถ้า register $s0 เก็บค่า 0x**0109502A** (ฐาน 16) แล้ว:

ก. ค่านี้เท่ากับค่าเท่าไรในเลขฐานสอง _____

ข. ถ้าค่านี้คือ คำสั่งในชุดคำสั่งของ `MIPS 32` ค่าของ `opcode` จะเป็นเท่าไร _____ **(ฐาน 2)**

2. Arithmetic for computers. (2 ข้อย่อย: 10 คำถาม)

2.1 สำหรับระบบ 4 bits (word size = 4 bits) แต่ละ word จะมีค่าได้ตั้งแต่ "0000" (ฐาน 2) ไปจนถึง "1111" (ฐาน 2)

(ก.) ค่าต่ำที่สุดและสูงที่สุดที่แต่ละ word สามารถแสดงได้จะคือเท่าไรบ้าง ถ้าค่า 2 ค่านี้แสดงในรูปแบบ unsigned number ค่าต่ำสุด = _____(ฐาน 10) . ค่าสูงสุด = _____ (ฐาน 10) .

(ข.) ค่าต่ำที่สุดและสูงที่สุดที่แต่ละ word สามารถแสดงได้จะคือเท่าไรบ้าง ถ้าค่า 2 ค่านี้แสดงในรูปแบบ signed number (2's complement)

ค่าต่ำสุด = _____ (ฐาน 2) = _____ (ฐาน 10).

ค่าสูงสุด = _____ (ฐาน 2) = _____ (ฐาน 10).

(ค.) ถ้าค่า 2 ค่านี้แสดงในรูปแบบ unsigned number แล้วบวกกัน จะเกิด overflow เมื่อผลบวกเกินเท่าไร (คำถามง่าย) _____

(ง.) ถ้าค่า 2 ค่านี้แสดงในรูปแบบ unsigned number แล้วลบกัน จะเกิด overflow เมื่อผลลบต่ำกว่าเท่าไร (คำถามง่าย) _____

2.2 จากรูป 1-bit ALU ดัง Figure 1 (ข้างล่าง) ถ้าต้องการทำ operation exclusive or ซึ่งมีค่าตรรกะดังตารางตรรกะข้างล่าง จะต้องกำหนดค่า Binvert และ Operation เป็นเท่าใด [hint: คำถามนี้ง่าย]

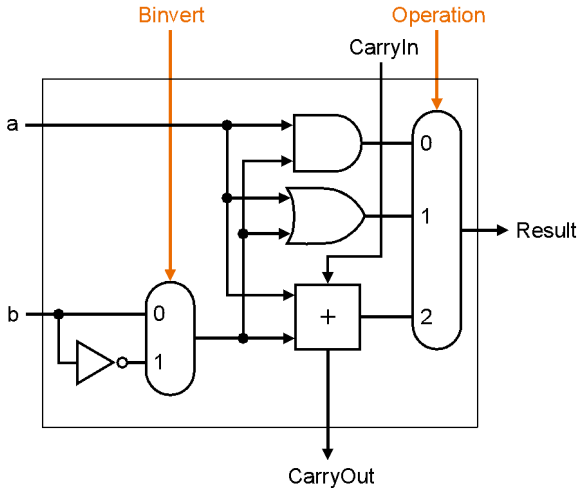

Figure 1: (Fig. 4.16 P&H 2e) A 1-bit ALU.

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0
ตารางตรรกะของ exclusive or.		

CarryIn = 0
Binvert = ____
Operation = ____

3. Performance measurement. (3 ข้อย่อย)

เครื่องคอมพิวเตอร์ A และ B ที่เป็นสถาปัตยกรรม MIPS โดย

เครื่อง A เป็น single-cycle machine รันที่ความถี่ 500 MHz ทุกคำสั่งใช้ 1 clock cycle เท่ากัน

เครื่อง B เป็น multi-cycle machine รันที่ความถี่ 2.1 GHz จำนวน clock cycles ที่ใช้สำหรับแต่ละคำสั่งจะเป็นดังตารางข้างล่าง

ชนิดคำสั่ง	จำนวน clock cycle ที่ใช้ในการทำคำสั่ง
arithmetic-logic instruction เช่น slt, sub	4
load instruction เช่น lw	5
store instruction เช่น sw	4
branch instruction เช่น beq	3
jump instruction เช่น j	3

ถ้าเครื่อง A และ B ถูกนำมารันชุดคำสั่งข้างล่าง

The program is:

```
                lw $t0, 0($s0)                # line 1
                lw $t1, 4($s0)                # line 2
                slt $t2, $t0, $t1             # line 3
                beq $t2, $zero, LAB_t1ELTt0   # line 4
                sub $t3, $t1, $t0             # line 5
                j LAB_SAVE                    # line 6
LAB_t1ELTt0:    sub $t3, $t0, $t1             # line 7
LAB_SAVE:       sw $t3, 8($s0)                # line 8
                lw $t0, 8($s0)                # line 9
```

ตอบคำถามดังนี้

3.1 ค่าเวลา clock period ของเครื่อง A และ B เป็นเท่าไร

A's clock period = _____; B's clock period = _____

3.2 จำนวน clock cycles ที่ใช้ในการรันโปรแกรมจะเป็นเท่าไร โดยถ้าเครื่องที่ใช้คือ เครื่อง A และ เครื่อง B เมื่อค่าโปรแกรมทำการ branch และไม่ทำการ branch เติมค่าลงในตารางให้สมบูรณ์

(ตัวอย่าง แถวสุดท้าย หลักที่ 3 แสดง จำนวน clock cycles เมื่อรันโดยเครื่อง A ถ้าโปรแกรมไม่ทำการ branch)

line	ชนิดคำสั่ง	เครื่อง A		เครื่อง B	
		branch is not taken	branch is taken	branch is not taken	branch is taken
1	lw	1			
2	lw	1			
3	slt	1			
4	beq	1			
5	sub	1			
6	j	1			

7	sub	-			
8	sw	1			
9	lw	1			
รวมจำนวน clock cycles =		8			

3.3 ถ้ารันโปรแกรมนี้ เครื่องไหนจะเร็วกว่ากัน

ก. เวลาที่ใช้ในการรันโปรแกรมนี้จะเป็นเท่าไร (เติมตาราง) ถ้ารันโดยเครื่อง A และเครื่อง B เมื่อไม่ทำและทำการ branch ดังตารางข้างล่าง

ข. ค่า CPI ของโปรแกรมนี้จะเป็นเท่าไร (เติมตาราง) ถ้ารันโดยเครื่อง A และเครื่อง B เมื่อไม่ทำและทำการ branch

ค. เวลาเฉลี่ยของการใช้โปรแกรมนี้ เมื่อรันด้วยเครื่อง A และ B เป็นเท่าไร (เติมตาราง)

ง. ค่า performance ของ A เมื่อเทียบกับ B จะเป็นเท่าไร (เติมตาราง)

	เครื่อง A		เครื่อง B	
	branch is not taken	branch is taken	branch is not taken	branch is taken
(ก.) เวลาในการรันโปรแกรม (execution time)				
(ข.) CPI				
(ค.) เวลาเฉลี่ยตามสัดส่วนการใช้งาน ถ้าสมมติ เปอร์เซ็นการใช้จากสถิติ: branch not taken 50%; branch taken 50%				

4. Datapath and control. (2 ข้อย่อย)

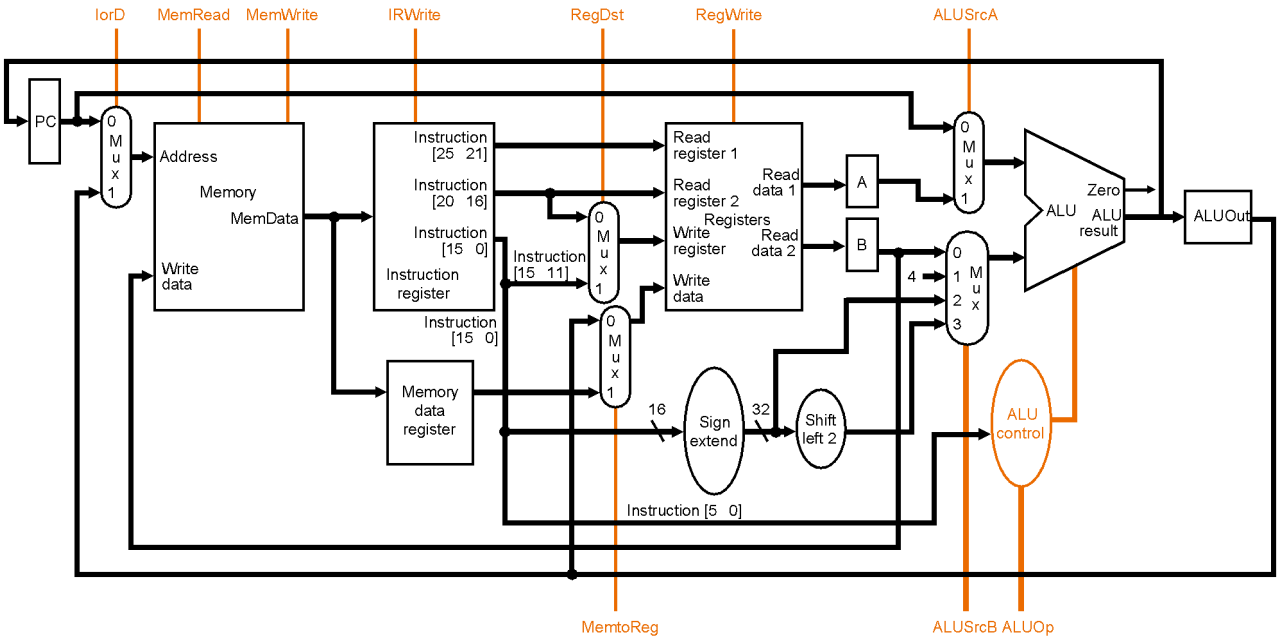

Figure 2: (Fig. 5.32 P&H 2e) The multicycle datapath and control for r-type, load and store instructions.

Step	Action for R-type instructions	Action for memory-reference instructions	
1	**IR <= Memory[PC]** **PC <= PC + 4**		
	`IorD = 0, MemRead, IRWrite, ALUSrcA = 0, ALUSrcB = 01, ALUOp =00, PCSource = 00, PCWrite`		
2	**A <= Reg[IR[25:21]]** **B <= Reg[IR[20:16]]** **ALUOut <= PC + (sign-extend (IR[15:0]) << 2)**		
	`ALUSrcA = 0,  ALUSrcB = 11, ALUOp =00`		

3	**ALUOut <= A op B**	**ALUOut <= A + sign-extend (IR[15:0])**
		`ALUSrcA = 1, ALUSrcB =10,`
		`ALUOp = 00`	

	`ALUSrcA = 1, ALUSrcB = 00, ALUOp = 10`		
4	**Reg[IR[15:11]] <= ALUOut**	**Load:** **MDR <= Memory[ALUOut]**	**Store:** **Memory[ALUOut]** **<= B**
	`RegDst = 1, MemtoReg = 0, RegWrite`	`IorD = 1, MemRead`	`IorD = 1, MemWrite`
5		**Load: Reg[IR[20:16]] <= MDR**	
		`RegDst = 0, MemtoReg = 1, RegWrite`	
Control Summary Table: Summary of the steps taken to execute any instruction class with corresponding control signals.			

จาก Figure 2 และ Control Summary table ข้างบน ซึ่งแสดง data path และ control signals ของฮาร์แวร์ implementing คำสั่ง r-type, load, และ store ตอบคำถาม 4.1 - 4.3.

4.1 Assigning control signal. (เติมตาราง)

เมื่อไร (step ไหน/คำสั่งอะไร) ที่สัญญาณ IorD ต้องถูกกำหนดเป็น '1', เป็น '0' หรือ จะถูกกำหนดเป็นอะไรก็ได้ ("don't care" ใช้ตัวย่อ 'x') เติมค่าลงในตารางข้างล่าง

ตัวอย่าง สัญญาณ IorD ใน Step 1 จะเป็นศูนย์ (ดังแสดงใน หลักที่ 2 ของตารางคำตอบ)

	step 1	step 2	step 3	step 4	step 5
r-type	0				
lw	0				
sw	0				

ตารางคำตอบ ข้อ 4.1 ค่าของสัญญาณควบคุม IorD.

4.2 Tracing data flow. (12 คำถามเล็ก)

สมมติ registers ต่างๆ เก็บค่าดังตารางข้างล่างนี้ (คอลัมน์ 1 แสดงเบอร์รีจีสเตอร์ คอลัมน์ 2 แสดงค่าที่เก็บอยู่ในรีจีสเตอร์ตัวนั้นๆ)

register index	register content
0	0x00000000
1	0x0B008004
:	:
16	0x00000008
17	0x00000009
18	0x000000AA
19	0x0000B00B
:	:
31	0x5000600D

ในการทำคำสั่ง add $s0, $s1, $s2 ซึ่งมี machine code เป็น 0x**02328020** (ฐาน 16) ตอบคำถามต่อไปนี้ (ดู Figure 2 และ ตาราง Control Summary Table ประกอบ)

ก. ค่าของ Instruction Register (IR) จะเป็นเท่าไร ใน step ที่ 2 ของการ execution (เขียนคำตอบในรูปของเลขฐาน 16 และ ฐาน 2) [hint: คำถามง่าย]

IR = _____ (ฐาน 16) หรือ _____ (ฐาน 2).

ข. ใน step ที่ 2 ค่า bits ที่ 25 - 21 ของ IR เป็นเท่าไร และค่านี้อ้างถึง register ตัวไหน

Bits 25 - 21 = _____ ซึ่งอ้างถึง รีจีสเตอร์ _____.

ค. ตอนปลายของ step ที่ 2 ค่าของ Read data 1 ของรีจีสเตอร์ไฟล์ เป็นเท่าไร

Read data 1 = _____ (ฐาน 16).

ง. ใน step ที่ 3 ค่าของ register A เป็นเท่าไร และ inputs ทั้งสองของ ALU มีค่าเท่าไร

A = _____(ฐาน 16).

Two inputs of ALU are _____ (ฐาน 16) and _____ (ฐาน 16).

จ. ใน step ที่ 4 ค่าของ register ALUOut เป็นเท่าไร ค่าที่ Write data พอร์ต ของ register file เป็นเท่าไร และ ค่า สัญญาณ RegDst กับ RegWrite เป็นเท่าไร [hint: คำถามไม่ยาก แต่ไล่ สัญญาณ ใน Figure 2 ร่วมกับ Control Summary Table.]

ALUOut = _____ (ฐาน 16). Write data port of the register file has _____(ฐาน 16).

RegDst = _____.

RegWrite = _____ (สมมติ RegWrite ถูก asserted เมื่อมีค่าเป็น '1'.)

5. Pipelining. (2 ข้อย่อย: 13 คำถาม)

5.1 Pipelining motivation: สมมติคอมพิวเตอร์ A คือ 5-cycle pipelining machine โดยใช้ clock ความถี่ 500 MHz.

ก. แต่ละ cycle จะใช้เวลาเท่าไร _____.

ข. แต่ละ instruction ที่รันด้วยคอมพิวเตอร์ A จะใช้เวลาเท่าไร _____.

ค. ถ้า pipelining สามารถทำได้อย่างสมบูรณ์แบบ (ไม่ต้องจัดการกับ hazard)

ค.1 ในช่วงเวลา 100 ns คอมพิวเตอร์ A จะทำงานได้กี่ cycles _____.

ค.2 ในช่วงเวลา 100 ns คอมพิวเตอร์ A จะสามารถทำคำสั่งเสร็จ(ตั้งแต่ cycle แรกจนจบ)ได้ทั้งหมดกี่คำสั่ง [hint: คำสั่งแรกสุดเสร็จที่ cycle ที่ 5, คำสั่งที่ 2 เสร็จที่ cycle ที่ 6, ...] _____.

ค.3 ค่า CPI ของ คอมพิวเตอร์ A เมื่อคำนวณภายในช่วงเวลา 100 ns นี้จะเป็นเท่าไร

_____.

ค.4 ในช่วงเวลา 100 ns คอมพิวเตอร์ B ก็สามารถทำคำสั่งได้เสร็จเท่ากับคอมพิวเตอร์ A โดยที่คอมพิวเตอร์ B ก็เป็น 5-cycle machine แต่ไม่ได้ทำ pipelining แล้ว คอมพิวเตอร์ B ใช้ clock ความถี่เท่าไร

_____.

5.2 Challenges in pipelining: สมมติคอมพิวเตอร์ A คือ MIPS 5-cycle pipelining machine ที่มี cycle การทำงานดังแสดงใน Figure 3

ห้า cycles คือ Instruction fetch (IF), Instruction Decode (ID), Execution using the ALU (EX), Memory access (MEM) และ Write back to the register file (WB).

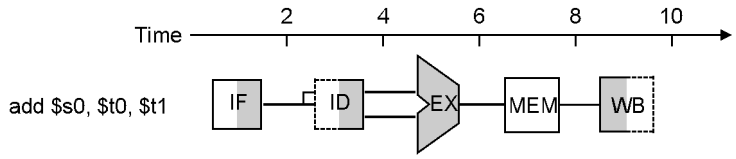

Figure 3: (Fig. 6.7 P&H2e) Graphical representation of the instruction pipeline.

เมื่อใช้ คอมพิวเตอร์ A รันโปรแกรมข้างล่างนี้ (ตัวเลขสำหรับ immediate field เขียนในฐาน 10):

```
addi $s1, $zero, 40
addi $s0, $zero, 18
add $s1, $s1, $s1
add  $s2, $s0, $s1
```

ก. ถ้าโปรแกรมนี้ทำงานถูกต้อง ค่าของ register $s2 จะเป็นเท่าไรหลังจบโปรแกรม

$s2 = _____ (ฐานสิบ).

ข. ถ้าไม่มีระบบสำหรับการจัดการกับ hazard ใดๆเลย แล้วหลังรันโปรแกรมจบ จะใช้เวลากี่ cycles ในการรันครบ 4 คำสั่งนี้ และ ค่า $s2 จะเป็นเท่าไร [hint: ดู Hint table ข้างล่าง]
หมายเหตุ สมมติว่าขั้นตอนการเขียนค่าในรีจีสเตอร์ (WB) สามารถทำเสร็จได้ตั้งแต่เริ่ม cycle ก่อนที่การคำนวณ (EX) ของคำสั่งหลังจะการเริ่มทำงาน.

$s2 = ____ (ฐานสิบ). และใช้ ___ **cycles**.

cycle	1	2	3	4	5	6	7	8	9	10
addi $s1, $zero, 40	IF	ID	EX	MEM	$s1 = 40					
addi $s0, $zero, 18		IF	ID	EX	MEM					
add $s1, $s1, $s1			IF	ID	EX	MEM				
add $s2, $s0, $s1				IF	ID	EX	MEM			

Hint table: ตารางนี้เป็นคำใบ้ ช่วยในการตอบคำถาม 5.2 ข. และ ตารางไม่ได้สมบูรณ์
ไม่ต้องสนใจตารางนี้ ถ้าตอบได้แล้วหรือถ้าไม่รู้ว่าตารางนี้เกี่ยวกับอะไร
!การตีความคำใบ้ผิด เป็นความเสี่ยงของคุณเอง!

ค. ถ้าระบบสำหรับการจัดการกับ hazard ทำโดย stall เท่านั้น แล้วหลังรันโปรแกรมจบค่า $s2 จะเป็นเท่าไร และ จะใช้เวลากี่ cycle ในการรับครบ 4 คำสั่งนี้

$s2 = ____ (ฐานสิบ). และใช้ _____ cycles.

ง. ถ้าระบบสำหรับการจัดการกับ hazard ทำโดย bypassing แล้วหลังรันโปรแกรมจบค่า $s2 จะเป็นเท่าไร

และ จะใช้เวลากี่ cycle ในการรับครบ 4 คำสั่งนี้

$s2 = ____ (ฐานสิบ). และใช้ _____ cycles.

6. Instruction Set.

แปลงโค้ดในภาษา C ข้างล่างนี้ให้เป็น MIPS assembly

```
for ( i = 0; i < h; i++ )
{
      a[i] = -a[i];
}
```

โดยสมมติว่าคอมไพเลอร์ใช้ registers $s3, $s4 กับ $s5 ในการการเก็บ base address ของอาร์เรย์ a, ตัวแปร i กับตัวแปร h ตามลำดับ [hint: เขียนคอมเมนต์ท้ายคำสั่งดังตัวอย่าง จะช่วยให้ตรวจสอบได้ง่าย และไม่ลืมคำสั่งใดๆ]

เติมโค้ดข้างล่างให้สมบูรณ์ (พื้นที่บรรทัดที่เว้นไว้ให้ควรจะพอ)

```
            add $s4, $zero, $zero     # i = 0
BEGFOR:     slt $t0, $s4, $s5         # i < h
            beq $t0, $zero, ENDFOR    # branch to ENDFOR when not (i < h)

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            _____________________________________________________

            j BEGFOR                  # branch to BEGFOR
ENDFOR:     nop                       # it's done.
```

========== สิ้นสุดข้อสอบ ==========

"*Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma — which is living with the results of other people's thinking. Don't let the noise of others' opinions drown out your own inner voice. And most important, have the courage to follow your heart and intuition. They somehow already know what you truly want to become. Everything else is secondary.*" - Steve Jobs