

ข้อสอบวิชา COS2108 โครงสร้างและสถาปัตยกรรมคอมพิวเตอร์

ภาคซ้อม 1/2565 สอนวันที่ 30 มกราคม 2567 เวลา 11:30 – 13:20 น.



คำแนะนำ

- ข้อสอบฉบับนี้เป็นข้อสอบแบบอัดแน่นจำนวน 7 ข้อ คะแนนรวม 100 คะแนน นักศึกษาจะต้องส่งคืนข้อสอบ และกระดาษคำตอบภายในเวลาที่กำหนด กรณีที่ไม่ประ拯救คำทำบนหน้ากระดาษคำตอบให้เรียบร้อยไว้ในกระดาษคำตอบนั้น แรกว่า “ไม่ประ拯救คำทำบนหน้ากระดาษคำตอบ”
- ให้นักศึกษาเขียนรหัสประจำตัวนักศึกษา ชื่อ-สกุล วันที่สอบ ห้องสอบ รวมถึงลงลายมือชื่อในช่องคำบัญญາณฯ บนกระดาษคำตอบและในการเขียนบรรยายความเขียนให้อ่านง่ายมิใชem อาจพิจารณาไม่ตรวจสอบไม่ตรวจกระดาษคำตอบ
- อนุญาตให้นำดินสอ ผู้ช่วย หรือเอกสารประกอบได้ทุกชนิด แต่ห้ามนำเครื่องเขียน หรืออุปกรณ์อิเล็กทรอนิกส์ ของทุกประเภท ให้ทำข้อสอบด้วยตนเองท่านนี้เพื่อศักดิ์ศรีความเป็นนักศึกษาของมหาวิทยาลัยรามคำแหงซึ่งจัด เป็นผู้ที่มี “ความรู้คุณธรรม”

คำถาม

1. จงเขียนวงจร Combinational ในรูปแบบ Sum-of-Products ที่ล็อกปูออย่างง่ายที่สุดแล้ว (5 คะแนน) โดยให้แสดงการใช้ Karnaugh Maps (7 คะแนน) ร่วมกับ Boolean Algebra (3 คะแนน) ในการลดรูปจาก Truth table ด้านล่าง

Truth Table

D	C	B	A	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
1	0	0	0	0
0	1	0	0	1
1	1	0	0	0
0	0	1	0	1
1	0	1	0	0
0	1	0	0	1
1	1	0	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	0
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0

ชื่อ-นามสกุล..... รหัสประจำตัวนักศึกษา.....

2. กำหนดให้เมมโมรี่เพรสเซอร์รุ่นหนึ่งมีรีจิสเตอร์ภายในจำนวน 7 ตัวคือ Program Counter (PC), Memory Address Register (MAR), Memory Buffer Register (MBR), Input / Output Address Register (I/O AR), Input / Output Buffer Register (I/O BR), Instruction Register (IR) และ Accumulator (AC) และมีรูปแบบคำสั่งขนาด 16 บิต ตามรูปที่ 2(a) ซึ่งมี Opcode ขนาด 4 บิต และสามารถถอดอ้างอิงตำแหน่งที่อยู่ได้ขนาด 12 บิต และใช้เทคนิคการอินเตอร์เฟชชนิด Memory mapped I/O โดย I/O เริ่มต้นที่ตำแหน่ง 0xE00h สำหรับการจัดเก็บข้อมูลจะอยู่ในรูปแบบจำนวนทศนิยมมาตรฐาน IEEE754 Half-precision ขนาด 16 บิตตามรูปที่ 2(b) บางส่วนของคำสั่งแสดงดังรูปที่ 2(c) ซึ่งตำแหน่งหน่วยความจำเริ่มต้นสำหรับโปรแกรมอยู่ที่ 0x100h และข้อมูลเริ่มต้นที่ 0xB00h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode	Address														

รูปที่ 2(a) Instruction format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sign	Exponent					Mantissa									

รูปที่ 2(b) Half-precision floating-point format

Assembly	Meaning	Operation	Opcode (Binary)
LDM AC, 0xXXXh	Load AC from Memory	AC <- (0xXXXh) _{mem}	0001
STM AC, 0xXXXh	Store AC to Memory	AC -> (0xXXXh) _{mem}	0010
LDIO AC, 0xXXXh	Load AC from I/O	AC <- (0xXXXh) _{I/O}	0101
STIO AC, 0xXXXh	Store AC to Memory	AC -> (0xXXXh) _{I/O}	0110
ADDF AC, 0xXXXh	Add float to AC from Memory	AC <- AC + (0xXXXh) _{mem}	1001

รูปที่ 2(c) Partial list of opcodes

a) จงแสดงวิธีการแปลงค่า 98.4 (5 คะแนน) และ -52.4 (5 คะแนน) ให้อยู่ในรูปแบบจำนวนทศนิยมมาตรฐาน IEEE754 Half-precision

b) จงเขียนโปรแกรมภาษา Assembly ตามลำดับคำสั่งต่อไปนี้

- i. นำค่าจาก I/O ตำแหน่ง 0xE01h (ซึ่งมีค่า 98.4) มาเก็บลงในรีจิสเตอร์ AC (1 คะแนน)
- ii. บอกค่าในหน่วยความจำตำแหน่ง 0xB01h (ซึ่งมีค่า -52.4) กับค่าที่อยู่ในรีจิสเตอร์ AC แล้วนำมาเก็บลงในรีจิสเตอร์ AC (1 คะแนน)
- iii. นำค่าในรีจิสเตอร์ AC เก็บลงในหน่วยความจำตำแหน่งที่ 0xB00h (1 คะแนน)
- iv. นำค่าในรีจิสเตอร์ AC เก็บลงที่ I/O ตำแหน่ง 0xE00h (1 คะแนน)

Memory	Register
0x000h	0xXXXXh
	...
0x100h	0xXXXXh
	...
0xB00h	0xXXXXh
	...
0xFFFFh	0xXXXXh
0xE00h	0xXXXXh
	...
0xFFFFh	0xXXXXh

ดัววย่างการเขียนตอบข้อ c) และ d)

- c) จงเขียนรูปเพื่อแสดงคำสั่งและข้อมูลทั้งหมดที่อยู่ในตำแหน่ง 0x000h-0xFFFFh ของ Memory และ I/O ก่อนโปรแกรมตามข้อ b) เริ่มทำงาน (ให้ใช้รูปแบบเลขฐาน 16 ทั้งหมด) (5 คะแนน)
- d) จงเขียนรูปเพื่อแสดงค่าใน Memory, I/O, Register AC, PC และ IR ที่เกิดขึ้นหลังกระบวนการ Fetch และ Execute ของโปรแกรมตามข้อ b) พร้อมเขียนคำอธิบายเหตุการณ์ที่เกิดขึ้นในแต่ละคำสั่ง (16 คะแนน)
- e) ผลลัพธ์ของโปรแกรมนี้ที่อยู่ในรีจิสเตอร์ AC มีค่าเท่าไหร่ (ให้ตอบโดยใช้รูปแบบเลขฐาน 16) (5 คะแนน)
3. จงหาจำนวน Block (2 คะแนน), ขนาดของ Tag (2 คะแนน), Set (2 คะแนน) และ Word (2 คะแนน) ของ 2-Way Set-Associative Mapping Cache Memory ขนาด 512 "บิต" ซึ่งมี line size ขนาด 8 "บิต" ของหน่วยความจำขนาด 12 bit address พร้อมทั้งวาระรูปโครงสร้างของ Memory และ Cache ดังกล่าวประกอบ (2 คะแนน)
4. จงอธิบายข้อดีและข้อจำกัดของ RAID Level 0, 1, 4 และ 5 (ควรคาดภาพประกอบ) (5 คะแนน)
5. จงอธิบายทั้งข้อดีและข้อจำกัดของการบริหารจัดการหน่วยความจำโดยใช้เทคนิค Partition (5 คะแนน) และ Demand Paging (5 คะแนน) โดยละเอียด (ควรคาดภาพประกอบ)
6. เมื่อ CPU 2 รุ่นซึ่งถูกออกแบบมาปฏิบัติภาระให้มีการอ้างอิงที่อยู่ของตัวถูกดำเนินการ (Operand) ดังตารางข้างล่างนี้ จงเขียนโปรแกรมที่สามารถคำนวนค่าคงเหลือ $Y = (J - K) / (L + M + N)$ สำหรับ CPU แต่ละรุ่น (5 คะแนน) และ อธิบายข้อดีและข้อจำกัดของการใช้การอ้างอิงที่อยู่ของตัวถูกดำเนินการทั้งสองรูปแบบ (5 คะแนน)

Three-address instructions CPU with 3 Register	
Instruction	Operation
ADD A, B, C	A <- B + C
SUB A, B, C	A <- B - C
MPY A, B, C	A <- B * C
DIV A, B, C	A <- B / C

Two-address instructions CPU with 2 Register	
Instruction	Operation
ADD A, B	A <- A + B
SUB A, B	A <- A - B
MPY A, B	A <- A * B
DIV A, B	A <- A / B
MOV A, B	A <- B

7. จงอธิบายการทำงานของ Addressing Mode ด้านล่างนี้ โดยให้ยกตัวอย่างคำสั่ง และเรียนรูปเพื่อแสดงค่าใน Register และ/หรือ Memory ที่เกี่ยวข้อง ก่อนและหลังการทำงานคำสั่งประกอบการอธิบาย
- a) Immediate Addressing (2 คะแนน)
- b) Direct Addressing (2 คะแนน)
- c) Indirect Addressing (2 คะแนน)
- d) Register Addressing (2 คะแนน)
- e) Register Indirect Addressing (2 คะแนน)

1. คณิตศาสตร์ Combinational ไม่ต่อไปนี้ Sum-of-Products ที่ต้องใช้ในการเขียนที่อยู่อาศัย (5 คะแนน) โดยมีผลลัพธ์การ
ที่ Karnaugh Maps (7 คะแนน) ตามกรอบ Boolean Algebra (3 คะแนน) ในการเขียนที่อยู่อาศัย Truth table คือ

Table 1. Karnaugh Maps (7 คะแนน) ตามกรอบ Boolean Algebra (3 คะแนน) ในการเขียนที่อยู่อาศัย Truth table คือ

Table 2. Truth Table (5 คะแนน)

D	C	B	A	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$\text{SOP} \quad A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}BCD$$

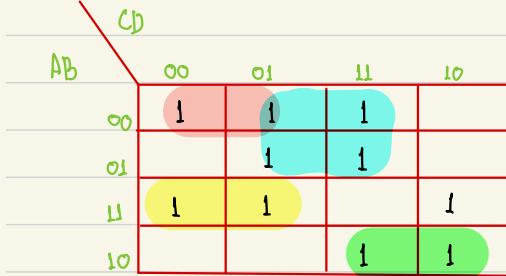


$$\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{D}(\bar{B}C + \bar{B}C) + ABC\bar{D} + \bar{A}\bar{B}C\bar{D} + BD(\bar{A}\bar{C} + A\bar{C}) + \bar{B}CD(\bar{A} + A) + \bar{A}BCD$$

$$\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{D}(BC \oplus BC) + ABC\bar{D} + \bar{A}\bar{B}C\bar{D} + BD(\bar{A}\bar{C} + A\bar{C}) + \bar{B}CD(\bar{A} + A) + \bar{A}BCD$$

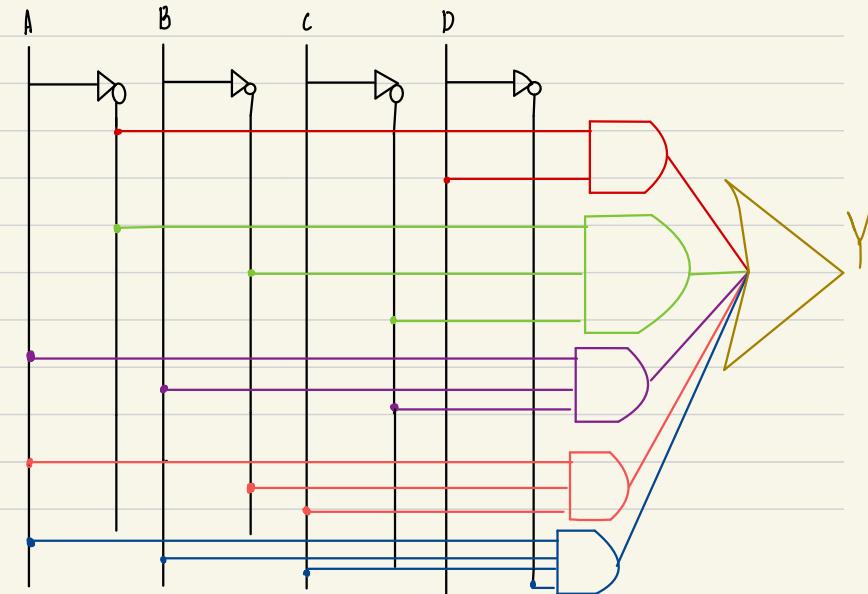
$$A\bar{D}(BC \oplus BC) + A\bar{D}(\bar{B}C + BC) + BD(\bar{A}\bar{C} + A\bar{C}) + \bar{A}D(\bar{B}C \oplus BC) + \bar{B}CD$$

5.1 k-map



$$\bar{A}\bar{D} + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC\bar{D}$$

6. วงจร



รูปที่ 2(c) Partial list of opcodes

a) จงแสดงวิธีการแปลงค่า 98.4 (5 บิตแนะนำ) และ -52.4 (5 บิตแนะนำ) ให้อยู่ในรูปแบบจำนวนทศนิยมมาตรฐาน IEEE754 Half-precision

IEEE754 Half-precision

ผลลัพธ์ 98.4 เป็น IEEE 754 Half

Sing	Ex	Man	หมายเหตุคร่าวๆ ของการเก็บทศนิยม
0	10101	1000100110	ผลลัพธ์

$$\text{Step 1 ถ้า Man of 98.4 = } 1100010.01100110 \div 1.1000100110$$

$$\begin{array}{r} 1100010.01100110 \\ \times 1.1000100110 \\ \hline 1100010.01100110 \\ 1100010.01100110 \\ \hline 0 \end{array}$$

ผลลัพธ์ที่ได้คือ 0.10101

$$\begin{array}{r} 1100010.01100110 \\ \times 1.1000100110 \\ \hline 1100010.01100110 \\ 1100010.01100110 \\ \hline 0 \end{array}$$

ผลลัพธ์ที่ได้คือ 0.10101

$$0.4 \times 2 = 0.8$$

$$\text{Exp } 98.4 = 10101$$

$$0.4 \times 2 = 1.6$$

ก็จะได้ในตารางให้มาอยู่ที่ 16

$$0.6 \times 2 = 1.2$$

$$0.10101001100110$$

$$0.2 \times 2 = 0.4$$

$$0.10101001100110$$

$$0.4 \times 2 = 0.8$$

$$0.10101001100110$$

$$0.4 \times 2 = 1.6$$

$$0.10101001100110$$

$$0.6 \times 2 = 1.2$$

$$0.10101001100110$$

$$0.2 \times 2 = 0.4$$

$$0.10101001100110$$

$$0.5 \quad 6 \quad 9 \quad 6$$

ผลลัพธ์ -54.4 เป็น IEEE 754 Half

Sing	Ex	Man
1	10100	1011001100

$$\text{ถ้า Man } -54.4 = 110110.01100110 \div 1.1011001100$$

$$\begin{array}{r} 110110.01100110 \\ \times 1.1011001100 \\ \hline 110110.01100110 \\ 110110.01100110 \\ \hline 0 \end{array}$$

$$\text{ถ้า Exp } 1.1011001100 \times 2^5 = 5 + 15 + 20 = 10100$$

$$21000$$

$$21100$$

$$2151$$

$$21200$$

$$1$$

$$11010011001100$$

$$D \quad 9 \quad C \quad C$$

รูปที่ 2(b) Half-precision floating-point format			
Assembly	Meaning	Operation	Opcode (Binary)
LDM AC, 0xXXXh	Load AC from Memory	$AC \leftarrow (0xXXXh)_{mem}$	0001 1
STM AC, 0xXXXh	Store AC to Memory	$AC \rightarrow (0xXXXh)_{mem}$	0010 2
LDIO AC, 0xXXXh	Load AC from I/O	$AC \leftarrow (0xXXXh)_{I/O}$	0101 5
STIO AC, 0xXXXh	Store AC to Memory	$AC \rightarrow (0xXXXh)_{I/O}$	0110 6
ADDF AC, 0xXXXh	Add float to AC from Memory	$AC \leftarrow AC + (0xXXXh)_{mem}$	1001 9

รูปที่ 2(c) Partial list of opcodes			
b)	จงเขียนนิพรักรรมภาษา Assembly ตามส่วนค่าส่วนต่อไปนี้		
i.	นำค่าจาก I/O ตำแหน่ง 0xE01h (ซึ่งมีค่า 98.4) มาเก็บลงในชีล์สตอเร็จ AC (1 คะแนน)	Memory	0x000h 0xFFFFh 0x100h 0xFFFFh 0xB00h 0xFFFFh 0xFFFh 0xFFFFh
ii.	นำผลค่าในหน่วยความจำตำแหน่ง 0xB01h (ซึ่งมีค่า -52.4) กับค่าที่อยู่ในชีล์สตอเร็จ AC และนำมารบกวนในชีล์สตอเร็จ AC (1 คะแนน)	I/O	0xE00h 0xFFFFh 0xFFFFh 0xFFFFh = ตัวอย่างการเขียนค่า
iii.	นำค่าในชีล์สตอเร็จ AC เก็บลงในหน่วยความจำตำแหน่งที่ 0xB00h (1 คะแนน)		
iv.	นำค่าในชีล์สตอเร็จ AC เก็บลงที่ I/O ตำแหน่ง 0xE00h (1 คะแนน)		

* คำแนะนำทางหมายเหตุที่สำคัญของภาษา Assembly

๕๖๙

i.

Ans LDIO AC, 0xE01h Load AC from I/O $AC \leftarrow (0xE01h)_{I/O}$

ii.

Ans ADDF AC, 0xB01h Add float to AC from Memory $AC \leftarrow AC + (0xB01h)_{mem}$

iii.

Ans STO AC, 0xB00h Store AC to Memory $AC \rightarrow (0xB00h)_{mem}$

iv.

Ans STIO AC, 0xE00h Store AC to I/O $AC \rightarrow (0xE00h)_{I/O}$

- c) จงเขียนรูปเพื่อแสดงคำสั่งและข้อมูลทั้งหมดที่อยู่ในคำແນ່ງ 0x000h-0xFFFFh ของ Memory และ I/O ก่อຈານ
โปรแกรมตามข้อ b) เริ่มทำงาน (ໃຫ້รูปແບບເຄົ້າສູ່ລາຍງານ 16 ທັງໝາດ) (5 ຄະແນນ)

Anb

Memory	Register
0x 000h	PC 0x xxxxh
0x 001h	AC 0x xxxxh
0x 002h	IR 0x xxxxh
0x 003h	
0x 004h	
.....

0x B00h	0x xxxxh
0x B01h	0x D11L
0x B02h	0x xxxxh

I/O

0x E00h	0x xxxxh
0x E01h	0x 5626
0x E02h	0x xxxxh
.....
0x FFFh	0x xxxxh

โปรแกรมตามข้อ b) เริ่มทำงาน (ให้ใช้รูปแบบเลขฐาน 16 กดหน้า) (5 คะแนน)

- c) จงเขียนรูปเพื่อแสดงค่าใน Memory, I/O, Register AC, PC และ IR ที่เกิดขึ้นหลังกระบวนการ Fetch และ Execute ของโปรแกรมตามข้อ b) พิริ่มเรียนคำอธิบายเหตุการณ์ที่เกิดขึ้นในแต่ละคำสั่ง (16 คะแนน)

Ans

Memory	Register
0x 000h	PC 0x xxxxh
0x 001h	AC 0x xxxxh
0x 002h	IR 0x xxxxh
0x 003h	
0x 004h	
...	...

0x B00h	0x xxxxh
0x B01h	0x D111
0x B02h	0x xxxxh

I/O

0x E00h	0x xxxxh
0x E01h	0x 5626
0x E02h	0x xxxxh
...	...

0x FFFFh

0x xxxxh

Step 1

Memory	Register
0x 000h	PC 0x 000h
0x 001h	AC 0x xxxh
0x 002h	IR 0x 5E01h
0x 003h	
0x 004h	
...	...

0x B0h	0x xxxxh
0x B01h	0x D2LL
0x B02h	0x xxxxh
I/O	
0x E00h	0x xxxxh
0x E01h	0x 5b2b
0x E02h	0x xxxxh
...	...
0x FFFh	0x xxxxh

"Step 2"

Memory	Register
0x 000h	PC 0x 001h
0x 001h	AC 0x 5626
0x 002h	IR 0x 5E01h
0x 003h	
0x 004h	
0x xxxxh	
....	

Load AC from i/o

I/O	Register
0x B00h	0x xxxxh
0x B01h	0x D22L
0x B02h	0x xxxxh
1/0	
0x E00h	0x xxxxh
0x E01h	0x 5626
0x E02h	0x xxxxh
....
0x FFFh	0x xxxxh

Step 3

Memory	Register
0x 000h	PC 0x 001h
0x 001h	AC 0x 5626
0x 002h	IR 0x 9B01h
0x 003h	
0x 004h	
...	...

0x B00h	0x xxxxh
0x B01h	0x D22L
0x B02h	0x xxxxh
I/O	
0x E00h	0x xxxxh
0x E01h	0x 5626
0x E02h	0x xxxxh
...	...
0x FFFh	0x xxxxh

" step 4 "

Memory

Ox 000h	Ox 5E01h
Ox 001h	Ox 9B01h
Ox 002h	Ox bB00h
Ox 003h	Ox 2E00h
Ox 004h	Ox xxxh
....

Register

PC	Ox 002h
AC	Ox 51C0
TR	Ox 9B01h

$$98.4 + (-59.4) = 46$$

Ox B00h	Ox xxxh
Ox B01h	Ox D92L
Ox B02h	Ox xxxh
I/O	
Ox E00h	Ox xxxh
Ox E01h	Ox 562b
Ox E02h	Ox xxxh
....
Ox FFFh	Ox xxxh

" step 5 "

Memory	Register
0x 000h	PC 0x 000h
0x 001h	AC 0x 51C0
0x 002h	JR 0x bB00h
0x 003h	
0x 004h	
...	...



0x B00h	0x xxxxh
0x B01h	0x D1C0
0x B02h	0x xxxxh
I/O	
0x E00h	0x xxxxh
0x E01h	0x 5626
0x E02h	0x xxxxh
...	...
0x FFFh	0x xxxxh

"
Step 6"

Memory	Register
0x 000h	PC 0x 003h
0x 001h	AC 0x 51C0
0x 002h	IR 0x 6B00h
0x 003h	
0x 004h	
...	
...	

0x B00h	0x 51C0
0x B01h	0x D22L
0x B02h	0x xxxxh
I/O	

0x E00h	0x xxxxh
0x E01h	0x 512L
0x E02h	0x xxxxh
...	...
0x FFFh	0x xxxxh

Step 4

Memory	Register
0x 000h	PC 0x 003h
0x 001h	AC 0x 51C0
0x 002h	IR 0x 2E00 h
0x 003h	
0x 004h	
...	



0x B00h	0x 51C0
0x B01h	0x D21L
0x B02h	0x xxxh
I/O	

0x E00h	0x xxxh
0x E01h	0x 512L
0x E02h	0x xxxh
...	
0x FFFh	0x xxxh

"Step 8"

Memory	Register
0x 000h	PC 0x 001h
0x 001h	AC 0x 51C0
0x 002h	TR 0x 2E00 h
0x 003h	
0x 004h	
0x xxxh	
...	

0x B00h	0x 51C0
0x B01h	0x D2CC
0x B02h	0x xxxh
I/O	

0x E00h	0x 51C0
0x E01h	0x 5626
0x E02h	0x xxxh
...	...
0x FFFh	0x xxxh

e) ผลลัพธ์ของโปรแกรมนี้ที่อยู่ในรีชีสเตอร์ AC มีค่าเท่ากับ (ให้ตอบโดยใช้รูปแบบเลขฐาน 16) (5 คะแนน)

An3 $46.4 + (-52.4) = 46$ * แก้แล้ว 46 ให้ยกให้ถูก IEEE 754 และ กำหนดฐาน 16 *

Sign	Ex	Mantissa
0	10100	0.111000000

๗) Man $46.0 = 101110.00000 \cdot 1.0111000000$

2 | 4 6 0

2 | 1 3 1

2 | 1 1

2 | 5 1

2 | 2 0

1

๗) Exp $1.0111000000 \times 2^5 \cdot 5+15=20$

$46 = 5120$ *

กำหนดเลข

0 10100 01 1100 0000
5 1 C 0

Demand Paging (5 คะแนน) โดยละเอียด (គរបាយការព្រមកប)

6. เมื่อ CPU 2 ຖຸ່ນຈູກອານມេសោមប្រឡង និងការចាយចັ້ງទີ່មີຢູ່ຂອງຕົວງຸດຕ່າເນີນການ (Operand) តີ່ສຳກັນຫຼັງນີ້
ຈະເຊີ່ນປັບປຸງການ (Program) ដີ່ສຳກັນການ (Data) ដີ່ມີຢູ່ $Y = (J - K) / (L + M + N)$ ສ້າງໃນ CPU ແລ້ວຮູ່ນີ້ (5 คะแนน) ແລ້ວ
ອື່ນນາ້ນຕີ່ແລ້ວຂຶ້ນຈຳດັບຂອງການໃກ້ຈົງຈາກອີງຫຼືຍູ້ຂອງຕົວງຸດຕ່າເນີນການທີ່ສອງປັບປຸງ (5 คะแนน)

Three-address instructions CPU with 3 Register		Two-address instructions CPU with 2 Register	
Instruction	Operation	Instruction	Operation
ADD A, B, C	$A \leftarrow B + C$	ADD A, B	$A \leftarrow A + B$
SUB A, B, C	$A \leftarrow B - C$	SUB A, B	$A \leftarrow A - B$
MPY A, B, C	$A \leftarrow B * C$	MPY A, B	$A \leftarrow A * B$
DIV A, B, C	$A \leftarrow B / C$	DIV A, B	$A \leftarrow A / B$
		MOV A, B	$A \leftarrow B$

An3

Two- address instruction CPU with 2 Register

Instruction	Operation
MOV Y, J	$Y \leftarrow J$
SUB Y, K	$Y \leftarrow Y - K$
ADD L, M	$L \leftarrow L + M$
ADD L, N	$L \leftarrow L + N$
DIV Y, L	$Y \leftarrow Y / L$

ចົກຕົວ

ຢັ້ງຕົກຕົວການກຳທຳການ ຂໍ້ມູນຄວາມຈຳເປົ້າໃນການເຫັນຕົວງຸດຕ່າເນີນການ ກຳທຳໄຟລ໌ແລ້ວເພີ້ນຂຶ້ນ

ຈຳທຳກົດ

ກົດ Register ກ່າວວ່ານີ້ມີຢູ່ຈຳທຳໃນການເຫັນຕົວງຸດຕ່າເນີນການ ມານາ-ຕົນໄຫວ້ເກົ່າກົດຈຳກັດການນຸ່ມກຳທຳກົ້ນ

Three- address instruction CPU

with 3 Register

Instruction	Operation
SUB Y, J, K	$Y \leftarrow J - K$
ADD L, L, M	$L \leftarrow L + M$
ADD L, L, N	$L \leftarrow L + N$
DIV Y, Y, L	$Y \leftarrow Y / L$

ຈົດຕັ້ງ

1. ຄວາມນັດຫັນໃຫຍ້ໃຫ້ໃນການ ໄກສະກິດການກຳທຳ ເພີ້ນຕົວງຸດຕ່າເນີນການ ແລ້ວຮັບຮັດ
ໄດ້ ເພີ້ນຕົວງຸດຕ່າເນີນການສະຫງົບໃນການເຫັນການກຳທຳຕົວງຸດຕ່າ ແນວດຫັນຫຼັມກຳທຳ

2. ຂໍ້ມູນຄວາມຈຳທຳກົດໃຫຍ້ໃຫ້ໃນການກຳທຳ (CPU) ໄກສະກິດການເຫັນການກຳທຳກົດ
ແລ້ວຮັບຮັດແລ້ວນັ້ນກົດຈຳກັດກົດ ເພີ້ນຕົວງຸດຕ່າ ຢັ້ງຕົວງຸດຕ່າເນີນການ

ຈຳທຳກົດ

ກົດ Register ກ່າວວ່ານີ້ມີຢູ່ຈຳທຳໃນການເຫັນຕົວງຸດຕ່າເນີນການ ຈັດກັນຫຼັມກຳທຳ

MOV A, B	A <- B
7. จงอธิบายการท่า้งานของ Addressing Mode ด้านล่างนี้ โดยให้ยกตัวอย่าง คำสั่ง และชี้ไปยังปุ่มเพื่อแสดงค่าใน Register และ/หรือ Memory ที่เกี่ยวข้องกับการท่า้งานคำสั่งประกอบการอธิบาย	

- a) Immediate Addressing (2 คะแนน)
- b) Direct Addressing (2 คะแนน)
- c) Indirect Addressing (2 คะแนน)
- d) Register Addressing (2 คะแนน)
- e) Register Indirect Addressing (2 คะแนน)

a) Immediate Addressing (2 คะแนน)

Ans ต้องการให้คำสั่งที่ ใส่ข้อมูลที่ใช้ตัวเริ่มต้นมาต่อคำสั่ง ทั้ง กำหนดให้ข้อมูลในคำสั่ง โดยไม่ต้องมีการอ่านจากหน่วยความจำ

เช่น ADD R1 นำตัวเลข 5 เที่ยวน Register

Register R1 ได้ผล

R1 เท่ากับ 5

คำสั่ง ADD R1, 5 R1 \leftarrow R1 + 5 // R1 = 5

b) Direct Addressing (2 คะแนน)

Ans รีบกัดตัวข้อมูลโดยตรงจากที่อยู่ในหน่วยความจำ โดยในคำสั่งจะนำที่อยู่ของข้อมูลที่ต้องการให้รวม

ยกตัวอย่าง Load X, 1000

หากนับตัวที่ Address 1000 ถึง Address 1000 นี้ข้อมูล

จะเป็น 1000

คำสั่ง

Op	Operand

Memory

1002
1001
1000
999

คำสั่ง Load #. 1000

Op	Operand

#.

1000

Memory

1002
1001
1000
999

จะเป็น 1000

c) Indirect Addressing (2 คะแนน)

Ans เป็นการอ่านข้อมูลโดยใช้ค่าที่อยู่ในชาร์จลูปัตต์ของให้ร้าน แต่ไม่ใช้กับค่าว่าง ในการหา ในการอ่าน ค่าที่อยู่นั้น เป็นที่อยู่ของที่อยู่ที่หนึ่งเดียวเท่านั้น สำหรับค่าที่อยู่นั้นจะถูกหักออก

ยก. คำสั่ง Load X,(1000)

คำสั่ง

Op Operand

Memory

1002
1001
1000
999

คำสั่ง Load

Op Operand

Memory

16:90 เนท: Address

1002 ← 1000 未必เข้าถึง

1001 n Address 1002

1000 nC: 1002 จัดอยู่

999 อยู่ต่อ 90 ต้องหัก n

n

เหลือ ไนเกิลชั้นดู Address 1002 16:90

d) Register Addressing (2 คะแนน)

Ans เป็นการอ่านข้อมูลโดยมาจาก Register ที่กำหนดไว้ในคำสั่ง ไม่ใช้ตัวอ่านค่าที่อยู่ในการอ่านก็จะเข้ามา

ยก. MOV // คำสั่ง

Register

A, B

MOV A,B

A ← B

Register A-B เนท: A 未必ค่าของ B ไนคำสั่ง Mov

e) Register Indirect Addressing (2 คะแนน)

Ans เป็นการอ่านข้อมูลโดยใช้ค่าที่อยู่ใน Register ที่กำหนดโดยหา คำสั่ง แต่ต้อง Register นั้นเป็นที่อยู่ของข้อมูลที่ต้องการให้ร้าน

ยก. Load X,(R1)

คำสั่ง

Op R1 ...

Memory

1002

Register

1001

n

20

1001

40

1000

1002

999