มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค ภาคการศึกษาที่ ๒ ปีการศึกษา ๒๕๕๒

วันที่ ๑๗ กุมภาพันธ์ ๒๕๕๓ เวลา ๑๓๓๐-๑๖๓๐

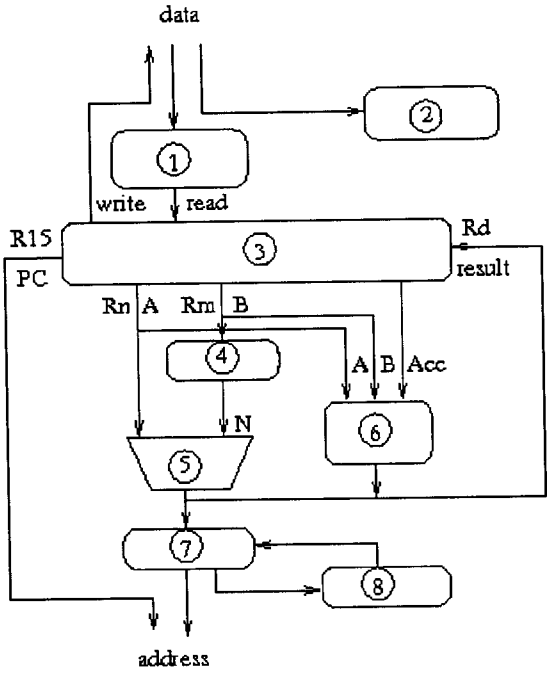
วิชา ๒๔๑-๒๑๐ Microprocessor Architecture and the Assembly Language ห้อง Robot, R300

๒๔๐-๓๐๔ Microprocessor Architecture and the Assembly Language ห้อง S201

- ข้อสอบมีทั้งหมด 7 แผ่นรวมปกและกระดาษคำตอบ, กระดาษคำตอบอยู่หน้าสุดท้าย
- ข้อสอบมี 90 ข้อ มีทั้งแบบตัวเลือกและเติมคำ ให้ทำทุกข้อ
- คำตอบแบบเติมคำข้อใด เขียนไม่ชัด อ่านไม่ออก จะไม่ให้คะแนน
- อนุญาตให้ดึงกระดาษคำตอบออกจากตัวข้อสอบได้
- ไม่อนุญาตให้นำหนังสือ, เอกสาร, หรือเครื่องคำนวณ เข้าห้องสอบ

ทุจริตในการสอบ โทษขั้นต่ำ ปรับตกในรายวิชาที่ทุจริตและพักการเรียน ๑ ภาคการศึกษา

ARM Fundamentals



รูปที่ 1 ARM Core Dataflow Model

จากรูปที่ 1 ซึ่งแสดง Dataflow Model ของ ARM Core CPU จงใช้รายชื่อของ Component ต่อไปนี้ สำหรับระบุว่าโมดูลใด (ตัวเลขในวงกลม (1)-(8)) ในรูปที่ 1 ตรงกับ Component ใด

(A) Address Register
(B) Data Register
(C) Register Files
(D) Incrementer
(E) Decrementer
(F) Instruction Decoder
(G) Data Decoder
(H) Signed Extend
(I) Unsigned Extend
(J) Barrel Shifter
(K) Multiply-Accumulate Unit
(L) Arithmetic-Logic Unit

ตารางที่ 1 รายชื่อ Component สำหรับ ARM CPU Core Dataflow Model

ระบุตัวเลือก (A – L) แทนชื่อ Component ลงในกระดาษคำตอบ **ข้อ 1-8**

ในข้อ 9-16 จะระบุฟังก์ชันการทำงานของ Component ซึ่งเป็นองค์ประกอบของ ARM Dataflow Model ให้ใช้รายชื่อของ Component ในตารางที่ 1 สำหรับระบุว่า ฟังก์ชันการทำงานในข้อ 9-16 ตรงหรือใกล้เคียงกับ ฟังก์ชันของ Component ใดมากที่สุด ให้ระบุตัวเลือก (A – L) แทนชื่อ Component ลงในกระดาษคำตอบ **ข้อ 9-16**

9. เก็บข้อมูลของ registers R0-R15
10. คำนวณทางด้านตรรกศาสตร์และคณิตศาสตร์พื้นฐาน
11. คำนวณทางด้านคณิตศาสตร์สำหรับคำสั่งคูณหารโดยเฉพาะ
12. กำหนด address ที่จะใช้ระบุตำแหน่งของหน่วยความจำ
13. ใช้สำหรับกระบวนการเลื่อน (Shift) หรือ หมุน (Rotate) บิทของข้อมูล
14. ใช้ในการคำนวณตำแหน่งของหน่วยความจำโดยเฉพาะสำหรับคำสั่ง load/store multiple
15. decode คำสั่งเพื่อสร้างสัญญาณในการควบคุมการทำงานของ module ต่างๆ
16. แปลงข้อมูลจากแบบ signed ขนาด byte หรือ half-word ให้เป็นข้อมูลแบบ signed ขนาด 1 word

ARM Mode

Mode การทำงานของ ARM Processor มี 7 Mode คือ

1. User 2. System 3. ___(17)____ 4. ___(18)___ 5. ____(19)____ 6. ____(20)____ 7. ____(21)____

ให้เขียนชื่อ mode ที่เหลือลงในกระดาษคำตอบข้อ 17-21

mode ___(22)____ ถือเป็น nonprivileged mode ส่วนที่เหลือถือเป็น privileged mode

โดยที่ใน User mode และ ___(23)___ mode จะใช้ Register ชุดเดียวกัน ต่างกันเพียงใน User mode ไม่สามารถแก้ไข (write) ค่าใน register ____(24)____ ได้

ARM Registers

ใน Register R0-R15 จะมี Register จำนวน 3 ตัวซึ่งใช้งานเป็น Register พิเศษคือ SP, LR, และ PC ซึ่ง

Register SP ตรงกับ Register ___(25)___

Register LR ตรงกับ Register ___(26)___

Register PC ตรงกับ Register ___(27)___

ใน CPSR Register สามารถแบ่งเป็น field หลักๆได้ 4 field คือ

1. ___(28)____ 2. ____(29)____ 3. Extension 4. ____(30)____

ส่วนของการควบคุมให้ ARM CPU อยู่ใน ARM State หรือ Thumb State จะควบคุมโดย bit ซึ่งอยู่ใน field __(31)___ โดยเมื่อ CPU อยู่ใน ARM State จะ execute คำสั่งที่มีขนาด ___(32)___ บิท เมื่ออยู่ใน Thumb State จะ execute คำสั่งที่มีขนาด ___(33)___ บิท

ส่วนของการควบคุมการเกิด normal interrupt request (IRQ) และ fast interrupt request (FIQ) สามารถควบคุมได้โดยการ set หรือ clear บิท ___(34)____ และบิท ____(35)____ ใน Control field ของ CPSR register

Condition Flags

จาก Condition Flag N, Z, V, C ให้ระบุชื่อเต็มของ Flag เหล่านี้

N ___(36)___

Z ___(37)___

V ___(38)___

C ___(39)___

Condition Execution

ในการ execute คำสั่งของ ARM โปรแกรมเมอร์สามารถควบคุมการ execute โดยใช้ Condition Execution ได้ซึ่งจะระบุเป็น Mnemonic 2 ตัวอักษรต่อท้ายในแต่ละคำสั่งของ ARM จากตารางที่สอง ซึ่งแสดง Condition Mnemonic บางส่วน ให้ตอบคำถามข้อ 36-41

Mnemomic	Flags: N Z C V
EQ	-,1,-,-
NE	-,0,-,-
CS	__(40)__
CC	__(41)__
MI	__(42)__
PL	__(43)__
VS	__(44)__
VC	__(45)__

ตารางที่ 2 Condition Mnemonic

Hint: Condition ที่ list มาทั้งหมดจะตรวจสอบจาก flag เพียงบิทเดียว

หมายเหตุ: flag บิทใดที่ไม่ส่งผลให้ใช้เครื่องหมายลบ (-), ถ้าค่า flag ของบิทนั้นจะต้อง set จึงจะตรงกับ condition นั้นให้ใช้เลข 1 และถ้า flag ของบิทนั้น clear จะต้องตรงกับ condition นั้น ให้ใช้เลข 0 และคั่นแต่ละบิทด้วยจุลภาค (,) ตามตัวอย่างของ Condition EQ และ NE

ARM Instructions Set

ชุดคำสั่งภาษา assembly ของ ARM สามารถแบ่งออกเป็นหมวดหมู่คร่าวๆได้เป็น

a) Data Processing Instructions
b) Branch Instructions
c) Load/Store Instructions
d) Software Interrupt Instructions
e) Program Status Register Instructions

จากคำสั่งต่อไปนี้ จงแยกหมวดหมู่ว่าคำสั่งนั้นอยู่ในหมวดหมู่ใด ตอบในกระดาษคำตอบ **ข้อ 46-50**

46. BX 47. LDR 48. MVN 49. MSR 50. SWI

กระบวนการเลื่อนบิท (shift) หรือ หมุนบิท (rotate) ข้อมูลแบบใด **ที่ไม่มี**กระบวนการ (Shift Operation) นี้บน Barrel Shifter ของ ARM จากรายการของ Shift Operation ต่อไปนี้ ใช้ตัวอักษร (a) – (h) สำหรับตอบลงใน กระดาษคำตอบ **ข้อ 51-54**

a) LSL b) LSR c) ASL d) ASR
e) ROL f) ROR g) RRX h) RLX

ให้ใช้รายการจาก shift operation ในรายการข้างต้น (a) – (h) สำหรับตอบคำถาม **ข้อ 55-60**

55. Operation ใดใช้สำหรับการเลื่อนบิทข้อมูลแบบ signed ไปทางด้านซ้าย
56. Operation ใดใช้สำหรับการเลื่อนบิทข้อมูลแบบ signed ไปทางด้านขวา
57. Operation ใดใช้สำหรับการเลื่อนบิทข้อมูลแบบ unsigned ไปทางด้านซ้าย
58. Operation ใดใช้สำหรับการเลื่อนบิทข้อมูลแบบ unsigned ไปทางด้านขวา
59. Operation ใดที่ไม่สามารถ เลื่อน/หมุน บิทข้อมูลได้มากกว่า 1 บิท
60. Operation ใดที่เมื่อเกิดขึ้นแล้วส่งผลต่อ flag field ใน CPSR

จงระบุค่าของ register R0 เมื่อ CPU ARM ทำตามคำสั่งเหล่านี้ เขียนคำตอบเป็น **เลขฐาน 16** ลงในกระดาษคำตอบ **ข้อ 61-64**

61. MOV R0, #1
62. MOV R0, #-1
63. MVN R0, #0
64. MVN R0, #-1

กำหนดค่าเริ่มต้นของของ Register R0, R1, R2, R3 ให้มีค่าดังต่อไปนี้

R0 = 0x00000000 R1 = 0x00000001 R2 = 0x00000020 R3 = 0x00000040

และค่าของ Carry Flag = 1 จงหาค่าของ Register R0 หลังจาก CPU ARM Execute คำสั่งดังต่อไปนี้ เขียนคำตอบเป็น **เลขฐาน 16** ลงในกระดาษคำตอบ **ข้อ 65-70**

65. ADD R0, R1, R2
66. ADD R0, R1, R2, LSL #4
67. ADC R0, R1, R2, LSR #4
68. SBC R0, R2, R1
69. RSB R0, R1, R2, LSR #4
70. RSC R0, R1, R2, LSR #4

สำหรับ Logical Operation ดังต่อไปนี้ให้เขียน ARM Instruction Mnemonic ของคำสั่งในชุดของ Logic Instruction ในกระดาษคำตอบ **ข้อ 71-73**

Operation	Mnemonic
Rd = Rn & N	AND
Rd = Rn | N	__(71)__
Rd = Rn ^ N	__(72)__
Rd = Rn & ~N	__(73)__

สำหรับ Compare Operation ต่อไปนี้ให้เขียน ARM Instruction Mnemonic ของคำสั่งในชุดของ Compare Instruction ในกระดาษคำตอบ **ข้อ 74-76**

Operation	Mnemonic
flag set as a result of Rn + N	___(74)___
flag set as a result of Rn – N	CMP
flag set as a result of Rn ^ N	___(75)___
flag set as a result of Rn & N	___(76)___

ในชุดคำสั่งของ ARM ไม่มีคำสั่ง CALL และ RET เมื่อเทียบกับชุดคำสั่งของ MCS51 แต่เราสามารถใช้คำสั่งในชุดของ Branch Instruction และ คำสั่งอื่นๆแทนการ return กลับจาก subroutine ได้ ให้เติมคำสั่ง และ operand ที่เหมาะสม เพื่อให้ได้โปรแกรมภาษา assembly ของ ARM ที่เทียบเคียงกันได้กับโปรแกรมภาษา assembly ของ MCS51 ในส่วนของโปรแกรมด้านล่าง ตอบลงในกระดาษคำตอบ**ข้อ 77-80**

MCS51	ARM	
main: lcall delay	main: __(77)__	delay
ljmp main	b	main
delay:	delay:	
nop	nop	
ret	__(78)__	__(79)__, __(80)__

กำหนดให้ค่าเริ่มต้นของ registers และค่าเริ่มในหน่วยความจำ มีค่าดังต่อไปนี้

R0 = 0x00000000 R1 = 0x40000000 R2 = 0x40000004 R3 = 0x40000008
R4 = 0x4000000C R5 = 0x00000001 R6 = 0x00000002 R7 = 0x00000003

ตำแหน่ง	ข้อมูล (word)			
0x40000000	0x10000001	0x10000002	0x10000003	0x10000004
0x40000010	0x10000005	0x10000006	0x10000007	0x10000008

ใช้ค่าของ register และ memory ข้างต้น **เป็นค่าเริ่มต้นก่อนที่จะ execute ทุกคำสั่ง** ในคำถามต่อไปนี้

LDR R0, [R1]
หลังจาก execute คำสั่งนี้แล้ว R0 = __(81)__
LDRB R0, [R2]
หลังจาก execute คำสั่งแล้ว R0 = __(82)__
STR R0, [R3, #4]!
หลังจาก execute คำสั่งแล้ว R0 = __(83)__ R3 = __(84)__ Mem32[0x4000000C] = __(85)__
STRB R0, [R4], #4
หลังจาก execute คำสั่งแล้ว R0 = __(86)__ R3 = __(87)__ Mem32[0x4000000C] = __(88)__
LDMIA R1!, {R2-R4}
หลังจาก execute คำสั่งแล้ว R1 = __(89)__ R3 = __(90)__

ฉัตรชัย จันทร์พริ้ม
2010-02-15

กระดาษคำตอบ

วิชา 241-210/240-305 Microprocessor Architecture and The Assembly Language

รหัสนศ. _____ **ชื่อ-สกุล** _____

1.	26.	51.	76.
2.	27.	52.	77.
3.	28.	53.	78.
4.	29.	54.	79.
5.	30.	55.	80.
6.	31.	56.	81.
7.	32.	57.	82.
8.	33.	58.	83.
9.	34.	59.	84.
10.	35.	60.	85.
11.	36.	61.	86.
12.	37.	62.	87.
13.	38.	63.	88.
14.	39.	64.	89.
15.	40.	65.	90.
16.	41.	66.	
17.	42.	67.	
18.	43.	68.	
19.	44.	69.	
20.	45.	70.	
21.	46.	71.	
22.	47.	72.	
23.	48.	73.	
24.	49.	74.	
25.	50.	75.	