รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

การสอบกลางภาค ภาคการศึกษาที่ ๒ ประจำปีการศึกษา ๒๕๕๐
วันที่ ๓๐ ธันวาคม พ.ศ. ๒๕๕๐ เวลา ๑๓๓๐-๑๖๓๐
วิชา 241-210 Microprocessor Architecture and The Assembly Language ห้อง หัวหุ่น

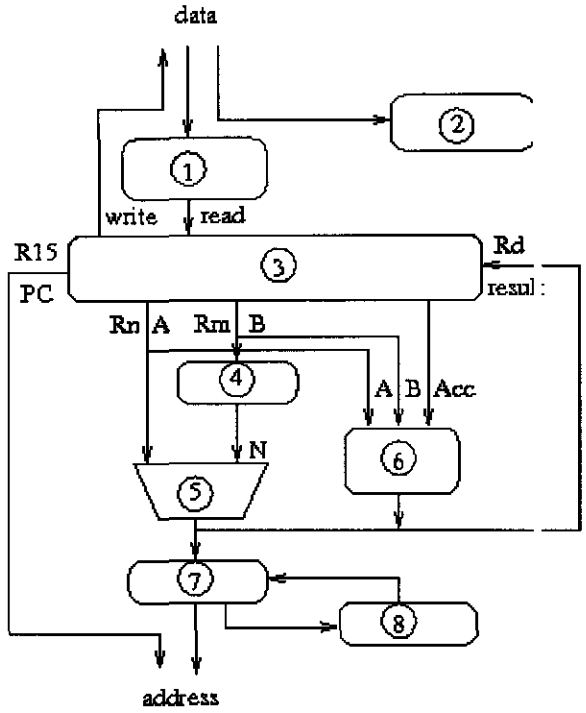
- ไม่อนุญาตให้นำเอกสารหรือเครื่องคำนวณเข้าห้องสอบ
- เขียนคำตอบให้อ่านง่าย ชัดเจน คำตอบข้อใดผู้ตรวจอ่านไม่ออก จะไม่ให้คะแนน ถ้าเนื้อที่สำหรับคำตอบไม่พอ อนุญาตให้เขียนต่อหน้าหลังของกระดาษคำตอบนั้น
- ข้อสอบมีทั้งหมด 9 หน้ารวมปก
- ข้อสอบมี 10 ข้อ คะแนนรวม 100 คะแนน ให้ทำทุกข้อ
- ให้เขียนชื่อ-สกุล รหัสนศ. **ทุกหน้าที่เป็นกระดาษคำตอบ**

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชานั้น และพักการเรียน ๑ ภาคการศึกษา

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

2. จาก dataflow model ของ ARM ในรูปต่อไปนี้ ให้จับคู่ระหว่าง ชื่อ component ต่อไปนี้ กับส่วนของ block (1) - (8) ที่แสดงในรูปที่ 1 (16 คะแนน)

a) Address Registers
b) Data Registers
c) Register Files
c) Incrementer
d) Decrementer
e) Instruction Decoder
f) Data Decoder
g) Sign Extend
h) Unsigned Extend
i) Barrel Shifter
j) Multiply Accumulator
k) Arithmetic Logic Unit

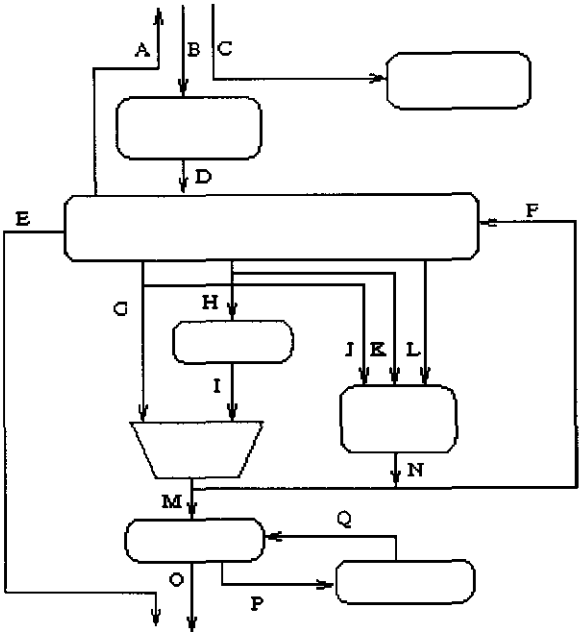


รูปที่ 1. dataflow diagram of ARM core

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

3. จากรูป dataflow diagram รูปที่ 2 ให้ใช้หมายเลข bus (A) - (P) สำหรับอธิบาย flow ของ data เมื่อตัว ARM processor execute คำสั่งต่อไปนี้ (16 คะแนน)



รูปที่ 2. dataflow diagram of ARM core

ตย. คำสั่ง xxx xx, xx, xx ... ข้อมูลจาก memory ส่งผ่าน bus C เข้าสู่ (2) จากนั้นข้อมูลจาก (3) จะส่งผ่าน bus G เข้าสู่ (5) และ ...

a) ADD R0, R1, R2

b) LDR R0, [R1, #4]

c) MOV R0, R0, SSL #4

d) MUL R0, R1, R2

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

4. จงเขียนตัวอักษรย่อที่ใช้ระบุ condition flag และ control flag และความหมายของ flag เหล่านั้นใน PSR register ให้ถูกต้อง (8 คะแนน)

บิทที่	Flags	ความหมายของ flag
31	_	_____
30	_	_____
29	_	_____
28	_	_____
...	...	...
7	_	_____
6	_	_____
5	_	_____
0-4	____	_____

5. จงอธิบายข้อแตกต่างระหว่าง ARM Mode กับ Thumb Mode (4 คะแนน)

_____
_____
_____
_____
_____

6. จงอธิบายความหมายของ Operation ของ Barrel Shifter ต่อไปนี้ (5 คะแนน)

LSL _____
LSR _____
ASR _____
ROR _____
RRX _____

7. จงอธิบายว่าทำไม**ไม่มี** Operation ASL ใน Barrel Shifter (5 คะแนน)

_____
_____
_____
_____
_____
_____

8. จงอธิบายข้อแตกต่างระหว่าง cache และ tightly couple memory (TCM) สำหรับ ARM processor (5 คะแนน)

_____
_____
_____
_____
_____
_____

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

9. จากตาราง interrupt vector table ต่อไปนี้ จงเติมข้อความในช่องว่าง ให้เหมาะสม (6 คะแนน)

Exception/Interrupt	Shorthand	Address
Reset	RESET	0x00000000
_____	UNDEF	_____
Software Interrupt	_____	_____
Prefetch abort	_____	_____
_____	DABT	_____
Reserved	--	0x00000014
_____	IRQ	0x00000018
_____	_____	_____

10. จงอธิบายข้อแตกต่างระหว่างคำสั่ง (5 คะแนน)

```
ldr   r0, [#constant]
```

กับ

```
ldr   r0, =constant
```

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

ARM Instruction Set

```
Syntax: <instruction>{<cond>}{S} Rd, N
MOV             Rd = N
MVN             Rd = ~N

Syntax: <instruction>{<cond>}{S} Rn, N
CMN             flag set : Rn + N
CMP             flag set : Rn - N
TEQ             flag set : Rn ^ N
TST             flag set : Rn & N

Syntax: <instruction>{<cond>}{S} Rd, Rn, N
ADC             Rd = Rn + N + carry
ADD             Rd = Rn + N
RSB             Rd = N - Rn
RSC             Rd = N - Rn - !{carry}
SBC             Rd = Rn - N - !{carry}
SUB             Rd = Rn - N
AND             Rd = Rn & N
ORR             Rd = Rn | N
EOR             Rd = Rn ^ N
BIC             Rd = Rn & ~N

Syntax: MLA{<cond>}{S} Rd, Rm, Rs, Rn
        MUL{<cond>}{S} Rd, Rm, Rs
MLA             Rd = (Rm * Rs) + Rn
MUL             Rd = Rm * Rs

Syntax: <instruction>{<cond>}{S} RdLo, RdHi, Rm, Rs
SMLAL           [RdHi,RdLo] = [RdHi,RdLo] + (Rm*Rs)
SMULL           [RdHi,RdLo] = Rm*Rs
UMLAL           [RdHi,RdLo] = [RdHi,RdLo] + (Rm*Rs)
UMULL           [RdHi,RdLo] = Rm*Rs

Syntax: B{<cond>} label
        BL{<cond>} label
        BX{<cond>} Rm
        BLX{<cond>} label | Rm
B               pc = label
BL              pc = label, lr = next instruction
BX              pc = Rm & 0xFFFFFFFE, T = Rm & 1
BLX             pc = label, T = 1, lr = next instruction
                pc = Rm & 0xFFFFFFFE, T = Rm & 1, lr = next instruction

Syntax: <LDR|STR>{<cond>}{B}   Rd, addressing1
        LDR{<cond>}SB|H|SH     Rd, addressing2
        STR{<cond>}H           Rd, addressing2
LDR             Rd <- mem32[address]
STR             Rd -> mem32[address]
LDRB            Rd <- mem8[address]
STRB            Rd -> mem8[address]
LDRH            Rd <- mem16[address]
STRH            Rd <- mem16[address]
LDRSB           Rd <- SignExtend(mem8[address])
LDRSH           Rd <- SignExtend(mem16[address])
```

index method	Data	Base address	Example
Preindex+writeback	mem[base+offset]	base+offset	LDR r0,[r1,#4]!
Preindex	mem[base+offset]	not update	LDR r0,[r1,#4]
Postindex	mem[base]	base+offset	LDR r0,[r1],#4

```
Syntax: <LDM|STM>{<cond>}<addressing mode> Rn{!}, <register>{^}
LDM             {Rd}*N <- mem32[start adress + 4*N] optional Rn updated
STM             {Rd}*N -> mem32[start adress + 4*N] optional Rn updated
```

Addressing Mode	Description	Start Address	End Address	Rn!
IA	increment after	Rn	Rn+4*N-4	Rn+4*N
IB	increment before	Rn+4	Rn+4*N	Rn+4*N
DA	decrement after	Rn-4*N+4	Rn	Rn-4*N
DB	decrement before	Rn-4*N	Rn-4	Rn-4*N

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

Addressing Mode	Description	Pop	=LDM	Push	=STM
FA	full ascending	LDMFA	LDMDA	STMFA	STMIB
FD	full decending	LDMFD	LDMIA	STMFD	STMDB
EA	empty ascending	LDMEA	LDMDB	STMEA	STMIA
ED	empty descending	LDMED	LDMIB	STMED	STMDA

```
Syntax: SWP{B}{<cond>} Rd, Rm, [Rn]
SWP             tmp = mem32[Rn], mem32[Rn] = Rm, Rd = tmp
SWPB            tmp = mem8[Rn], mem8[Rn] = Rm, Rd = tmp

Syntax: SWI{<cond>} SWI_number
SWI             lr_svc = next instruction
                spsr_svc = cpsr
                pc = vectors + 8
                cpsr mode = svc
                cpsr I = 1

Syntax: MRS{<cond>} Rd, <cpsr|spsr>
        MSR{<cond>} <cpsr|spsr>_<fields>, Rm
        MSR{<cond>} <cpsr|spsr>_<fields>, #immediate
MRS             Rd = psr
MSR             psr{field} = Rm
MSR             psr{field} = immediate

Syntax: LDR Rd, =constant
        ADR Rd, label
LDR             Rd = 32-bit constant
ADR             Rd = 32-bit relative address

Condition Mnemonic
```

Mnemonic	Name
EQ	equal
NE	not equal
CS/HS	carry set/unsigned higher or same
CC/LO	carry clear/unsigned lower
MI	minus/negative
PL	plus/positive or zero
VS	overflow
VC	no overflow
HI	unsigned higher
LS	unsigned lower or same
GE	signed greater than or equal
LT	signed less than
GT	signed greater than
LE	signed less than or equal
AL	always(unconditional)

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

1. จงเขียน subroutine โปรแกรมภาษา assembly ของ ARM สำหรับหาค่ามากที่สุดและค่าน้อยที่สุดของข้อมูลชุดหนึ่ง โดยกำหนดตำแหน่งเริ่มต้นของข้อมูล ให้เก็บอยู่ใน register R0, จำนวนของข้อมูลเก็บอยู่ใน register R1 ใส่ค่า max กลับมาใน register R2 และค่า min กลับมาใน register R3 หลังจากเรียกใช้ subroutine แล้วค่าของ R0, R1 จะต้องคงเดิม ข้อมูลที่ใช้มีขนาดเป็น byte
(30 คะแนน)

```
.text
.global _startup
_startup:
        ldr     sp, =stack_start
        ldr     r0, =data_start
        ldr     r1, =data_end
        sub     r1, r1, r0

@ call subroutine
        bl      findmaxmin
@ using data below, should return R2 = 10, R3 = 1
@ terminate
       swi         2

@ ------------------------------------------------------------------
@ findmaxmin   - find maximum, minimum value in the data set
@                return value in R2, R3
@ input        R0 - start of data
@              R1 - number of byte
@ output       R2 - max value
@              R3 - min value
@ ------------------------------------------------------------------
findmaxmin:

@ ... put your code here ...

.bss
.align    4

stack_start:    .word     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                .word     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

.data
.align    4

data_start:     .byte     6,2,7,10,1,4,3,5,8,9
data_end:

.end
```