รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

การสอบกลางภาค ภาคการศึกษาที่ ๑ ประจำปีการศึกษา ๒๕๕๑
วันที่ ๒ สิงหาคม พ.ศ. ๒๕๕๑ เวลา ๐๙๐๐-๑๒๐๐
วิชา 240-305 Microprocessor Architecture and The Assembly Language ห้อง R300

- o **ไม่อนุญาต** ให้นำเอกสารหรือเครื่องคำนวณเข้าห้องสอบ
- o เขียนคำตอบให้อ่านง่าย ชัดเจน คำตอบข้อใดผู้ตรวจอ่านไม่ออก จะไม่ให้คะแนน ถ้าเนื้อที่สำหรับคำตอบไม่พอให้เขียนต่อหน้าหลังของกระดาษคำตอบนั้น
- o ข้อสอบมีทั้งหมด 8 หน้ารวมปก
- o ข้อสอบมี 6 ข้อ คะแนนรวม 50 คะแนน ให้ทำทุกข้อ
- o ให้เขียนชื่อ-สกุล รหัสนศ. **ทุกหน้าที่เป็นกระดาษคำตอบ**

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชานั้น
และพักการเรียน ๑ ภาคการศึกษา

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

1. คุณลักษณะของไมโครโปรเซสเซอร์ต่อไปนี้ ให้ระบุว่าข้อใดเป็นลักษณะ ARM Microprocessor ข้อใดที่ไม่สามารถระบุได้ชัดเจนว่าใช่ หรือ ไม่ใช่ ให้อธิบายประกอบ (10 คะแนน)
(หมายเหตุ: คำตอบ ใช่/ไม่ใช่ เพียงแต่อย่างเดียวถึงแม้จะ **ถูก** แต่ **อาจจะ** ไม่ได้คะแนน)

1) เป็นโปรเซสเซอร์ขนาด 32 bit

2) มีชุดคำสั่งขนาด 16 bit

3) สถาปัตยกรรมของการออกแบบเป็นแบบ RISC

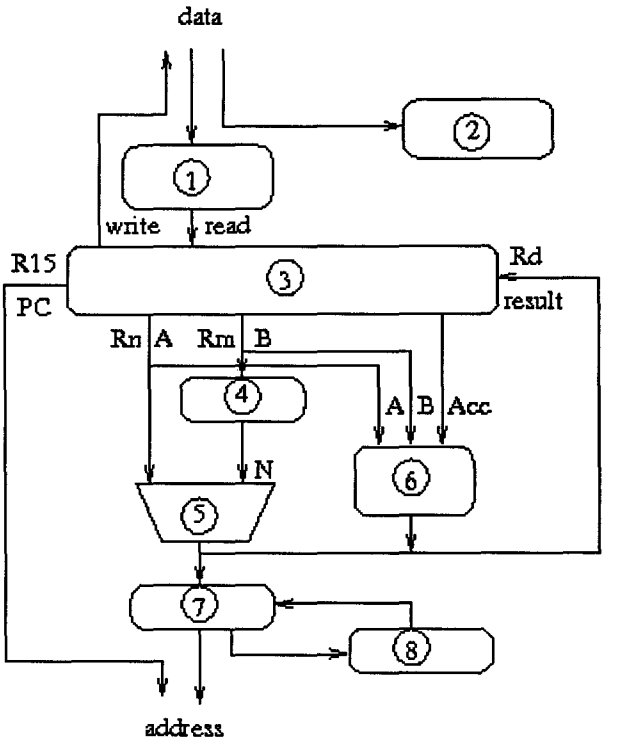
4) ใช้สถาปัตยกรรมในการเข้าถึงหน่วยความจำแบบ Load/Store

5) ใช้สถาปัตยกรรมหน่วยความจำแบบ von Nuemann

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

2. จาก dataflow model ของ ARM ในรูป ให้จับคู่ระหว่างชื่อ component (a) - (k) กับส่วนของ block (1) - (8) ใน dataflow ที่แสดงในรูปที่ 1 (10 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. _____

จาก รายการของ component (a) – (k) จงบอกชื่อ component ที่ไม่มีอยู่ใน dataflow model ของ ARM มา 2 ข้อ

9. _____
10. _____

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

3. จาก dataflow model ของ ARM ในรูปที่ 1 จงอธิบายการ flow ของข้อมูลเมื่อ ARM ทำตามคำสั่งต่อไปนี้ (เฉพาะขั้นตอนของการ execute คำสั่ง ไม่ต้องรวม fetch/decode) (10 คะแนน)

ต.ย. MOV R0, R1

ข้อมูลของ R1 ถูกส่งผ่านจาก (3) ผ่านทาง bus A สู่ (5)
หลังจากนั้น (5) ส่งข้อมูลกลับไปยัง (3) ผ่านทาง bus result ไปเก็บในตำแหน่ง R0

MVN R0, R1, LSL #1

ADD R0, R1, R2

TST R1, R2

MUL R1, R2, R3

B LABEL

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

4. จงอธิบายว่าทำไม**ไม่มี** Operation ASL ใน Barrel Shifter (2 คะแนน)

_____

_____

_____

_____

5. กำหนดให้ค่าตั้งต้นของ Register ของ ARM เป็นดังนี้ (8 คะแนน)

```
R0  = 0x00000000  R1  = 0x00000001  R2  = 0x00000002 R3  = 0x00000003
R4  = 0x00000004  R5  = 0x00000005  R6  = 0x00000006 R7  = 0x00000007
R8  = 0x80000001  R9  = 0x90000001  R10 = 0x10000000 R11 = 0x11000000
R12 = 0x12000000  R13 = 0x13000001  R14 = 0x00000000 R15 = 0x40000000
```

จงแสดงว่าค่า Register ใดเปลี่ยนแปลงไปและเปลี่ยนเป็นค่าใด เมื่อ ARM execute คำสั่งต่อไปนี้ กำหนดให้คำสั่งในแต่ละข้อไม่เกี่ยวข้องกัน และทุกคำสั่งเริ่มต้นที่ตำแหน่ง 0x40000000

1) MVN R0, #1

_____

_____

2) MOV R0, R1, LSL #1

_____

_____

3) MOV R0, R1, LSL R2

_____

_____

4) MOV R0, R8, ASR R3

_____

_____

5) ADD R0, R1, R2 LSL R3

_____

_____

6) BIC R0, R9, R3 ROR R1

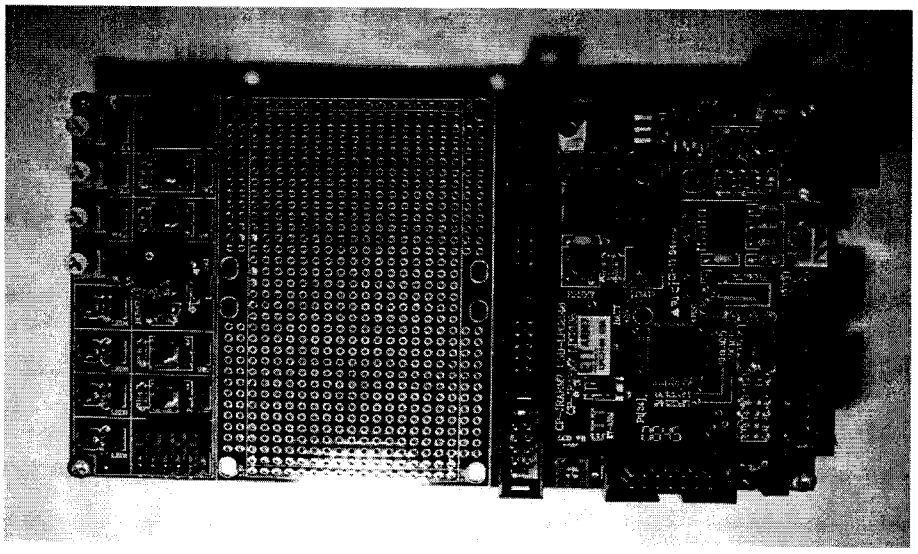
_____

_____

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

6. จากการทดลองใน Lab จงเติมคำในช่องว่าง และ ระบุตำแหน่งของอุปกรณ์ (10 คะแนน)
   1. บอร์ดทดลองใช้ CPU ARM รุ่น LPC_____ ของ บริษัท _____
   2. Crystal ซึ่งอยู่บนบอร์ดทดลองให้ความถี่ _____ MHz ตัว CPU สามารถทำงานที่ความถี่สูงสุด _____ MHz
   3. ส่วนของวงจรซึ่งใช้ในการกำหนดความถี่ให้กับ CPU คือวงจร _____
   4. Baudrate ซึ่งใช้ในการโปรแกรม Flash บนตัว CPU คือ _____
   5. โปรแกรมที่ใช้ในการพัฒฯาซอฟแวร์ชื่อว่า _____
      โปรแกรมที่ใช้ในการโปรแกรม hexfile ลง flash ชื่อว่า _____
   6. จากรูปจงระบุตำแหน่งของ อุปกรณ์ต่อไปนี้บนบอร์ดทดลอง (เขียนหมายเลข ลากเส้นไปชี้ที่อุปกรณ์บนรูป)

      1. CPU
      2. ปุ่ม RESET
      3. ปุ่ม LOAD
      4. Port 1 Pin 24 (P1[24])
      5. USB connector
      6. Serial port connector
      7. Power supply connector
      8. Push button switch สำหรับ input
      9. LED สำหรับการแสดงผล output
      10. Buzzer/Speaker



ฉัตรชัย จันทร์พริ้ม
2008-07-31

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

ARM Instruction Set

Barrel shifter operations

Mnemonic	Description	Shift	Result
LSL	Logical shift left	x LSL y	x << y
LSR	Logical shift right	x LSR y	(unsigned)x >> y
ASR	Arithmetic shift right	x ASR y	(signed)x >> y
ROR	Rotate right	x ROR y	((unsigned)x >> y)|(x << (32-y))
RRX	Rotate right extended	x RRX	(c flag << 31)|((unsigned)x>>1)

Barrel Shifter Operations

N Shift operations	Syntax
Immediate	#immediate
Register	Rm
Logical shift left by immediate	Rm, LSL #shift_imm
Logical shift left by register	Rm, LSL Rs
Logical shift right by immediate	Rm, LSR #shift_imm
Logical shift right by register	Rm, LSR Rs
Arithmetic shift right by immediate	Rm, ASR #shift_imm
Arithmetic shift right by register	Rm, ASR Rs
Rotate right by immediate	Rm, ROR #shift_imm
Rotate right by register	Rm, ROR Rs
Rotate right with extend	Rm, RRX

```
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]
```

รหัส นศ. _____ ชื่อ-สกุล _____ ตอน ____

```
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

AddrMode	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)