ชื่อ-สกุล ..... รหัสนักศึกษา .....

# มหาวิทยาลัยสงขลานครินทร์

## คณะวิศวกรรมศาสตร์

ข้อสอบปลายภาค: ภาคเรียนที่ 1 ปีการศึกษา 2550

วันสอบ: 7 ตุลาคม 2550 เวลา: 09:00 – 12:00

วิชา: 240- 305 Microprocessor and assembly language ห้อง A400

---

**คำสั่ง**

- ข้อสอบมีทั้งหมด 7 ข้อ 80 คะแนน ให้นักศึกษาทำข้อสอบทุกข้อ
- ให้นักศึกษาเขียนคำตอบลงในข้อสอบ
- อนุญาตให้นำเอกสาร และเครื่องคิดเลขเข้าห้องสอบได้

ชื่อ-สกุล ..... รหัสนักศึกษา .....

1. จงเติมค่าลงในช่องว่าง (10 คะแนน)

| Index method | Data | Base address register | จงยกตัวอย่าง |
|---|---|---|---|
| Preindex with writeback | Mem[base + offset] | Base + offset | (1.1) |
| Preindex | Mem[base + offset] | Not updated | (1.2) |
| Postindex | Mem[base] | Base + offset | (1.3) |

ก่อน r0 = 0x00000000 r1 = 0x00090000

Mem32[0x00090000] = 0x01010101 mem32[0x00090004] = 0x02020202

**LDR r0, [r1, #4]!**

หลัง r0 = .....(1.4) r1 = ..... (1.5)

ก่อน r0 = 0x00000000 r1 = 0x00090000

Mem32[0x00090000] = 0x01010101 mem32[0x00090004] = 0x02020202

**LDR r0, [r1, #4]**

หลัง r0 = .....(1.6) r1 = ..... (1.7)

ก่อน r0 = 0x00000000 r1 = 0x00090000

Mem32[0x00090000] = 0x01010101 mem32[0x00090004] = 0x02020202

**LDR r0, [r1], #4**

หลัง r0 = .....(1.8) r1 = ..... (1.9)

ก่อน r1 = 0x00000002 r4 = 0x00000003

sp = 0x00080014

**STMFD sp!, {r1, r4}**

หลัง sp = .....(1.10)

2. จงเขียนส่วนของโปรแกรมภาษาแอสเซมบลี ARM ให้สอดคล้องกับส่วนของโปรแกรมภาษาซีที่กำหนดให้ (20 คะแนน)

2.1 while (a != b)

```
{
   if (a > b) a -= b; else b -= a;
}
```

2.2 จงเติมส่วนของโปรแกรมภาษาแอสเซมบลีในส่วนที่ขาดหายไป เพื่อให้สอดคล้องกับส่วนของโปรแกรมภาษาซีด้านซ้ายมือ

```
short add_v1(short a, short b)
{
  return a + (b >> 1);
}
```

```
add_v1
    .....
    MOV   r0, r0, lsl #16
    MOV  r0, r0, asr #16  ; r0 = (short)r0
    .....
```

2.3 จงเขียนส่วนของโปรแกรมภาษาแอสเซมบลีทางด้านขวามือต่อให้สมบูรณ์และสอดคล้องกับส่วนของโปรแกรมภาษาซีด้านซ้ายมือ {กำหนดให้ใช้ r1 เก็บค่า N, r2 เก็บค่า sum)

```
int checksum_v8 (int *data, unsigned int N)
{
   int sum = 0;
   do
   {
       sum += *(data++);
   } while (--N != 0);
  return sum;
}
```

```
checksum_v8
    MOV   r2, #0     ;sum =0
checksum_v8_loop
```

2.4 จง rearrange การประกาศตัวแปรต่อไปนี้เพื่อให้ใช้พื้นที่ในหน่วยความจำประหยัดที่สุด

```
struct {
        short a;
        int b;
        char c;
        short d;
        char e;
}
```

3. สมมติว่าโปรแกรมภาษาแอสเซมบลีที่กำลังพัฒนาเป็นของไมโครโพรเซสเซอร์ ARM9TDMI ที่มีไปป์ไลน์ทั้งหมด 5 สเตจการทำงานได้แก่ fetch, decode, ALU, LS1 และ LS2

```
Checksum_v4
        MOV   r2, #0                ;sum = 0
        MOV   r1, #0                ;i = 0
Checksum_v4_loop
        LDRSH   r3, [r0], #2        ; r3 = *(data++)
        ADD     r1, r1, #1          ;i++
        CMP     r1, #0x40           ; compare  i, 64
        ADD     r2, r3, r2          ; sum += r3
        BCC     Checksum_v4_loop    ; if (sum<64) goto loop
        MOV     r0, r2, lsl #16
        MOV     r0, r0, asr #16     ;r0 = (short)sum
        MOV     pc, r14             ;return r0
```

3.1 จงปรับเปลี่ยนส่วนของโปรแกรมที่ให้มาเพื่อให้มีประสิทธิภาพมากขึ้นโดยใช้วิธี Load scheduling by preloading (10 คะแนน)



... vector table ... exception

(3 คะแนน)

3.2 จงปรับเปลี่ยนส่วนของโปรแกรมที่ให้มาเพื่อให้มีประสิทธิภาพมากขึ้นโดยใช้วิธี Load scheduling by unrolling (10 คะแนน)

4. การจัดการกับโหมด Exception ในโพรเซสเซอร์ ARM มีค่า Vector Table มาเกี่ยวข้อง

4.1 จงเติมคำลงในช่องว่างของ vector table ในแต่ละโหมดการทำงาน (2 คะแนน)

| **Exception** | **Mode** | **Vector table offset** |
|---|---|---|
| Reset | SVC | ..... |
| Undefined Instruction | UND | ..... |
| ..... (SWI) | SVC | +0x08 |
| ..... | ABT | +0x0C |
| ..... | ABT | ..... |
| Not assigned | - | +0x14 |
| IRQ | IRQ | ..... |
| ..... | FIQ | +0x1C |

4.2 ยกตัวอย่างมา 3 คำสั่งในการ branch ไปยังตำแหน่งตามที่ระบุไว้ใน vector table เมื่อเกิด exception (3 คะแนน)

ชื่อ-สกุล ..... รหัสนักศึกษา .....

5. จงอธิบายวิธีการของ interrupt handling ต่อไปนี้

5.1 A nonnested interrupt handler (5 คะแนน)

5.2 A re-entrant interrupt handler (5 คะแนน)

5.3 A prioritized simple interrupt handler (5 คะแนน)

6. จงอธิบายหลักการทำงานของ $I^2C$ มาพอสังเขป (5 คะแนน)

7. จงอธิบายหลักการทำงานของ SPI มาพอสังเขป (5 คะแนน)