มหาวิทยาลัยสงขลานครินทร์

คณะวิศวกรรมศาสตร์

สอบกลางภาค : ภาคการศึกษาที่ 1 **ปีการศึกษา** : 2549

วันที่สอบ : 5 สิงหาคม 2549 **เวลาสอบ** : 09.00 – 12.00 น.

รหัสวิชา : 240-305 **ห้องสอบ** : A301,A303,A305

ชื่อวิชา : Microprocessor Architecture and the Assembly Language

คำสั่ง : อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต : เครื่องเขียนต่าง ๆ เช่น ปากกา หรือ ดินสอ

ไม่อนุญาต : เครื่องคิดเลข, หนังสือ หรือ เอกสารใด ๆ เข้าห้องสอบ

เวลา : 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 2 ตอน แบ่งออกเป็น
  - ตอนที่ 1 เป็นรายละเอียดของข้อสอบ มีจำนวน 11 หน้า (รวมใบปะหน้า)
  - ตอนที่ 2 เป็นรายละเอียดของชุดคำสั่ง MCS-51 มีจำนวน 6 หน้า (รวมใบปะหน้า)
- ในส่วนของข้อสอบ แบ่งออกเป็น 2 Part ตามเนื้อหา คะแนนรวม 90 คะแนน
  - Part 1 มี 21 ข้อ 45 คะแนน
  - Part 2 มี 5 ข้อ 45 คะแนน
- เขียนคำตอบลงในข้อสอบเท่านั้น
- ใช้ดินสอทำข้อสอบได้ กรณีเขียนไม่ชัดหรืออ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- **อย่าลืม เขียน ชื่อ-นามสกุล และรหัสนักศึกษา ลงในข้อสอบทุกแผ่น**

-- ทุจริตในการสอบมีโทษขั้นต่ำปรับตกในรายวิชานี้ และพักการเรียน 1 ภาคการศึกษา --

-- โทษสูงสุดคือ ไล่ออก --

ส่วนที่ 1 เนื้อหา บทที่ 1 และ 4 (45 คะแนน)

1. จงเติมคุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ต่อไปนี้ให้สมบูรณ์ (4 คะแนน)

- หน่วยประมวลผลกลางขนาด_____บิต
- ใช้สถาปัตยกรรมแบบ_____
- หน่วยความจำข้อมูลภายในขนาด_____ไบต์
- ต่อหน่วยความจำข้อมูลภายนอกได้สูงสุด_____ไบต์
- ต่อหน่วยความจำโปรแกรมภายนอกได้สูงสุด_____ไบต์
- มีพอร์ตอนุกรมแบบ_____จำนวน_____พอร์ท
- มีพอร์ตขนานจำนวน_____พอร์ท
- มีวงจรนับ/จับเวลาขนาด_____บิต จำนวน_____ชุด
- มีวงจรรับอินเตอร์รัพท์ได้_____แหล่ง กำหนดความสำคัญได้_____ระดับ

2. จงอธิบายความแตกต่างระหว่างไมโครคอนโทรลเลอร์กับไมโครโปรเซสเซอร์ (2 คะแนน)

_____

_____

_____

_____

_____

3. จงอธิบายความหมายของคำต่อไปนี้ (อย่างสั้นๆ) (3 คะแนน)

- Data memory_____
- Program memory_____
- Instruction Register_____
- Instruction cycle_____
- Machine cycle_____
- Instruction Set_____

4. จงคำนวณเวลาที่ใช้ในการทำงานคำสั่ง DIV AB โดยกำหนดให้ไมโครคอนโทรลเลอร์ต่อกับคริสตอลขนาด 16 MHz (2 คะแนน)

- 1 pulse (crystal oscillator) = _____second
- 1 state = _____second
- 1 MC (machine cycle) = _____second
- 1 Instruction Cycle = _____second

5. คำสั่งที่ใช้ในการกระโดด ได้แก่ SJMP, AJMP และ LJMP ทั้ง 3 คำสั่ง มีระยะในการกระโดดแตกต่างกัน อย่างไร (2 คะแนน)

_____

_____

_____

_____

_____

จากรูปที่ 1 แสดงข้อมูลภายในหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 ใช้ตอบคำถามข้อ 6 - 21

- คำถามสำหรับข้อ 6-16

จงหาว่า หลังจากทำคำสั่งในแต่ละข้อแล้ว ค่าในรีจิสเตอร์และหน่วยความจำใดเปลี่ยนแปลง และเปลี่ยนเป็นค่าอะไร กำหนดให้คำสั่งในแต่ละข้อไม่เกี่ยวข้องกัน

6.
```
MOVX    A,@DPTR
MOVX    @R1,A
```
(2 คะแนน)

_____

_____

_____

7.
```
MOV     A,#05H
MOVC    A,@A+DPTR
```
(1 คะแนน)

_____

_____

8.
```
MOV     PSW,#08H
MOV     A,R4
```
(1 คะแนน)

_____

_____

9.
```
MOV     05H,90H
MOV     R0,#90H
MOV     A,@R0
```
(2 คะแนน)

10.
```
SETB    C
SUBB    A,R0
MOV     10H,C
```
(3 คะแนน)

11.
```
ADD     A,#0A8H
MOV     0F0H,A
DA      A
```
(3 คะแนน)

12.
```
MOV     A,#05
MOV     DPTR,#1000H
JMP     @A+DPTR
```
(2 คะแนน)

13.

```
ANL     A,#0FH
MOV     32H,A
ORL     33H,A
XRL     34H,#0F0H
MOV     A,@R1
RL      A
MOV     35H,A
RRC     A
MOV     36H,A
SWAP    A
XCH     A,31H
```

(7 คะแนน)

14. INC @R1 (1 คะแนน)

15. LCALL 2FFFH (2 คะแนน)

16.

```
MOV     SP,#11H
POP     B
PUSH    P1
```

(3 คะแนน)

_____

_____

_____

_____

_____

<u>คำถามสำหรับข้อ 17-21</u>

คำสั่งต่อไปนี้เป็นคำสั่งกระโดดแบบมีเงื่อนไข จงเขียนเงื่อนไขที่ใช้ในการกระโดดแต่ละข้อ และตรวจสอบเงื่อนไขดังกล่าวว่า ทำให้เกิดการกระโดดเกิดหรือไม่

17. JNB 05H,04H (1 คะแนน)

เงื่อนไขที่ตรวจเช็ค_____

เงื่อนไขเป็น (True / False)_____การกระโดด (เกิดขึ้น / ไม่เกิดขึ้น)_____

18. JC 20H (1 คะแนน)

เงื่อนไขที่ตรวจเช็ค_____

เงื่อนไขเป็น (True / False)_____การกระโดด (เกิดขึ้น / ไม่เกิดขึ้น)_____

19. JZ 20H (1 คะแนน)

เงื่อนไขที่ตรวจเช็ค_____

เงื่อนไขเป็น (True / False)_____การกระโดด (เกิดขึ้น / ไม่เกิดขึ้น)_____

20. CJNE R7,#99,20H (1 คะแนน)

เงื่อนไขที่ตรวจเช็ค_____

เงื่อนไขเป็น (True / False)_____การกระโดด (เกิดขึ้น / ไม่เกิดขึ้น)_____

21. DJNZ 0EH,20H (1 คะแนน)

เงื่อนไขที่ตรวจเช็ค_____

เงื่อนไขเป็น (True / False)_____การกระโดด (เกิดขึ้น / ไม่เกิดขึ้น)_____

ส่วนที่ 2 เนื้อหา บทที่ 2 และ 3 (45 คะแนน)

1. จงบอกหน้าที่การทำงานของขาสัญญาณของ MCS-51 ต่อไปนี้ (5 คะแนน)

/EA _____

/RD, /WR _____

/PSEN _____

ALE _____

RXD, TXD _____

2. จงตอบคำถามต่อไปนี้ (10 คะแนน)

2.1) ในการรีเซ็ทไมโครคอนโทรลเลอร์ MCS-51 จะต้องป้อนสัญญาณให้กับขาใด และใช้ลอจิกใด

_____

_____

2.2) วงจร 'Power-On-Reset' มีลักษณะการทำงานอย่างไร

_____

_____

_____

_____

_____

2.3) ไมโครคอนโทรลเลอร์ MCS-51 สามารถต่อเพิ่มหน่วยความจำภายนอกได้เท่าไร

_____

_____

2.4) บิท RS0 และ RS1 ของรีจิสเตอร์ PSW มีความเกี่ยวข้องกับ รีจิสเตอร์แบงค์ R0-R7 อย่างไร

_____

_____

_____

_____

2.5) ส่วนของหน่วยความจำข้อมูลภายใน ส่วนที่เรียกว่า bit addressable memory แตกต่างจากส่วนอื่นๆ อย่างไร

_____

_____

_____

_____

_____

2.6) การสื่อสารโดยใช้พอร์ทสื่อสารแบบอนุกรมระหว่าง MCS51 กับ PC ทำไมถึงไม่สามารถเชื่อมต่อกันได้โดยตรง

2.7) สำหรับ MCS-51 ซึ่งมีหน่วยความจำข้อมูลภายใน 256 ไบต์ (ไม่นับรวม SFR) วิธีการใช้งานหน่วยความจำในช่วงตำแหน่ง 0x00-0x7F จะแตกต่างจากการใช้ในช่วง 0x80-0xFF อย่างไร

2.8) SFR ตัวใดบ้างที่สามารถใช้งานในระดับบิทได้ บอกมา 4 ชื่อ

2.9) รีจิสเตอร์ DPTR ต่างจากรีจิสเตอร์อื่น ๆ อย่างไร

2.10) พอร์ท P0 และ P2 เกี่ยวข้องกับการใช้งานหน่วยความจำภายนอกอย่างไร

3. จงวาดรูปของวงจร MCS-51 ให้มีรายละเอียดของวงจรถึงระดับของการต่อตัวต้านทานและตัวเก็บประจุ (โดยไม่จำเป็นต้องระบุค่า R และ C) โดยมีส่วนของวงจรต่อไปนี้ (10 คะแนน)

3.1) บิท 0 ของพอร์ท P1 สำหรับการควบคุมการ ON/OFF ของ LED แบบ active LOW

3.2) Power-On-Reset

3.2) On-Chip Oscillator โดยใช้ Crystal ความถี่ 12 MHz

4. จงบอกข้อแตกต่างของหน่วยความจำชนิดต่าง ๆ ดังต่อไปนี้ (5 คะแนน)

1) ROM _____

_____

2) PROM _____

_____

3) EPROM _____

_____

4) EEPROM _____

_____

5) FLASH _____

_____

5. ถ้าไมโครคอนโทรลเลอร์ MCS-51 ต้องการหน่วยความจำ **โปรแกรม** ภายนอก โดยมี memory map ดังต่อไปนี้ จงออกแบบวิธีการต่อระบบหน่วยความจำดังกล่าว ให้วาดรูปประกอบแสดงไอซีและการเชื่อมต่อกับสัญญาณต่างๆ ที่จำเป็น (15 คะแนน)

0x0000 – 0x1FFF	EPROM#1 (8K)
0x2000 – 0x3FFF	EPROM#2 (8K)
0x4000 – 0x7FFF	EPROM#1 (16K)

8051 Instruction Set

และข้อมูลในหน่วยความจำ

สำหรับใช้ตอบคำถามใน ส่วนที่ 1 ข้อที่ 6 – 21

จงใช้ข้อมูลจากรูปต่อไปนี้ในการตอบคำถามข้อที่ 6 - 21

Internal Data Memory

17H	33H	7FH	93H	0FFH	0A3H
16H	⋮	⋮	⋮	⋮	⋮
15H	09H	35H	00H	0F1H	56H
14H	19H	34H	13H	0F0H	15H
13H	2AH	33H	2AH	⋮	⋮
12H	4BH	32H	3BH	0E0H	2BH
11H	65H	31H	55H	⋮	⋮
10H	0DEH	30H	0EH	90H	0DH
0FH	78H	⋮	⋮	8FH	79H
0EH	00H	2BH	30H	8EH	60H
0DH	0FFH	2AH	6FH	8DH	2FH
0CH	99H	29H	98H	8CH	97H
0BH	0F3H	28H	03H	8BH	0FH
0AH	0AH	27H	0A5H	8AH	0AAH
09H	03H	26H	03H	89H	03H
08H	11H	25H	1AH	88H	1FH
07H	99H	24H	99H	87H	99H
06H	80H	23H	87H	86H	8BH
05H	0FFH	22H	20H	85H	40H
04H	0EFH	21H	0F0H	84H	0DFH
03H	85H	20H	5AH	83H	17H
02H	90H	1FH	56H	82H	0F0H
01H	82H	⋮	⋮	81H	25H
00H	0F8H	18H	02H	80H	03H

External Data Memory

0FFFFH	3BH
⋮	⋮
2007H	59H
2006H	10H
2005H	27H
2004H	45H
2003H	66H
2002H	00H
2001H	87H
2000H	90H
1FFFH	44H
1FFEH	96H
⋮	⋮
0101H	07H
0100H	03H
00FFH	19H
⋮	⋮
0082H	8DH
0081H	60H
0080H	0FEH
⋮	⋮
0002H	54H
0001H	33H
0000H	0CH

Program Memory

0FFFFH	0D3H
⋮	⋮
2007H	63H
2006H	89H
2005H	2FH
2004H	49H
2003H	77H
2002H	0EEH
2001H	70H
2000H	0B0H
1FFFH	78H
1FFEH	95H
⋮	⋮
0101H	0AH
0100H	03H
00FFH	90H
⋮	⋮
0082H	8FH
0081H	90H
0080H	0F0H
⋮	⋮
0002H	53H
0001H	73H
0000H	22H

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2		CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH				87

DPL 01H DPH 20H PSW 00H

TCON 09H SP 40H PC 2000H

ACC 0C3H B 01H P0 25H

P1 03H P2 02H P3 2FH

PSW	C	AC	F0	RS1	RS0	OV	-	P

** P = even parity

80C51 FAMILY INSTRUCTION SET

Table 7. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings[1]

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data	8-bit constant included in the instruction.
#data 16	16-bit constant included in the instruction
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	12
ADDC	A,#data	Add immediate data to A_{CC} with carry	2	12
SUBB	A,Rn	Subtract Register from A_{CC} with borrow	1	12
SUBB	A,direct	Subtract direct byte from A_{CC} with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from A_{CC} with borrow	1	12
SUBB	A,#data	Subtract immediate data from A_{CC} with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12

All mnemonics copyrighted © Intel Corporation 1980

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS (Continued)				
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A and B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator left	1	12
RLC	A	Rotate Accumulator left through the carry	1	12
RR	A	Rotate Accumulator right	1	12
RRC	A	Rotate Accumulator right through the carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12

All mnemonics copyrighted © Intel Corporation 1980

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
DATA TRANSFER (Continued)				
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	RN,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A_{CC}	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to A_{CC}	1	24
MOVX	A,@Ri	Move external RAM (8-bit addr) to A_{CC}	1	24
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A_{CC}	1	24
MOVX	A,@Ri,A	Move A_{CC} to external RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move A_{CC} to external RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A_{CC}	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to carry	2	24
ANL	C,/bit	AND complement of direct bit to carry	2	24
ORL	C,bit	OR direct bit to carry	2	24
ORL	C,/bit	OR complement of direct bit to carry	2	24
MOV	C,bit	Move direct bit to carry	2	12
MOV	bit,C	Move carry to direct bit	2	24
JC	rel	Jump if carry is set	2	24
JNC	rel	Jump if carry not set	2	24

All mnemonics copyrighted © Intel Corporation 1980

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
BOOLEAN VARIABLE MANIPULATION (Continued)				
JB	rel	Jump if direct bit is set	3	24
JNB	rel	Jump if direct bit is not set	3	24
JBC	bit,rel	Jump if direct bit is set and clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute subroutine call	2	24
LCALL	addr16	Long subroutine call	3	24
RET		Return from subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute jump	2	24
LJMP	addr16	Long jump	3	24
SJMP	rel	Short jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is zero	2	24
JNZ	rel	Jump if Accumulator is not zero	2	24
CJNE	A,direct,rel	Compare direct byte to A_{CC} and jump if not equal	3	24
CJNE	A,#data,rel	Compare immediate to A_{CC} and jump if not equal	3	24
CJNE	RN,#data,rel	Compare immediate to register and jump if not equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	24
DJNZ	Rn,rel	Decrement register and jump if not zero	2	24
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	24
NOP		No operation	1	12

All mnemonics copyrighted © Intel Corporation 1980