 ชื่อ.....

มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบกลางภาค: ภาคการศึกษาที่ 1 **ปีการศึกษา:** 2545

วันที่สอบ: กันยายน 2545 **เวลาสอบ:** น.

รหัสวิชา: 240-235 **ห้องสอบ:**

ชื่อวิชา: MICROPROCESSOR ARCHITECTURE AND SYSTEM DESIGN

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

เวลา: 3 ชั่วโมง (180 นาที)

รายละเอียดของข้อสอบ: ข้อสอบมีทั้งหมด 13 หน้า

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือและสมุดโน้ตใดๆ เข้าห้องสอบ

คำสั่ง:

- ให้ทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ
- เขียนชื่อและรหัสให้ชัดเจนในข้อสอบ**ทุกแผ่น**
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งเพิ่มเติมในแต่ละข้อให้ชัดเจน

1. จงเติมคำในช่องว่างต่อไปนี้ (10 คะแนน)

- ขา PSEN ใน MCS-51 มีหน้าที่.....
- ขา EA ใน MCS-51 มีหน้าที่.....
- หน่วยความจำของ MCS-51 แบ่งออกเป็น .....ประเภท ได้แก่.....

.....

- สถาปัตยกรรมแบบ Harvard คือ.....

.....

- MCS-51 มีพอร์ตในตัวจำนวน.....พอร์ตหากพอร์ตภายในไม่พอสามารถเพิ่มพอร์ตได้โดยการ.....

.....

.....

- EPROM Emulator คืออะไร มีประโยชน์อย่างไร.....

.....

.....

.....

- In-Circuit Emulator คืออะไร มีประโยชน์อย่างไร.....

.....

.....

.....

- จงยกตัวอย่างชื่อโปรแกรมซิมูเลเตอร์ (Simulator) และแอสเซมเบลอร์ (Assembler) ของ MCS-51 มาอย่างละ 1 ตัว.....
- เพราะเหตุใดจึงถือว่า MCS-51 ใช้ระบบ I/O แบบ Memory mapped I/O แทนที่จะเป็น Isolated I/O.....

.....

.....

.....

.....

- จงยกตัวอย่างคำสั่งแบบ Read-Modify-Write instruction มาอย่างน้อย 3 คำสั่ง.....

.....

- Machine Cycle คือ.....

.....

.....สำหรับกรณีของ MCS-51 นั้น 1 Machine Cycle ใช้เวลาเท่ากับ.....คาบเวลาออสซิเลเตอร์

- ความสามารถในการเลือกรีจิสเตอร์แบงก์ใน MCS-51 มีประโยชน์อย่างไร.....

.....

.....

- Remote monitor คือ.....

.....

2. จากความรู้เรื่องไมโครโพรเซสเซอร์ Z-80 และ MCS-51 ที่ได้เรียนมาจงเปรียบเทียบคุณสมบัติ รวมทั้งข้อดีข้อเสียของไมโครโพรเซสเซอร์ทั้งสองตระกูล พร้อมทั้งยกตัวอย่างงานที่เหมาะกับไมโครโพรเซสเซอร์แต่ละแบบ (5 คะแนน)

 ชื่อ.....

3. จากแต่ละโปรแกรมที่แสดงต่อไปนี้ จงหาว่าหลังจากการทำงานของแต่ละคำสั่งเสร็จสิ้นแล้ว ค่ารีจิสเตอร์อะไรเปลี่ยนแปลงบ้าง และจงหาค่าหลังจากการเปลี่ยนแปลงของแต่ละรีจิสเตอร์และหน่วยความจำข้อมูลภายใน (20 คะแนน)

CY	AC	F0	RS1	RS0	OV	—	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).
OV	PSW.2	Overflow Flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

ตัวช่วย : โครงสร้างของบิตข้อมลภายในรีจิสเตอร์ PSW

ก่อนรันโปรแกรม

ACC	75h
PSW	C7h
B	A8h
SP	C7h
P0	10h
P1	09h
P2	10h
P3	05h

0Ah	a9h
09h	37h
08h	62h
07h	34h
06h	c5h
05h	67h
04h	07h
03h	00h
02h	03h
01h	10h
00h	05h

โปรแกรมที่1

MOV P1,B

MOV A,#00

INC R1

ANL A,05H

หลังรันโปรแกรม

ACC	
PSW	
B	
SP	
P0	
P1	
P2	
P3	

0Ah	
09h	
08h	
07h	
06h	
05h	
04h	
03h	
02h	
01h	
00h	

ก่อนรันโปรแกรม

ACC	75h
PSW	C7h
B	A8h
SP	07h
P0	10h
P1	09h
P2	10h
P3	05h

0Ah	a9h
09h	37h
08h	62h
07h	34h
06h	c5h
05h	67h
04h	07h
03h	00h
02h	03h
01h	10h
00h	05h

โปรแกรมที่2

SETB C

SETB RS0

MOV R1,A

PUSH 0AH

หลังรันโปรแกรม

ACC	
PSW	
B	
SP	
P0	
P1	
P2	
P3	

0Ah	
09h	
08h	
07h	
06h	
05h	
04h	
03h	
02h	
01h	
00h	

 ชื่อ.....

ก่อนรันโปรแกรม

ACC	75h
PSW	07h
B	A8h
SP	C7h
P0	10h
P1	09h
P2	10h
P3	05h

0Ah	a9h
09h	37h
08h	62h
07h	34h
06h	c5h
05h	67h
04h	07h
03h	00h
02h	03h
01h	10h
00h	05h

โปรแกรมที่3

INC R0

MOV PSW,08h

INC R1

MOV P1,00

หลังรันโปรแกรม

ACC	
PSW	
B	
SP	
P0	
P1	
P2	
P3	

0Ah	
09h	
08h	
07h	
06h	
05h	
04h	
03h	
02h	
01h	
00h	

ก่อนรันโปรแกรม

ACC	75h
PSW	C7h
B	A8h
SP	07h
P0	10h
P1	09h
P2	10h
P3	05h

0Ah	a9h
09h	37h
08h	62h
07h	34h
06h	c5h
05h	67h
04h	07h
03h	00h
02h	03h
01h	10h
00h	05h

โปรแกรมที่4

MOV A,B

INC ACC

ADDC A,04H

MOV B,#00h

หลังรันโปรแกรม

ACC	
PSW	
B	
SP	
P0	
P1	
P2	
P3	

0Ah	
09h	
08h	
07h	
06h	
05h	
04h	
03h	
02h	
01h	
00h	

ก่อนรันโปรแกรม

ACC	75h
PSW	07h
B	A8h
SP	C7h
P0	10h
P1	09h
P2	10h
P3	05h

0Ah	a9h
09h	37h
08h	62h
07h	34h
06h	c5h
05h	67h
04h	07h
03h	00h
02h	03h
01h	10h
00h	05h

โปรแกรมที่5

SETB C

SUBB A,#03

INC @R0

DEC P2

หลังรันโปรแกรม

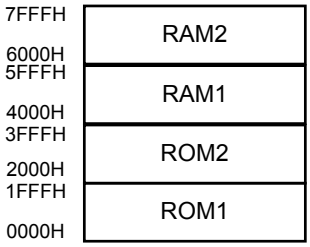
ACC	
PSW	
B	
SP	
P0	
P1	
P2	
P3	

0Ah	
09h	
08h	
07h	
06h	
05h	
04h	
03h	
02h	
01h	
00h	

4. จงเขียนโปรแกรมไมโครคอนโทรลเลอร์ MCS-51 เพื่อบวกค่าตัวเลขขนาด 8 บิทจำนวน 25 ตัวที่เก็บอยู่ในหน่วยความจำข้อมูลตำแหน่งเริ่มต้นที่แอดเดรส 2000H โดยให้ผลลัพธ์จากการบวกมีขนาด 16 บิท หลังจากบวกเสร็จให้เก็บผลลัพธ์การบวกไว้ในหน่วยความจำตำแหน่ง 1000H และ 1001H โดยเก็บไบต์ต่ำและไบต์สูงตามลำดับ (10 คะแนน)

 ชื่อ.....

5. จงเขียนแผนภาพของระบบหน่วยความจำภายนอกของ MCS-51 โดยใช้ไอซีรอม 2 ตัวและมีแรม 2 ตัวโดยให้มี Memory map ดังรูป (15 คะแนน)



6. จงออกแบบวงจรขยายพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 โดยใช้ไอซีขยายพอร์ต 8255 ทำการ Map กับแอดเดรส C000H-C003H (10 คะแนน)

7. อินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ MCS-51 มีกี่แบบ อะไรบ้าง จงอธิบายรายละเอียดกลไกการอินเตอร์รัพต์แต่ละแบบพร้อมทั้งยกตัวอย่างประกอบคำอธิบายอย่างละเอียด (10 คะแนน)

อ. ปัญญยศ ไชยกาฬ ผู้ออกข้อสอบ
วันที่ 18 กันยายน 2545

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.

Note 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 instruction.

#data 16 — 16-bit constant included in instruction.

addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 — 11-bit destination address. Used by ACALL & 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 Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct 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

All mnemonics copyrighted ©Intel Corporation 1980

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & 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

Mnemonic		Description	Byte	Oscillator Period
LOGICAL OPERATIONS (Continued)				
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
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

All mnemonics copyrighted © Intel Corporation 1980

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (Continued)				
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 Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc 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 Acc	1	12

Mnemonic		Description	Byte	Oscillator Period
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
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & 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

All mnemonics copyrighted © Intel Corporation 1980

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
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 Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
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

82C55A