Document No : UniKL MFI_SD_AC41
Revision No: 02
Effective Date: 01 December 2008

CONFIDENTIAL

SETA

UNIVERSITI KUALA LUMPUR
Malaysia France Institute

FINAL EXAMINATION
JANUARY 2010 SESSION

SUBJECT CODE	:	FSB23203
SUBJECT TITLE	:	MICROCONTROLLER
LEVEL	:	BACHELOR
TIME / DURATION	:	12.00pm – 3.00pm (3 HOURS)
DATE	:	26 April 2010

INSTRUCTIONS TO CANDIDATES

1. Please read the instructions given in the question paper CAREFULLY.
2. This question paper is printed on both sides of the paper.
3. Please write your answers on the answer booklet provided.
4. Answer should be written in blue or black ink except for sketching, graphic and illustration.
5. This question paper consists of TWO (2) sections, Section A and B. Answer all questions in Section A. For Section B, answer three (3) questions only.
6. Answer all questions in English.
7. Instruction set and data sheet are attached in Appendixes

THERE ARE 6 PAGES OF QUESTIONS AND 7 APPENDIXES, EXCLUDING THIS PAGE.

SECTION A (Total: 40 marks)

INSTRUCTION: Answer ALL questions.

Please use the answer booklet provided.

Question 1

(a) Describe briefly two criteria to be considered for choosing a microcontroller for a target application.

(4 marks)

(b) What is a register and list down two registers in 8051?

(2 marks)

(c) List two devices which can be attached to a general purpose microprocessor to make it functional and explain their purpose.

(4 marks)

(d) A 16 bits data equals to _____ nibble(s) or _____ word(s).

(3 marks)

(e) What is the most important factor in choosing a microcontroller for a PC interface embedded product?

(1 mark)

Question 2

(a) What is the hexadecimal value in register A and the status of CY, OV and P flags after the following instructions executed?

```
MOV   A,#75H
MOV   B,#15
DIV   AB
```

(5 marks)

(b) State the selected register bank and range of RAM location after the following program:

```
SETB  RS0
MOV   R2,#78
MOV   R6,#05H
MOV   R1,#08
```

(3 marks)

(c) State the data (in hexadecimal) contents in RAM locations 0x0E after the program in Question 2(b) executed.

(2 marks)

Question 3

(a) What type of addressing mode, indicated by **arrow 1**?

```
MOV     R1, #23
MOV     A, #10
MOV     @R1, #10h
SUBB    A, @R1
```

(1 mark)

(b) What will be the final hexadecimal value in Accumulator and explain the process?

(3 marks)

Question 4

```
CLR     C
MOV     A, #04CH
SUBB    A, #06EH
```

(a) The program above shows the subtraction process for unsigned number. Show the calculation manually.

(2 marks)

(b) What are the values of carry flag and accumulator A after SUBB instruction has been executed?

(4 marks)

(c) Comparing with manual calculation, determine whether the subtraction result in the accumulator A is correct. Justify your answer.

(1 mark)

(d) If the answer in accumulator A is incorrect, modify the above program to obtain the correct final result.

(2 marks)

(e) What are the flags that will be involved in solving unsigned and sign number addition?

(3 marks)

SECTION B (Total: 60 marks)

INSTRUCTION: Answer only THREE (3) questions.
Please use the answer booklet provided.

Question 5

(a) Explain briefly the serial and parallel communication.

(5 marks)

(b) Which timer register will be involved and calculating the baud rate. Calculate the value to be loaded into the register for *baud rate 4800.* Assume Crystal clock: *11.0592MHz*

(5 marks)

(c) Figure 1 shows the serial communication between a computer and microcontroller 8051. Write program for 8051 to receive characters M, F and I serially at baud rate 4800. Use serial mode 1 and crystal clock 11.0592Mz.

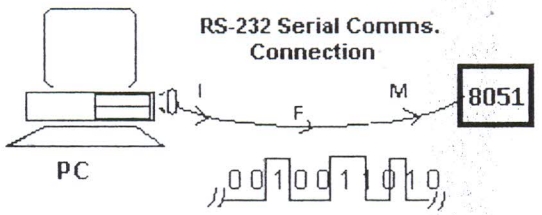


Figure 1: Serial interfacing

(10 marks)

Question 6

(a) List the two advantages of interrupt method in microcontroller.

(2 marks)

(b) Explain the different between polling and interrupt method.

(2 marks)

(c) Program IP register to assign highest priority to serial port interrupt.

(2 marks)

(d) What happens if INT0, serial port interrupt, TF0 activated at the same time? Assume interrupts are at same triggered.

(2 marks)

(2 marks)

(e) Create an alarm system using assembly language programming with the following features:

Assume that INT1 pin is connected to a discrete-sensor at the entrance of a bank restricted room. The discrete-sensor gives normally high condition. Whenever it goes low, it should turn the buzzer ON. The buzzer is connected to P1.1 and it stays ON until an authorized person press a reset button (normally high condition), connected to P1.2.

(12 marks)

Question 7

(a) List the two timers of the 8051 and draw their associated registers.

(4 marks)

(b) Which timer is selected in the instruction "**MOV TMOD, #20H**"?

(1 mark)

(c) Assume that XTAL=11.0592, TMOD=#01
What value do we need to load into the timer's registers if we want to have delay of 2ms (milliseconds)?

(4 marks)

(d) A pulse width modulation (PWM) method is used to control the speed of DC motor by changing the width of the pulse. Use *Timer 0 Mode 1* to generate a square wave of 2.5kHz frequency on pin P1.1 continuously. A switch is provided to start the motor when it is pressed. Create a program to perform the task. Assume that XTAL=11.0592MHz.

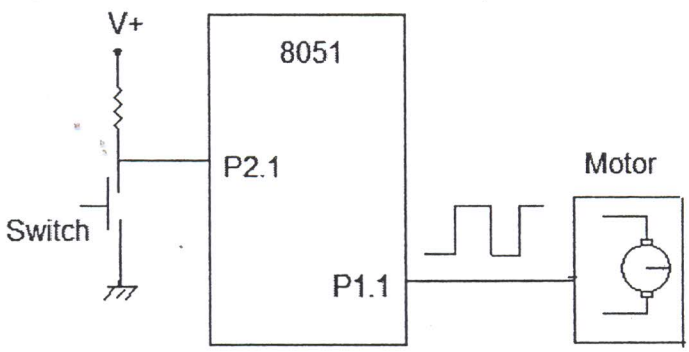


Figure 2: Controlling DC Motor speed

(12 marks)

Question 8

Consider the following program and answer the questions.

```
        MOV     A, #FFH
        MOV     P1, A
        ACALL   DELAY
        MOV     P1, #0
DELAY:  MOV     R2, #130
AGAIN:  NOP
        DJNZ    R2, AGAIN
        RET
```

Assuming the crystal frequency (XTAL) = 11.0592 MHz and Clocks per Machine Cycle =6.

(a) For an 8051 system of 11.0592 MHz, find the time delay for the delay subroutine. (5 marks)

(b) Find how long the port P1 is high? (3 marks)

(c) A swarm robot is programmed to move forward using 8051 microcontroller with XTAL=11.0592 MHz and two DC motors as shown in Figure 3 and 4. An Infra red sensor is located in front of robot chassis to detect the obstacle and it is connected to the interrupt of the microcontroller. If the sensor is triggered, then mobile robot will stop and turn left (for a certain degree). Then, the robot will move forward again. Write program to perform the robot movement as mention above.

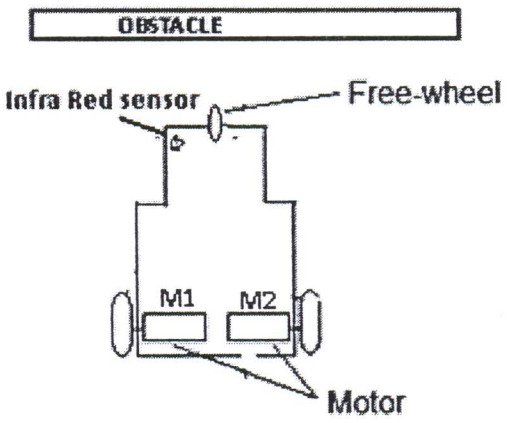


Figure 3: Swarm robot design

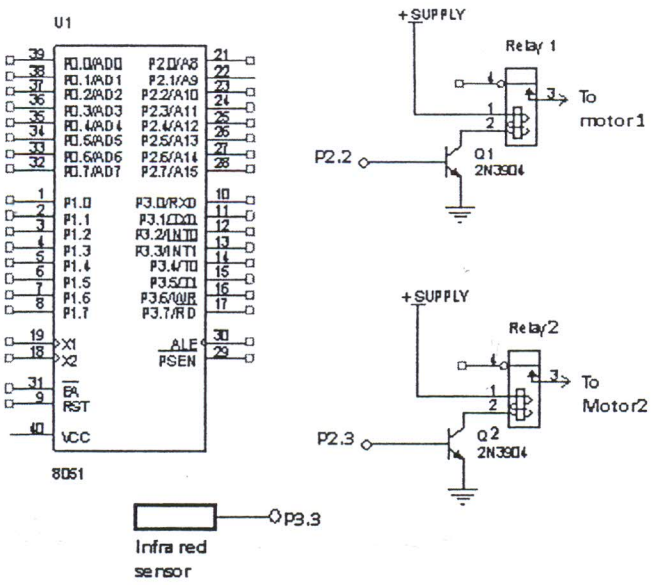


Figure 4: Input/Output connection

(12 marks)

END OF QUESTION

APPENDIX 1

DATA SHEET AND INSTRUCTION SET

ARITHMETIC OPERATORS

MNEMONIC	DESCRIPTION	BYTES	CYCLES	C	OV	AC
ADD A, Rn	Add register to ACC	1	1	x	x	x
ADD A, direct	Add direct byte to ACC	2	1	x	x	x
ADD A, @Ri	Add indirect RAM to ACC	1	1	x	x	x
ADD A, #data	Add immediate data to ACC	2	1	x	x	x
ADDC A, Rn	Add register to ACC with Carry	1	1	x	x	x
ADDC A, direct	Add direct byte to ACC with Carry	2	1	x	x	x
ADDC A, @Ri	Add indirect RAM to ACC with Carry	1	1	x	x	x
ADDC A, #data	Add immediate data to ACC with Carry	2	1	x	x	x
SUBB A, Rn	Subtract Register from ACC with borrow	1	1	x	x	x
SUBB A, direct	Subtract indirect RAM from ACC with borrow	2	1	x	x	x
SUBB A, @Ri	Subtract indirect RAM from ACC with borrow	1	1	x	x	x
SUBB A, #data	Subtract immediate data from ACC with borrow	2	1	x	x	x
INC A	Increment ACC	1	1			
INC Rn	Increment register	1	1			
INC direct	Increment direct byte	2	1			
INC @Ri	Increment direct RAM	1	1			
DEC A	Decrement ACC	1	1			
DEC Rn	Decrement Register	1	1			
DEC direct	Decrement direct byte	2	1			
DEC @Ri	Decrement indirect RAM	1	1			
INC DPTR	Increment Data Pointer	1	2			
MUL AB	Multiply A and B	1	4	0	x	
DIV AB	Divide A by B	1	4	0	x	
DAA	Decimal Adjust ACC	1	1	x		

BOOLEAN OPERATORS

MNEMONIC	DESCRIPTION	BYTES	CYCLES	C	OV	AC
CLR C	Clear carry flag	1	1	0		
CLR bit	Clear direct bit	2	1			
SETB C	Set carry flag	1	1	1		
SETB bit	Set direct bit	2	1			
CPL C	Complement carry flag	1	1	x		
CPL bit	Complement direct bit	2	1			
ANL C,bit	AND direct bit to carry	2	2	x		
ANL C,/bit	AND complement of direct bit to carry	2	2	x		
ORL C,bit	OR direct bit to carry	2	2	x		
ORL C,/bit	OR complement of direct bit to carry	2	2	x		
MOV C,bit	Move direct bit to carry	2	1	x		
MOV bit,C	Move carry to direct bit	2	2			
JC rel	Jump if carry is set	2	2			
JNC rel	Jump if carry is NOT set	2	2			
JB bit,rel	Jump if direct bit is set	3	2			
JNB bit,rel	Jump if direct bit is NOT set	3	2			
JBC bit,rel	Jump if direct bit is set and clear that bit	3	2			

JUMPS AND BRANCHES

MNEMONIC	DESCRIPTION	BYTES	CYCLES	C	OV	AC
ACALL addr11	Absolute call within 2K page	2	2			
LCALL addr16	Absolute call (Long call)	3	2			
RET	Return from subroutine	1	2			
RETI	Return from interrupt	1	2			
AJMP addr11	Absolute jump within 2K page	2	2			
LJMP addr16	Absolute jump (Long jump)	3	2			
SJMP rel8	Relative jump within +/- 127 bytes (Short jump)	2	2			
JMP @A+DPTR	Jump direct relative to DPTR	1	2			
JZ rel8	Jump if ACC is zero	2	2			
JNZ rel8	Jump if ACC is NOT zero	2	2			
CJNE A,direct,rel8	Compare direct byte to ACC, jump if NOT equal	3	2	x		
CJNE A,#data,rel8	Compare immediate to ACC, jump if NOT equal	3	2	x		
CJNE Rn,#data,rel8	Compare immediate to register, jump if NOT equal	3	2	x		
CJNE @Ri,#data,rel8	Compare immediate to indirect, jump if NOT equal	3	2	x		
DJNZ Rn,rel8	Decrement register, jump if NOT zero	2	2			
DJNZ direct,rel8	Decrement direct byte, jump if NOT zero	3	2			
NOP	No operation (Skip to next instruction)	1	1			

LOGICAL OPERATIONS

MNEMONIC	DESCRIPTION	BYTES	CYCLES	C	OV	AC
ANL A,Rn	AND register to ACC	1	1			
ANL A,direct	AND direct byte to ACC	2	1			
ANL A,@Ri	AND indirect RAM to ACC	1	1			
ANL A,#data	AND immediate data to ACC	2	1			
ANL direct,A	AND ACC to direct byte	2	1			
ANL direct,#data	AND immediate data to direct byte	3	2			
ORL A,Rn	OR register to ACC	1	1			
ORL A,direct	OR direct byte to ACC	2	1			
ORL A,@Ri	OR indirect RAM to ACC	1	1			
ORL A,#data	OR immediate data to ACC	2	1			
ORL direct,A	OR ACC to direct byte	2	1			
ORL direct,#data	OR immediate data to direct byte	3	2			
XRL A,Rn	XOR register to ACC	1	1			
XRL A,direct	XOR direct byte to ACC	2	1			
XRL A,@Ri	XOR indirect RAM to ACC	1	1			
XRL A,#data	XOR immediate data to ACC	2	1			
XRL direct,A	XOR ACC to direct byte	2	1			
XRL direct,#data	XOR immediate data to direct byte	3	2			
CLR A	Clear the ACC	1	1			
CPL A	Complement the ACC	1	1			
RL A	Rotate the ACC left	1	1			
RLC A	Rotate the ACC left through Carry	1	1	x		
RR A	Rotate the ACC right	1	1			
RRC A	Rotate the ACC right through Carry	1	1	x		
SWAP A	Swap nibbles in the ACC	1	1			

DATA TRANSFER

MNEMONIC	DESCRIPTION	BYTES	CYCLES	C	OV	AC
MOV A,Rn	Move Register to ACC	1	1			
MOV A,direct	Move Direct byte to ACC	2	1			
MOV A,@Ri	Move Indirect byte to ACC	1	1			
MOV A,#data	Move Immediate data to ACC	2	1			
MOV Rn,A	Mov ACC to Register	1	1			
MOV Rn,direct	Move Direct byte to Register	2	2			
MOV Rn,#data	Move Immediate data to Register	2	1			
MOV direct,A	Move ACC to Direct byte	2	1			
MOV direct,Rn	Move Register to Direct byte	2	2			
MOV direct,direct	Move Direct byte to Direct byte	3	2			
MOV direct,@Ri	Mov Indirect RAM to Direct byte	3	2			
MOV direct,#data	Move Immediate data to Direct byte	3	2			
MOV @Ri,A	Move ACC to Indirect RAM	1	1			
MOV @Ri,direct	Move direct byte to indirect RAM.	2	2			
MOV @Ri,#data	Move Immediate data to Indirect RAM	2	1			
MOV DPTR,#data16	Load datapointer with 16 bit constant	3	2			
MOVC A,@A+DPTR	Move code byte at ACC+DPTR to ACC	1	2			
MOVC A,@A+PC	Move code byte at ACC+PC to ACC	1	2			
MOVX A,@Ri	Move external RAM to ACC	1	2			
MOVX @Ri,A	Move ACC to external RAM	1	2			
MOVX A,@DPTR	Move external RAM to ACC	1	2			
MOVX @DPTR,A	Move ACC to external RAM	1	2			
PUSH direct	Push direct byte to stack	2	2			
POP direct	Pop direct byte from stack	2	2			
XCH A,Rn	Exchange register with ACC	1	1			
XCH A,direct	Exchange direct byte with ACC	2	1			
XCH A,@Ri	Exchange indirect RAM with ACC	1	1			
XCHD A,@Ri	Exchange low order digit indirect RAM with ACC	1	1			

APPENDIX 2

SUPPORT DATA FOR 8051 PROGRAMMING

8051 SFR REGISTERS

Byte address	Bit address b7 b6 b5 b4 b3 b2 b1 b0	
FFh		
F0h	B	*
E0h	A (accumulator)	*
D0h	PSW	*
B8h	IP	*
B0h	Port 3 (P3)	*
A8h	IE	*
A0h	Port 2 (P2)	*
99h	SBUF	
98h	SCON	*
90h	Port 1 (P1)	*
8Dh	TH1	
8Ch	TH0	
8Bh	TL1	
8Ah	TL0	
89h	TMOD	
88h	TCON	*
87h	PCON	
83h	DPH	
82h	DPL	
81h	SP	
80h	Port 0 (P0)	*

SFR register layout

** indicates the SFR registers which are bit addressable*

8051 TIMER/COUNTER PROGRAMMING SUPPORT DATA

The TMOD register (**Timer Mode** Control) is an SFR and is used to define the Timer/Counter mode of operation.

TMOD register

Gate msb	C/T	M1	M0	Gate	C/T	M1	M0 *lsb*
--------- timer 1 ----------|				----------timer 0 ----------			

The **Gate** bit can be set to 0 for these examples.
The **C/T** bit is set to 1 for COUNTER operation and it is set to 0 for TIMER operation.

MI and M2 bits define different modes i.e.:

M1	**M0**	
0	0	mode 0: 13 bit mode seldom used these days. Ignore.
0	1	mode 1: 16-bit mode
1	0	mode 2: 8-bit mode (with auto reload feature)
1	1	mode 3: ignore for now

TCON Timer Control Register

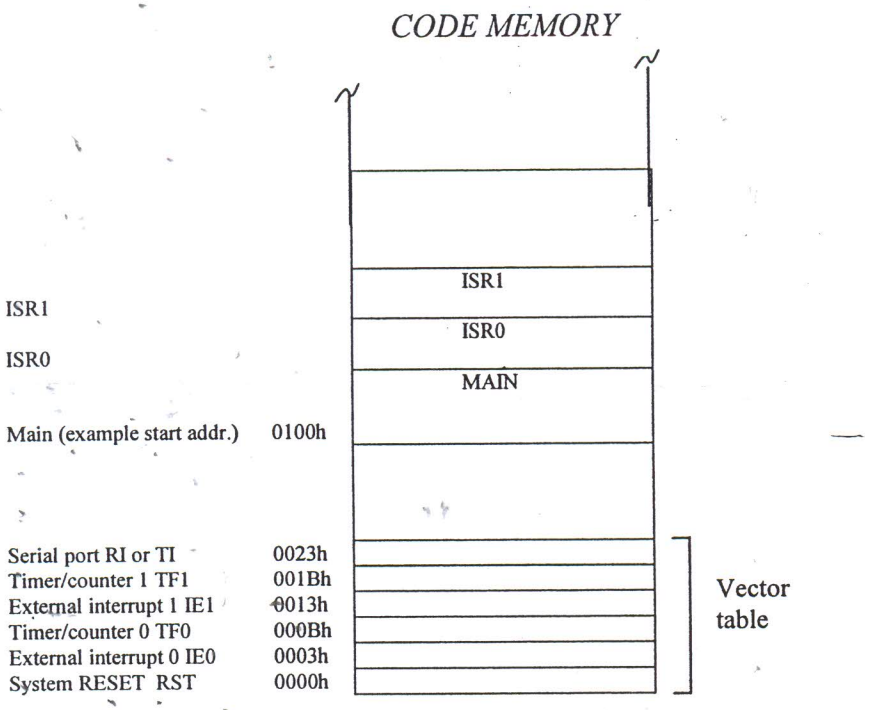
TCON register

TF1 msb	TR1	TF0	TR0	IE1	IT1	IE0	IT0 lsb

TF1 Timer 1 overflow flag. Set when timer overflows. Clear by software
TR1 Set to enable Timer 1
TF0 Timer 0 overflow flag. Set when timer overflows. Clear by software
TR0 Set to enable Timer 0
IEI Interrupt flag for interrupt 1
ITI Set for negative edge trigger for interrupt 1, clear for level trigger
IE0 Interrupt flag for interrupt 0
IT0 Set for negative edge trigger for interrupt 0, clear for level trigger

8051 INTERRUPT PROGRAMMING SUPPORT DATA

THE INTERRUPT VECTOR TABLE



IE Interrupt Enable register

EA msb			ES	ET1	EX1	ET0	EX0 lsb

EA	Enable all	**ES**	Enable serial port
ET1	Enable Timer/counter 1	**EX1**	Enable external interrupt 1
ET0	Enable Timer/counter 0	**EX0**	Enable external interrupt 0

8051 SERIAL PORT PROGRAMMING SUPPORT DATA

The SCON is an SFR register, used for configuring and monitoring the serial port.

SCON register

SM0 msb	SM1	SM2	REN	TB8	RB8	TI	RI lsb

SM0, SM1 bits define the mode of operation, such as the number of data bits (8 or 9), the clock source etc. Our examples will use *mode 3,* which specifies 9 data bits (8 data plus a parity bit) with the clock source being Timer/Counter 1.

SM2 is set to 0 for normal operation

REN is set to 1 to enable reception, 0 to disable reception

TB8 is the ninth bit (parity bit) to be transmitted

RB8 is the ninth bit received (parity bit)

TI Transmit Interrupt flag. 1 indicates that transmit buffer (SBUF) is empty. This flag must be cleared by software.

RI Receive Interrupt flag. 1 indicates that data has been received in the receive buffer (SBUF). This flag must be cleared by software.

Serial port baud rate configuration.

For exam purposes assume that the following values for TH1 to achieve the listed baud rates.

Baud rate	Timer/Counter1 TH1 value
300	A0h
1,200	D0h
2,400	FAh
9,600	FDh