UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan Pendidikan Berterusan (UTMSPACE)

FINAL EXAMINATION / *PEPERIKSAAN AKHIR*
SEMESTER 2 – SESSION 2015 / 2016
PROGRAM KERJASAMA

COURSE CODE *KOD KURSUS*	:	DDPE 2803/DDE 2334
COURSE NAME *NAMA KURSUS*	:	MICROPROCESSOR *PEMPROSES MIKRO*
YEAR / PROGRAMME *TAHUN / PROGRAM*	:	DDPB/DDPE/DDPK/DDPP
DURATION *TEMPOH*	:	2 HOURS 30 MINUTES/ *2 JAM 30 MINIT*
DATE *TARIKH*	:	APRIL 2016

INSTRUCTION/*ARAHAN* :

1. Answer **ALL** questions in the answer booklet(s) provided.
   *Jawab **SEMUA** soalan di dalam buku jawapan yang disediakan.*

2. Students are provided with the 8051 instruction set and specification by the college.
   Pelajar-pelajar dibekalkan dengan set suruhan dan spesifikasi 8051 oleh pihak kolej.

(You are required to write your name and your lecturer's name on your answer script)
(Pelajar dikehendaki tuliskan nama dan nama pensyarah pada skrip jawapan)

NAME / *NAMA*	:	.....
I.C NO. / *NO. K/PENGENALAN*	:	.....
YEAR / COURSE *TAHUN / KURSUS*	:	.....
COLLEGE NAME *NAMA KOLEJ*	:	.....
LECTURER'S NAME *NAMA PENSYARAH*	:	.....

This examination paper consists of 12 pages including the cover
Kertas soalan ini mengandungi 12 *muka surat termasuk kulit hadapan*

S1. Every 8051 family member starts its program at address _____ when it is powered up.
Setiap ahli keluarga 8051 memulakan aturcara pada alamat _____ apabila ianya dihidupkan.

(2 marks/*markah*)

S2. On power-up, the 8051 uses bank _____ for registers R0 to R7.
Ketika bekalan kuasa dihidupkan, 8051 menggunakan bank _____ untuk daftar-daftar R0 hingga R7.

(2 marks/*markah*)

S3. Determine the ROM memory address range of each of the following 8051 variant chips.
a) AT89C51 with 4 KB ROM
b) DS89C420 with 16 KB ROM
Tentukan julat alamat ingatan ROM untuk setiap serpih jenis 8051 berikut.
a) AT89C51 dengan 4 KB ROM
b) DS89C420 dengan 16 KB ROM

(4 marks/*markah*)

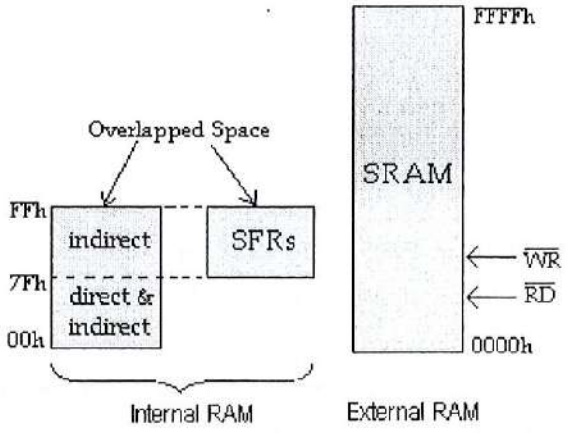

Figure 1/*Rajah 1*

S4. Refer to Figure 1. Write an instruction(s) to clear the content of internal RAM at byte address location of 88H.
Rujuk Rajah 1. Tulis satu arahan/arahan-arahan untuk mengosongkan kandungan RAM dalaman pada lokasi alamat byte 88H.

(4 marks / *markah*)

S5. Show the status of the CY, AC, and P flags after the execution of the following arithmetic instructions.
Tunjukkan status bendera CY, AC dan P setelah perlaksanaan arahan aritmetik berikut.

```
MOV   A,#9CH
ADD   A,#64H
```

(4 marks/*markah*)

S6. State the address of R5 after the execution of the following instruction.
Nyatakan alamat bagi R5 setelah arahan berikut dilaksanakan.

```
SETB  PSW.4
```

(2 marks/*markah*)

S7. Write an 8051 program based on the following sequence of requirements:
Tuliskan satu program 8051 berdasarkan kepada turutan keperluan-keperluan berikut:

i). Clear the accumulator and complement its contents.
Bersihkan pengumpuk dan terbalikkan logik kandungannya.

ii). Add 47H.
Tambah 47H.

iii) Subtract 92H.
Tolak 92H

iv) Add the content of address 64H.
Tambah dengan kandungan alamat 64H.

v) If the result equals zero, clear Port2 and the end of the program. Otherwise, jump to the first instruction.
Jika hasil keputusan bersamaan sifar, bersihkan Liang2 dan program tamat. Sebaliknya, lompat ke arahan pertama.

(7 marks/*markah*)

S8. Write a program to clear ACC and then add the value 3 to the ACC for ten times. Save the result in external RAM as memory location 3000H.
Tuliskan satu program untuk membersihkan ACC dan kemudian menambahkan ACC dengan nilai 3 sebanyak sepuluh kali. Simpan keputusannya dalam lokasi RAM luaran beralamat 3000H.

(6 marks/*markah*)

S9. For a machine cycle of 1 μs, calculate the time delay in the following subroutine.
Untuk kitaran mesin 1 μs, kirakan lengah masa dalam subrutin berikut.

```
                                    ;Machine Cycle/Kitaran Mesin
        MOV   R5,#9H        ;   1
HERE1:  MOV   R4,#64H       ;   1
HERE2:  MOV   R3,#255       ;   1
HERE3:  DJNZ  R3,HERE3      ;   2
        DJNZ  R4,HERE2      ;   2
        DJNZ  R5,HERE1      ;   2
        RET                 ;   2
```

(6 marks/*markah*)

S10. The stack pointer contains 7H, accumulator A contains 55H and register B (F0H) contains 44H. What internal RAM locations are altered and what are the new values after executing the following instructions?
Penunjuk tindanan mengandungi nilai 7H, pengumpuk A mengandungi 55H dan daftar B (F0H) mengandungi nilai 44H. Apakah lokasi dalaman RAM yang berubah dan apakah nilai barunya setelah arahan-arahan berikut terlaksana?

```
PUSH  ACC
PUSH  FOH
```

(4 marks/*markah*)

S12. Indicate which mode and which timer are selected after the following instruction is executed.

Tentukan ragam dan pemasa yang dipilih setelah arahan berikut terlaksana.

```
MOV   TMOD,#12H
```

(4 marks/*markah*)

S13. Convert **Program 11** into machine codes (hex codes) with starting address of 10H.

*Tukarkan **Program 11** ke dalam kod mesin dengan alamat yang bermula pada 10H.*

```
      MOV   DPTR,#9000H
      MOVC  A,@A+DPTR
      CJNE  A,#10H,SINI
      INC   A
SINI: MOVX  @R1,A
```

Program 11

(6 marks/*markah*)

S14. Suppose you are required to generate a square waveform on P1.0 with a frequency of 21 kHz having a duty cycle of 10 %. What

S15. **Program 15** is a delay subroutine in assembly language. Assuming a 12 MHz oscillator is used. How long does the timer take to overflow? Show your calculation to support your answer

Program 15 *adalah suatu subrutin lengah dalam bahasa himpunan. Anggap pengayun 12 MHz digunakan. Berapa lamakah masa diambil untuk pemasa tersebut mengalami limpahan? Tunjukkan pengiraan anda untuk menyokong jawapan yang diberikan.*

(5 marks/*markah*)

```
DELAY:  MOV  TMOD,#10H
        MOV  R0,#200
ULANG:  MOV  TH1,#9EH
        MOV  TL1,#58H
        SETB TR1
LOOP:   JNB  TF1,LOOP
        CLR  TR1
        CLR  TF1
        DJNZ R0,ULANG
        RET
```

Program 15 : Delay Subroutine/ *Subrutin Lengah*

S16. What address in the interrupt vector table is assigned to INT0 and INT1? How about the pin numbers on Port3 regarding both interrupts?

Apakah alamat dalam jadual sampukan vector dikhususkan untuk INT0 dan INT1? Bagaimana pula nombor pin pada Port3 bagi bagi kedua-dua sampukan berkenaan?

(4 marks/*markah*)

S17. What register keeps track of interrupt priority in the 8051? Explain its role.

Apakah daftar yang memantau keutamaan sampukan dalam 8051? Terangkan tugasnya.

(3 marks/*markah*)

S18. Write the instructions to enable the serial interrupt, Timer 0 interrupt and external interrupt 1 (INT1).

Tuliskan arahan-arahan untuk menghidupkan sampukan sesiri, sampukan Pemasa 0 dan sampukan luaran 1 (INT1).

(3 marks/*markah*)

S19. The following instruction is executed by an 8051 micro-controller. Discuss the sequence in which the interrupts are serviced.

Arahan berikut dilaksanakan oleh mikropengawal 8051. Bincangkan aturan jujukan di mana sampukan dilayan.

MOV IP,#00001100B

(5 marks/markah)

S20. A 4-bit DIP switch and a common-anode 7-segment display are connected to 8051 as shown in **Figure 20**. Write a program that continually reads a 4-bit code from the DIP switch and updates the segments to display the appropriate hexadecimal character. For example, if the code 1010B is read, the hexadecimal character "A" should appear.

*Satu suis DIP 4 bit dan paparan 7-ruas anod sepunya disambungkan kepada 8051 seperti yang ditunjukkan pada **Rajah 20**. Tuliskan satu program yang membaca secara berterusan kod 4-bit daripada suis DIP dan memgemaskini paparan pada ruas-ruas untuk memaparkan huruf heksadesimel yang tertentu. Sebagai contoh, jika kod 1010B dibaca, huruf heksadesimel "A" akan muncul.*

(20 marks/*markah*)

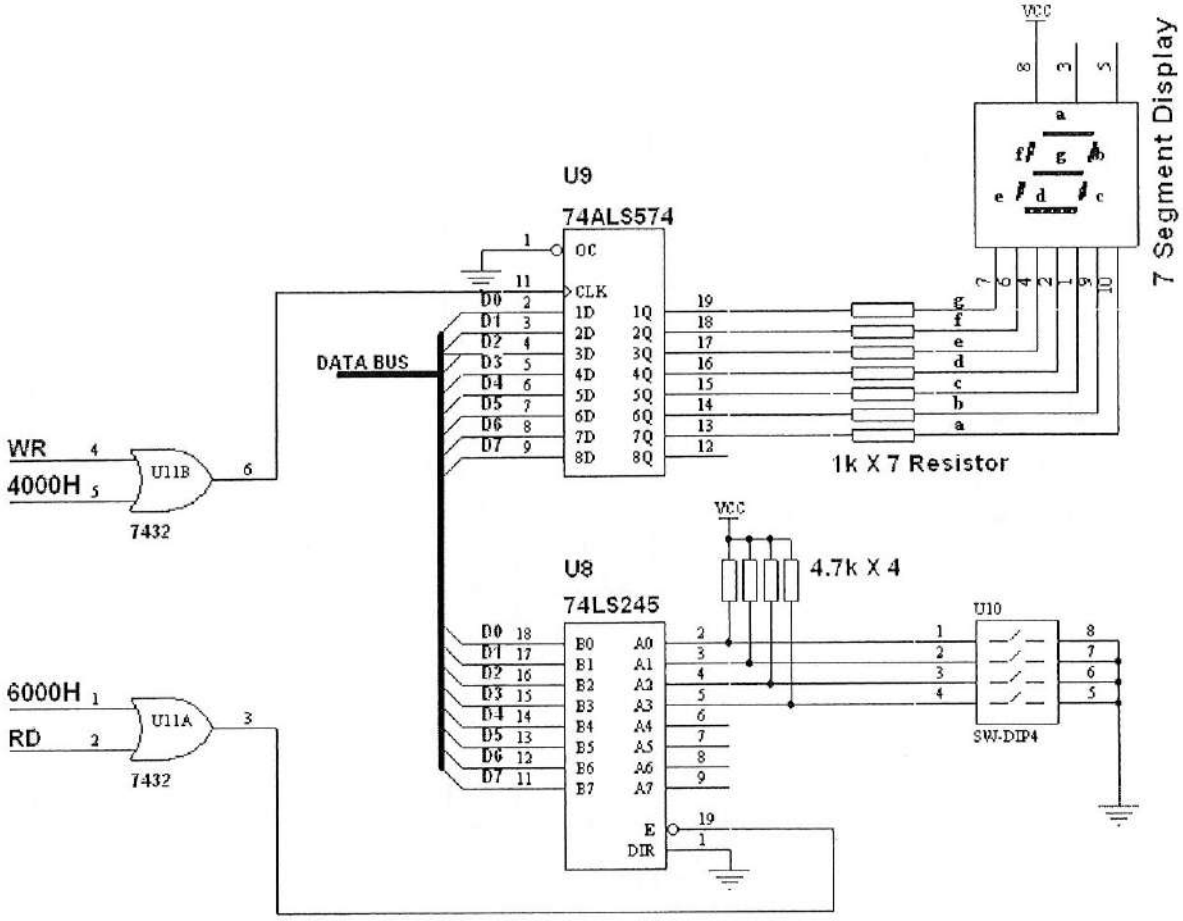

Figure 20/ ***Rajah 20***

Instruction Code Summary

H / L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	JBC bit, rel	JB bit, rel	JNB bit, rel	JC rel	JNC rel	JZ rel	JNZ rel	SJMP rel	MOV DPTR, # data 16	ORL C, /bit	ANL C, /bit	PUSH dir	POP dir	MOVX A, @DPTR	MOVX @DPTR, A
1	AJMP (P0)	ACALL (P0)	AJMP (P1)	ACALL (P1)	AJMP (P2)	ACALL (P2)	AJMP (P3)	ACALL (P3)	AJMP (P4)	ACALL (P4)	AJMP (P5)	ACALL (P5)	AJMP (P6)	ACALL (P6)	AJMP (P7)	ACALL (P7)
2	LJMP addr16	LCALL addr16	RET	RETI	ORL dir, A	ANL dir, A	XRL dir, A	ORL C, bit	ANL C, bit	MOV bit, C	MOV C, bit	CPL bit	CLR bit	SETB bit	MOVX A, @R0	MOVX @R0, A
3	RR A	RRC A	RL A	RLC A	ORL dir, # data	ANL dir, # data	XRL dir, # data	JMP @A+DPTR	MOVC A, @A+PC	MOVC A, @A+DPTR	INC DPTR	CPL C	CLR C	SETB C	MOVX A, @R1	MOVX @R1, A
4	INC A	DEC A	ADD A, # data	ADDC A, # data	ORL A, # data	ANL A, # data	XRL A, # data	MOV A, # data	DIV AB	SUBB A, # data	MUL AB	CJNE A, # data, rel	SWAP A	DA A	CLR A	CPL A
5	INC dir	DEC dir	ADD A, dir	ADDC A, dir	ORL A, dir	ANL A, dir	XRL A, dir	MOV dir, # data	MOV dir, dir	SUBB A, dir		CJNE A, dir, rel	XCH A, dir	DJNZ dir, rel	MOV A, dir	MOV dir, A
6	INC @R0	DEC @R0	ADD A, @R0	ADDC A, @R0	ORL A, @R0	ANL A, @R0	XRL A, @R0	MOV @R0, # data	MOV dir, @R0	SUBB A, @R0	MOV @R0, dir	CJNE @R0, # data, rel	XCH A, @R0	XCHD A, @R0	MOV A, @R0	MOV @ R0, A
7	INC @R1	DEC @R1	ADD A, @R1	ADDC A, @R1	ORL A, @R1	ANL A, @R1	XRL A, @R1	MOV @R1, # data	MOV dir, @R1	SUBB A, @R1	MOV @R1, dir	CJNE @R1, # data, rel	XCH A, @R1	XCHD A, @R1	MOV A, @R1	MOV @ R1, A
8	INC R0	DEC R0	ADD A, R0	ADDC A, R0	ORL A, R0	ANL A, R0	XRL A, R0	MOV R0, # data	MOV dir, R0	SUBB A, R0	MOV R0, dir	CJNE R0, # data, rel	XCH A, R0	DJNZ R0, rel	MOV A, R0	MOV R0, A
9	INC R1	DEC R1	ADD A, R1	ADDC A, R1	ORL A, R1	ANL A, R1	XRL A, R1	MOV R1, # data	MOV dir, R1	SUBB A, R1	MOV R1, dir	CJNE R1, # data, rel	XCH A, R1	DJNZ R1, rel	MOV A, R1	MOV R1, A
A	INC R2	DEC R2	ADD A, R2	ADDC A, R2	ORL A, R2	ANL A, R2	XRL A, R2	MOV R2, # data	MOV dir, R2	SUBB A, R2	MOV R2, dir	CJNE R2, # data, rel	XCH A, R2	DJNZ R2, rel	MOV A, R2	MOV R2, A
B	INC R3	DEC R3	ADD A, R3	ADDC A, R3	ORL A, R3	ANL A, R3	XRL A, R3	MOV R3, # data	MOV dir, R3	SUBB A, R3	MOV R3, dir	CJNE R3, # data, rel	XCH A, R3	DJNZ R3, rel	MOV A, R3	MOV R3, A
C	INC R4	DEC R4	ADD A, R4	ADDC A, R4	ORL A, R4	ANL A, R4	XRL A, R4	MOV R4, # data	MOV dir, R4	SUBB A, R4	MOV R4, dir	CJNE R4, # data, rel	XCH A, R4	DJNZ R4, rel	MOV A, R4	MOV R4, A
D	INC R5	DEC R5	ADD A, R5	ADDC A, R5	ORL A, R5	ANL A, R5	XRL A, R5	MOV R5, # data	MOV dir, R5	SUBB A, R5	MOV R5, dir	CJNE R5, # data, rel	XCH A, R5	DJNZ R5, rel	MOV A, R5	MOV R5, A
E	INC R6	DEC R6	ADD A, R6	ADDC A, R6	ORL A, R6	ANL A, R6	XRL A, R6	MOV R6, # data	MOV dir, R6	SUBB A, R6	MOV R6, dir	CJNE R6, # data, rel	XCH A, R6	DJNZ R6, rel	MOV A, R6	MOV R6, A
F	INC R7	DEC R7	ADD A, R7	ADDC A, R7	ORL A, R7	ANL A, R7	XRL A, R7	MOV R7, # data	MOV dir, R7	SUBB A, R7	MOV R7, dir	CJNE R7, # data, rel	XCH A, R7	DJNZ R7, rel	MOV A, R7	MOV R7, A

2Byte	3Byte
2Cycle	4Cycle

The Program Status Word (PSW)

Bit	Symbol	Address	Description
PSW.7	CY	D7H	Carry flag
PSW.6	AC	D6H	Auxiliary carry flag
PSW.5	F0	D5H	Flag 0
PSW.4	RS1	D4H	Register bank select 1
PSW.3	RS0	D3H	Register bank select 0
PSW.2	OV	D2H	Overflow flag
PSW.1	--	D1H	Reserved
PSW.0	P	D0H	Parity Flag.

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

Timer Mode (TMOD) register summary

Bit	Name	Timer	Description
7	GATE	1	When this bit is set the timer will only run when INT1(P3.3) is high(hardware control). When this bit is cleared the timer will run regardless of the state of INT1(software control.
6	C/T	1	Counter/timer select bit. 1 = counter operation 0 = timer operation
5	M1	1	Mode bit 1
4	M0	1	Mode bit 0
3	GATE	0	Timer 0 gate bit
2	C/T	0	Timer 0 counter/timer select bit
1	M1	0	Timer 0 M1 bit
0	M0	0	Timer 0 M0 bit

M1	M0	Mode	Description
0	0	0	13-bit timer mode (8048 mode)
0	1	1	16-bit timer mode
1	0	2	8-bit auto-reload mode
1	1	3	Split timer mode Timer 0: TL0 is an 8-bit timer controlled by timer 0 mode bits; TH0, the same except controlled by timer 1 mode bits Timer1: stopped

Timer Control (TCON) register summary

Bit	SYMBOL	BIT ADDRESS	DESCRIPTION
TCON.7	TF1	8FH	Timer 1 overflow flag. Set by hardware upon overflow; cleared by software, or by hardware when processor vectors to interrupt service routine
TCON.6	TR1	8EH	Timer 1 run-control bit. Set/cleared by software to turn timer on/off
TCON.5	TF0	8DH	Timer 0 overflow bit. Do the same function as TF1 but for Timer 0
TCON.4	TR0	8CH	Timer 0 run-control bit. Do the same function as TR1 but for Timer 0
TCON.3	IE1	8BH	External interrupt 1 edge flag. Set by hardware when a falling edge is detected on INT1;cleared by software, or by hardware when CPU vectors to interrupt service routine
TCON.2	IT1	8AH	External interrupt 1 type flag. Set/cleared by software for falling edge/low-level activated external interrupt.
TCON.1	IE0	89H	External interrupt 0 edge flag. Do the same function as IE1 but for external interrupt-0.
TCON.0	IT0	88H	External interrupt 0 type flag. Do the same function as IT1 but for external interrupt-0.

Table 6-1 Interrupt Enable (IE) register summary

Bit	Symbol	Bit Address	Description (1 = enable, 0 = disable)
IE.7	EA	AFH	Global enable/disable. EA = 1, each individual source is enabled/disabled by setting/clearing its enable bit. EA= 0, disable all interrupts.
IE.6	-	AEH	Undefined
IE.5	ET2	ADH	Enable Timer 2 interrupt(8052)
IE.4	ES	ACH	Enable serial port interrupt
IE.3	ET1	ABH	Enable Timer 1 interrupt
IE.2	EX1	AAH	Enable external 1 interrupt
IE.1	ET0	A9H	Enable Timer 0 interrupt
IE.0	EX0	A8H	Enable external 0 interrupt

Interrupt Priority (IP) Register

Bit	Symbol	Bit Address	Description (1 = Higher level, 0 = lower level)
IP.7	-	-	Undefined
IP.6	-	-	Undefined
IP.5	PT2	BDH	Priority for Timer 2 interrupt(8052)
IP.4	PS	BCH	Priority for serial port interrupt
IP.3	PT1	BBH	Priority for Timer 1 interrupt
IP.2	PX1	BAH	Priority for external 1 interrupt
IP.1	PT0	B9H	Priority for Timer 0 interrupt
IP.0	PX0	B8H	Priority for external 0 interrupt

Interrupt Vectors

Interrupt	Flag	Bit Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI or TI	0023H
Timer 2	TF2 or EXF2	002BH

Register values after system reset(power-up)

REGISTER(S)	CONTENTS
Program Counter	0000H
Accumulator	00H
B register	00H
PSW	00H
SP	07H
DPTR	0000H
Ports 0 – 3	FFH
IP	XXX00000B
Timer Register	00H
SCON	00H
SBUF	00H
PCON(HMOS)	0XXXXXXXB
PCON(CMOS)	0XXX0000B