UNIVERSITI MALAYSIA PERLIS

Peperiksaan Akhir Semester Pertama
Sidang Akademik 2020/2021

Disember 2020

DKT218 – Microcontroller
[Mikropengawal]

Masa: 3 jam

Please make sure that this question paper has **FOURTEEN (14)** printed pages including this cover before you begin this exam.
(*Sila pastikan kertas soalan ini mengandungi **EMPAT BELAS (14)** mukasurat bercetak termasuk muka hadapan sebelum anda memulakan peperiksaan ini.*).

This question paper contains **TWO (2)** parts:
*[Kertas soalan ini mengandungi **DUA (2)** bahagian:]*

PART A : This part has **FOUR (4)** questions. Answer **ALL** questions (80 marks).
*[**BAHAGIAN A** : Bahagian ini ada **EMPAT (4)** soalan. Jawab **SEMUA** soalan (80 markah).]*

PART B : This part has **TWO (2)** questions. Answer any **ONE (1)** question (20 marks).
*[**BAHAGIAN B** : Bahagian ini ada **DUA (2)** soalan. Jawab mana-mana **SATU (1)** soalan (20 markah).]*

Reference for 8051 machine codes and Special Function Registers are given in Appendices.
[Rujukan kod-kod mesin dan Pendaftar-pendaftar Fungsi Khas 8051 disediakan dalam helaian Lampiran.]

Dicetak oleh Unit Peperiksaan & Pengijazahan, Bahagian Pengurusan Akademik, Pusat Pengurusan Akademik, UniMAP

PART A
[Bahagian A]

Answer ALL FOUR (4) questions
[Jawab SEMUA EMPAT (4) soalan]

Question 1
[Soalan 1]

(a) A microcontroller is a device similar to a microprocessor, but has the ability to perform certain specific task in a much better way in terms of speed, performance and cost.
[Sebuah mikropngawal adalah peranti yang menyerupai sebuah mikropemproses, tetapi mempunyai keupayaan untuk melaksanakan tugas khas yang lebih baik dari segi kepantasan, prestasi dan kos.]

i) Explain **THREE (3)** distinct features that differentiates a microcontroller from a microprocessor.
*[Jelaskan **TIGA (3)** perbezaan-perbezaan ketara di antara sebuah mikropemproses dan sebuah mikropengawal.]*

(3 Marks/ *Markah*)

ii) Draw the block diagram of a microcontroller and a microprocessor.
[Lukiskan gambarajah blok sebuah mikropengawal dan sebuah mikropemproses.]

(2 Marks/ *Markah*)

(b) The Intel 8051 microcontroller has a certain number of instruction types that a developer can use to create their assembly code programs. These instructions are categorized to 5 addressing modes and 5 operation types.
[Mikropengawal Intel 8051 mempunyai sejumlah arahan-arahan bagi seseorang peneroka guna untuk membina aturcara-aturcara kod himpunan. Arahan-arahan ini dikategorikan kepada 5 mod-mod pengalamatan dan 5 jenis operasi.]

i) Explain any **THREE (3)** addressing modes and provide **TWO (2)** examples for each mode.
*[Jelaskan mana-mana **TIGA (3)** mod-mod pengalamatan dan berikan **DUA (2)** contoh bagi setiap mod.]*

(3 Marks/ *Markah*)

ii) Explain any **THREE (3)** operation types and provide **TWO (2)** examples for each operation.
*[Jelaskan mana-mana **TIGA (3)** mod-mod pengalamatan dan berikan **DUA (2)** contoh bagi setiap mod.]*

(3 Marks/ *Markah*)

(c) Assuming the initial value inside the accumulator of an 8051 microcontroller is 8AH, and the initial value inside the program status word is 88H is used in every operation below, determine the program status word's hexadecimal value after executing each operation.
[Dengan menganggapkan nilai mula "accumulator" sesebuah mikropengawal 8051 adalah 8AH, dan nilai mula "program status word" adalah 88H digunakan dalam setiap operasi dibawah, tentukan nilai "hexadecimal program status word" selepas melaksanakan setiap operasi.]

i) CJNE A, #9BH, LOOP3
[CJNE A,#9BH,LOOP3.]

ii) CPL A
[CPL A]

iii) ADDC A, #34H
[ADDC A,#34H]

(3 Marks/ *Markah*)

(d) An 8051 microcontroller is to perform the following arithmetic routine below. Draw the appropriate flowchart to represent the given task.
[Sebuah 8051 mikropengawal perlu melaksanakan rutin aritmetik seperti dibawah. Lukiskan carta alir yang bersesuaian bagi menggambarkan tugas yang diberi.

i) *Add 5 for thirteen(13) times*
[Tambah 5 sebanyak tiga belas(13) kali]

(6 Marks/ *Markah*)

....4/

Question 2
[Soalan 2]

(a) Given an 8051 stack operation as in **Figure 1.**
*[Diberi suatu operasi tindanan 8051 seperti dalam **Rajah 1**.]*

```
        ORG   00H
        MOV   SP,#2FH
        MOV   PSW,#16
        MOV   R0,#0A9H
        MOV   R1,#58
        MOV   R2,#242
        MOV   R3,#140
        PUSH  13H
        PUSH  10H
        PUSH  11H
        PUSH  12H
        POP   10H
        POP   13H
        POP   11H
        END
```

Figure 1 : 8051 Stack Operation
***[Rajah 1** : Operasi Tindanan 8051]*

After running this operation, show the resulting contents of registers R0, R1, R2, R3 and the stack pointer.
[Selepas operasi ini dilaksanakan, tunjukkan hasil kandungan bagi daftar-daftar R0, R1, R2, R3 dan penunjuk alamat tindan]

(5 Marks/ *Markah*)

(b) Draw a flowchart to represent the following partial assembly language program as given in **Figure 2.**
*[Lukis carta alir untuk mewakili petikan aturcara yang berikut seperti dalam **Rajah 2**.]*

(5 Marks/ *Markah*)

```
        ORG   00H
          :
          :
        SETB  P0.0
AGAIN:  MOV   A,P0
        ANL   A,#1
        JNZ   AGAIN
          :
          :
        END
```

Figure 2 : Partial program
***[Rajah 2** : Petikan aturcara]*

(c) A certain hex code program was retrieved from an 8051 microcontroller as shown in **Table 1**. Convert this machine language into assembly language to determine the program installed.
*[Satu aturcara berbentuk kod hex telah diperolehi daripada suatu mikropengawal 8051 sepertimana ditunjukkan dalam **Jadual 1**. Terjemahkan bahasa mesin tersebut kepada bahasa perhimpunan untuk mengetahui jenis program yang telah digunakan.]*

(10 Marks/ *Markah*)

Table 1
[Jadual 1]

Address	Code	Address	Code
0030	7B	0037	A1
0031	00	0038	50
0032	74	0039	01
0033	13	003A	0B
0034	78	003B	D8
0035	25	003C	F9
0036	24	003D	F9

Question 3
[Soalan 3]

(a) Based on the 8051 assembly language program shown in **Table 2**, complete the address and hex code columns.
*[Dengan merujuk kepada aturcara bahasa himpunan 8051 yang ditunjukkan dalam **Jadual 2**, lengkapkan lajur-lajur alamat dan kod hex.]*

(10 Marks/ *Markah*)

Table 2
[Jadual 2]

Assembly Program			Address	Hex Codes		
				Byte 1	Byte 2	Byte 3
	ORG	0050H				
ST:	MOV	P1,#0FFH				
L1:	MOV	A,P1				
	CJNE	A,#63H,L1				
	MOV	R2,#13H				
L2:	MOV	P2,A				
	CPL	A				
	DJNZ	R2,L2				
L3:	SJMP	L3				
	END					

....7/

(b) An 8051 microcontroller subroutine is shown below.
[Satu subrutin 8051 mikropengawal ditunjukkan seperti di bawah.]

```
DELAY:  MOV R0, #14
LOOP3:  MOV R1, #76
LOOP2:  MOV R2, # 255
LOOP1:  DJNZ  R2, LOOP1
        DJNZ  R1, LOOP2
        DJNZ  R0, LOOP3
        RET
```

i) State the machine cycle for each instruction and calculate the total machine cycles that have been used to complete this subroutine.
[Nyatakan kitaran mesin bagi setiap arahan dan kirakan jumlah kitaran mesin yang telah digunakan untuk melengkapkan subrutin ini]

(8 Marks/ *Markah*)

ii) If the 11.0592 MHz crystal oscillator clock source is supplied to this 8051 microcontroller, calculate the total time delay to execute this subroutine. Show the calculation method in detail.
[Sekiranya sumber detik berkelajuan 11.0592 MHz dibekalkan ke mikropengawal 8051 ini, kirakan hasil jumlah masa untuk melaksanakan subrutin ini. Tunjukkan kaedah pengiraan secara terperinci.]

(2 Marks/ *Markah*)

....8/

Question 4
[Soalan 4]

(a) Write an assembly language program to generate a square wave of 1050 µs period on pin P2.4 by using interrupt for Timer 0. Assume that XTAL = 11.0592 MHz. **Figure 3** illustrates this operation.
*[Tuliskan satu program bahasa himpunan untuk menjana gelombang segiempat sama 1050 µs tempoh pada pin P2.4 dengan menggunakan sampukan untuk Pemasa 0. Andaikan XTAL=11.0592 MHz. **Rajah 3** menggambarkan operasi ini.]*

(10 Marks/ *Markah*)

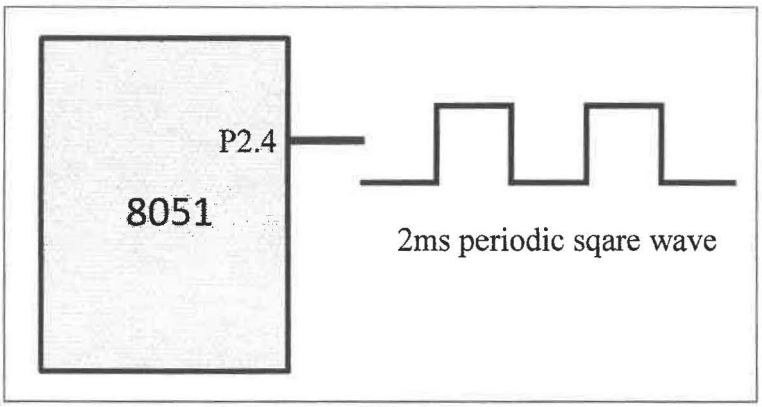

Figure 3 : Operation block diagram
***[Rajah 3** : Gambarajah blok operasi]*

(b) A push button is connected to pin P0.1. Write an assembly language program to monitor its status and transmits the following message "**Hello**" continuously using the 8051 microcontroller serial port when the button is pressed. This message will transmit at 9600 baud rate, 8-bit data, and 1 stop bit. **Figure 4** illustrates this operation.
*[Suis tekan disambungkan kepada pin P0.1. Tuliskan program bahasa himpunan untuk memantau status suis dan menghantar mesej "**Hello**" secara berterusan melalui port siri mikropengawal 8051. Mesej ini dihantar pada 9600 kadar baud,data 8-bit dan 1 bit berhenti. **Rajah 4** menggambarkan operasi ini.]*

(10 Marks/ *Markah*)

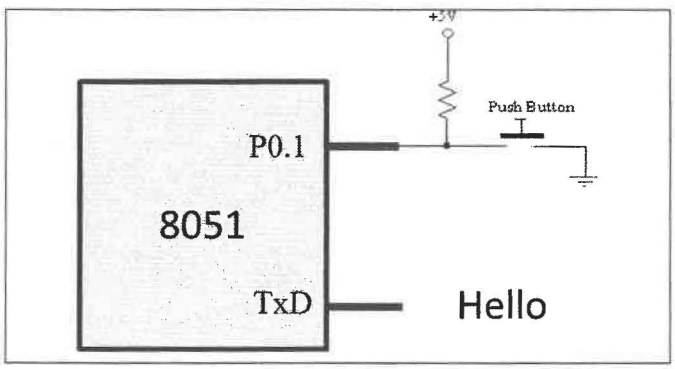

Figure 4 : Operation block diagram
***[Rajah 4** : Gambarajah blok operasi]*

....9/

PART B
[Bahagian B]

Answer ONE (1) question only
[Jawab SATU (1) soalan sahaja]

Question 5
[Soalan 5]

A binary coded decimal (BCD) counter is to be designed that will count from 0 to 15 using two common cathode 7 segment displays connected through a 74LS373 octal latch to P1 of an 8051 microcontroller. The grounding of the two 7 segment display A and B will be connected to P2.0 and P2.1 respectively. **Table 3** lists the 7 seven segment data for each digit between 0 to 9 to be used for display.
[Satu pembilang perpuluhan berkod binary (BCD) perlu dibina untuk membilang dari 0 ke 15 menggunakan dua paparan 7 segmen katod seragam yang disambung melalui sebuah selak 8-bit 74LS373 ke P1 mikropengawal 8051. Sambungan bumi paparan 7 segmen A dan B masing-masing ke P2.0 dan P2.1. ***Jadual 3*** *menunjukkan data 7 segmen bagi setiap digit untuk paparan setiap digit di antara 0 hingga 9 yang hendak di paparkan]*

Table 3
[Jadual 3]

Bit / Digit	7	6	5	4	3	2	1	0
	a	b	c	d	e	f	g	dp
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
3	1	1	1	1	0	0	1	0
4	0	1	1	0	0	1	1	0
5	1	0	1	1	0	1	1	0
6	1	0	1	1	1	1	1	0
7	1	1	1	0	0	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	1	1	0	1	1	0

(a) If the microcontroller is supplied with an 11.0592 MHz clock source, show the delay subroutine program code for a 1 second time delay.
[Jika mikropengawal dibekalkan dengan sumber detik 11.0592 MHz, tunjukkan kod program subrutin lengah untuk melengahkan masa 1 saat.]

(8 Marks/ *Markah*)

(b) Using the subroutine delay program built in **5(b),** create the assembly language program that will display the counting sequence as mentioned above.
[Dengan menggunakan program subrutin lengah yang dibina dalam ***5(b)****, bina sebuah program bahasa himpunan yang akan memapar siri bilangan seperti yang telah disebut di atas.]*

(12 Marks/ *Markah*)

....10/

Question 6
[Soalan 6]

An 8051 microcontroller system is needed to store the resulting binary coded decimal (BCD) number received from a 4x3 matrix keypad. **Figure 5** shows the connection and direction flow for this 4x3 matrix keypad.
[Sebuah sistem mikropengawal diperlukan untuk menyimpan hasil nombor perpuluhan berkod binari (BCD) yang diterima daripada sebuah papan kunci 4x3. ***Rajah 5*** *menunjukkan sambungan dan arah alir untuk papan kunci 4x3 ini.]*

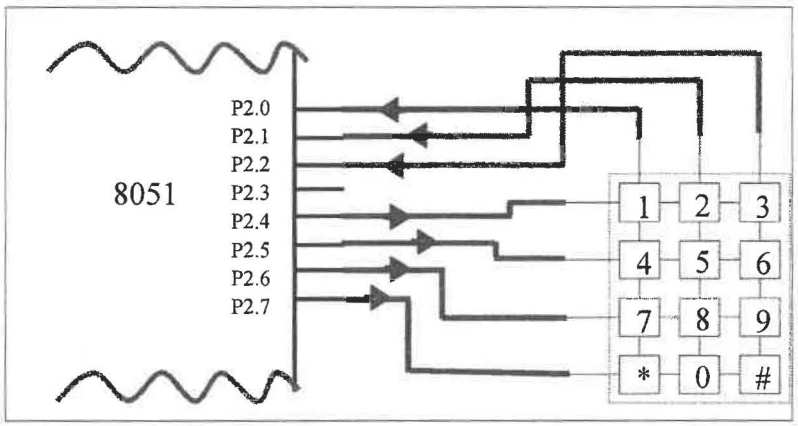

Figure 5 : 4x3 Keypad connection
***[Rajah 5** : Sambungan papan kunci 4x3]*

(a) Create the program to initialize the system and followed by the code to detect whether a button has been pressed or not.
[Bina satu program untuk membuat penetapan awalan sistem ini dan kemudian diikuti dengan kod untuk mengesan samada satu butang telah ditekan ataupun tidak.]

(8 Marks/ *Markah*)

(b) Show the program to store the BCD number associated with the pressed button into the multipurpose register R6.
[Tunjukkan program untuk menyimpan nombor BCD yang berkaitan dengan butang yang ditekan ke dalam daftar serbaguna R6]

(12 Marks/ *Markah*)

-oooOooo-

Appendix 1
[Lampiran1]

8051 HEX CODE

HEX Code	Mnemonic	Operand	Byte	Cycle	C	OV	AC
00	NOP		1	1			
01	AJMP	addr11	2	2			
02	LJMP	addr16	3	2			
03	RR	A	1	1			
04	INC	A	1	1			
05	INC	direct	2	1			
06	INC	@R0	1	1			
07	INC	@R1	1	1			
08	INC	R0	1	1			
09	INC	R1	1	1			
0A	INC	R2	1	1			
0B	INC	R3	1	1			
0C	INC	R4	1	1			
0D	INC	R5	1	1			
0E	INC	R6	1	1			
0F	INC	R7	1	1			
10	JBC	bit, offset	3	2			
11	ACALL	addr11	2	2			
12	LCALL	addr16	3	2			
13	RRC	A	1	1	x		
14	DEC	A	1	1			
15	DEC	direct	2	1			
16	DEC	@R0	1	1			
17	DEC	@R1	1	1			
18	DEC	R0	1	1			
19	DEC	R1	1	1			
1A	DEC	R2	1	1			
1B	DEC	R3	1	1			
1C	DEC	R4	1	1			
1D	DEC	R5	1	1			
1E	DEC	R6	1	1			
1F	DEC	R7	1	1			
20	JB	bit, offset	3	2			
21	AJMP	addr11	2	2			
22	RET		1	2			
23	RL	A	1	1			
24	ADD	A, #immed	2	1	x	x	x
25	ADD	A, direct	2	1	x	x	x
26	ADD	A, @R0	1	1	x	x	x
27	ADD	A, @R1	1	1	x	x	x
28	ADD	A, R0	1	1	x	x	x
29	ADD	A, R1	1	1	x	x	x
2A	ADD	A, R2	1	1	x	x	x
2B	ADD	A, R3	1	1	x	x	x
2C	ADD	A, R4	1	1	x	x	x
2D	ADD	A, R5	1	1	x	x	x
2E	ADD	A, R6	1	1	x	x	x
2F	ADD	A, R7	1	1	x	x	x
30	JNB	bit, offset	3	2			
31	ACALL	addr11	2	2			
32	RETI		1	2			
33	RLC	A	1	1	x		
34	ADDC	A, #immed	2	1	x	x	x
35	ADDC	A, direct	2	1	x	x	x
36	ADDC	A, @R0	1	1	x	x	x
37	ADDC	A, @R1	1	1	x	x	x
38	ADDC	A, R0	1	1	x	x	x
39	ADDC	A, R1	1	1	x	x	x
3A	ADDC	A, R2	1	1	x	x	x
3B	ADDC	A, R3	1	1	x	x	x
3C	ADDC	A, R4	1	1	x	x	x
3D	ADDC	A, R5	1	1	x	x	x
3E	ADDC	A, R6	1	1	x	x	x
3F	ADDC	A, R7	1	1	x	x	x

HEX Code	Mnemonic	Operand	Byte	Cycle	C	OV	AC
40	JC	offset	2	2			
41	AJMP	addr11	2	2			
42	ORL	direct, A	2	1			
43	ORL	direct, #immed	3	2			
44	ORL	A, #immed	2	1			
45	ORL	A, direct	2	1			
46	ORL	A, @R0	1	1			
47	ORL	A, @R1	1	1			
48	ORL	A, R0	1	1			
49	ORL	A, R1	1	1			
4A	ORL	A, R2	1	1			
4B	ORL	A, R3	1	1			
4C	ORL	A, R4	1	1			
4D	ORL	A, R5	1	1			
4E	ORL	A, R6	1	1			
4F	ORL	A, R7	1	1			
50	JNC	offset	2	2			
51	ACALL	addr11	2	2			
52	ANL	direct, A	2	1			
53	ANL	direct, #immed	3	2			
54	ANL	A, #immed	2	1			
55	ANL	A, direct	2	1			
56	ANL	A, @R0	1	1			
57	ANL	A, @R1	1	1			
58	ANL	A, R0	1	1			
59	ANL	A, R1	1	1			
5A	ANL	A, R2	1	1			
5B	ANL	A, R3	1	1			
5C	ANL	A, R4	1	1			
5D	ANL	A, R5	1	1			
5E	ANL	A, R6	1	1			
5F	ANL	A, R7	1	1			
60	JZ	offset	2	2			
61	AJMP	addr11	2	2			
62	XRL	direct, A	2	1			
63	XRL	direct, #immed	3	2			
64	XRL	A, #immed	2	1			
65	XRL	A, direct	2	1			
66	XRL	A, @R0	1	1			
67	XRL	A, @R1	1	1			
68	XRL	A, R0	1	1			
69	XRL	A, R1	1	1			
6A	XRL	A, R2	1	1			
6B	XRL	A, R3	1	1			
6C	XRL	A, R4	1	1			
6D	XRL	A, R5	1	1			
6E	XRL	A, R6	1	1			
6F	XRL	A, R7	1	1			
70	JNZ	offset	2	2			
71	ACALL	addr11	2	2			
72	ORL	C, bit	2	2	x		
73	JMP	@A+DPTR	1	2			
74	MOV	A, #immed	2	1			
75	MOV	direct, #immed	3	2			
76	MOV	@R0, #immed	2	1			
77	MOV	@R1, #immed	2	1			
78	MOV	R0, #immed	2	1			
79	MOV	R1, #immed	2	1			
7A	MOV	R2, #immed	2	1			
7B	MOV	R3, #immed	2	1			
7C	MOV	R4, #immed	2	1			
7D	MOV	R5, #immed	2	1			
7E	MOV	R6, #immed	2	1			
7F	MOV	R7, #immed	2	1			

HEX Code	Mnemonic	Operand	Byte	Cycle	C	OV	AC
80	SJMP	offset	2	2			
81	AJMP	addr11	2	2			
82	ANL	C, bit	2	2	x		
83	MOVC	A, @A+PC	1	2			
84	DIV	AB	1	4	0	x	
85	MOV	direct, direct	3	2			
86	MOV	direct, @R0	2	2			
87	MOV	direct, @R1	2	2			
88	MOV	direct, R0	2	2			
89	MOV	direct, R1	2	2			
8A	MOV	direct, R2	2	2			
8B	MOV	direct, R3	2	2			
8C	MOV	direct, R4	2	2			
8D	MOV	direct, R5	2	2			
8E	MOV	direct, R6	2	2			
8F	MOV	direct, R7	2	2			
90	MOV	DPTR, #immed	3	2			
91	ACALL	addr11	2	2			
92	MOV	bit, C	2	2			
93	MOVC	A, @A+DPTR	1	2			
94	SUBB	A, #immed	2	1	x	x	x
95	SUBB	A, direct	2	1	x	x	x
96	SUBB	A, @R0	1	1	x	x	x
97	SUBB	A, @R1	1	1	x	x	x
98	SUBB	A, R0	1	1	x	x	x
99	SUBB	A, R1	1	1	x	x	x
9A	SUBB	A, R2	1	1	x	x	x
9B	SUBB	A, R3	1	1	x	x	x
9C	SUBB	A, R4	1	1	x	x	x
9D	SUBB	A, R5	1	1	x	x	x
9E	SUBB	A, R6	1	1	x	x	x
9F	SUBB	A, R7	1	1	x	x	x
A0	ORL	C, /bit	2	2	x		
A1	AJMP	addr11	2	2			
A2	MOV	C, bit	2	1	X		
A3	INC	DPTR	1	2			
A4	MUL	AB	1	4	0	x	
A5	RESERVED						
A6	MOV	@R0, direct	2	2			
A7	MOV	@R1, direct	2	2			
A8	MOV	R0, direct	2	2			
A9	MOV	R1, direct	2	2			
AA	MOV	R2, direct	2	2			
AB	MOV	R3, direct	2	2			
AC	MOV	R4, direct	2	2			
AD	MOV	R5, direct	2	2			
AE	MOV	R6, direct	2	2			
AF	MOV	R7, direct	2	2			
B0	ANL	C, /bit	2	2	X		
B1	ACALL	addr11	2	2			
B2	CPL	bit	2	1			
B3	CPL	C	1	1	X		
B4	CJNE	A, #immed, offset	3	2	x		
B5	CJNE	A, direct, offset	3	2	x		
B6	CJNE	@R0, #immed, offset	3	2	x		
B7	CJNE	@R1, #immed, offset	3	2	x		
B8	CJNE	R0, #immed, offset	3	2	x		
B9	CJNE	R1, #immed, offset	3	2	x		
BA	CJNE	R2, #immed, offset	3	2	x		
BB	CJNE	R3, #immed, offset	3	2	x		
BC	CJNE	R4, #immed, offset	3	2	x		
BD	CJNE	R5, #immed, offset	3	2	x		
BE	CJNE	R6, #immed, offset	3	2	x		
BF	CJNE	R7, #immed, offset	3	2			

HEX Code	Mnemonic	Operand	Byte	Cycle	C	OV	AC
C0	PUSH	direct	2	2			
C1	AJMP	addr11	2	2			
C2	CLR	bit	2	1			
C3	CLR	C	1	1	0		
C4	SWAP	A	1	1			
C5	XCH	A, direct	2	1			
C6	XCH	A, @R0	1	1			
C7	XCH	A, @R1	1	1			
C8	XCH	A, R0	1	1			
C9	XCH	A, R1	1	1			
CA	XCH	A, R2	1	1			
CB	XCH	A, R3	1	1			
CC	XCH	A, R4	1	1			
CD	XCH	A, R5	1	1			
CE	XCH	A, R6	1	1			
CF	XCH	A, R7	1	1			
D0	POP	direct	2	2			
D1	ACALL	addr11	2	2			
D2	SETB	bit	2	1			
D3	SETB	C	1	1	1		
D4	DA	A	1	1	x		
D5	DJNZ	direct, offset	3	2			
D6	XCHD	A, @R0	1	1			
D7	XCHD	A, @R1	1	1			
D8	DJNZ	R0, offset	2	2			
D9	DJNZ	R1, offset	2	2			
DA	DJNZ	R2, offset	2	2			
DB	DJNZ	R3, offset	2	2			
DC	DJNZ	R4, offset	2	2			
DD	DJNZ	R5, offset	2	2			
DE	DJNZ	R6, offset	2	2			
DF	DJNZ	R7, offset	2	2			
E0	MOVX	A, @DPTR	1	2			
E1	AJMP	addr11	2	2			
E2	MOVX	A, @R0	1	2			
E3	MOVX	A, @R1	1	2			
E4	CLR	A	1	1			
E5	MOV	A, direct	2	1			
E6	MOV	A, @R0	1	1			
E7	MOV	A, @R1	1	1			
E8	MOV	A, R0	1	1			
E9	MOV	A, R1	1	1			
EA	MOV	A, R2	1	1			
EB	MOV	A, R3	1	1			
EC	MOV	A, R4	1	1			
ED	MOV	A, R5	1	1			
EE	MOV	A, R6	1	1			
EF	MOV	A, R7	1	1			
F0	MOVX	@DPTR, A	1	2			
F1	ACALL	addr11	2	2			
F2	MOVX	@R0, A	1	2			
F3	MOVX	@R1, A	1	2			
F4	CPL	A	1	1			
F5	MOV	direct, A	2	1			
F6	MOV	@R0, A	1	1			
F7	MOV	@R1, A	1	1			
F8	MOV	R0, A	1	1			
F9	MOV	R1, A	1	1			
FA	MOV	R2, A	1	1			
FB	MOV	R3, A	1	1			
FC	MOV	R4, A	1	1			
FD	MOV	R5, A	1	1			
FE	MOV	R6, A	1	1			
FF	MOV	R7, A	1	1			

APPENDIX 2
[LAMPIRAN 2]

8051 Special Function Registers

Reg Type									Byte address
SCON	9F	9E	8D	9C	98	9A	99	98	98
P1	97	96	95	94	93	92	91	90	90
TH1	not bit addressable								8D
TH0	not bit addressable								8C
TL1	not bit addressable								8B
TL0	not bit addressable								8A
TMOD	not bit addressable								89
TCON	8F	8E	8D	8C	8B	8A	89	88	88
PCON	not bit addressable								87
DPH	not bit addressable								83
DPL	not bit addressable								82
SP	not bit addressable								81
P0	87	86	85	84	83	82	81	80	80

Reg Type									Byte address
									FF
B	F7	F6	F5	F4	F3	F2	F1	F0	F0
ACC	E7	E6	E5	E4	E3	E2	E1	E0	E0
PSW	D7	D6	D5	D4	D3	D2	--	D0	D0
IP	--	--	--	BC	BB	BA	B9	B8	B8
P3	B7	B6	B5	B4	B3	B2	B1	B0	B0
IE	AF	--	--	AC	AB	AA	A9	A8	A8
P2	A7	A6	A5	A4	A3	A2	A1	A0	A0
SBUF	not bit addressable								99

SPECIAL FUNCTION REGISTER - PSW

BYTE	BIT							
D0	D7	D6	D5	D4	D3	D2	D1	D0
	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
	CY	AC	F0	RS1	RS0	OV	--	P

SPECIAL FUNCTION REGISTER – TIMER

TCON REGISTER

BYTE	BIT							
88	8F	8E	8D	8C	8B	8A	89	88
	TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IT_X – external interrupt type ; $edge/\overline{level}$ triggered
IE_X – external flag; HIGH everytime detects active low
TR_X – timer run ; $start/\overline{stop}$ bit
TF_X – timer flag; HIGH everytime rollover to 00H

TMOD REGISTER

BYTE	BIT							
89	--	--	--	--	--	--	--	--
	TMOD							
	G	$C/\overline{T}$	M1	M0	G	$C/\overline{T}$	M1	M0

G – gate control; '1' – timer runs when IE_X='1' and TR_X='1'
'0' – timer runs when TR_X='1' only
$C/\overline{T}$ – whether to use $\overline{internal}/external$ clock soure
M1 M0 – modes "00"-13 bit, "01"-16 bit, "10"-8 bit, "11"-split timer

Dicetak oleh Unit Peperiksaan & Pengijazahan, Bahagian Pengurusan Akademik, Pusat Pengurusan Akademik, UniMAP

SPECIAL FUNCTION REGISTER – INTERRUPT

IE REGISTER

BYTE	BIT							
A8	AF	AE	AD	AC	AB	AA	A9	A8
	IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
	EA	--	--	ES	ET1	EX1	ET0	EX0

Interrupt Vector Table

Type	Reset	Ex0	T0	Ex1	T1	S
Address	0000	0003	000B	0013	001B	0023

IP REGISTER

BYTE	BIT							
B8	BF	BE	BD	BC	BB	BA	B9	B8
	IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
	--	--	--	PS	PT1	PX1	PT0	PX0

SPECIAL FUNCTION REGISTER – SERIAL COMMUNICATION

SCON REGISTER

BYTE	BIT							
98	9F	9E	9D	9C	9B	9A	99	98
	SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

RI – Sets when a byte has been received in SBUFF
TI – Sets when a byte has been transmitted
RB8 – Receive 9th bit for mode 2 and 3.
TB8 – Transmit 9th bit for mode 2 and 3
REN – Receiver enable; HIGH to receive data on RxD pin
SM2 – Enables multiprocessor comm in mode 2 and 3
SM0 SM1 – "00" – Shift Reg baud rate OSC/12
"01" – 8-bit UART variable baud rate
"10" – 9-bit UART baud rate OSC/32 or OSC/64
"11" – 9-bit UART variable baud rate

PCON REGISTER

BYTE	BIT							
87	--	--	--	--	--	--	--	--
	PCON							
	SMOD	--	--	--	GF1	GF0	PD	IDL

SMOD – '0' -f/64 , '1' –f/32