

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 2000/2001

Februari/Mac 2001

ZAT 281/4 – Pengantar Mikropemproses

Masa : 3 jam

Sila pastikan bahawa kertas peperiksaan ini mengandungi **DUAPULUH SATU** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.

Jawab kesemua LIMA soalan. Semua soalan mesti dijawab di ruang yang disediakan di dalam kertas soalan ini. Ringkasan set arahan mikropemproses 68000, jadual kebenaran bagi IC 74LS138 dan 74LS139, serta keadaan pemapar tujuh ruas disediakan di Lampiran A.

1) Pilih satu sahaja jawapan yang paling sesuai dan tuliskan pilihan anda di ruang jawapan yang disediakan untuk setiap persoalan berikut:-

I) Mikropemproses 68000 mempunyai 24-bit bas alamat. Sekiranya bit-23, 22, dan 21 dijadikan sebagai bit pembahagi blok maka peta ingatan mikropemproses tersebut menjadi seperti berikut; kecuali,

- A. nombor blok ialah 3 bit
- B. saiz setiap blok ialah 2 Mbyte
- C. bilangan blok ialah 8
- D. blok alamat ialah 21 bit
- E. nombor blok terendah ialah 1

JAWAPAN:.....(10/100)

II) Pernyataan berikut adalah benar mengenai format fail heksadesimal rekod-S untuk CPU 68000 berikut; kecuali,

S2144004002079004005022279004005063E3900402A

- A. rekod data dengan 32-bit alamat
- B. bilangan byte ialah 20
- C. alamat muatan bermula daripada \$00400400
- D. ceksum \$2A
- E. byte data yang pertama ialah \$20

JAWAPAN:.....(10/100)

... 2/-

- III) Sekiranya kandungan awal alatdaftar D3 ialah \$00400500, persamaan arahan (a) MOVE.B \$5F, D3 dan arahan (b) MOVE.W \$5F, D3 adalah seperti berikut; kecuali,
- kedua arahan akan memindahkan \$5F ke dalam byte bawah alat daftar D3
 - kedua arahan memberikan kandungan akhir alatdaftar D3 adalah sama
 - kedua arahan tidak mengubah byte atas alatdaftar D3
 - kedua arahan akan menukarkan byte bawah alatdaftar D3 kepada 5F
 - kedua arahan akan mengekalkan nombor 400 dalam alatdaftar D3

JAWAPAN:.....(10/100)

- IV) Kandungan awal PC ialah \$00001000 dan kandungan awal D5 ialah \$12345678. Perubahan yang berlaku selepas arahan MOVE.W \$1020(PC), D5 adalah seperti berikut; kecuali,
- data di alamat \$00001021 akan dipindahkan ke byte bawah D5
 - perkataan bawah yang terkandung dalam D5 adalah kandungan data di alamat \$00001020 dan \$00001021
 - perkataan atas dalam alatdaftar D5 tidak berubah
 - nilai "offset" dalam arahan tersebut ialah \$1020
 - kandungan PC selepas pelaksanaan arahan ialah \$00001004
 -

JAWAPAN:.....(10/100)

- V) Jika kandungan awal A5 ialah \$00400550 dan kandungan D0 ialah \$0000DDDD, perubahan yang berlaku selepas pelaksanaan arahan MOVE.W (A5)+, D0 adalah
- kandungan D0 menjadi \$00400550
 - kandungan A5 menjadi \$0000DDDD
 - byte bawah D0 sama dengan kandungan di alamat \$00400551
 - byte bawah D0 sama dengan kandungan di alamat \$00400550
 - kandungan A5 meningkat menjadi \$00400554
 -

JAWAPAN:.....(10/100)

- VI) Kandungan A3 = \$0040ABDF dan A0 = \$00400400 sebaik sahaja sebelum pelaksanaan arahan MOVEA A3, A0. Kandungan A0 selepas pelaksanaan arahan tersebut adalah;
- Tidak berubah
 - \$FFFFABDF
 - \$0000ABDF
 - \$0040ABDF
 - \$ABDF0400

JAWAPAN:.....(10/100)

VII) Jika kandungan awal D0 ialah \$00056789 dan kandungan A4 ialah \$00400516. Pelaksanaan arahan MOVE.W D0, -(A4) akan menghasilkan perubahan berikut; kecuali,

- A. A4 menjadi \$00400512
- B. kandungan D0 tidak berubah
- C. kandungan data di alamat \$00400514 bernilai \$67
- D. kandungan data di alamat \$00400515 bernilai \$89
- E. kandungan data di alamat \$00400516 tidak berubah
- F.

JAWAPAN:.....(10/100)

VIII) Arahan MOVE.B #1, CCR akan menyebabkan perubahan berikut; kecuali,

- A. alatdaftar CCR bernilai \$01
- B. semua bit keadaan di reset kecuali bit pembawa
- C. bit pembawa diset manakala semua bit lain direset
- D. bit pembawa direset manakala bit tanda diset
- E. byte #01 dipindah ke alat daftar CCR

JAWAPAN:.....(10/100)

IX) Jika kandungan awal D0 ialah \$10204FFF dan D3 ialah \$1034F8AA. Perubahan yang berlaku selepas pelaksanaan arahan MOVE.B D0, D3 ialah

- A. kandungan D3 menjadi \$10204FFF
- B. kandungan D3 menjadi \$FFFFFFFF
- C. kandungan D0 menjadi \$10204FFE
- D. kandungan D3 menjadi \$10344FFF
- E. kandungan D3 menjadi \$1034F8FF

JAWAPAN:.....(10/100)

X) Jika kandungan awal D0 ialah \$00400400 dan D5 ialah \$004D0500. Perubahan yang akan berlaku selepas pelaksanaan arahan MOVE.W D0, D5 ialah

- A. kandungan D5 menjadi \$00400400
- B. kandungan D5 menjadi \$00400500
- C. byte bawah D5 tidak berubah
- D. byte bawah D0 meningkat
- E. pekataan bawah D5 tidak berubah

JAWAPAN:.....(10/100)

2) Pilih satu sahaja jawapan yang paling sesuai dan tuliskan pilihan anda di ruang jawapan yang disediakan untuk setiap persoalan berikut:-

I) Jika arahan MOVE.W \$0440(PC), D4 berada di alamat ingatan RAM \$00400430, perkataan tambahan operasi sumber ialah

- A. \$000E
- B. \$0440
- C. \$0400
- D. \$0010
- E. \$FFFE

JAWAPAN:.....(10/100)

II) Kandungan A0 sebelum pelaksanaan arahan MOVE.W #10, (A0)+ ialah \$00400408. Kandungan A0 selepas pelaksanaan arahan tersebut ialah

- A. \$0040040A
- B. \$00400410
- C. \$00400418
- D. \$00000010
- E. \$0000000A

JAWAPAN:.....(10/100)

III) Kandungan A4 sebaik sahaja sebelum pelaksanaan arahan MOVE.L #1, -(A4) ialah \$00400400. Kandungan A4 selepas pelaksanaan arahan tersebut ialah

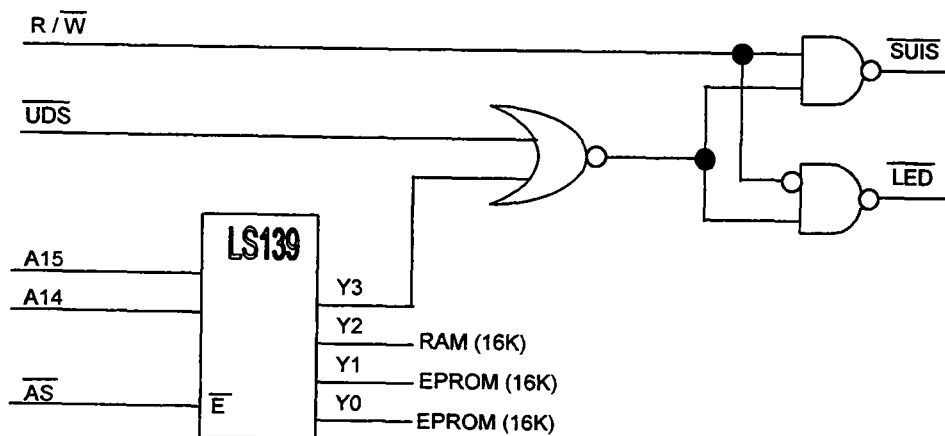
- A. \$00000001
- B. \$004003FC
- C. \$00000004
- D. \$004003FE
- E. \$00400404

JAWAPAN:.....(10/100)

IV) Kandungan D5 sebaik sahaja sebelum pelaksanaan arahan MOVEQ #1, D5 ialah \$AAAAAAAA. Kandungan D5 selepas pelaksanaan arahan tersebut ialah

- A. \$AAAAAAAA01
- B. \$00000001
- C. \$AAAA0001
- D. \$AAAAAAAA1
- E. \$00000010

JAWAPAN:.....(10/100)



Rajah 1

Berdasarkan litar sistem mikropemproses dalam Rajah 1 selesaikan permasalahan V - VIII

V) Arahan yang akan membaca keadaan SUI5 dan memaparkannya pada LED ialah

- i. `MOVE.B $0000F000,D0`
- ii. `MOVE.B $0000B000,D0`
- iii. `MOVE.B D0,$0000C000`
- iv. `MOVE.B D0,$0000B000`
- v. `MOVE.B D0,$0000E000`

- A. i dan ii
- B. i dan iii
- C. ii dan iv
- D. ii dan v
- E. iii dan iv

JAWAPAN:.....(10/100)

VI) Arahan yang akan membaca 2-byte data daripada RAM ialah

- i. `MOVE.W $00407000, D1`
- ii. `MOVE.W $0000B000, D1`
- iii. `MOVE.W $2020A000, D1`
- iv. `MOVE.W $B0D09000, D1`

- A. i sahaja
- B. i dan ii
- C. i, ii, iii
- D. ii, iii, dan iv
- E. tiada kombinasi jawapan yang betul

JAWAPAN:.....(10/100)

VII) Bilangan pantulan alamat LED ialah

- A. 2^{10}
- B. 2^{20}
- C. 2^{21}
- D. 2^{22}
- E. 2^{24}

JAWAPAN:.....(10/100)

VIII) Alamat berikut dinyahkodkan untuk EPROM oleh sistem mikropemproses dalam Rajah 1 kecuali

- A. FF070000 - FF07FFFF
- B. FF000000 - FF007FFF
- C. 00000000 - 00007FFF
- D. 00FF0000 - 00FF7FFF
- E. FF000000 - FF0F7FFF

JAWAPAN:.....(10/100)

IX) Kandungan D5 sebaik sahaja sebelum pelaksanaan arahan MOVEQ #9, D5 ialah \$B0D0B0D0. Kandungan D5 selepas pelaksanaan arahan tersebut ialah

- A. \$B0D0B0D9
- B. \$00000009
- C. \$B0D0FFF9
- D. \$FFFFFFF9
- E. \$FFFFFFF09

JAWAPAN:.....(10/100)

X) Berdasarkan bahagian aturcara dalam Rajah 2 nilai offset bagi pembilang program (PC) ialah

- A. \$3E
- B. \$1E
- C. \$40
- D. \$38
- E. \$42

JAWAPAN:.....(10/100)

MULA	EQU	\$00400400
DATA	EQU	\$00400440
ORG	MULA	
MOVE.W	BIL(PC), D5	
ORG	DATA	
BIL	DC.W	\$ABCD
END		

Rajah 2

- 3) Pilih satu sahaja jawapan yang paling sesuai dan tuliskan pilihan anda di ruang jawapan yang disediakan untuk setiap persoalan berikut:-

Berdasarkan aturcara bahasa pengumpulan dalam Rajah 3, selesaikan permasalahan I - III

- I) Pernyataan yang benar mengenai pembolehubah tempatan yang disediakan oleh arahan LINK dalam Rajah 3 adalah
- Berada di alamat mulai daripada alamat \$0040450
 - Bilangannya ialah 4 perkataan
 - Kedudukannya ditunjukkan oleh penunjuk stack
 - Berada di alamat \$00400444 dan \$00400442
 - Kedudukannya ditunjuk oleh alatdaftar alamat A7

JAWAPAN:.....(10/100)

- II) Antara nilai pembolehubah yang disediakan oleh arahan LINK tersebut adalah
- \$20
 - \$FF
 - \$CD
 - \$AB
 - \$40

JAWAPAN:.....(10/100)

```

MULA   EQU $00400400
        ORG MULA
        MOVE.L #$00400450,A7
        MOVE.W #$2020,D0
        MOVE.W D0,D1
        MOVE.W D0,-(SP)
        MOVEA.L #$ABCDABCD,A3
        BSR   SUBRTN
        MOVE.W A0,$0000C000
SUBRTN LINK   A3,#-4
        MOVEM.W D0-D1,-(A3)
        MOVE.W 14(A3),D1
        MOVE.W D0,$0000C000
        ADD   D1,D0
        UNLK  A3
        RTS
        END

```

Rajah 3

- III) Alamat arahan yang akan dilaksanakan oleh mikropemproses selepas arahan RTS ialah
- \$00400400
 - \$ABCDABCD
 - \$00400418
 - \$00400450
 - \$00400436

JAWAPAN:.....(10/100)

... 8/-

- IV) Penyataan yang benar mengenai arahan a) MOVEA.L #ARRAY, A5 dan arahan b) LEA ARRAY, A5 ialah
- arahan a) memindahkan kod operasi ke alatdaftar A5 manakala arahan b) memindahkan alamat kod operasi ke alatdaftar A5
 - arahan b) memindahkan kod operasi ke alatdaftar A5 manakala arahan a) memindahkan alamat kod operasi ke alatdaftar A5
 - kedua arahan memindahkan kod operasi ke alat daftar A5
 - kedua arahan memindahkan alamat berkesan kod operasi
 - tiada jawapan yang betul

JAWAPAN:.....(10/100)

- V) Kandungan D6 dan D7 selepas pelaksanaan bahagian aturcara dalam Rajah 4 ialah

	D6	D7
A.	\$12345678	\$ABCDABCD
B.	\$ABCDABCD	\$12345678
C.	\$5678ABCD	\$1234ABCD
D.	\$5678ABCD	\$ABCD1234
E.	\$1234ABCD	\$5678ABCD

JAWAPAN:.....(10/100)

MULA	EQU	\$00400400
	ORG	MULA
	MOVE.L	#\$1234ABCD, D6
	MOVE.L	#\$ABCD5678, D7
	SWAP	D7
	EXG	D6, D7
	END	

Rajah 4

- VI) Jika kandungan D3 ialah \$00000080, kandungannya sebaik sahaja selepas pelaksanaan arahan EXT.L D3 ialah

A.	\$0000FF80
B.	\$FFFFFF80
C.	\$FFFF0080
D.	\$00800000
E.	\$00000080

JAWAPAN:.....(10/100)

MULA	EQU	\$00400400
	ORG	MULA
	MOVE.L	#\$00010000, SP
	MOVE.L	#\$00009000, D1
	ADD.L	D7, D1
	BSR	\$00400480

Rajah 5

VII) Kandungan A7 sebaik sahaja selepas pelaksanaan bahagian aturcara dalam Rajah 5 ialah

- A. \$0000FFFC
- B. \$0000FFFE
- C. \$00010000
- D. \$00009000
- E. \$00400480

JAWAPAN:.....(10/100)

VIII) Jika D0 = \$0000003E dan D1 = \$000000E0, arahan berikut akan menseset bit pembawa (C) kecuali

- A. ADD.B D0, D1
- B. ROR.B #4, D0
- C. LSR.B #6, D1
- D. MOVE.B #1, CCR
- E. ASL.B #4, D1

JAWAPAN:.....(10/100)

IX) Jika D2 = \$FFFF000E dan D3 = \$FFFFFF0F, arahan berikut akan menseset bit sifar (Z) kecuali

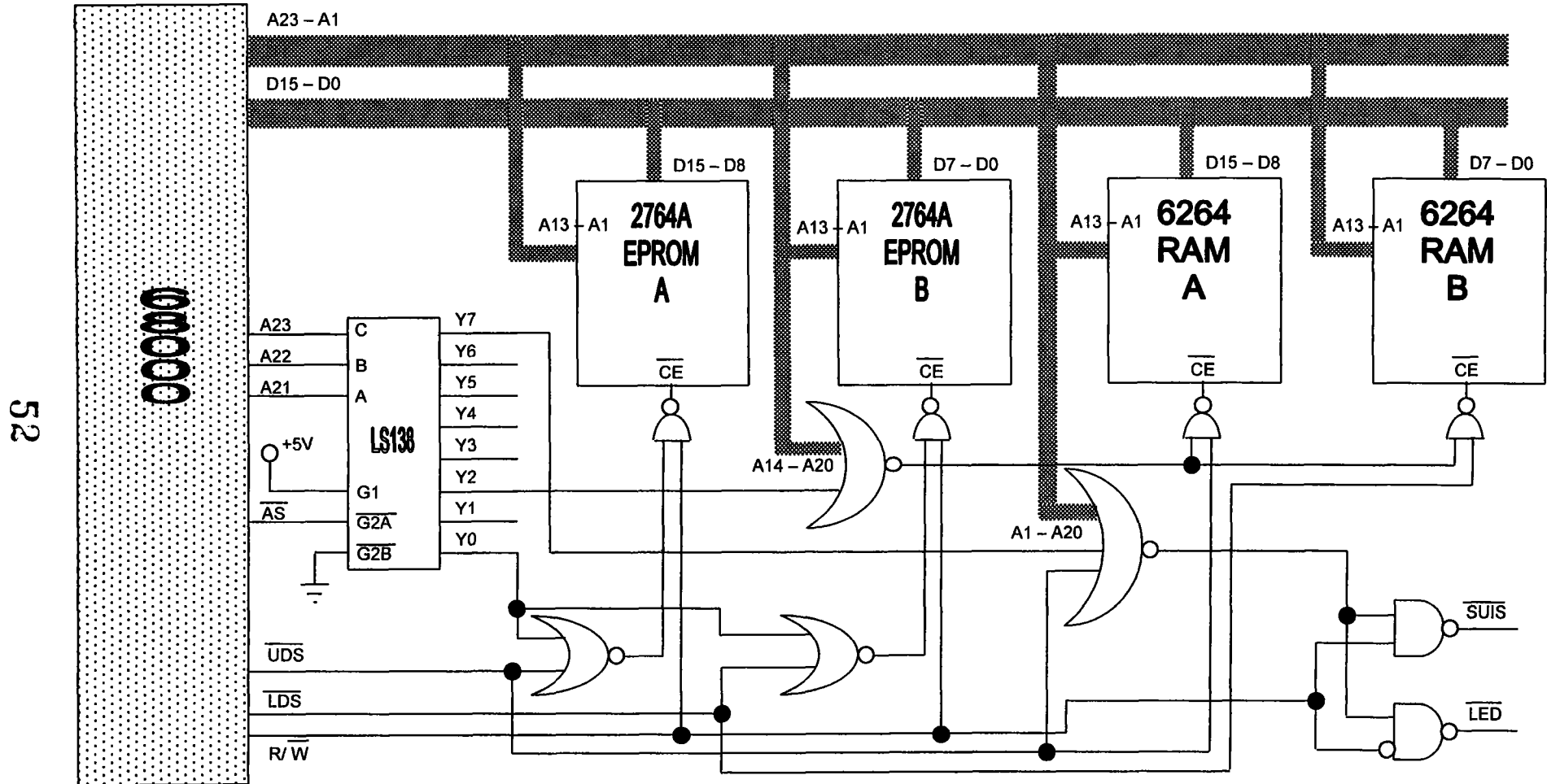
- A. CMP.B D2,D2
- B. SUBI.W #14, D2
- C. ASL.B D2, D3
- D. ROXR.B #4, D3
- E. XOR.B D3, D3

JAWAPAN:.....(10/100)

X) Kandungan D5 sebaik sahaja sebelum pelaksanaan arahan ROR.B #4, D0 ialah \$12345678. Kandungan D5 selepas pelaksanaan arahan tersebut ialah

- A. \$12345687
- B. \$12345678
- C. \$56781234
- D. \$00001234
- E. \$87654321

JAWAPAN:.....(10/100)



Rajah 6

4) Rajah 6 menunjukkan litar sistem mikropemproses dengan ingatan RAM, EPROM, SUIS, dan LED di antara mukakan dengan mikropemproses pada suatu lokasi alamat yang dinyahkodkan oleh penyahkod 74LS138 dan get TAKATAU. Berdasarkan Rajah 6 tersebut selesaikan perkara berikut:

i) Saiz ingatan yang boleh diberi oleh mikropemproses 68000 ialah (10/100)

.....
.....
.....

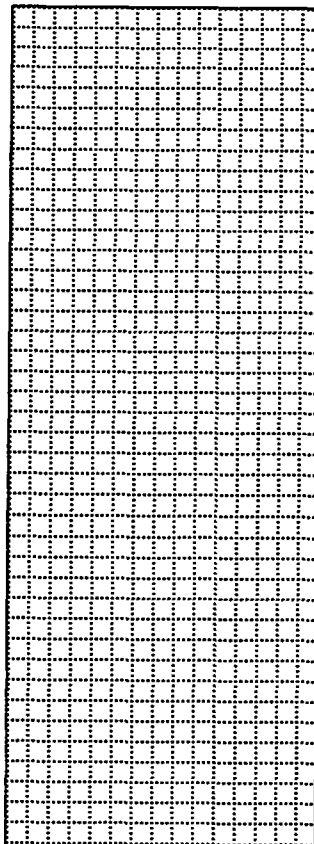
ii) Bilangan blok ingatan yang dinyahkodkan oleh penyahkod LS138 ialah (10/100)

.....
.....
.....

iii) Saiz setiap blok pula ialah (10/100)

.....
.....
.....

iv) Lengkapkan peta ingatan mikropemproses tersebut dengan memberikan alamat awal dan akhir setiap bahagian blok. (Gunakan Rajah 7 yang disediakan) (10/100)



Rajah 7

- v) Jumlah keseluruhan ingatan RAM sistem mikropemproses tersebut adalah (10/100)
.....
.....
.....
- vi) Jika semua garis alamat yang tiada sambungan fizikal bernilai rendah tentukan lokasi ingatan RAM sistem mikropemproses tersebut. (10/100)
.....
.....
.....
- vii) Jumlah keseluruhan ingatan EPROM sistem mikropemproses tersebut ialah (10/100)
.....
.....
.....
- viii) Tuliskan bahagian aturcara yang membolehkan mikropemproses membaca keadaan SUIS kemudian memaparkannya di LED dan menyimpannya di lokasi ingatan RAM \$00402020 (30/100)
.....
.....
.....
.....
.....
.....
.....
.....

5) Fahamkan aturcara bahasa penghimpunan berikut; keadaan logik sistem pemapar tujuh ruas yang digunakan dalam aturcara ini di sediakan di Lampiran A.

```

*   EXPERIMEN 8
*   MEMANDU SATU PEMAPAR TUJUH RUAS
*   EXP8.SRC
*   INCLUDE IOEQU.INC -- MEMASUKKAN ALAMAT IO
    INCLUDE IOEQU.INC
PROGRAM   EQU   $400400           ALAMAT ATURCARA
DATA     EQU   $400500           ALAMAT DATA
    ORG     PROGRAM
MOVE.B   #$FF,PCDDR             SET POT C SEBAGAI OUTPUT
MOVE.B   #$FF,PBDDR             SET POT B SEBAGAI OUTPUT
MOVE.L   #DIGIT,A1             DAPATKAN DIGIT UNTUK DIPAPARKAN
MOVE.B   0(A1),D0
MOVE.L   #PATNS,A0             AO MENUNJUK KE RUPABENTUK
MOVE.B   0(A0,D0),D1           MENDAPATKAN RUPABENTUK SETARA
MOVE.B   D1,PBDR               OUPUTKAN RUPABENTUK
MOVE.B   #4,D0                 PILIH DIGIT L.S.
MOVE.B   D0,PCDR               ONKAN SUIS
TRAP     #11                    KEMBALI KE MONITOR
DC.W     0
    ORG     DATA               RUPABENTUK TUJUH RUAS
PATNS DC.B   $3F,$06,$5B,$4F,$66,$6D,$7D,$07,$7F,$67,$5F,$7C,$39,
           $5E,$79,$71,$76,$38,$73,$3E
DIGIT DC.B   $11
    END
    
```

- i) Nyatakan alamat permulaan aturcara ditulis (10/100)

- ii) Nyatakan alamat permulaan data ditulis (10/100)

- iii) Nilai PATNS dalam nombor perenambelasan ialah (10/100)

- iv) Nilai DIGIT dalam nombor perenambelasan ialah (10/100)

- v) Nyatakan bentuk digit yang dipaparkan oleh aturcara (10/100)

LAMPIRAN A

Mnemonic	Assembler Syntax	Operand Size	Allowable Addressing Modes		Condition Codes
			Source	Destination	X N Z V C
ABCD	ABCD Dy,Dx ABCD -(Ay),-(Ax)	8 8	Dn -(An)	Dn -(An)	. U . U . . U . U .
ADD	ADD <ea>,Dn ADD Dn,<ea>	8, 16, 32 8, 16, 32	All (1) Dn	Dn Alterable
ADDA	ADD <ea>,An	16, 32	All	An	- - - - -
ADDI	ADDI #d,<ea>	8, 16, 32	#d	Data Alterable
ADDQ	ADDQ #d,<ea>	8, 16, 32	#d (2)	Alterable (1)
ADDX	ADDX Dy,Dx ADDX -(Ay),-(Ax)	8, 16, 32 8, 16, 32	Dn -(An)	Dn -(An)
AND	AND <ea>,Dn AND Dn,<ea>	8, 16, 32 8, 16, 32	Data Dn	Dn Alterable	- . . 0 0 - . . 0 0
ANDI	ANDI #d,<ea> ANDI #d,SR (3)	8, 16, 32 8, 16	#d #d	Data Alterable SR	- . . 0 0
ASL	ASL Dx,Dy ASL #d,Dn ASL <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable
ASR	ASR Dx,Dy ASR #d,Dn ASR <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable
Bcc	Bcc <label>	8, 16	If cc, then PC + d → PC		- - - - -
BCHG	BCHG Dn,<ea> BCHG #d,<ea>	8, 32 8, 32	-Dn #d	Data Alterable Data Alterable	- - . . - - - . . -
BCLR	BCLR Dn,<ea> BCLR #d,<ea>	8, 32 8, 32	Dn #d	Data Alterable Data Alterable	- - . . - - - . . -
BRA	BRA <label>	8, 16	PC + d → PC,		- - - - -
BSET	BSET Dn,<ea> BSET #d,<ea>	8, 32 8, 32	Dn #d	Data Alterable Data Alterable	- - . . - - - . . -
BSR	BSR <label>	8, 16	PC → -(SP); PC + d → PC		- - - - -
BTST	BTST Dn,<ea> BTST #d,<ea>	8, 32 8, 32	Dn #d	Data, Except Immediate Data, Except Immediate	- - . . - - - . . -
CHK	CHK <ea>,Dn	16	If Dn < 0 or Dn > (ea), then TRAP		- . U U U
CLR	CLR <ea>	8, 16, 32	Data Alterable		- 0 1 0 0
CMP	CMP <ea>,Dn	8, 16, 32	All (1)		-
CMPA	CMPA <ea>,An	16, 32	All		-
CMPI	CMPI #d,<ea>	8, 16, 32	#d		-
CMPM	CMPM (Ay)+,(Ax)+	8, 16, 32	(An)+		-

Mnemonic	Assembler Syntax	Operand Size	Allowable Addressing Modes		Condition Codes
			Source	Destination	X N Z V C
DBcc	BDcc Dn,<label>	16	If cc, then Dn - 1 → Dn; If Dn ≠ -1, then PC + d → PC		-----
DIVS	DIVS <ea>,Dn	16	Data	Dn	- * * * 0
DIVU	DIVU <ea>,Dn	16	Data	Dn	- * * * 0
EOR	EOR Dn,<ea>	8, 16, 32	Dn	Data Alterable	- * * 0 0
EORI	EORI #d,<ea> EORI #d,SR (3)	8, 16, 32 8, 16	#d #d	Data Alterable SR	- * * 0 0 * * * 0 *
EXG	EXG Rx,Ry	32	Dn or An	Dn or An	-----
EXT	EXT Dn	16, 32	Dn		- * * 0 0
JMP	JMP <ea>		<ea> → PC	Control	-----
JSR	JSR <ea>		PC → -(SP); <ea> → PC	Control	-----
LEA	LEA <ea>,An	32	Control	An	-----
LINK	LINK An,#d	Unsize	An		-----
LSL	LSL Dx,Dy LSL #d,Dn LSL <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	* * * 0 * * * * 0 * * * * 0 *
LSR	LSR Dx,Dy LSR #d,Dn LSR <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	* 0 * 0 * * 0 * 0 * * 0 * 0 *
MOVE	MOVE <ea>,<ea> MOVE <ea>,CCR MOVE <ea>,SR (6) MOVE SR,<ea> MOVE USP,An (6) MOVE An,USP (6)	8, 16, 32 16 16 16 32 32	All (1) Data Data SR USP An	Data Alterable CCR SR Data Alterable An USP	- * * 0 0 * * * * * * * * * * ----- ----- -----
MOVEA	MOVEA <ea>,An	16, 32	All	An	-----
MOVEM	MOVEM <list>,<ea> MOVEM <ea>,<list>	16, 32 16, 32	 Control or (An)+	Control Alterable or -(An)	----- -----
MOVEP	MOVEP Dx,d(Ay) MOVEP d(Ay),Dx	16, 32 16, 32	Dn d(An)	d(An) Dn	----- -----
MOVEQ	MOVEQ #d,Dn	32	#d (7)	Dn	- * * 0 0
MULS	MULS <ea>,Dn	16	Data	Dn	- * * 0 0
MULU	MULU <ea>,Dn	16	Data	Dn	- * * 0 0
NBCD	NBCD <ea>	8		Data Alterable	* U * U *
NEG	NEG <ea>	8, 16, 32	Data Alterable		* * * * *
NEGX	NEGX <ea>	8, 16, 32	Data Alterable		* * * * *
NOP	NOP		PC + 2 - PC		-----
NOT	NOT <ea>	8, 16, 32		Data Alterable	- * * 0 0

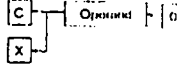

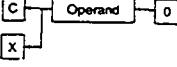
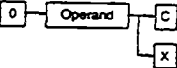

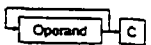
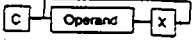
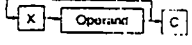
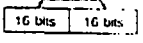
Mnemonic	Assembler Syntax	Operand Size	Allowable Addressing Modes		Condition Codes
			Source	Destination	
OR	OR <ea>, Dn OR Dn,<ea>	8, 16, 32 8, 16, 32	Data Dn	Dn Alterable	X N Z V C - . . 0 0 - . . 0 0
ORI	ORI #d,<ea> ORI #d,SR (3)	8, 16, 32 8, 16	#d #d	Data Alterable SR	- . . 0 0
PEA	PEA <ea>	32	Control		- - - - -
RESET (6)	RESET				- - - - -
ROL	ROL Dx,Dy ROL #d,Dn ROL <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	- . . 0 . - . . 0 . - . . 0 .
ROR	ROR Dx,Dy ROR #d,Dn ROR <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	- . . 0 . - . . 0 . - . . 0 .
ROXL	ROXL Dx,Dy ROXL #d,Dn ROXL <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	. . . 0 0 0 .
ROXR	ROXR Dx,Dy ROXR #d,Dn ROXR <ea>	8, 16, 32 8, 16, 32 16	Dn (4) #d (5)	Dn Dn Memory Alterable	. . . 0 0 0 .
RTE (6)	RTE		(SP) + → SP; (SP) + → PC	
RTR	RTR		(SP) + → CCR; (SP) + → PC	
RTS	RTS		(SP) + → PC		- - - - -
SBCD	SBCD Dy,Dx SBCD -(Ay),-(Ax)	8 8	Dn -(An)	Dn -(An)	. U . U . . U . U .
Scc	Scc <ea>	8	If cc, then 1s → (ea); otherwise 0s → (ea)	Data Alterable	- - - - -
STOP (6)	STOP #d	16	#d → SR, then STOP	
SUB	SUB <ea>,Dn SUB Dn,<ea>	8, 16, 32 8, 16, 32	All (1) Dn	Dn Alterable
SUBA	SUBA <ea>,An	16, 32	All	An	- - - - -
SUBI	SUBI #d,<ea>	8, 16, 32	#d	Data Alterable
SUBQ	SUBQ #d,<ea>	8, 16, 32	#d (2)	Alterable (1)
SUBX	SUBX Dy,Dx SUBX -(Ay),-(Ax)	8, 16, 32 8, 16, 32	Dn -(An)	Dn -(An)
SWAP	SWAP Dn	16	Dn		- - - - -
TAS	TAS <ea>	8	Data Alterable		- - - - -
TRAP	TRAP #<vector>		PC → -(SP); SR → -(SP); #<vector> → PC		- . . 0 0 - - - - -
TRAPV	TRAPV		If V = 1, then TRAP		- - - - -
TST	TST <ea>	8, 16, 32	Data Alterable		- - - - -
UNLK	UNLK An	Unsize		An	- . . 0 0 - - - - -

Footnotes:
 (1) If the operation size is byte, the address register direct addressing mode is not allowed.
 (2) Immediate operand, with a value from 1 to 8.
 (3) If the operation size is word, the instruction is privileged.
 (4) Source data register contains the shift count. Count = 0 to 63, where 0 produces a count of 64.
 (5) The data is the shift count, 1 to 8.
 (6) This operation is privileged.
 (7) Eight bits of immediate data, which are sign-extended to a 32-bit long operand.

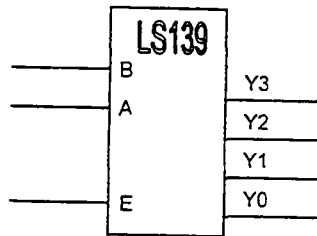
Effective Addressing Mode Categories

Addressing Mode	Addressing Categories				Assembler Syntax
	Data	Memory	Control	Alterable	
Data register direct.	X			X	Dn
Address register direct.				X	An
Register indirect.	X	X	X	X	(An)
Register indirect with postincrement.	X	X		X	(An)+
Register indirect with predecrement.	X	X		X	-(An)
Register indirect with displacement.	X	X	X	X	d(An)
Register indirect with index.	X	X	X	X	d(An,RI)
Absolute short.	X	X	X	X	xxxx
Absolute long.	X	X	X	X	xxxxxxxx
PC relative with displacement.	X	X	X		d
PC relative with index.	X	X	X		d(RI)
Immediate.	X	X			#xxxx

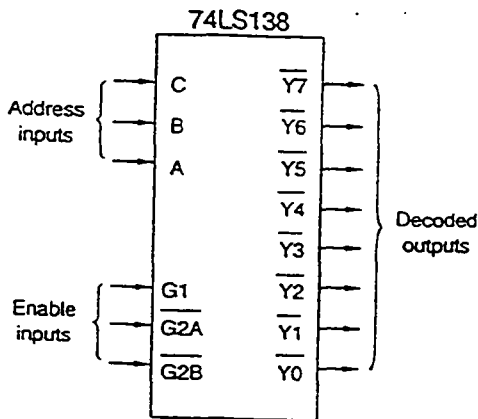
Table 3-6. Shift and Rotate Instructions.

Instruction	Operation	Bit Movement
ASL	Arithmetic shift left	
ASR	Arithmetic shift right	
LSL	Logical shift left	
LSR	Logical shift right	
ROL	Rotate left	
ROR	Rotate right	
ROXL	Rotate left with extend bit	
ROXR	Rotate right with extend bit	
SWAP	Swap words of a longword	

Jadual Kebenaran untuk Penyahkod 2 ke 4 (74LS139) dan 3 ke 8 (74LS138)



E	A	B	Y0	Y1	Y2	Y3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



	G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	x	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1	0

(a)

(b)

The 74LS138 3-line to 8-line decoder. (a) symbol, (b) truth table.

8. SEVEN SEGMENT DISPLAY

The seven segment display on the Applications Board consists of two, two digit modules providing a total of four digits. No decoding is provided on board so the decoding function must be performed in software. In this experiment, a single digit decimal value is read from the a defined memory location, decoded then displayed on a single seven segment display. The bit allocation for the display is shown below:

D3	D2	D1	D0	Function	D7	D6	D5	D4	D3	D2	D1	D0
Port B					Port C							
				7 Seg. Disp Segments	dp	g	f	e	d	c	b	a
D3	D2	D1	D0	7 Seg. Disp Digit En.								

A logic 1 turns the segment ON and a logic 1 enables the digit. For example to display the number 8 on digit 2 (third from the right) the program must output \$FF on port C and \$04 on port B. The task of deriving the seven segment patterns is left to the student. The resulting values can be seen in the program listing starting at the label "PATNS". This experiment introduces the 'address register indirect with index addressing' mode of the 68000 microprocessor. The address register is used to point to the start of the pattern table and a data register used as the index (or offset) value which is the value to be displayed. Thus if the value to display is 6, this is loaded to the data register, added to the address register (which points to the start of the table) this causes the pattern for 6 to be obtained (hence 6 has been decoded to the equivalent seven segment pattern).

This program takes a value passed to it via a defined memory location, obtains the equivalent seven segment pattern and output it to the least significant digit.