

Angka Giliran:

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 2003/2004

Februari/Mac 2004

ZAT 281/4 - Pengantar Mikropemproses

Masa : 3 jam

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

Jawab kesemua **LIMA** soalan.

Angka Giliran:

1. Rajah 1 merupakan suatu aturcara bahasa penghimpunan sistem mikropemproses 68000 yang dilaksanakan kepada sistem mikropemproses di makmal fizik gunaan USM.

1	*	ATURCARA PERLENGAHAN DAN PEMANGGILAN SUBRUTIN	
2	*	MENSET POT-B SEBAGAI OUTPUT DAN MENSET PEMBILANG	
3	*	PELENGAHAN SETIAP PENURUNAN	
4	*	E103.SRC	
5	00A00007	PBDDR EQU \$A00007	
6	00A00013	PBDR EQU \$A00013	
7	00400420	PROGRAM EQU \$400420	KAWASAN RAM ATURCARA
8	00400470	SUBRTN EQU \$400470	KAWASAN RAM SUBRUTIN
9	00400500	DATA EQU \$400500	
10	004006FE	SPLOC EQU \$4006FE	
11	00400500	ORG DATA	
12	00400500 000144AD	MASLENG DC.L \$144AD	TETAPKAN BILANGAN PERLENGAHAN
13	00400420	ORG PROGRAM	ALAMAT PERMULAAN ATURCARA
14	00400420 13FC00FF	MULA MOVE.B #\$FF,PBDDR	SET POT-B SEBAGAI OUTPUT
	00400424 00A00007		
15	00400428 103C00FF	MOVE.B #\$FF,D0	SET BILANGAN MAKSIMUM
16	0040042C 3E790040	MOVEA SPLOC,SP	TETAPKAN LOKASI PENUNJUK STACK
	00400430 06FE		
17	00400432 13C000A0	ULANG MOVE.B D0,PBDR	TUNJUKKAN BILANGAN PADA LED
	00400436 0013		
18	00400438 6136	BSR.S DELAY	DELAY .5 SAAT
19	0040043A 56C8FFF6	DBNE D0,ULANG	TURUN DAN ULANG JIKA Z=0
20	0040043E 13C000A0	MOVE.B D0,PBDR	
	00400442 0013		
21	00400444 4E4B	TRAP #11	
22		*****	
23	*	SUBRUTIN PERLENGAHAN 0.5 SAAT	
24	00400470	ORG SUBRTN	
25	00400470 22390040	DELAY MOVE.L MASLENG,D1	SET PELENGAHAN 0.5 SAAT
	00400474 0500		
26	00400476 5381	BERIKUT SUBQ.L #1,D1	BILANG MENURUN HINGGA SIFAR
27	00400478 66FC	BNE.S BERIKUT	
28	0040047A 4E75	RTS	KELUAR DARIPADA SUBRUTIN
29	0040047C	END	

Rajah 1 Aturcara bahasa penghimpunan

Angka Giliran:

a) Secara ringkas terangkan apakah yang dilaksanakan oleh aturcara tersebut?

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

(10/100)

b) Apabila aturcara dimuatkan ke dalam RAM ia akan ditulis di alamat

(10/100)

c) Saiz aturcara (bilangan byte) ialah

(10/100)

d) Nilai bagi pembolehubah MASLENG ialah

(10/100)

e) Kandungan permulaan alatdaftar A7 ialah

(10/100)

f) Kandungan alatdaftar A7 semasa mikropemproses melaksanakan subrutin ialah

.....

(10/100)

g) Kandungan alatdaftar D0 selepas arahan MOVE.B #\$FF,D0 dilaksanakan ialah

.....

(10/100)

h) Kandungan alatdaftar D0 selepas aturcara selesai dilaksanakan ialah

.....

(10/100)

Angka Giliran:

- i) Tentukan masa yang diambil untuk aturcara berhenti (terminate) dengan menganggap masa pelaksanaan aturcara sangat pendek berbanding dengan masa perlangahan

.....

.....

.....

.....

.....

.....

(20/100)

```
S007000065323033FE
S213400400207C004005003018D05850804E4B00001D
S9030000FC
```

Rajah 2. Fail perenambelasan arahan bahasa penghimpunan

2.

- a) Rajah 2 merupakan fail perenambelasan suatu arahan aturcara bahasa penghimpunan. Bilangan dan semakan jumlah dalam fail tersebut mungkin berlaku kesilapan. Tentukan bilangan byte dan semakan jumlah yang betul serta berikan alamat permulaan dan alamat akhir aturcara tersebut.

Bilangan byte ialah

Alamat permulaan ialah

Alamat akhir ialah

Semakan jumlah ialah

(50/100)

- b) Dua perkataan data berada di alamat \$400500 dan \$400502. Tuliskan aturcara bahasa penghimpunan yang boleh mencampur dua perkataan tersebut dan simpan jumlahnya di alamat \$400504.

.....

.....

.....

.....

.....

...5/-

Angka Giliran:

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

(50/100)

3. Rajah 3 menunjukkan litar sistem mikropemproses dengan ingatan EPROM yang di antaramukakan dengan mikropemproses pada suatu lokasi alamat yang dinyahkodkan oleh penyahkod 74LS138. Berdasarkan Rajah 3 tersebut selesaikan persoalan berikut:

a) Saiz ingatan yang boleh diberikan oleh mikropemproses 68000 ialah

.....
(10/100)

b) Saiz setiap blok ingatan yang dinyahkodkan oleh penyahkod 74LS138 ialah

.....
(10/100)

c) Jumlah keseluruhan ingatan EPROM mikropemproses tersebut ialah

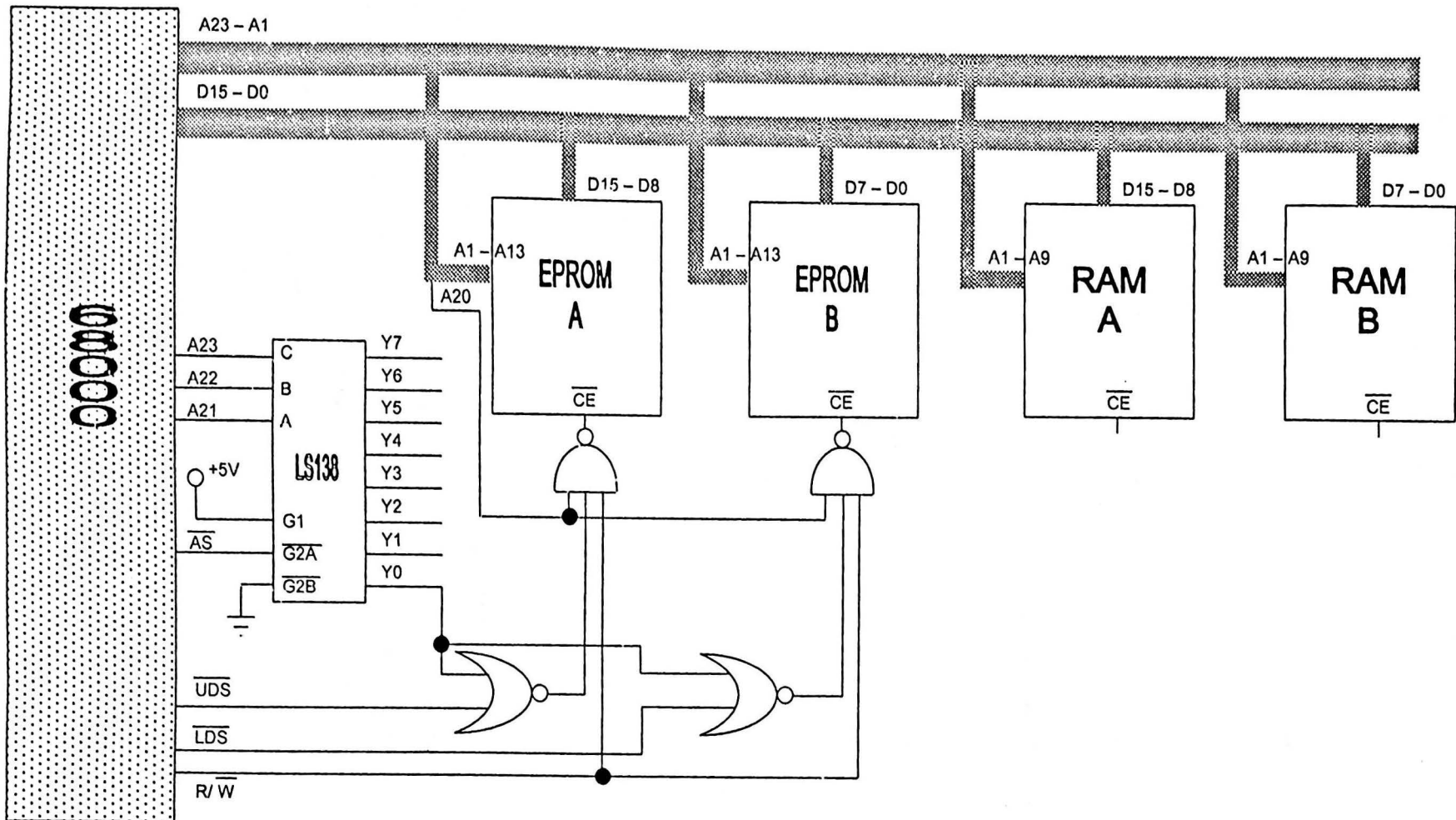
.....
(10/100)

d) Ingatan EPROM tersebut dikatakan terpantul. Bilangan pantulannya ialah

.....
(20/100)

e) Ingatan RAM A dan RAM B hendak diantaramukakan kepada sistem mikropemproses dengan alamat permulaan \$400400 tanpa alamat yang berlipat. Dengan menggunakan get-get logik yang bersesuaian rekabentuk litar penyahkod yang sesuai di dalam Rajah 3 yang mengantaramukakan kedua RAM tersebut.

(50/100)



54

Rajah 3

Angka Giliran:

1	*	SET BIT-6 POT C SEBAGAI INPUT DAN YANG LAINNYA SEBAGAI OUTPUT	
2	*	POT B SEBAGAI OUTPUT. MENGANTAR DENYUT PELAHAN MELALUI	
3	*	GENTIAN OPTIK KEPADA PENERIMA OPTIK DAN DENYUT KEPADA LED D7	
4	*	E403.SRC	
5	00AC0007	PBDDR EQU \$A00007	
6	00AC0009	PCDDR EQU \$A00009	
7	00A00013	PBDR EQU \$A00013	
8	00A00019	PCDR EQU \$A00019	
9	00400400	PROGRAM EQU \$400400	KAWASAN ATURCARA
10	0000D000	MASLENG EQU \$D000	MASA PELENGAHAN
11	0040067E	SPLOC EQU \$40067E	LOKASI ALAMAT STACK
12	00400400	ORG PROGRAM	ALAMAT PERMULAAN
13	00400400 13FC00BE	START MOVE.B #SBE,PCDDR	SET BIT 6 POT C SEBAGAI INPUT [20]
	00400404 00A00009		
14	00400408 13FC00FF	MOVE.B #\$FF,PBDDR	SET POT B SEBAGAI OUTPUT [20]
	0040040C 00A00007		
15	00400410 2E7C0040	MOVE.L #SPLOC,A7	SET LOKASI ALAT DAFTAR A7 [28]
	00400414 067E		
16	00400416 13FC0000	REPEAT MOVE.B #0,PBDR	OUTPUT LOGIK 0 KE GENTIAN OPTIK TX [20]
	0040041A 00A00013		
17	0040041E 6100001C	BSR UJIBIT	UJI BIT DAN OUTPUTKAN KE BIT-1 POT C [18]
18	00400422 13FC0001	MOVE.B #1,PBDR	OUTPUT LOGIK 1 KE GENTIAN OPTIK TX [20]
	00400426 00A00013		
19	0040042A 61000010	BSR UJIBIT	UJI BIT DAN OUTPUTKAN KE BIT-1 POT C [18]
20	0040042E 1A3900A0	MOVE.B PCDR,D5	[20]
	00400432 0019		
21	00400434 02050001	ANDI.B #01,D5	[8]
22	00400438 66DC	BNE REPEAT	ULANG JIKA DO ON [10(8)]
23	0040043A 4E4B	TRAP #11	[34]
24		*****	
25	*	SUBROUTIN UNTUK MENGUJI PENERIMA GENTIAN OPTIK	
26	*	DAN KEADAAN OUTPUT KEPADA BIT-0 POT C	
27	0040043C 61000036	UJIBIT BSR SETTLE	MASA UNTUK SETTLE [18]
28	00400440 103900A0	MOVE.B PCDR,D0	BACA POT C [16]
	00400444 0019		
29	00400446 02000040	ANDI.B #\$40,D0	TOPENGGAN BIT PENETIMA GENTIAN OPTIK
30	0040044A 6700000E	BEQ SIFAR	JIKA SIFAR OUTPUTKAN 0 KE BIT-0 [10(8)]
31	0040044E 13FC0080	MOVE.B #\$80,PBDR	JIKA TIDAK OUTPUTKAN 1 KE BIT-7 [20]
	00400452 00A00013		
32	00400456 6000000A	BRA TERUS	TERUS [10]
33	0040045A 13FC0000	SIFAR MOVE.B #C,PBDR	OUTPUT 0 TO BIT 7 [20]
	0040045E 00AC0013		
34	00400462 61000004	TERUS BSR LENGAH	PELENGAHAN [18]
35	00400466 4E75	RTS	[16]
36		*****	
37	*	SUBROUTIN PELENGAHAN	
38	00400468 223C0000	LENGAH MOVE.L #MASLENG,D1	SET PELENGAHAN [12]
	0040046C D000		
39	0040046E 5381	BERIKUT SUBQ.L #1,D1	BILANG MENURUN HINGGA SIFAR [8]
40	00400470 66FC	BNE.S BERIKUT	[10(12)]
41	00400472 4E75	RTS	KELUAR DARIPADA SUNROUTIN [16]
42		*****	
43	*	SUBROUTIN PELENGAHAN SETTLE	
44	00400474 223C0000	SETTLE MOVE.L #400,D1	SET PELENGAHAN [12]
	00400478 0190		
45	0040047A 5381	NEXTS SUBQ.L #1,D1	BILANG MENURUN HINGGA SIFAR
46	0040047C 66FC	BNE.S NEXTS	[10(12)]
47	0040047E 4E75	RTS	KELUAR DARIPADA SUBROUTIN [16]
48	00400480	END	

Rajah 4 Aturcara penghantaran denyut melalui gentian optik

Angka Giliran:

d) Terangkan dengan ringkas bagaimana data dihantar melalui suatu gentian optik

.....

.....

.....

.....

.....

.....

.....

.....

.....

(20/100)

e) Tentukan lebar denyut isyarat yang diperlukan bagi menghantar satu byte data dalam masa 4 ns melalui suatu gentian optik

.....

.....

.....

.....

(10/100)

1		*			
2	00E0000E	PSUIS	EQU	\$B0000E	;POT SUIS
3	00400400		ORG	\$400400	;LOKASI ATURCARA
4	00400400	MULA			
5	00400400 DB85		ECR.L	D5,D5	;arahan A
6	00400402 2E7C0040		MOVE.L	#\$4004FE,SP	;arahan B
	00400406 04FE				
7	00400408 3A3900B0	ULANG	MOVE.W	PSUIS,D5	;arahan C
	0040040C 000E				
8	0040040E 4EF90040		JMP	ULANG	;arahan D
	00400412 0408				
9	00400414 4E4B		TRAP	#11	;arahan E
10	00400416 0000		DC.W	0	;arahan F
11	00400418		END		

Rajah 5 Aturcara pengujian pot suis

5. Aturcara dalam Rajah 5 digunakan untuk untuk menguji samaada sistem mikropemproses boleh membaca barisan enambelas suis input yang di antaramukakan melalui suatu pot. Berdasarkan atur cara tersebut selesaikan permasalahan berikut.

Angka Giliran:

a) Nyatakan alamat bagi pot tersebut.

.....


(20/100)

b) Terangkan kenapa isyarat pelaksanaan bahagian aturcara di alamat \$00400408 hingga \$00400412 dapat diperhatikan pegun di layar osiloskop serta nyatakan syarat untuk suatu kitaran isyarat pegun tersebut boleh dicapai sekiranya frekuensi sistem mikropemproses ialah 1MHz.

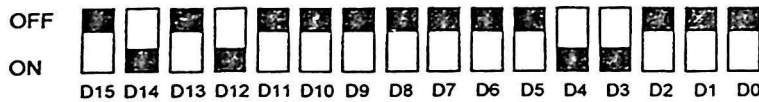
.....
.....
.....
.....
.....
.....

(30/100)

c) Sekiranya sistem tersebut berfungsi dengan baik dan keadaan barisan suis seperti ditunjukkan dalam Rajah 6, lengkapkan bentuk isyarat (R/\overline{W} , A20, D12, dan D4) serta nyatakan kandungan bus alamat dan data pada kitaran jam yang diperhatikan dalam Rajah 7.

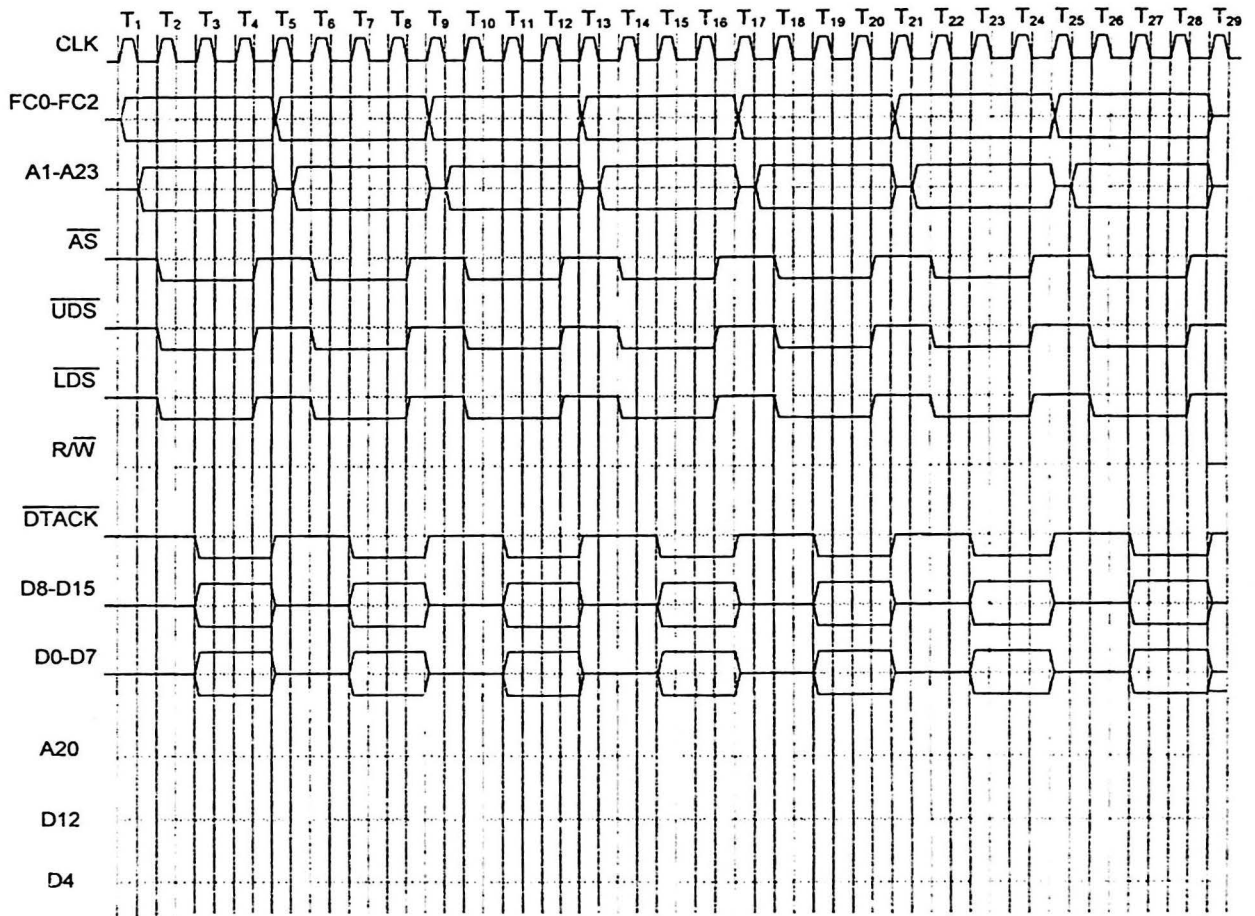
(kawasan berlorek () menandakan keadaan suis berkenaan)

(50/100)



Rajah 6 Barisan suis

Angka Giliran:



Rajah 7 Isyarat pegun di layar osiloskop

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	Data	- . U U U
CLR	CLR <ea>	8, 16, 32	Data Alterable		- 0 1 0 0
CMP	CMP <ea>,Dn	8, 16, 32	All (1)	Dn	-
CMPA	CMPA <ea>,An	16, 32	All	An	-
CMPI	CMPI #d,<ea>	8, 16, 32	#d	Data Alterable	-
CMPM	CMPM (Ay)+,(Ax)+	8, 16, 32	(An)+	(An)+	-

Mnemonic	Assembler Syntax	Operand Size	Allowable Addressing Modes		Condition Ccdes
			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 * * * * *
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 6, 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

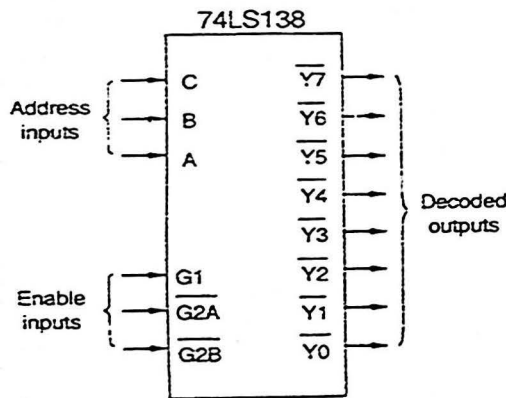
Mnemonic	Assembler Syntax	Operand Size	Allowable Addressing Modes		Condition Codes
			Source	Destination	X N Z V C
OR	OR <ea>, Dn OR Dn,<ea>	8, 16, 32 8, 16, 32	Data Dn	Dn Alterable	- * * 0 0 - * * 0 0
ORI	ORI #d,<ea> ORI #d,SR (3)	8, 16, 32 8, 16	#d #d	Data Alterable SR	- * * 0 0 * * * . .
PEA	PEA <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		- * * 0 0
TRAP	TRAP #<vector>		PC -> -(SP); SR -> -(SP); #<vector> -> PC		- - - - -
TRAPV	TRAPV		If V = 1, then TRAP		- - - - -
TST	TST <ea>	8, 16, 32	Data Alterable		- * * 0 0
UNLK	UNLK An	Unsize		An	- - - - -

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



G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	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.

Masa pelaksanaan beberapa arahan mikropemproses 68000

Table D-2. Move Byte and Move Word Instruction Execution Times.

Source	Destination								
	Dn	An	(An)	(An)+	-(An)	d16(An)	d8(An,Xn)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
(An)	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
(An)+	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
-(An)	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
d16(An)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d8(An,Xn)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
d16(PC)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
d8(PC,Xn)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#data	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

* The size of the index register (Xn) does not affect execution time.

Table D-3. Move Long Instruction Execution Times.

Source	Destination								
	Dn	An	(An)	(An)+	-(An)	d16(An)	d8(An,Xn)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	12(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
(An)	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
(An)+	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
-(An)	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
d16(An)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
d8(An,Xn)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
d16(PC)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
d8(PC,Xn)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	24(4/2)	28(5/2)
#data	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	30(5/2)	34(6/2)

* The size of the index register (Xn) does not affect execution time.

Table D-10. JMP, JSR, LEA, PEA, and MOVEM Instruction Execution Times.

Inst.	Size	(An)	(An)+	-(An)	d16(An)	d8(An,Xn)*	xxx.W	xxx.L	d16(PC)	d8(PC,Xn)*
JMP	-	8(2/0)	-	-	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	-	16(2/2)	-	-	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	-	4(1/0)	-	-	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	-	12(1/2)	-	-	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM	Word	12+4n	12+4n	-	16+4n	18+4n	16+4n	20+4n	16+4n	18+4n
M→R		(3+n/0)	(3+n/0)	-	(4+n/0)	(4+n/0)	(4+n/0)	(5+n/0)	(4n/0)	(4+n/0)
	Long	12+8n	12+8n	-	16+8n	18+8n	16+8n	20+8n	16+8n	18+8n
		(3+2n/0)	(3+2n/0)	-	(4+2n/0)	(4+2n/0)	(4+2n/0)	(5+2n/0)	(4+2n/0)	(4+2n/0)
MOVEM	Word	8+4n	-	8+4n	12+4n	14+4n	12+4n	16+4n	-	-
R→M		(2/n)	-	(2/n)	(3/n)	(3/n)	(3/n)	(4/n)	-	-
	Long	8+8n	-	8+8n	12+8n	14+8n	12+8n	16+8n	-	-
		(2/2n)	-	(2/2n)	(3/2n)	(3/2n)	(3/2n)	(4/2n)	-	-

n is the number of registers to move

* The size of the index register (Xn) does not affect the instruction's execution time

