

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Tambahan

Sidang 1986/1987

CSS101 - Pengantar Sistem Komputer

Tarikh: 26 Jun 1987

Masa: 9.00 pagi - 12.00 tgh.
(3 Jam)

Jawab semua soalan. Semua soalan mempunyai markah yang sama.

Peringatan: Semua nombor adalah dalam perlawanan melainkan dinyatakan sebaliknya.

Semua soalan mesti dijawab dalam Bahasa Malaysia.

1. (a) Nyatakan samada arahan-arahan MACRO-11 berikut sah atau tidak. Berikan penjelasan untuk menyokong jawapan anda.

- (i) TSTB 12346
- (ii) DEC @(RO)
- (iii) CMP #101, #JAD
- (iv) SUB SUM, #25
- (v) DIV 10(R2), R1

(20/100)

(b) Nyatakan perbezaan di antara (i) dan (ii) di bawah jika PET ialah simbol bagi lokasi memori di alamat 1254 dan isikandungan PET ialah 214.

- (i) MOV PET, RO
DEC (RO)+
DEC (RO)+
DEC RO
- (ii) MOV #PET, RO
DEC (RO)+
DEC (RO)+
DEC RO

(10/100)

(c) Nyatakan kesan daripada setiap arahan berikut. Anggapkan bahawa sebelum setiap arahan dilaksanakan, kandungan R1 ialah 123605.

- (i) NEG R1
- (ii) MOVB R1, R2
- (iii) ROL R1
- (iv) ASR R1
- (v) BIT #063721, R1

(20/100)

(d) Terangkan dengan terperinci apa yang dibuat oleh dua daripada arahan berikut:

- (i) JSR PC, SUBR
- (ii) JSR PC, @(SP)+
- (iii) RTS PC

(10/100)

(e) Aturcara di bawah bertujuan untuk menjumlahkan n nombor dalam satu jadual bermula di JAD dan meletakkan keputusannya dalam lokasi memori JUM.

```

.PSECT CODE
MOV #JAD, R1
GLG: ADD R1, JUM
ADD #2, R1
CMP R1, #JUM
BNE GLG
HALT
.PSECT DATA
JAD: .WORD n1, n2, ... , nn
JUM: .WORD

```

Jelaskan adakah aturcara di atas mencapai tujuannya? Nyatakan kelemahan-kelemahannya (jika ada) serta cadangkan cara untuk mengatasinya. Kemudian tuliskan kod aturcara yang anda cadangkan itu.

(20/100)

(f) Terangkan mod pengalamatan relatif dan bagaimana ia dilaksanakan dalam bahasa mesin dengan menunjukkan satu contoh.

(10/100)

(g) Terjemahkan arahan-arahan berikut ke dalam kod mesin PDP-11.

(i) MOV (RO)+, 12(R1)

(ii) CLRB @#365

(iii) ADD @-(R3), R5

(10/100)

2. (a) Dibawah ialah keratan kod aturcara bagi menunjukkan teknik siaran hujah lokasi tetap di mana nilai hujah diletakkan dalam daftar. DARAB ialah satu subrutin untuk mendarab dua nombor yang disiarkan dalam daftar R1 dan R2 masing-masingnya dan mengembalikan keputusannya dalam R3. Dalam tatatanda sisipan apa yang dilaksanakan oleh aturcara ini ialah $(C) \leftarrow (A) * (B)$.

```

MOV    A,R1
MOV    B,R2
JSR    PC,DARAB
MOV    R3,C
.....
.....
DARAB: CLR    R3
GLG:   DEC    R2
      BMI    TAMAT;
      ADD    R1,R3
      BR    GLG
TAMAT: RTS    PC
.....
.....
A:     .WORD a
B:     .WORD b
C:     .BLKW 1

```

} (R3) ← (R1) * (R2)

Tuliskan kod-kod aturcara yang sama dengan menggunakan teknik siaran hujah seperti berikut:

- (i) Teknik lokasi tetap dengan meletakkan alamat (bukan nilai) hujah dalam daftar
- (ii) Teknik lokasi berjujukan
- (iii) Teknik rujukan struktur

(60/100)

- (b) Terangkan secara menyeluruh apakah yang dilakukan oleh aturcara di bawah. Terangkan juga tentang bentuk input (data) dan outputnya.

```
.PSECT DATA
JAK: .ASCII / / ; satu rentetan aksara
AKHIR: .ASCII / / ; aksara penamat
INDEK: .WORD -1
DOT: .ASCII ./

.PSECT CODE
MULA: MOV #JAK,RO
      DEC RO
GELUNG: INC RO
        CMP RO,#AKHIR
        BEQ TAMAT
        CMPB @RO,DOT
        BNE GELUNG
        INC RO
        CMPB @RO,DOT
        BNE GELUNG
        DEC RO
        MOV RO,INDEK
TAMAT: HALT
      .END MULA
```

(40/100)

3. (a) Nyatakan komponen-komponen satu sistem komputer yang minimal dan terangkan peranan komponen-komponen tersebut.

(20/100)

- (b) Lukiskan satu gambarajah yang menunjukkan komponen-komponen dan perhubungan antara komponen di dalam satu sistem komputer yang minimal.

(20/100)

- (c) Kita mempunyai satu sistem komputer perkataan 5-bit (5-bit word). Terangkan secara ringkas sistem nombor bertanda dan tidak bertanda yang terdapat pada sistem tersebut. Nyatakan juga bit tandanya dan bit paling bererti.

(20/100)

- (d) Sistem komputer hanya dapat membuat pengiraan di dalam suatu julat yang tertentu. Jelaskan perkara ini dan apakah yang akan berlaku jika pengiraan melampaui batas ini.

(20/100)

- (e) Terangkan secara ringkas kitar pelaksanaan arahan (instruction execution cycle).

(20/100)

4. (a) Perkataan 16-bit 0010010101111111 berada di dalam lokasi ingatan yang beralamat 026442. Kedua-dua arahan berikut:

(i) tambah 1 kepada perkataan di alamat 026442

dan

(ii) tambah 1 kepada bait di alamat 026442

akan menghasilkan perkataan 0010010110000000.

Walaupun kedua-dua operasi ini nampak sama, tetapi ada sedikit perbezaan. Terangkan perbezaan tersebut.

(20/100)

- (b) Tuliskan satu makro yang akan menerima dua nilai iaitu M dan N. M ialah satu nilai integer positif. Manakala N pula bernilai 1 atau 0. Jika N bernilai 1 maka makro perlu mengira nilai ungkapan $M \times 2$. Tetapi jika N bernilai 0 maka makro perlu mengira nilai ungkapan $M/2$.

(30/100)

- (c) Di bawah ini ialah satu stek dengan nilai-nilai yang terkandung di dalamnya. R3 merupakan penunjuk stek.

4	← R3
172303	
12	
10422	

Nyatakan kandungan stek ini selepas pelaksanaan setiap arahan mengikut jujukan berikut:

- (i) MOV (R3)+,R5
- (ii) MOV #17,-(R3)
- (iii) NEG (R3)
- (iv) ADD 4(R3),(R3)
- (v) ADD (R3)+,2(R3)

(25/100)

- (d) Terangkan perbezaan di antara subrutin dan makro sepertimana ditakrifkan di dalam bahasa perhimpunan MACRO-11. (Di antara perkara yang anda harus bincangkan ialah dari segi takrif, aliran kawalan, kebaikan dan keburukannya).

(25/100)

...ooOoo...

(PDP-II INSTRUCTIONS)

DD = 6-bit code for destination

SS = 6-bit code for source

R = 3-bit code for register

X = 0 for word instructions, 1 for byte instructions

	<i>Mnemonic</i>	<i>Code</i>	<i>Description</i>	<i>Condition Codes Affected</i>
	CLR(B)	x050DD	clear dst	N,V,C clr; Z set
	DEC(B)	x053DD	subtract 1 from dst	N,Z,V set/clr by result
	INC(B)	x052DD	add 1 to dst	N,Z,V set/clr by result
	NEG(B)	x054DD	negate dst	N,Z,V set/clr by result
OPERAND	TST(B)	x057DD	set condition codes	C clr if result = 0, else set N,Z set/clr by dst contents V,C cleared
	COM(B)	x051DD	complement dst	N,Z set/clr by result, V clr, C set
	ASR(B)	x062DD	shift dst 1 place right replicate high-order bit	N,Z set/clr by result C ← old low-order bit of dst V ← exclusive OR of N, C bits
	ASL(B)	x063DD	shift dst 1 place left put 0 in low-order bit	N,Z, set/clr by result C ← old high-order bit of dst V ← exclusive OR of N,C bits
SINGLE	ADC(B)	x055DD	add C bit to dst	all set/clr by result
	SBC(B)	x056DD	subtract C bit from dst	all set/clr by result
	SXT	0067DD	all dst bits to value of N bit	Z set if N bit clr, V clr
	ROR(B)	x060DD	rotate dst right 1 bit, via C bit	N,Z set/clr by result C ← old low-order bit of dst V ← exclusive OR of N,C bits
	ROL(B)	x061DD	rotate dst left 1 bit, via C bit	N,Z set/clr by result C ← old high-order bit of dst V ← exclusive OR of N,C bits
	SWAB	0003DD	swap dst bytes	N,Z set/clr by old dst value V,C clr

BRANCHES: PC ← PC + (2 x offset). Coded as code + offset. Condition codes unaffected.

	<i>Mnemonic</i>	<i>Code</i>	<i>Condition</i>	<i>Mnemonic</i>	<i>Code</i>	<i>Condition</i>
BRANCHES	BR	000400	always			
	BEQ	001400	Z = 1	BNE	001000	Z = 0
	BMI	100400	N = 1	BPL	100000	N = 0
	BCS	103400	C = 1	BCC	103000	C = 0
	BVS	102400	V = 1	BVC	102000	V = 0
	BLT	002400	N ∨ V = 1	BGE	002000	N ∨ V = 0
	BLE	003400	Z ∨ (N ∨ V) = 1	BGT	003000	Z ∨ (N ∨ V) = 0
	BLOS	101400	C ∨ Z = 1	BHI	101000	C = 0 and Z = 0

	<i>Mnemonic</i>	<i>Code</i>	<i>Description</i>	<i>Condition Codes Affected</i>
DOUBLE OPERAND	MOV(B)	x1SSDD	move src to dst	N,Z set/clr by result, V clr
	ADD	06SSDD	add src to dst	all set/clr by result
	SUB	16SSDD	subtract src from dst	all set/clr by result
	CMP(B)	x2SSDD	form (src - dst)	all set/clr by result
	BIS(B)	x5SSDD	put (src OR dst) in dst	N,Z set/clr by result, V clr
	BIT(B)	x3SSDD	form (src AND dst)	N,Z set/clr by result, V clr
	BIC(B)	x4SSDD	put (~ src AND dst) in dst	N,Z set/clr by result, V clr
	MUL	070RSS	multiply } result in R, and	all set/clr by result
	DIV	071RSS	divide } next reg if R odd	all set/clr by result
	XOR	074RDD	exclusive OR, result in dst	N,Z set/clr by result, V clr
PROGRAM CONTROL	JSR	004RDD	reg → stack, PC → reg, dst → PC	
	RTS	00020R	reg → PC, stack → reg	
	SPL	00023L	set CPU priority level to L	
	JMP	0001DD	dst → PC	
	SOB	077RXX	XX = offset; subtract 1 from reg. contents, if ≠ 0 branch back	
	EMT	104000	PS, PC → stack,	all loaded from 32
		104377	new PC, PS from 30,32	
	TRAP	104400	PS, PC → stack,	all loaded from 36
		104777	new PC, PS from 34,36	
	BPT	000003	PS, PC → stack,	all loaded from 16
		new PC, PS from 14,16		
RTI	000002	load PC, PS from stack	all loaded from stack	
RTT	000006	PC, PS from stack, delay T trap	all loaded from stack	
MFPI	0065SS	move word from previous space to current stack	N,Z set/clr by src value, V clr	
MTPI	0066DD	move word from current stack to previous space	N,Z set/clr by result V clr	
CPU	HALT	000000	stop CPU	
	WAIT	000001	wait for interrupt	
	RESET	000005	reset UNIBUS	

	<i>Mnemonic</i>	<i>Code</i>	<i>Description</i>	<i>Mnemonic</i>	<i>Code</i>	<i>Description</i>
CONDN. CODE OPS	CLC	000241	clear C bit	SEC	000261	set C bit
	CLV	000242	clear V bit	SEV	000262	set V bit
	CLZ	000244	clear Z bit	SEZ	000264	set Z bit
	CLN	000250	clear N bit	SEN	000270	set N bit
	CCC	000257	clear all codes	SCC	000277	set all codes
	ASH	072RSS	SHIFT ARIT. NN PLACES. (SS) = NN - 32 ≤ NN ≤ + 31 -ve ⇒ right			

ASCII Control Characters

0	NUL	Null	20	DLE	Data link escape
1	SOH	Start of heading	21	DC1	Device control 1
2	STX	Start of text	22	DC2	Device control 2
3	ETX	End of text	23	DC3	Device control 3
4	EOT	End of transmission	24	DC4	Device control 4
5	ENQ	Enquiry	25	NAK	Negative acknowledge
6	ACK	Acknowledge	26	SYN	Synchronous idle
7	BEL	Bell	27	ETB	End of transmission block
10	BS	Backspace	30	CAN	Cancel
11	HT	Horizontal tab	31	EM	End of medium
12	LF	Line feed	32	SUB	Substitute
13	VT	Vertical tab	33	ESC	Escape
14	FF	Form feed	34	FS	File separator
15	CR	Carriage return	35	GS	Group separator
16	SO	Shift out	36	RS	Record separator
17	SI	Shift in	37	US	Unit separator

ASCII Character Code (octal)

40 (Space)	60	0	100	@	120	P	140		160	p
41	61	1	101	A	121	Q	141	a	161	q
42	62	2	102	B	122	R	142	b	162	r
43	63	3	103	C	123	S	143	c	163	s
44	64	4	104	D	124	T	144	d	164	t
45	65	5	105	E	125	U	145	e	165	u
46	66	6	106	F	126	V	146	f	166	v
47	67	7	107	G	127	W	147	g	167	w
50	70	8	110	H	130	X	150	h	170	x
51	71	9	111	I	131	V	151	i	171	y
52	72	:	112	J	132	Z	152	j	172	z
53	73	;	113	K	133	I	153	k	173	{
54	74	<	114	L	134	\	154	l	174	
55	75	=	115	M	135]	155	m	175	}
56	76	>	116	N	136	^	156	n	176	~
57	77	?	117	O	137	_	157	o	177 (Delete)	