



**FINAL EXAMINATION / PEPERIKSAAN AKHIR  
SEMESTER II – SESSION 2017 / 2018**

COURSE CODE : DDWC 1223  
KOD KURSUS

COURSE NAME : COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE  
NAMA KURSUS : ORGANISASI KOMPUTER & BAHASA HIMPUNAN

YEAR / PROGRAMME : 1 DDWC / 1 DDWZ  
TAHUN / PROGRAM

DURATION : 2 HOURS 30 MINUTES  
TEMPOH

DATE : APRIL 2018

TARIKH

INSTRUCTION/ARAHAN :

Answer **ALL** questions in the spaces provided in this question paper.

Jawab **SEMUA** soalan di ruang yang disediakan dalam kertas soalan ini.

(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	:	.....
SECTION SEKSYEN	:	.....
LECTURER'S NAME NAMA PENSYARAH	:	.....

This examination paper consists of ... 19... pages including the cover  
Kertas soalan ini mengandungi ..... 19..... muka surat termasuk kulit hadapan



**PUSAT PROGRAM KERJASAMA**

**PETIKAN DARIPADA PERATURAN AKADEMIK  
ARAHAN AM - PENYELEWENGAN AKADEMIK**

**1. SALAH LAKU SEMASA PEPERIKSAAN**

1.1 Pelajar tidak boleh melakukan mana-mana salah laku peperiksaan seperti berikut :-

- 1.1.1 memberi dan/atau menerima dan/atau memiliki sebarang maklumat dalam bentuk elektronik, bercetak atau apa jua bentuk lain yang tidak dibenarkan semasa berlangsungnya peperiksaan sama ada di dalam atau di luar Dewan Peperiksaan melainkan dengan kebenaran Ketua Pengawas; atau
- 1.1.2 menggunakan makluman yang diperolehi seperti di atas bagi tujuan menjawab soalan peperiksaan; atau
- 1.1.3 menipu atau cuba untuk menipu atau berkelakuan mengikut cara yang boleh ditafsirkan sebagai menipu semasa berlangsungnya peperiksaan; atau
- 1.1.4 lain-lain salah laku yang ditetapkan oleh Universiti (seperti membuat bising, mengganggu pelajar lain, mengganggu Pengawas menjalankan tugasnya).

**2. HUKUMAN SALAH LAKU PEPERIKSAAN**

2.1 Sekiranya pelajar didapati telah melakukan pelanggaran mana-mana peraturan peperiksaan ini, setelah diperakukan oleh Jawatankuasa Peperiksaan Fakulti dan disabitkan kesalahannya, Senat boleh mengambil tindakan dari mana-mana satu yang berikut :-

- 2.1.1 memberi markah SIFAR (0) bagi keseluruhan keputusan peperiksaan kursus yang berkenaan (termasuk kerja kursus); atau
- 2.1.2 memberi markah SIFAR (0) bagi semua kursus yang didaftarkan pada semester tersebut.

2.2 Jawatankuasa Akademik Fakulti boleh mencadangkan untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999 bergantung kepada tahap kesalahan yang dilakukan oleh pelajar.

2.3 Pelajar yang didapati melakukan kesalahan kali kedua akan diambil tindakan seperti di perkara 2.1.2 dan dicadang untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999.

**SECTION A / BAHAGIAN A**  
**21 MARKS / 21 MARKAH**

**MULTIPLE CHOICE / ANEKA PILIHAN**

Choose the most appropriate answer. Write your answer in the table provided on page 7.  
*Pilih satu jawapan yang paling tepat. Tulis jawapan anda pada jadual di mukasurat 7.*

1. Which of the following is lowest in memory hierarchy?  
*Manakah antara berikut merupakan yang terendah dalam hirarki ingatan?*

A. Cache memory	/	Ingatan cache
B. Secondary storage	/	Storan sekunder
C. Registers	/	Daftar
D. RAM	/	RAM
  
2. CPU fetches the instruction from memory according to the value of \_\_\_\_\_.  
*CPU mengambil arahan dari memori mengikut nilai \_\_\_\_\_.*

A. instruction pointer	/	penunjuk arahan
B. status register	/	daftar status
C. instruction register	/	daftar arahan
D. program status word	/	perkataan status aturcara
  
3. A CPU has 16 bit program counter (PC). This means that the CPU can address \_\_\_\_\_.  
*Satu CPU mempunyai pembilang aturcara 16 bit (PC). Ini bermakna CPU ini boleh mengalamatkan \_\_\_\_\_.*

A. 16K memory locations	/	16K lokasi ingatan
B. 32K memory locations	/	32K lokasi ingatan
C. 64K memory locations	/	64K lokasi ingatan
D. 256K memory locations	/	256K lokasi ingatan
  
4. The assembler stores the object code in \_\_\_\_\_.  
*Penghimpun menyimpan kod objek di dalam \_\_\_\_\_.*

A. main memory	/	ingatan utama
B. cache	/	cache
C. RAM	/	RAM
D. hard disk	/	cakera keras



5. The Pentium microprocessor has \_\_\_\_\_ execution units.  
*Pemproses mikro Pentium mempunyai \_\_\_\_\_ unit pelaksanaan.*
- A. 1
  - B. 2
  - C. 3
  - D. 4
6. When CPU perform subtraction on -7 and -5 the answer in 2's compliment form is \_\_\_\_\_.  
*Apabila CPU melakukan penolakan ke atas -7 dan -5 jawapannya dalam pelengkap-2 ialah \_\_\_\_\_.*
- A. 11110
  - B. 1110
  - C. 1010
  - D. 0010.
7. Which directives below identify the memory configuration?  
*Mana direktif berikut yang menentukan konfigurasi ingatan utama?*
- A. .686
  - B. .CODE directive
  - C. .MODEL directive
  - D. .DATA
8. In Intel IA-32 microprocessor, the value of the most significant bit of the result following the execution of any arithmetic or Boolean instruction is stored in the \_\_\_\_\_.  
*Dalam Intel IA-32 mikropemproses, nilai untuk bit paling bermakna bagi hasil selepas dari pelaksanaan sebarang arahan aritmetik atau Boolean disimpan dalam \_\_\_\_\_.*
- |                         |   |                       |
|-------------------------|---|-----------------------|
| A. carry flag           | / | bendera bawa          |
| B. auxiliary carry flag | / | bendera bawa tambahan |
| C. sign flag            | / | bendera tanda         |
| D. zero flag            | / | bendera sifar         |

9. Choose **illegal** instruction based on the following data definition:  
*Pilih arahan yang tidak sah berdasarkan pentakrifan data berikut:*

```
.DATA  
arrayW WORD 1020h,3040h, 5060h  
arrayD DWORD 1,2,3,4
```

- A. mov ax, arrayW+2  
B. mov ax, arrayW [4]  
C. mov ax, arrayD+4  
D. mov eax, arrayD
10. Four contiguous bytes in memory contain the hexadecimal values 12, 34, 56 and 78 in the order of increasing address. If these bytes are used to store a 32-bit integer in **little-endian** format, what is the integer? (Give your answer in hexadecimal.)  
*Empat bait berdampingan dalam memori mengandungi nilai perenambelasan 12, 34, 56 dan 78 dalam turutan peningkatan alamat. Sekiranya bait ini digunakan untuk menyimpan integer 32-bit dalam format kecil-endian, apakah integer tersebut? (Berikan jawapan anda dalam nombor perenambelasan.)*

- A. 1234h  
B. 12345678h  
C. 78563412h  
D. 3412h

11. Choose the **incorrect** statement regarding real-address mode.  
*Pilih pernyataan yang salah mengenai mod alamat-nyata.*

- A. Only 1 MB of memory can be addressed and the address ranges from 0000 to FFFFF (hex)  
*Hanya 1 MB ingatan boleh dialamatkan dan julat alamat daripada 0000 hingga FFFFF (hex)*
- B. In 8086 processor, a program can access up to six segments at any time and the segment size is 64KB.  
*Dalam pemproses 8086, satu aturcara boleh mencapai sehingga enam segmen pada bila-bila masa dan saiz satu segmen adalah 64KB.*
- C. The logical address consists of a segment value and an offset.  
*Alamat logikal mengandungi nilai segmen dan jarak.*
- D. CS, DS and SS are the only registers that can be used to hold the value of the segment.  
*Hanya daftar CS, DS dan SS yang boleh digunakan untuk menyimpam nilai segmen.*

Trace the following program fragment and answer question 12 and 13,

```
.data
    array word 7,6,5,4
    count word 4
.code
    xor eax,eax
    xor ecx,ecx
    stc
    mov cx,count
    mov esi,offset array
label1:  adc ax,word ptr [esi]
        add esi,2
        loop label1
label2:
```

12. What will be the value in AX when control reaches label2?  
*Apa nilai dalam AX apabila kawalan sampai ke label2?*

- A. 0016h
- B. 0017h
- C. 0022h
- D. 0023h

13. How many times will the loop statement be executed?  
*Berapa kali penyata gelung akan dilaksanakan?*

- A. 8
- B. 6
- C. 4
- D. 2

14. Identify the addressing mode of the destination operand for the instruction: **CMP WORD PTR [BX+DI],10**  
*Kenal pasti mod pengalamatan bagi operan destinasi dalam arahan: **CMP WORD PTR [BX+DI],10***

- A. BASE+INDEX
- B. IMMEDIATE
- C. INDEX + DISP
- D. BASE + DISP



**SECTION B/ BAHAGIAN B**

**79 marks / 79 markah**

**ANSWER ALL QUESTIONS. ANSWER IN THE PROVIDED SPACES IN THIS EXAM PAPER.**

**JAWAB SEMUA SOALAN. JAWAB PADA RUANG YANG DISEDIAKAN DALAM KERTAS SOALAN INI.**

- Q1. a) Given two decimal numbers  $a = -105$  and  $b = 28$ . Change them into 8-bit binary numbers and then perform  $a - b$ . Determine whether there is an overflow. Show your working. Use 8-bit binary representation.

[3 M]

*Diberi dua nombor perpuluhan  $a = 105$  dan  $b = 28$ , Tukarkan kepada nombor perduaan 8-bit dan kemudian lakukan  $a - b$ . Tentukan sama ada terdapat limpahan. Tunjukkan jalan kerja anda.*

- b) Explain the concept of a stored program computer . List the names of the basic components of the computer that employed this concept

[5 M]

*Terangkan konsep komputer aturcara tersimpan. Senaraikan nama-nama komponen asas komputer yang menggunakan konsep ini.*

- c) What is the name of the computer's component that executes program instructions and what are its main components.

[2 M]

*Apakah nama komponen komputer yang melaksanakan program arahan dan apakah komponennya yang utama.*



- Q2. a) Suppose that a bus has 16 data lines and requires 4 cycles of 250 ns each to transfer data. The bandwidth of this bus is 2 Megabytes/sec. If the cycle time of the bus was reduced to 125 ns and the number of cycles required for transfer stayed the same, what would the bandwidth of the bus be?

[3 M]

*Katakan bahawa satu bus ada 16 talian data dan memerlukan 4 kitaran 250 ns setiap satu untuk memindahkan data. Jalur-lebar bus ini adalah 2 Megabait/sec. Jika masa kitaran bus telah dikurangkan kepada 125 ns dan bilangan kitaran yang diperlukan untuk pemindahan kekal sama, apakah nilai jalur-lebar bus?*

- b) Computer AXY has 32MB of main memory. How many bits are needed to address any single byte of its memory?

[2 M]

*Komputer AXY mempunyai 32MB ingatan utama. Berapa bit yang diperlukan untuk mengalamatkan setiap bait ingatannya?*

- c) When the CPU needs data from the memory, the CPU will perform Memory Read Cycle. Write steps (in sequence) involved in a Memory Read Cycle.

[5 M]

*Apabila CPU memerlukan data dari ingatan, CPU akan melakukan Kitar Baca Ingatan. Tuliskan langkah-langkah (mengikut turutan) yang terlibat dalam Kitar Baca Ingatan.*

- Q3. a) What is the main function of I/O interface module? [1 M]  
*Apakah fungsi utama modul antaramuka I/O?*
- b) List steps involve in I/O transfer via DMA technique. [4 M]  
*Terangkan langkah-langkah dalam pemindahan I/O melalui teknik DMA.*
- c) What is the average time to read or write a **1024-byte** sector for a typical disk rotating at **8000 RPM**? Assuming the advertised average seek time is **8 ms**, the transfer rate is **20 MB/sec**. [3 M]  
*Apakah masa purata untuk baca atau tulis satu sektor bersaiz **1024-bait** untuk satu cakera biasa yang berputar pada **8000 RPM**? Andaikan purata masa cari adalah **8 ms**, dan kadar perpindahan adalah **20 MB/sec**.*

Q4. a) Give the data declaration statements for the following:

*Memberi penyata deklarasasi data bagi perkara-perkara berikut:*

- i) Declare a 16-bit signed integer variable and initialize it with the smallest possible negative decimal value. [2 M]

*Mengisytiharkan pembolehubah 16-bit integer bertanda dan diberi nilai awalan dengan nilai perpuluhan negatif yang terkecil.*

- ii) Declare a string variable containing "Good Luck" repeated 30 times, and terminated with the null character. [2 M]

*Mengisytiharkan rentetan pembolehubah mengandungi "Good Luck" yang diulang 30 kali, dan diakhiri dengan null.*

b) Indicate the content of register AL, DL and CF (in hex value) after the execution of the following program fragment.

*Tunjukkan kandungan daftar AL, DL dan CF (dalam nilai heksa) selepas pelaksanaan keratan aturcara berikut:*

```
MOV    DX, -80
SAR    DL, 1
MOV    AL, 6Bh
SHR    AL, 1
SHL    AL, 3
```

AL = \_\_\_\_\_ [2 M]

DL = \_\_\_\_\_ [2 M]

CF = \_\_\_\_\_ [1 M]

Q5. Given the following definitions:

*Diberi definisi data seperti berikut:*

```
.DATA
wval      LABEL WORD
barray    BYTE 10h, 20h, 30h, 6 DUP (0Ah)
ALIGN 4
warray    WORD 5 DUP(1000h)
pressKey  EQU  <"Press any key to continue ...",0>
darray    DWORD 5 DUP(56789ABh),
            7 DUP(12345678h)
dval      LABEL DWORD
prompt    BYTE  pressKey
```

What will be the value of EAX, AX, and AL after executing each of the following instructions? Assume that the address of barray is 404000h.

*Apakah nilai EAX, AX, dan AL selepas melaksanakan setiap arahan berikut? Anggap bahawa alamat barray adalah 404000h.*

- a) mov eax, TYPE warray            eax = \_\_\_\_\_
- b) mov eax, LENGTHOF barray        eax = \_\_\_\_\_
- c) mov eax, SIZEOF darray          eax = \_\_\_\_\_
- d) mov eax, OFFSET warray          eax = \_\_\_\_\_
- e) mov eax, OFFSET darray          eax = \_\_\_\_\_
- f) mov eax, OFFSET prompt          eax = \_\_\_\_\_
- g) mov eax, DWORD PTR barray        eax = \_\_\_\_\_
- h) mov al, BYTE PTR darray          al = \_\_\_\_\_
- i) mov ax, wval                    ax = \_\_\_\_\_
- j) mov eax, dval                    eax = \_\_\_\_\_





Q7. Trace the following program:

*Jejak aturcara berikut:*

```
TITLE Summing an Array                (SumArray.asm)
; This program sums an array of words.

INCLUDE Irvine32.inc .....[1]
.data
intarray DWORD 10000h,20000h,30000h,40000h
sum      DWORD ?
.code
main PROC

mov  edi,OFFSET intarray                .....[2]
mov  ecx,LENGTHOF intarray              .....[3]
mov  eax,0
L1:
add  eax,[edi]
add  edi,TYPE intarray                  .....[4]
loop L1                                  .....[5]
mov  sum, eax
exit
main ENDP
END main
```

Assume the program will be loaded into starting address 405000H for execution,

*Andaikan aturcara akan dimuatkan ke alamat 405000H untuk pelaksanaan.*

a) What is the use of **INCLUDE** directive in instruction labeled [1] in the program above? [2 M]  
*Apakah kegunaan direktif **INCLUDE** pada arahan bertlabel [1] dalam program di atas?*

b) What is the content of register **edi** after the execution of instruction [2]? [2 M]  
*Apa kah kandungan dafter **edi** selepas pelaksanaan arahan [2]?*

- c) What is the content of **eax** after the execution of instruction labeled [3]? [1 M]  
*Apakah kandungan **eax** selepas pelaksanaan arahan bertanda [3]?*

eax = \_\_\_\_\_

- d) How many times the **Loop** instruction [labeled 5] be executed? [1 M]  
*Berapa kalikah arahan **Loop** [label 5] akan dilaksanakan?*

\_\_\_\_\_

- e) What will be the content of **sum** after the program completes execution? [1 M]  
*Apakah kandungan **sum** selepas aturcara selesai dilaksanakan?*

sum = \_\_\_\_\_

- Q8. Given students' test scores as follows: 88, 67, 54, 78, 100, 65 and 45. Use an array named **MARKS** to store all these test scores. Write an assembly program to:

*Diberi skor ujian seperti berikut: 88, 67, 54, 78, 100, 65 dan 45. Guna tatasusunan bernama **MARKS** untuk menyimpan semua skor ujian ini. Tulis aturcara bahasa himpunan untuk:*

- i. find the average test score and put the result in **AVERAGE**  
*mencari purata skor ujian dan letak hasil di dalam **AVERAGE***
- ii. find the highest test score and put it in **HIGHEST**  
*mencari skor ujian tertinggi dan simpan hasil dalam **HIGHEST***
- iii. display the output as follows:  
*paparkan keluaran seperti berikut:*

-----  
**Average test score is: xx**  
**Highest test score is: xx**  
-----

Note: Value **xx** are in decimal. *Nilai xx adalah dalam desimal.*

Your program **SHOULD** use **CMP** and Conditional Jumps & Loops for example **JGE, JNE, JZ, LOOP** and etc. Refer **Appendix B** for **Irvine32.lib** for calling procedures and functions.

*Aturcara **HARUS** menggunakan **CMP**, lompatan bersyarat & gelung contohnya **JGE, JNE, JZ, LOOP** dan lain-lain. Rujuk **Lampiran B** untuk **Irvine32.lib** untuk panggilan prosidur dan fungsi.*

[12 M]

Intentionally left blank - space for answering Q8

*Dengan sengaja dibiarkan kosong - ruang untuk menjawab Q8*



APPENDIX A  
ASCII TABLE / JADUAL ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

**APPENDIX B**  
**Link library Irvine32.lib**

**1. Dumping Registers and Memory**

<b>Procedure</b>	<b>Description</b>
DumpRegs	Writes EAX, EBX, ECX, and EDX on first line in hexadecimal Writes ESI, EDI, EBP, and ESP on second line in hexadecimal Writes EIP, EFLAGS, CF, SF, ZF, and OF on third line
DumpMem	Writes a range of memory to standard output in hexadecimal ESI = starting address ECX = number of elements to write EBX = element size (1, 2, or 4)

**2. Input procedures: ReadInt, ReadChar, ReadString, Dec**

<b>Procedure</b>	<b>Description</b>
ReadChar	Reads a char from keyboard and returns it in the AL register. The character is NOT echoed on the screen.
ReadHex	Reads a 32-bit hex integer and returns it in the EAX register. Reading stops when the user presses the [Enter] key. No leading spaces. No error checking is performed.
ReadInt	Reads a 32-bit signed integer and returns it in EAX. Leading spaces are ignored. Optional + or - is allowed. Error checking is performed (error message) for invalid input.
ReadDec	Reads a 32-bit unsigned integer and returns it in EAX.
ReadString	Reads a string of characters from keyboard. Additional null-character is inserted at the end of the string. EDX = address of array where input characters are stored. ECX = maximum characters to be read + 1 (for null byte) Return EAX = count of non-null characters read.

3. Output procedures: **Clrscr**, **WriteInt**, **WriteHex**, **WriteString**, **WriteHex**, **WriteInt**, **WriteDec**, **WriteBin**

Procedure	Description
Clrscr	Clears screen, locates cursor at upper left corner.
Crlf	Writes end of line sequence (CR,LF) to standard output.
WriteChar	Writes character in register AL to standard output.
WriteString	Writes a null-terminated string to standard output. String address should be passed in register EDI.
WriteHex	Writes EAX in hexadecimal format to standard output.
WriteInt	Writes EAX in signed decimal format to standard output.
WriteDec	Writes EAX in unsigned decimal format to standard output.
WriteBin	Writes EAX in binary format to standard output.

END OF QUESTIONS / SOALAN TAMAT

**Mukasurat ini sengaja dibiarkan kosong**

*[ This page is purposely left blank ]*