

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama  
Sidang Akademik 1996/97

Oktober/November 1996

**CSY301 - Mikropemproses dan Perantaramukaan**

Masa: [3 jam]

---

**ARAHAN KEPADA CALON:**

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **TUJUH** soalan di dalam **EMPAT** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
  - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. Jawab semua soalan di bawah:

- Berapakah jenis sampukan (interrupts) yang terdapat dalam pemproses 8086?
- Namakan semua sampukan tersebut.
- Apakah jenis dan nombor sampukan limpah atas pembahagian (divide overflow)?
- Tulis satu contoh kod dalam bahasa pemrograman himpunan yang akan menyebabkan sampukan limpah atas pembahagian (divide overflow). (15%)

2. Tuliskan kod himpunan untuk menggerakkan 10 bait data nombor integer tidak bertanda (unsigned integer) dari lokasi bermula dengan 01AC0 ke lokasi bermula dengan A1000. Data yang disimpan ke lokasi baru mestilah dua kali ganda data permulaan. Kalau saiz data yang diganda melebihi saiz satu bait (contoh FF x 2), maka lokasi baru itu akan diisi dengan 0. Gunakan penyataan LOOP dan program anda mestilah sekecil mungkin. Anda juga dinasihati menggunakan pengalamatan tak terus daftar (register indirect addressing). (15%)

3. Sampukan 20 (sampukan jenis 20) adalah suatu sampuk (interrupt) yang mengambil nilai dalam AL dan menjanakan nilai  $2^{AL}$  berkenaan. Jawapannya disimpan dalam AX (nilai rendah) dan BX (nilai tinggi). Program sampukan tersebut berada di bahagian memori yang bermula dengan alamat 3C502H.
- Isikan kandungan sampukan ini di dalam jadual vektor di bawah. (Jangan tulis program, hanya alamat sahaja). Jawapan mesti dalam bentuk perenambelasan (hexadecimal).

Jadual Vektor (Vector Table)

Alamat Memori (Address)	Kandungan (8 bit)

- Apabila sampukan ini berlaku, apakah yang akan terjadi kepada tindanan (stack) [iaitu, adakah tindanan (stack) berubah, dan jikalau berubah, nilai apakah yang diisi ke dalamnya]?
- Lukis carta alir kod sampukan tersebut. (Pengistiharan dan program utama tidak perlu di buat.)
- Tulis program himpunan untuk sampukan di atas. (20%)

...3/-

4. Tulis kod yang akan menginput data dari port I/O beralamat A6H. Kalau data ini mempunyai nilai lebih daripada 0A, paparkan mesej berikut ke skrin.

"Input lebih daripada 0A"

Selepas ini program akan tamat. Kalau nilainya sama ataupun kurang daripada 0A, maka simpan data ini di lokasi memori B12AC. (Anggap DS=B010).

(10%)

5. Jawab semua soalan di bawah:

- Berapakah segmen yang terdapat dalam memori 8086 dan namakan segmen-segmen itu.
- Apakah saiz setiap segmen itu? Apakah saiz minimum dan saiz maksimum yang dapat diliputi oleh semua segmen itu pada sesuatu masa?
- Tukarkan "ADD AX, 120AH" ke kod perduaan. Gunakan carta yang diberi di mukasurat 4.

(15%)

6. Jawab semua soalan di bawah:

- Berapakah bilangan kitaran bus (bus cycles) yang diperlukan untuk pencapaian 16 bit dari memori:
  - Jika alamat memori itu genap?
  - Jika alamat memori itu ganjil?
- Apakah saiz memori yang dapat dicapai oleh pemproses 8086? Sila terangkan jawapan anda dengan menggunakan saiz bus alamat. Tunjukkan semua pengiraan anda.

(10%)

7. Jawab semua soalan di bawah:

- Apakah saiz giliran suruhan (instruction queue) 8086?
- Apakah yang berlaku ke giliran suruhan apabila arahan JMP dilaksanakan?
- Secara ringkas, terangkan apa yang akan berlaku kepada IP, CS dan kandungan SS apabila kod "call <intrasegment>" dilaksanakan.
- Apakah perbezaan antara lompatan antara segmen (intersegment jump) dan lompatan dalam segmen (intrasegment jump).
- Terangkan fungsi BIU (Bus Interface Unit) dengan ringkas.

(15%)

REG	W = 0	W = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

**ARITHMETIC**

**ADD = Add:**

Reg/memory with register to either

0 0 0 0 0 0 d w	mod reg r/m	(DISP-LO)	(DISP-HI)		
1 0 0 0 0 0 s w	mod 0 0 0 r/m	(DISP-LO)	(DISP-HI)	data	data if s: w=01
0 0 0 0 0 1 0 w		data	data if w=1		

Immediate to register/memory

Immediate to accumulator

**ADC = Add with carry:**

Reg/memory with register to either

0 0 0 1 0 0 d w	mod reg r/m	(DISP-LO)	(DISP-HI)		
1 0 0 0 0 0 s w	mod 0 1 0 r/m	(DISP-LO)	(DISP-HI)	data	data if s: w=01
0 0 0 1 0 1 0 w		data	data if w=1		

Immediate to register/memory

Immediate to accumulator

**INC = Increment:**

Register/memory

1 1 1 1 1 1 1 w	mod 0 0 0 r/m	(DISP-LO)	(DISP-HI)
-----------------	---------------	-----------	-----------

Register

0 1 0 0 0 reg
---------------

AAA = ASCII adjust for add

0 0 1 1 0 1 1 1
-----------------

DAA = Decimal adjust for add

0 0 1 0 0 1 1 1
-----------------

**Converting Source Programs to Machine Code**