

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang 1989/90

Mac/April 1990

EET 410 - Pemprosesan Isyarat Digit

Masa : [3 jam]

ARAHAN KEPADA CALON:

Sila pastikan bahawa kertas peperiksaan ini mengandungi 6 muka surat beserta Lampiran (2 muka surat) bercetak dan LIMA (5) soalan sebelum anda memulakan peperiksaan ini.

Jawab EMPAT (4) soalan.

Agihan markah bagi setiap soalan diberikan di sisi sebelah kanan sebagai peratusan daripada markah keseluruhan yang diperuntukkan bagi soalan berkenaan.

Jawab kesemua soalan dalam Bahasa Malaysia.

1. (a) Huraikan 4 kelebihan penuras digit.

(5%)

- (b) Fungsi pindah bagi suatu penuras IIR tertib 2 diberikan seperti berikut:

$$H(Z) = \frac{1 + 2Z^{-1} + Z^2}{1 - 0.75Z^{-1} + 0.125Z^{-2}}$$

Lakarkan struktur "Direct Form I" dan "Direct Form II" bagi penuras tersebut.

(20%)

- (c) Berikan persamaan beza bagi output $y(n)$ untuk kedua-dua struktur di atas.

(20%)

- (d) Tulis aturcara bagi TMS32010 untuk melaksanakan penuras "Direct Form II" tersebut.

(30%)

- (e) Beri anggaran frekuensi persampelan maksimum yang dapat dilaksanakan, jika TMS32010 dikendalikan pada kadar 20MHz.

(25%)

2. (a) Huraikan teknik-teknik rekabentuk penuras digit IIR berikut:

- (i) Dedenyut tak-varian.
(ii) Jelmaan z-dwilelurus.

(40%)

- (b) Dengan menggunakan teknik jelmaan z-dwilelurus dapatkan pekali-pekali bagi penuras laluan rendah yang dapat memenuhi spesifikasi berikut:
- (i) Frekuensi (digit) potong jalur laluan = 0.5π rad
 - (ii) Attenuasi - 15dB pada frekuensi (digit) 0.7π rad
 - (iii) Sambutan monoton
- (Andaian: $T = 1$ saat)

(30%)

- (c) Dapatkan fungsi $H(z)$ menerusi teknik dedenut tak-varian bagi fungsi analog berikut:

$$H(s) = \frac{2s}{(s+1)(s+3)}$$

(30%)

3. (a) Rajah dalam lampiran 1 menunjukkan proses jelmaan Fourier diskrit secara grafiks. Lengkapkan bahagian (b) - (g).

(30%)

- (b) Jujukan $x(n)$ dan $h(n)$ ditakrifkan seperti berikut:

$$x(n) = \cos(\pi n/2), \quad n = 0, 1, 2, 3$$

$$h(n) = 2^n, \quad n = 0, 1, 2, 3$$

- (i) Hitungkan DFT 4 - titik $X(k)$.

(20%)

- (ii) Hitungkan DFT 4 - titik $H(k)$.

(20%)

- (iii) Dapatkan hasil pelingkaran pekeliling di antara $x(n)$ dengan $h(n)$, menerusi domain frekuensi.

(30%)

4. (a) Terbitkan persamaan "butterfly" asas bagi algoritma "Decimation-in-Time" (radiks - 2) yang dapat mempercepatkan pengiraan jelmaan Fourier diskrit.

(20%)

- (b) Lakarkan graf alir isyarat bagi algoritma tersebut, untuk jelmaan 8 - titik.

(15%)

- (c) Merujuk kepada graf alir isyarat di atas:

- (i) Ada berapakah laluan di antara suatu masukan $x(n)$ dan suatu keluaran $X(k)$? (contoh: di antara $x(7)$ dengan $X(2)$ dan sebagainya).

(15%)

- (ii) Dengan mengesan laluan dalam graf alir di atas tunjukkan bahawa setiap sampel input menyumbangkan nilai yang tertentu untuk menghasilkan $X(2)$, iaitu buktikan bahawa:

$$X(2) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)2n}$$

(30%)

- (d) Spesifikasi bagi 2 operasi aritmetik penting untuk 2 jenis pemikroproses adalah seperti berikut:

Operasi (nyata)	MC68000	TMS320C25
Darab	7μs	0.1μs
Tambah	1μs	0.1μs

Beri anggaran kasar berapa kali gandakah kepentasan TMS320C25 jika dibandingkan dengan MC68000, apabila mengira FFT 2048-titik ("Decimation-in-time")

(20%)

5. (a) Jujukan $x(n)$ terdiri dari L sampel manakala $h(n)$ mengandungi M sampel. Apakah kekangan yang mesti dikenakan kepada L dan M , supaya hasil pelingkaran pekeliling adalah sama dengan pelingkaran lelurus, apabila pengiraan dijalankan menerusi DFT titik- N .

(5%)

- (b) Dengan bantuan gambarajah,uraikan teknik "overlap-add".

(30%)

(c) Suatu jujukan yang terdiri dari 10000 sampel perlu dilingkarkan (secara lelurus) dengan sambutan dedenyut FIR 100-titik. Operasi ini hendak dilaksanakan dengan menggunakan DFT 256-titik.

(i) Jika kaedah "overlap-add" digunakan, berapakah jumlah operasi DFT dan DFT sonsang 256-titik yang diperlukan.

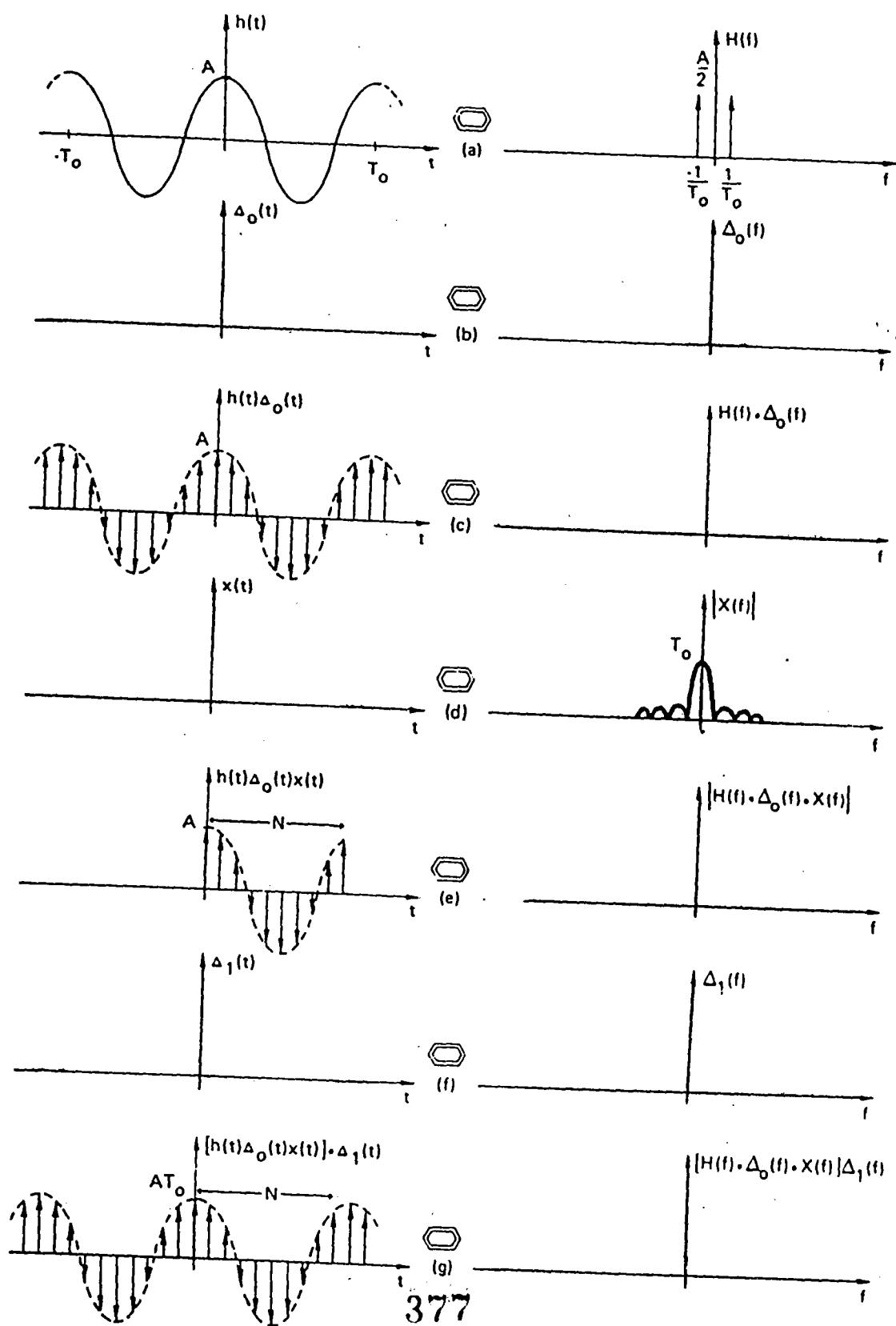
(20%)

(ii) Sebaliknya, jika "overlap-save" digunakan, berapakah jumlah DFT dan DFT sonsang 256-titik yang diperlukan.

(20%)

(iii) Andaikan pengiraan DFT dijalankan menerusi algoritma FFT ("decimation-in-time"), bandingkan jumlah operasi aritmetik (iaitu pendaraban dan penambahan) yang diperlukan oleh kaedah "overlap-add", "overlap-save" dan pelingkaran terus (domain masa), untuk melengkapkan proses penurasan tersebut.

(25%)



AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
LAR	Load auxiliary register	1	1	0	0	1	1	1	1	0	R	I	←	D	→				
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	R	←	K	→						
LARP	Load auxiliary register immediate page pointer	1	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	←	D	→					
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	K	X
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	I	←	D	→					
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	R	I	←	D	→					

BRANCH INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
BGEZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
BGZ	Branch if accumulator < 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
BHZ	Branch on HTO = 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
BLEZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
RET	Return from subroutine	2	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
MPY	Multiply T register with P register	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1
MPYK	Multiply T register with immediate operand and store product in P register	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	0	0	1	0	1	0	0	0

CONTROL INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
LST	Load status register	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
NOP	No operation	2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
POP	Pop stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
PUSH	Push stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
SST	Store status register	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	0	0

I/O AND DATA MEMORY OPERATIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
DMOV	Copy contents of data memory location into next location	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0
IN	Input data from port	2	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0
OUT	Output data to port	2	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0
TSLR	Table read from program memory to data RAM	3	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

ACCUMULATOR INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
ADOH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	0
ADOS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	0
AND	AND with accumulator	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0
LAC	Load accumulator with shift	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	0
SUB	Subtract from accumulator with shift	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0</									