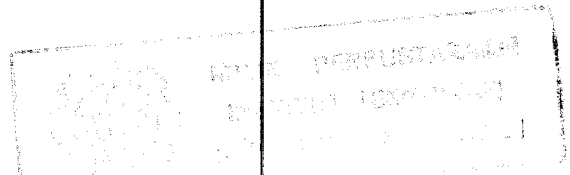


3891/115/4/91 ✓



IMPLEMENTASI ALGORITMA FAST FOURIER TRANSFORM DENGAN PROSESOR TMS 32010



PSE
621.395
KOT
I-1
1990

Oleh :

YONATHAN KRISTRIANTO

NRP. 2852200330


JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1990

IMPLEMENTASI ALGORITMA FAST FOURIER TRANSFORM DENGAN PROSESOR TMS 320210

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar
Sarjana Teknik Elektro
pada
Bidang Studi Elektronika
Di
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui
Dosen Pembimbing



(DR.Ir. Handayani Tjandrasa MSc)

S u r a b a y a

J u l i 1990

ABSTRAK



BILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Algoritma Fast Fourier Transform adalah algoritma untuk mendapatkan Diskrit Fourier Transform secara cepat. Untuk data sample yang cukup besar, pengolahan data ini masih memerlukan banyak operasi perkalian. Oleh karena itu diperlukan prosesor khusus yang memiliki hardware multiplier seperti halnya TMS32010.

Dengan dipakainya digital signal processor TMS32010 maka FFT bisa didapatkan relatif lebih cepat dibandingkan dengan penghitungan FFT melalui general purpose processor seperti halnya 8088.

Untuk menampilkan data hasil olahan TMS32010 diperlukan interfacing dengan IBM PC/XT. Melalui IBM PC/XT data tersebut bisa ditampilkan pada layar monitor.

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan yang telah membimbing penyusun menyelesaikan Tugas Akhir yang berjudul:

IMPLEMENTASI ALGORITMA FAST FOURIER TRANSFORM

DENGAN PROSESOR TMS32010

sebagai salah satu syarat kelulusan program S1 pada Jurusan Teknik Elektro ITS.

Terima kasih yang mendalam penyusun ucapkan kepada pihak-pihak yang telah membantu terselesaikannya Tugas Akhir ini. Secara khusus ucapan terima kasih kami tujukan kepada:

- DR. Ir. Handayani Tjandrasa selaku dosen pembimbing, dosen wali dan koordinator bidang studi elektronika.
- Seluruh staff pengajar maupun administrasi Jurusan Teknik Elektro ITS.
- Semua rekan yang telah mendukung terselesaikannya tugas akhir ini.

Penyusun berharap tugas akhir ini bisa bermanfaat bagi pembaca terutama mereka yang menaruh minat pada pengolahan sinyal digital dan semoga tugas akhir ini bermanfaat bagi penelitian lebih lanjut.

Surabaya, 20 Juli 1990

Penyusun

DAFTAR ISI

BAB	HALAMAN
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
I. PENDAHULUAN	1
1. 1. LATAR BELAKANG	1
1. 2. PERMASALAHAN	2
1. 3. PEMBatasan MASALAH	3
1. 4. METODOLOGI	4
1. 5. PEMBAHASAN	4
II. TRANSFORMASI FOURIER DAN ALGORITMA FAST FOURIER TRANSFORM	6
2. 1. PENDAHULUAN	6
2. 2. TRANSFORMASI FOURIER	7
2. 2. 1. DASAR ANALISA TRANSFORMASI FOURIER	7
2. 2. 2. INTEGRAL FOURIER	10
2. 2. 3. TRANSFORMASI FOURIER INVERSE	10

2.3.	SIFAT-SIFAT TRANSFORMASI FOURIER	11
2.3.1.	SIFAT-SIFAT UMUM	11
2.3.2.	INTEGRAL KONVOLUSI	12
2.3.3.	INTEGRAL KORELASI	17
2.4.	TRANSFORMASI FOURIER PADA SINYAL PERIODIK DAN TEORI SAMPLING	17
2.4.1.	TRANSFORMASI FOURIER PADA SINYAL PERIODIK	17
2.4.2.	PENYAMPLINGAN GELOMBANG	18
2.4.3.	TEORI SAMPLING	23
2.5.	TRANSFORMASI FOURIER DISKRIT	23
2.5.1.	PENJELASAN SECARA GRAFIS	23
2.5.2.	PENJELASAN TEORITIS	26
2.5.3.	TRANSFORMASI FOURIER INVERSE DISKRIT	31
2.5.4.	PENGARUH SAMPLING DAN PEMOTONGAN PADA TRANSFORMASI FOURIER DISKRIT	32
2.6.	ALGORITMA FAST FOURIER TRANSFORM	39
2.6.1.	PENGEMBANGAN TEORITIS ALGORITMA FFT BASIS DUA	41
2.6.2.	PENURUNAN ALGORITMA COOLEY-TUKEY UNTUK $N = 2^r$	46
2.6.3.	BENTUK LAIN DARI ALGORITMA FFT	48
2.6.4.	SIGNAL FLOW GRAPH	49

III. DIGITAL SIGNAL PROCESSOR TMS32010 DAN INTERFACING-NYA PADA IBM PC/XT	52
3. 1. PENDAHULUAN	52
3. 2. PIN-PIN TMS32010 DAN PENJELASANNYA	53
3. 3. BLOK DIAGRAM TMS32010	55
3. 4. ORGANISASI MEMORY	61
3. 4. 1. MEMORY MAP	61
3. 4. 2. PROGRAM MEMORY	62
3. 4. 3. DATA MEMORY	63
3. 4. 4. AUXILIARY REGISTER	64
3. 5. CENTRAL ARITHMETIC LOGIC UNIT (CALU)	66
3. 5. 1. PENGGESER (SHIFTER)	67
3. 5. 2. ALU DAN ACCUMULATOR	68
3. 5. 3. PENGALI, REGISTER T DAN REGISTER P	70
3. 6. SISTEM KONTROL	71
3. 6. 1. PROGRAM COUNTER DAN STACK	71
3. 6. 2. RESET	73
3. 6. 3. REGISTER STATUS	74
3. 6. 4. INTERRUPT	76
3. 7. FUNGSI INPUT DAN OUTPUT	78
3. 7. 1. OPERASI INPUT/OUTPUT	78
3. 7. 2. OPERASI PADA EKSTERNAL MEMORY	79
3. 8. BAHASA ASSEMBLY TMS32010	81
3. 8. 1. MODE DIRECT ADDRESSING	81
3. 8. 2. MODE INDIRECT ADDRESSING	83
3. 8. 3. MODE IMMEDIATE ADDRESSING	84

3. 8. 4. INSTRUKSI-INSTRUKSI TMS32010	85
3. 9. TEORI PENUNJANG UNTUK INTERFACING KE IBM PC/XT	88
3. 9. 1. MEMORY IBM PC/XT	91
3. 9. 2. INPUT/OUTPUT PORT IBM PC/XT	93
3. 9. 3. OPERASI INTERRUPT IBM PC/XT	95
IV. PERENCANAAN HARDWARE DAN SOFTWARE	98
4. 1. PENDAHULUAN	98
4. 1. 1. BLOK DIAGRAM	98
4. 1. 2. CARA KERJA RANGKAIAN SECARA UMUM	100
4. 2. PERENCANAAN RANGKAIAN DEKODER IBM PC/XT	101
4. 2. 1. PERENCANAAN RANGKAIAN DEKODER MEMORY IBM PC/XT	101
4. 2. 2. PERENCANAAN DEKODER I/O PORT IBM PC/XT	103
4. 2. 3. RANGKAIAN UNTUK MENGUBAH LEVEL RESET TMS32010	105
4. 3. PERENCANAAN RANGKAIAN BUFFER DAN MEMORY	106
4. 3. 1. PETA PEMAKAIAN MEMORY TMS32010	106
4. 3. 2. PERENCANAAN BUFFER	106
4. 3. 3. HUBUNGAN SINYAL PEMILIH KE MEMORY	108
4. 3. 4. PERHITUNGAN WAKTU AKSES MAKSIMUM MEMORY	110
4. 4. PERENCANAAN RANGKAIAN UNTUK OPERASI I/O TMS32010	113

4. 5. PERENCANAAN RANGKAIAN UNTUK SARANA KOMUNIKASI	115
4. 5. 1. PERENCANAAN PEMBANGKIT SINYAL BIO DAN INT TMS32010	115
4. 5. 2. PERENCANAAN INTERRUPT UNTUK IBM PC/XT	116
4. 6. PERENCANAAN RANGKAIAN ADC	118
4. 6. 1. ANALOG TO DIGITAL CONVERTER ADC0820	118
4. 6. 2. RANGKAIAN OP-AMP PENGUBAH TEGANGAN ANALOG	121
4. 6. 3. RANGKAIAN MONOSTABLE MULTIVIBRATOR	123
4. 7. PERENCANAAN PERANGKAT LUNAK (SOFTWARE)	125
4. 7. 1. PERENCANAAN PERANGKAT LUNAK UNTUK IBM PC/XT	125
4. 7. 2. PERENCANAAN PERANGKAT LUNAK UNTUK TMS32010	128
4. 7. 3. PERENCANAAN PROGRAM BERDASARKAN ALGORITMA FFT	132
V. ANALISIS DATA DAN HASIL PENGUKURAN	137
5. 1. ANALISIS FUNGSI-FUNGSI YANG TERSEDIA	137
5. 1. 1. ANALISIS FUNGSI SINUS	137
5. 1. 2. ANALISIS FUNGSI KOTAK	139
5. 1. 3. ANALISIS FUNGSI SEGITIGA	143

	x
5. 2. PENGUKURAN SPEKTRUM FREKUENSI FUNGSI YANG TERSEDIA	148
5. 2. 1. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI SINUS	148
5. 2. 2. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI KOTAK	151
5. 2. 3. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI SEGITIGA	152
VI. KESIMPULAN	156
DAFTAR PUSTAKA	158
LAMPIRAN I. GAMBAR RANGKAIAN LENGKAP	160
LAMPIRAN II. KARAKTERISTIK KOMPONEN	161
LAMPIRAN III. USULAN TUGAS AKHIR	191

DAFTAR GAMBAR

GAMBAR	HALAMAN
2. 1. INTERPRETASI TRANSFORMASI FOURIER	8
2. 2. TRANSFORMASI FOURIER SINYAL PERIODIK DAN NON PERIODIK	9
2. 3. KONVOLUSI FUNGSI IMPULSE	14
2. 4. APLIKASI PERSAMAAN (2-10)	15
2. 5. APLIKASI PERSAMAAN (2-11)	16
2. 6. TRANSFORMASI FOURIER UNTUK SINYAL PERIODIK	19
2. 7. PENYAMPLINGAN GELOMBANG DAN TRANSFORMASI FOURIER-NYA	20
2. 8. TERJADINYA ALIASING PADA FUNGSI TERSAMPLING JIKA PERIODE SAMPLING TERLALU BESAR	22
2. 9. PENURUNAN TRANSFORMASI FOURIER DISKRIT SECARA GRAFIS	24
2. 10. TRANSFORMASI FOURIER DISKRIT DENGAN INTERVAL PEMOTONGAN SAMA DENGAN PERIODE SINYAL INPUT	34
2. 11. TRANSFORMASI FOURIER DISKRIT DENGAN INTERVAL PEMOTONGAN TIDAK SAMA DENGAN PERIODE SINYAL INPUT	36
2. 10. TRANSFORMASI FOURIER DISKRIT UNTUK SINYAL INPUT YANG TIDAK PERIODIK	38
2. 11. BUTTERFLY ALGORITMA FFT DECIMATION IN TIME	49
2. 12. BUTTERFLY ALGORITMA FFT DECIMATION IN FREQUENCY	50
2. 13. SINYAL FLOW GRAPH FFT DECIMATION IN TIME UNTUK $N = 8$	50

2. 14.	SINYAL FLOW GRAPH FFT DECIMATION IN FREQUENCY UNTUK N = 8	50
3. 1.	PIN-PIN TMS32010	53
3. 2.	BLOK DIAGRAM TMS32010	56
3. 3.	MEMORY MAP TMS32010	61
3. 4.	AUXILIARY REGISTER	65
3. 5.	CENTRAL ARITHMETIC LOGIC UNIT (CALU)	66
3. 6.	INSTRUKSI RET DAN PUSH	72
3. 7.	INSTRUKSI POP	73
3. 8.	REGISTER STATUS DALAM DATA MEMORY	76
3. 9.	SINKRONISASI SINYAL INTERRUPT PADA TMS32010	77
3. 10.	DIAGRAM WAKTU UNTUK INSTRUKSI IN	79
3. 11.	DIAGRAM WAKTU UNTUK INSTRUKSI OUT	79
3. 12.	DIAGRAM WAKTU INSTRUKSI TBLR	80
3. 13.	DIAGRAM WAKTU INSTRUKSI TBLW	80
3. 14.	BLOK DIAGRAM MODE DIRECT ADDRESSING	82
3. 15.	KONFIGURASI PIN SLOT IBM PC/XT	89
3. 16.	PETA MEMORY IBM PC/XT	92
3. 17.	PETA I/O PORT PADA SYSTEM BOARD IBM PC/XT	94
3. 18.	PETA PEMAKAIAN I/O PORT PADA SLOT IBM PC/XT	95
4. 1.	BLOK DIAGRAM RANGKAIAN SECARA LENGKAP	99
4. 2.	PERBANDINGAN ADDRESS MEMORY PADA IBM PC/XT DAN TMS32010	102
4. 3.	RANGKAIAN DEKODER UNTUK MEMORY DAN I/O PORT	104
4. 4.	RANGKAIAN UNTUK MENGUBAH LEVEL RESET TMS32010	105

4. 5.	PETA PEMAKAIAN MEMORY TMS32010	106
4. 6.	HUBUNGAN SINYAL PEMILIH PADA MEMORY	110
4. 7.	RANGKAIAN LENGKAP BUFFER DAN MEMORY	111
4. 8.	RANGKAIAN DEKODER UNTUK OPERASI OUTPUT TMS32010	114
4. 9.	RANGKAIAN SINKRONISASI SINYAL BIO DAN INT	116
4. 10.	RANGKAIAN PEMBANGKIT SINYAL INTERRUPT IBM PC/XT	117
4. 11.	KONFIGURASI PIN-PIN ADC0820	119
4. 12.	DIAGRAM WAKTU KONVERSI ADC0820 MODE WRITE-READ	121
4. 13.	RANGKAIAN OP-AMP PENGUBAH TEGANGAN ANALOG	122
4. 14.	RANGKAIAN ANALOG TO DIGITAL CONVERTER	124
4. 15.	FLOWCHART PERANGKAT LUNAK IBM PC/XT	126
4. 16.	PETA PEMAKAIAN MEMORY TMS32010	129
4. 17.	FLOWCHART PERANGKAT LUNAK TMS32010	130
4. 18.	SIGNAL FLOW GRAPH FFT DECIMATION IN TIME $N = 8$	132
4. 19.	FLOWCHART PENGOLAHAN DATA SESUAI DENGAN ALGORITMA FFT	134
5. 1.	FUNGSI SINUS TERSAMPLING 32 DATA	137
5. 2.	FUNGSI KOTAK TERSAMPLING 32 DATA	139
5. 3.	PENGHITUNGAN TRANSFORMASI FOURIER FUNGSI KOTAK PERIODIK SECARA GRAFIS	142
5. 4.	FUNGSI SEGITIGA TERSAMPLING 32 DATA	144
5. 5.	PENGHITUNGAN TRANSFORMASI FOURIER FUNGSI SEGITIGA PERIODIK SECARA GRAFIS	146
5. 6 (a).	HASIL PENGUKURAN FREKUENSI REAL FUNGSI SINUS	149
5. 6 (b).	HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI SINUS	149

5.7(a).	HASIL PENGUKURAN FREKUENSI REAL FUNGSI KOTAK	151
5.7(b).	HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI SINUS	152
5.8(a).	HASIL PENGUKURAN FREKUENSI REAL FUNGSI SEGITIGA	154
5.8(b).	HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI SEGITIGA	154

DAFTAR TABEL

TABEL	HALAMAN
2. 1. SIFAT-SIFAT TRANSFORMASI FOURIER	11
3. 1. OPERASI LOGIKA PADA ALU	69
3. 2. SIMBOL DAN SINGKATAN PADA PENULISAN ASSEMBLY TMS32010	85
3. 3. KUMPULAN INSTRUKSI TMS32010	86
3. 4. LEVEL PRIORITY INTERRUPT IBM PC/XT	96
5. 1. PENGUKURAN TRANSFORMASI FOURIER FUNGSI SINUS 32 DATA	150
5. 2. PENGUKURAN TRANSFORMASI FOURIER FUNGSI KOTAK 32 DATA	153
5. 3. PENGUKURAN TRANSFORMASI FOURIER FUNGSI SEGITIGA 32 DATA	155

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Menjelang akhir abad ke 20 ini, perkembangan teknologi digital sangat pesat dan mempengaruhi hampir semua bidang. Pengolahan sinyal tidak ketinggalan juga memasuki era digital.

Pengolahan sinyal secara umum dipakai dalam berbagai bidang yang mencakup teknologi biomedik, akustik, sonar, radar, seismologi, komunikasi data dan masih banyak lagi. Fungsi pengolahan sinyal ini antara lain untuk menghilangkan noise dari suatu sinyal masukan atau memodulasi suatu sinyal dengan sinyal yang lain.

Sejak tahun 1950-an pemrosesan sinyal sudah berkembang sehingga data yang diproses sudah sedemikian banyak. Oleh karena itu dibutuhkan suatu sistem baru yang melibatkan komputer sebagai alat bantu. Dari situ lah perkembangan pengolahan sinyal digital dimulai. Dibandingkan dengan pengolahan sinyal analog, pengolahan sinyal digital mempunyai keunggulan utama yaitu tidak banyak dipengaruhi noise.

Salah satu bagian dalam pengolahan sinyal yang banyak dipakai adalah Transformasi Fourier. Transformasi Fourier

ini berfungsi untuk mentransformasikan suatu sinyal dari daerah waktu (time domain) ke daerah frekuensi (frequency domain). Transformasi ini penting terutama untuk analisis spektrum. Dalam pengolahan sinyal digital, transformasi ini dilakukan secara diskrit sehingga disebut Transformasi Fourier Diskrit (DFT). Penerapan DFT secara langsung dalam mikrokomputer yang ada saat ini sangat tidak efisien karena membutuhkan waktu proses yang lama.

Pada tahun 1965, James W. Cooley dan John W. Tukey mempublikasikan suatu algoritma baru untuk mendapatkan DFT secara lebih efisien. Algoritma itu sekarang dikenal dengan nama Fast Fourier Transform (FFT). Dengan dipakainya algoritma ini maka DFT bisa didapatkan dengan lebih cepat sehingga Transformasi Fourier bisa didapatkan melalui komputer secara real time.

1.2. PERMASALAHAN

Pengolahan sinyal digital yang mengolah data dalam jumlah yang sangat besar, misalnya data yang diterima stasiun bumi dari satelit ruang angkasa, akan membutuhkan waktu proses yang lama. Pengolahan data tersebut membutuhkan waktu yang sangat panjang bila dikerjakan dengan komputer biasa yang memakai general purpose prosesor.

Untuk mendapatkan waktu olah yang cepat dibutuhkan suatu prosesor khusus yang disebut digital signal processor.

Ada beberapa jenis digital signal processor yang dijual bebas. Satu diantaranya adalah TMS32010 produksi Texas Instrument. Prosesor ini dipakai karena keunggulannya dalam pengolahan data aritmatik. Dengan memakai prosesor ini proses perkalian hanya membutuhkan satu cycle saja sehingga waktu pemrosesan data, terutama yang banyak melakukan operasi perkalian, dapat direduksi.

1.3. PEMBATAAN MASALAH

Dalam implementasi algoritma FFT dengan prosesor TMS32010, ada beberapa hal yang perlu diperhatikan sebagai batas permasalahan.

Prosesor TMS32010 di sini berfungsi hanya sebagai pengolah data sehingga masih perlu dihubungkan dengan komputer IBM PC. IBM PC dipergunakan sebagai pengontrol untuk inisialisasi dan juga untuk menampilkan data yang diolah serta hasil pengolahannya.

Jumlah data yang bisa diolah oleh TMS32010 bisa diprogram melalui IBM PC. Banyaknya data maksimum adalah 256 yang terdiri dari data real saja. Data ini diperoleh melalui proses sampling yang dilakukan dengan ADC0820. Frekuensi penyampling maksimum adalah 7 KHz.

Algoritma FFT yang dipakai adalah algoritma decimation in time basis 2. Hal ini berarti bahwa jumlah sample harus merupakan eksponensial dari 2. Algoritma ini merupakan algoritma FFT yang paling sederhana tetapi sudah cukup banyak mereduksi waktu pemrosesan.

1.4. METODOLOGI

Rangkaian TMS32010 yang dibuat dalam tugas akhir ini didesain dalam bentuk eksperimental. Dalam hal ini seluruh memory yang dipergunakan berupa RAM Statik sehingga program dapat disimpan dalam diskette dan pemrograman dapat dilakukan melalui komputer IBM PC. Keuntungan dari desain ini terletak pada keluwesannya karena seluruh sistem dapat diprogram melalui komputer IBM PC, sehingga dengan mengubah softwarenya maka sistem hardware dapat dipergunakan untuk keperluan lain seperti implementasi untuk filter digital.

1.5. PEMBAHASAN

Teori-teori penunjang yang mendasari algoritma FFT dasar 2 dibahas dalam bab II. Dalam bab tersebut dipelajari juga mengenai hubungan FFT dengan Transformasi

Fourier kontinyu, efek-efek aliasing dan ripple. Tugas akhir ini lebih ditekankan pada pengolahan data sehingga aplikasi algoritma FFT tidak dibahas.

Pada bab III dibahas mengenai prosesor TMS32010 yang dipakai. Hal ini diperlukan karena prosesor ini mempunyai kemampuan yang berbeda dengan prosesor lain yang banyak dipakai seperti 8088 atau Z80.

Pada bab IV dibahas mengenai perencanaan hardware dan software dari seluruh rangkaian. Perencanaan hardware yang dibahas mencakup pembuatan minimum sistem TMS32010 dan sistem interfacing pada IBM PC/XT. Perencanaan software dibagi menjadi dua yaitu software untuk TMS32010 dan software untuk IBM PC/XT.

Pada bab V ditunjukkan hasil pengolahan data dari rangkaian beserta analisisnya.

Kesimpulan dan saran diberikan pada bab VI.

B A B II

TRANSFORMASI FOURIER DAN ALGORITMA FAST FOURIER TRANSFORM

2.1. PENDAHULUAN

Sebelum perencanaan pembuatan rangkaian implementasi Fast Fourier Transform dengan TMS32010 ini dipelajari, lebih dulu akan dibahas mengenai transformasi Fourier serta algoritma Fast Fourier Transform.

Transformasi Fourier dapat dibandingkan dengan bentuk transformasi lain misalnya transformasi sistem koordinat dari bentuk koordinat rectanguler (X, Y) menjadi koordinat kutub (R, θ). Transformasi sistem koordinat ini ditujukan untuk memudahkan suatu perhitungan atau analisis. Demikian juga dengan transformasi Fourier, hanya saja transformasi Fourier lebih kompleks karena variabel yang ditransformasikan berada dalam range $-\infty$ ke $+\infty$. Secara sederhana transformasi Fourier dapat dikatakan sebagai transformasi yang mengubah suatu sinyal dari daerah waktu (time domain) ke daerah frekuensi (frequency domain). Secara fisik transformasi ini berfungsi untuk memisah-misahkan suatu bentuk gelombang menjadi sejumlah sinyal sinusoida dengan frekuensi dan amplitudo yang berbeda-beda.

Dalam bentuk digital, transformasi Fourier diperoleh melalui suatu proses yang disebut Transformasi Fourier

Diskrit (DFT). DFT sebenarnya merupakan penghitungan transformasi Fourier kontinyu secara diskrit sehingga dapat diproses pada komputer digital. Fast Fourier Transform adalah sebuah algoritma untuk menghitung Discrete Fourier Transform secara lebih cepat. Dengan algoritma ini maka jumlah operasi perkalian untuk menghitung DFT bisa direduksi. Bila suatu sinyal disampling sebanyak N kali maka untuk menghitung DFT dibutuhkan operasi perkalian kompleks sebanyak N^2 kali. Dengan algoritma FFT operasi perkalian bisa direduksi menjadi $0,5N \times \log_2(N)$ kali.

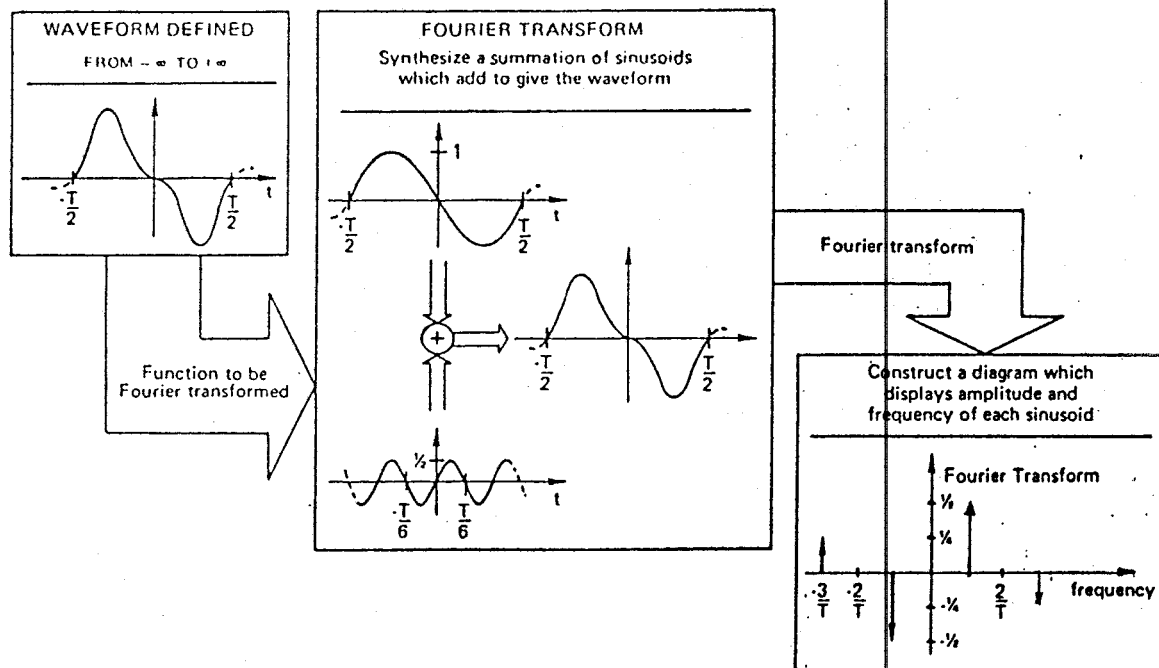
Perangkat keras yang digunakan adalah suatu sistem minimum dari prosesor TMS32010. Dalam sistem minimum ini pemrosesan data dilakukan secara cepat karena prosesor TMS32010 melakukan operasi aritmatik dengan hanya membutuhkan satu clock. Cara kerja prosesor ini akan dibahas lebih lanjut setelah pembahasan mengenai transformasi Fourier dan algoritma Fast Fourier Transform.

2.2. TRANSFORMASI FOURIER

2.2.1. DASAR ANALISIS TRANSFORMASI FOURIER

Transformasi Fourier merupakan salah satu metode penganalisis yang penting dalam beberapa bidang ilmu pada akhir abad XX ini. Transformasi Fourier sebenarnya dapat

diinterpretasikan sebagai suatu transformasi untuk mengubah suatu bentuk gelombang menjadi penjumlahan dari beberapa gelombang sinusoida yang mempunyai frekuensi dan amplitudo yang bervariasi.



GAMBAR. 2.1 1)

INTERPRETASI TRANSFORMASI FOURIER

Secara matematis transformasi Fourier dapat ditulis sebagai berikut: 2)

$$S(f) = \int_{-\infty}^{+\infty} s(t)e^{-j2\pi f t} dt \quad (2-1)$$

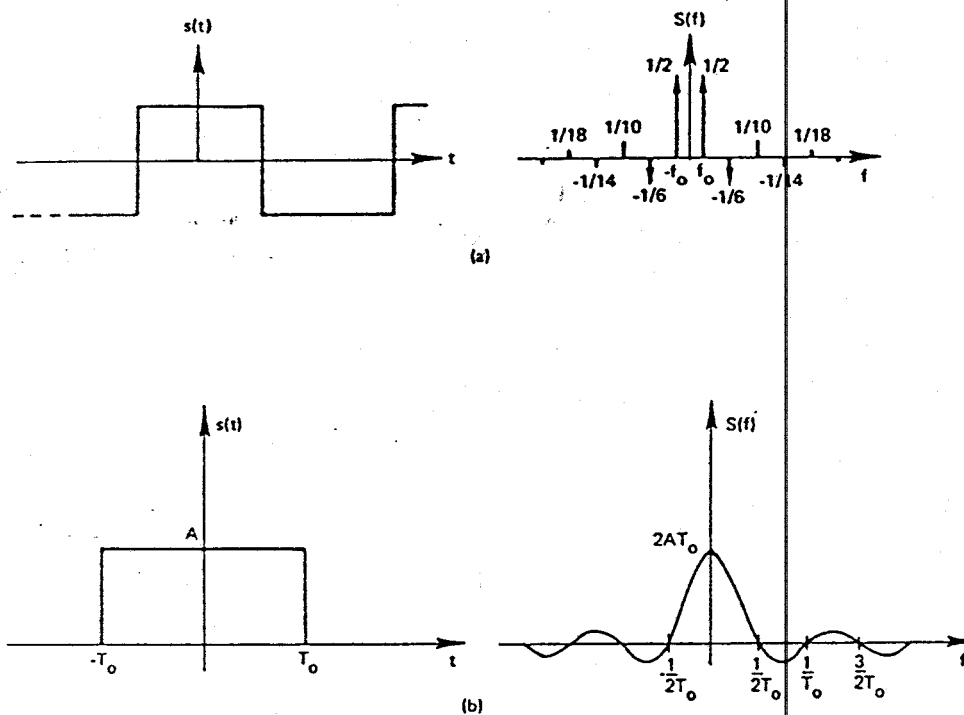
1) Brigham, E.O., "The Fast Fourier Transform", Prentice Hall Inc., hal 4

2) Ibid, hal 3

dengan $s(t)$ adalah gelombang yang ditransformasikan dan $j = \sqrt{-1}$.

Transformasi Fourier dikatakan sempurna bila penjumlahan gelombang sinusoida itu menghasilkan gelombang aslinya. Secara praktis untuk bentuk gelombang tertentu hal ini sulit dilakukan karena transformasi itu dilakukan untuk daerah $-\infty$ sampai $+\infty$.

Pada gelombang yang periodik hasil dari transformasi Fourier merupakan suatu deret yang diskrit, sedangkan untuk gelombang yang nonperiodik, hasil transformasi Fourier akan berbentuk sinyal yang kontinu.



GAMBAR 2.2 ³⁾

- (a). TRANSFORMASI FOURIER SINYAL PERIODIK
 (b). TRANSFORMASI FOURIER SINYAL NONPERIODIK

³⁾ Ibid, hal 5 s.d. 6

2. 2. 2 INTEGRAL FOURIER

Integral Fourier didefinisikan sebagai berikut: ⁴⁾

$$H(f) = \int_{-\infty}^{+\infty} h(t) e^{-j2\pi ft} dt \quad (2-2)$$

Jika harga integral tersebut ada untuk setiap harga f maka H(f) adalah transformasi Fourier dari h(t).

Umumnya hasil transformasi Fourier berupa bilangan kompleks:

$$H(f) = R(f) + jI(f) = |H(f)| e^{j\Theta(f)} \quad (2-3)$$

dengan R(f) adalah bagian real hasil transformasi Fourier, dan I(f) adalah bagian imajineranya. |H(f)| adalah amplitudo atau *Spektrum Fourier* H(f) yang besarnya adalah $\sqrt{R^2(f) + I^2(f)}$, sedangkan $\Theta(f)$ disebut sudut phase H(f) yang besarnya adalah $\tan^{-1} [I(f)/R(f)]$.

2. 2. 3. TRANSFORMASI FOURIER INVERSE

Transformasi Fourier inverse adalah transformasi untuk mengembalikan suatu spektrum frekuensi menjadi fungsi waktunya. Jadi transformasi tersebut merupakan kebalikan transformasi Fourier.

Transformasi Fourier Inverse didefinisikan sebagai: ⁵⁾

$$h(t) = \int_{-\infty}^{+\infty} H(f) e^{j2\pi ft} df \quad (2-4)$$

Fungsi h(t) dan H(f) yang merupakan suatu fungsi dan hasil

4) Ibid, hal 11

5) Ibid, hal 13

transformasi Fourier-nya disebut pasangan transformasi Fourier dan dinotasikan $h(t) \leftrightarrow H(f)$.

2.3. SIFAT-SIFAT TRANSFORMASI FOURIER

2.3.1. SIFAT-SIFAT UMUM

Pada tabel 2.1 berikut ini ditunjukkan sifat-sifat transformasi Fourier kontinyu.

TABEL 2.1 ⁶⁾

SIFAT-SIFAT TRANSFORMASI FOURIER

Time domain	Frequency domain
Linear addition $x(t) + y(t)$	Linear addition $X(f) + Y(f)$
Symmetry $H(f)$	Symmetry $h(-t)$
Time scaling $h(kt)$	Inverse scale change $\frac{1}{k} H\left(\frac{f}{k}\right)$
Inverse scale change $\frac{1}{k} h\left(\frac{t}{k}\right)$	Frequency scaling $H(kf)$
Time shifting $h(t - t_0)$	Phase shift $H(f)e^{-j2\pi ft_0}$
Modulation $h(t)e^{j2\pi ft_0}$	Frequency shifting $H(f - f_0)$
Even function $h_e(t)$	Real function $H_e(f) = R_e(f)$
Odd function $h_o(t)$	Imaginary $H_o(f) = jI_o(f)$
Real function $h(t) = h_e(t)$	Real part even Imaginary part odd $H(f) = R_e(f) + jI_o(f)$
Imaginary function $h(t) = jh_o(t)$	Real part odd Imaginary part even $H(f) = -R_o(f) + jI_e(f)$

⁶⁾ Ibid, hal 46

Dalam penerapannya ada sifat-sifat yang sering dipakai. Misalnya sifat ketujuh dan kedelapan dapat diterapkan pada setiap bentuk gelombang karena suatu bentuk fungsi tertentu dapat dipisahkan menjadi penjumlahan fungsi genap dan fungsi ganjil. Hal ini bisa dibuktikan sebagai berikut.

$$\begin{aligned} h(t) &= \frac{h(t)}{2} + \frac{h(t)}{2} \\ &= \left[\frac{h(t)}{2} + \frac{h(-t)}{2} \right] + \left[\frac{h(t)}{2} - \frac{h(-t)}{2} \right] \\ &= h_{\text{even}}(t) + h_{\text{odd}}(t) \end{aligned}$$

Jadi transformasi Fourier dari $h(t)$ adalah

$$H(f) = H_{\text{even}}(f) + H_{\text{odd}}(f) \quad (2-5)$$

Selain sifat-sifat umum di atas ada dua lagi sifat yang banyak dipergunakan secara praktis yaitu *konvolusi* dan *korelasi*.

2.3.2. INTEGRAL KONVOLUSI

Integral konvolusi didefinisikan sebagai berikut: ⁷⁾

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau) d\tau \quad (2-6)$$

Fungsi $y(t)$ disebut *konvolusi* fungsi $x(t)$ dan $h(t)$ dan $y(t)$ biasa dituliskan sebagai $x(t)*h(t)$.

Salah satu penerapan yang penting dari integral

7) Ibid, hal 50

Konvolusi adalah konvolusi untuk fungsi impulse $\delta(t)$. Fungsi impulse ini mempunyai sifat yang khusus bila diintegrasikan sebagai berikut:

$$\int_{-\infty}^{+\infty} \delta(t-t_0) x(t) dt = x(t_0) \quad (2-7)$$

Misalkan $h(t)$ adalah deret fungsi impulse $h(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT)$ maka $h(t)*x(t)$ dapat dihitung sebagai berikut:

$$\begin{aligned} y(t) &= \int_{-\infty}^{+\infty} \left[\sum_{n=-\infty}^{\infty} \delta(\tau-nT) \right] x(t-\tau) d\tau \\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{+\infty} \delta(\tau-nT) x(t-\tau) d\tau \end{aligned} \quad (2-8)$$

Dari persamaan (2-7) maka persamaan (2-8) dapat ditulis menjadi:

$$y(t) = \sum_{n=-\infty}^{\infty} [x(t-nT)] \quad (2-9)$$

Jadi dapat disimpulkan bahwa konvolusi suatu fungsi $x(t)$ dengan fungsi impulse bisa langsung didapat dengan menempatkan fungsi $x(t)$ pada posisi impulse. Pada gambar 2.3 ditunjukkan cara mendapatkan fungsi periodik melalui konvolusi.

Operasi konvolusi bisa diterapkan di daerah waktu maupun daerah frekuensi. Jika konvolusi diterapkan di daerah waktu maka disebut *time convolution* sedangkan pada daerah frekuensi, konvolusi disebut *frequency convolution*.

Integral konvolusi ini menjadi sangat penting dalam transformasi Fourier dengan adanya dua buah teorema

berikut ini.

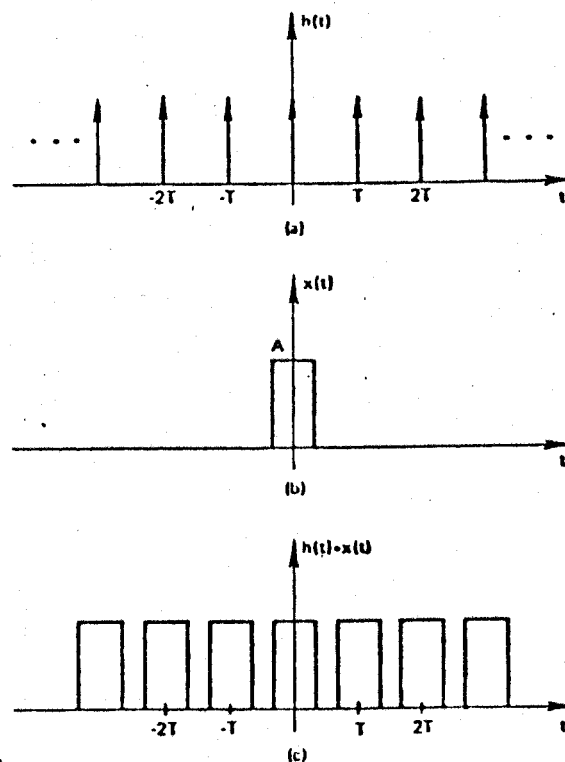
Teorema konvolusi waktu:

$$x(t) * h(t) \quad \langle\langle \underline{\quad} \rangle\rangle \quad X(f)H(f) \quad (2-10)$$

Teorema konvolusi frekuensi:

$$x(t)h(t) \quad \langle\langle \underline{\quad} \rangle\rangle \quad X(f) * H(f) \quad (2-11)$$

Pada gambar 2.4 dan 2.5 ditunjukkan aplikasi dari persamaan (2-10) dan (2-11) yang dihubungkan dengan konvolusi fungsi impulse.

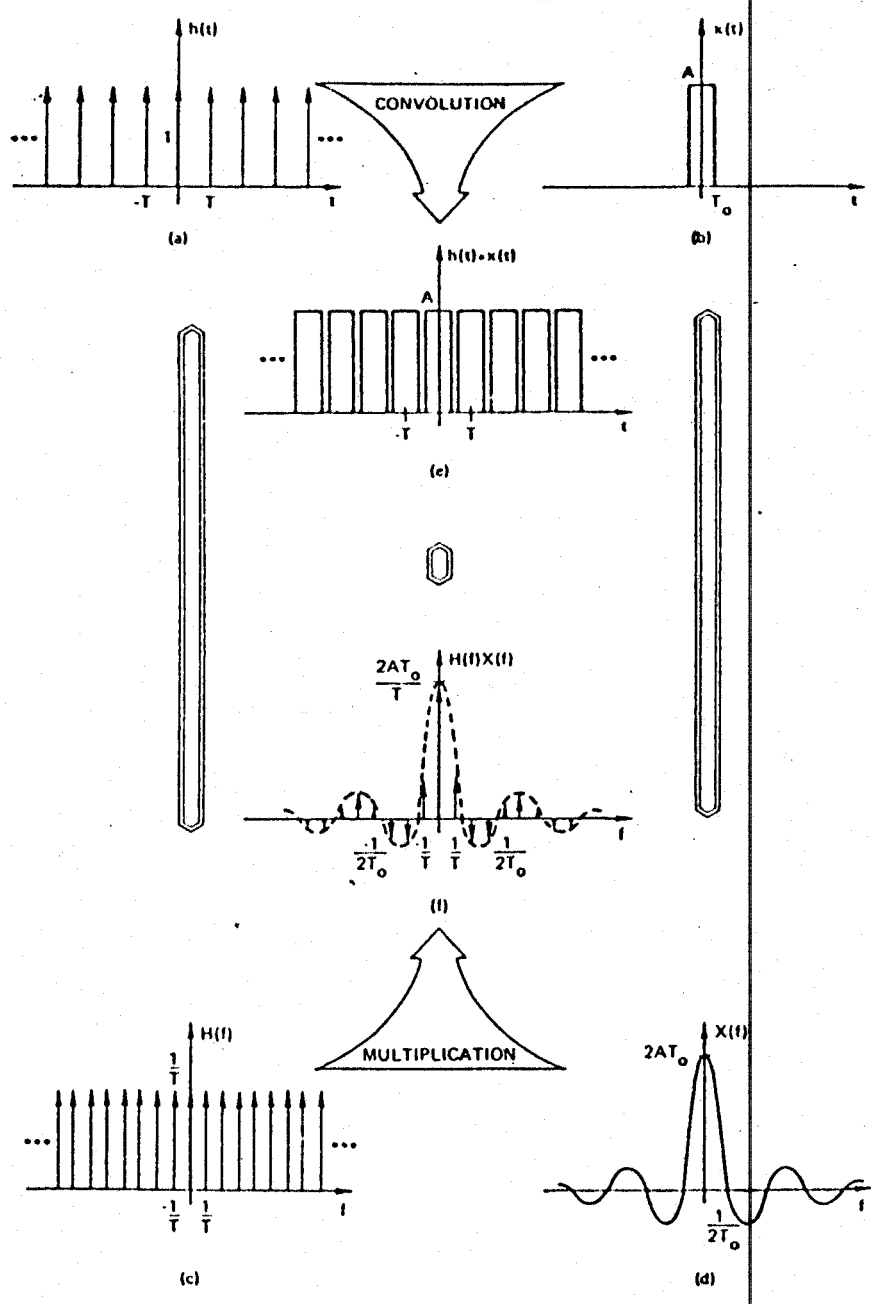


GAMBAR 2.3 8)

KONVOLUSI FUNGSI IMPULSE

- (a). FUNGSI IMPULSE YANG BERPERIODE T
- (b). FUNGSI KOTAK TUNGGAL
- (c). HASIL KONVOLUSI FUNGSI TUNGGAL DENGAN FUNGSI IMPULSE

8) Ibid, hal 59

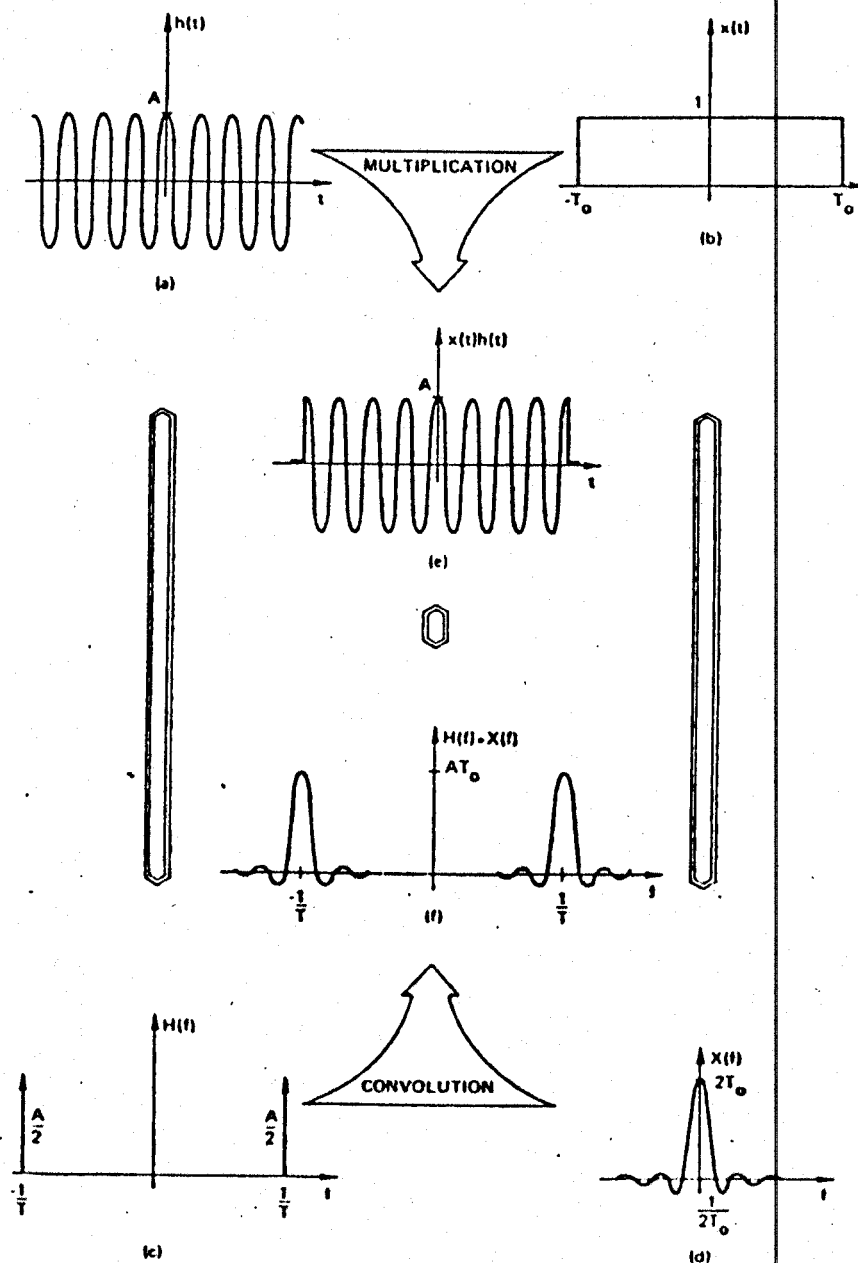


GAMBAR 2.4 9)

APLIKASI PERSAMAAN (2-10)

- (a). FUNGSI IMPULSE SEBAGAI OPERAND PERTAMA
- (b). FUNGSI KOTAK TUNGGAL SEBAGAI OPERAND KEDUA
- (c). HASIL TRANSFORMASI FOURIER FUNGSI (a)
- (d). HASIL TRANSFORMASI FOURIER FUNGSI (b)
- (e). HASIL KONVOLUSI FUNGSI (a) DENGAN FUNGSI (b)
- (f). HASIL PERKALIAN FUNGSI (c) DENGAN FUNGSI (d)

9) Ibid, hal 62

GAMBAR 2.5 ¹⁰⁾

APLIKASI PERSAMAAN (2-11)

- (a). FUNGSI SINUS SEBAGAI OPERAND PERTAMA
- (b). FUNGSI KOTAK TUNGGAL SEBAGAI OPERAND KEDUA
- (c). HASIL TRANSFORMASI FOURIER FUNGSI SINUS
- (d). HASIL TRANSFORMASI FOURIER FUNGSI KOTAK TUNGGAL
- (e). HASIL PERKALIAN FUNGSI (a) DENGAN FUNGSI (b)
- (f). HASIL KONVOLUSI FUNGSI (c) DENGAN FUNGSI (d)

10) Ibid, hal 63

2.3.3. INTEGRAL KORELASI

Persamaan integral lain yang penting adalah integral korelasi yang didefinisikan sebagai berikut: ⁱⁱ⁾

$$z(t) = \int_{-\infty}^{+\infty} x(\tau)h(t+\tau) d\tau \quad (2-12)$$

Hubungan antara korelasi dengan transformasi Fourier adalah sebagai berikut:

$$Z(f) = H(f)X^*(f) \quad (2-13)$$

dengan $X^*(f)$ adalah konjugat dari $X(f)$.

2.4. TRANSFORMASI FOURIER PADA SINYAL PERIODIK DAN TEORI SAMPLING

2.4.1. TRANSFORMASI FOURIER PADA SINYAL PERIODIK

Suatu sinyal dengan periode T_0 dapat dituliskan sebagai konvolusi dari sinyal tunggalnya dengan sinyal impulse yang berperiode T_0 . Misalkan sinyal periodik itu $y(t)$, sinyal tunggal $h(t)$ dan sinyal impulse $x(t)$ maka

$$y(t) = h(t)*x(t)$$

Transformasi Fourier pada kedua sisi persamaan di atas menghasilkan

$$Y(f) = H(f)X(f)$$

$$= H(f) \frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_0}\right)$$

ii) Ibid, hal64

$$= \frac{1}{T_0} \sum_{n=-\infty}^{\infty} H\left(\frac{n}{T_0}\right) \delta\left(f - \frac{n}{T_0}\right) \quad (2-14)$$

Jika didefinisikan

$$\alpha_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} h(t) e^{-j\omega n t} dt$$

$$\alpha_n = \frac{1}{T_0} H(nf_0)$$

maka didapat

$$Y(f) = \sum_{n=-\infty}^{\infty} \alpha_n \delta\left(f - \frac{n}{T_0}\right) \quad (2-15)$$

Pada gambar 2.6 ditunjukkan transformasi Fourier dari sinyal periodik.

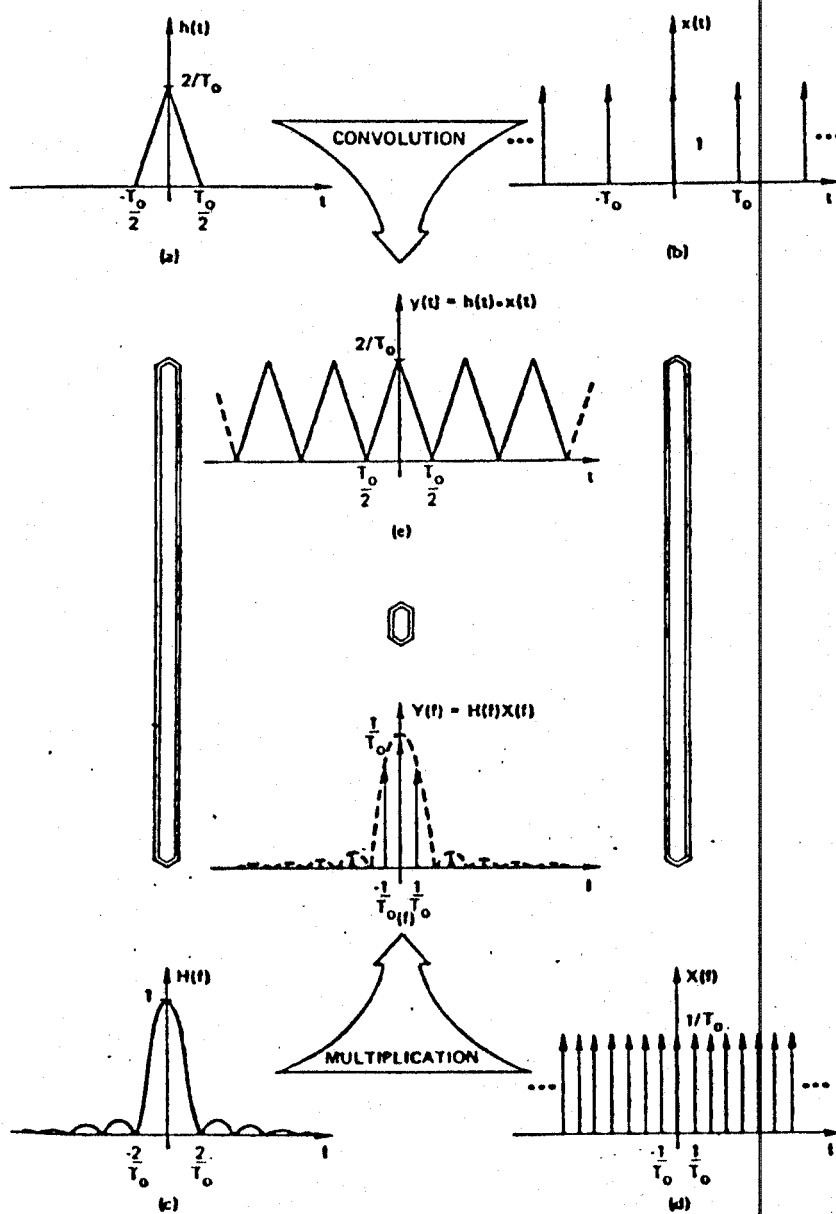
2.4.2 PENYAMPLINGAN GELOMBANG

Pada bagian sebelumnya sudah dijelaskan mengenai integral konvolusi yang dihubungkan dengan fungsi impulse. Jika suatu gelombang disampling dengan periode T detik maka hasil yang sama bisa didapat dengan cara mengalikan gelombang tersebut dengan sinyal impulse dengan periode T detik. Secara matematis hal ini dituliskan sebagai berikut

$$h'(t) = h(t) \left[\sum_{n=-\infty}^{\infty} \delta(t-nT) \right] \quad (2-16)$$

Jika $h(t)$ kontinu pada $t=nT$ maka persamaan (2-16) bisa ditulis:

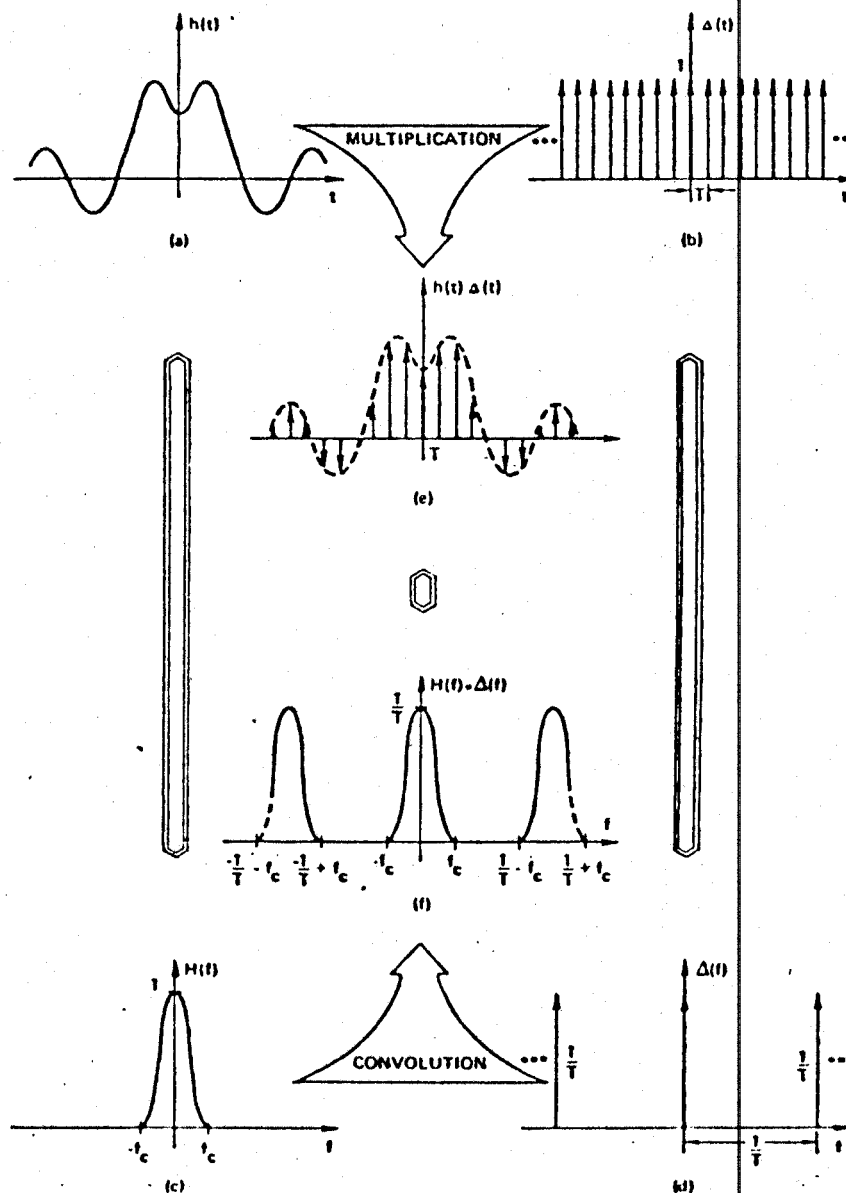
$$h'(t) = \sum_{n=-\infty}^{\infty} [h(nT) \delta(t-nT)] \quad (2-17)$$

GAMBAR 2.6 ¹²⁾

TRANSFORMASI FOURIER UNTUK SINYAL PERIODIK

- (a). FUNGSI TUNGGAL PADA DAERAH WAKTU
 (b). FUNGSI IMPULSE DENGAN PERIODE T_0
 (c). HASIL TRANSFORMASI FOURIER FUNGSI (a)
 (d). HASIL TRANSFORMASI FOURIER FUNGSI (b)
 (e). HASIL KONVOLUSI FUNGSI (a) DENGAN FUNGSI (b)
 (f). HASIL PERKALIAN FUNGSI (c) DENGAN FUNGSI (d)

¹²⁾ Ibid, hal 79



GAMBAR 2. 7 13)

PENYAMPLINGAN GELOMBANG DAN TRANSFORMASI FOURIERNYA

- (a). FUNGSI YANG AKAN DISAMPLING
- (b). FUNGSI IMPULSE SEBAGAI FUNGSI PENYAMPLING
- (c). HASIL TRANSFORMASI FOURIER FUNGSI (a)
- (d). HASIL TRANSFORMASI FOURIER FUNGSI IMPULSE
- (e). HASIL PERKALIAN FUNGSI (a) DENGAN FUNGSI (b)
- (f). HASIL KONVOLUSI FUNGSI (c) DENGAN FUNGSI (d)

13) Ibid, hal 91

Pada gambar 2.7 ditunjukkan bahwa konvolusi berperan membentuk sinyal yang periodik dari hasil transformasi Fourier gelombang tersampling. Hal ini terjadi karena transformasi Fourier sinyal impulse dengan periode T adalah sinyal impulse dengan periode $1/T$ sehingga konvolusi gelombang hasil transformasi Fourier menghasilkan sinyal yang periodik dengan periode $1/T$.

Perlu diperhatikan juga bahwa jika periode sampling terlalu besar maka transformasi Fourier akan menghasilkan sinyal yang terdistorsi. Distorsi ini disebut dengan *aliasing*. *Aliasing* terjadi bila frekuensi sampling lebih kecil dari frekuensi yang disebut dengan *Nyquist sampling rate*. *Nyquist sampling rate* dirumuskan sebagai berikut:¹⁴⁾

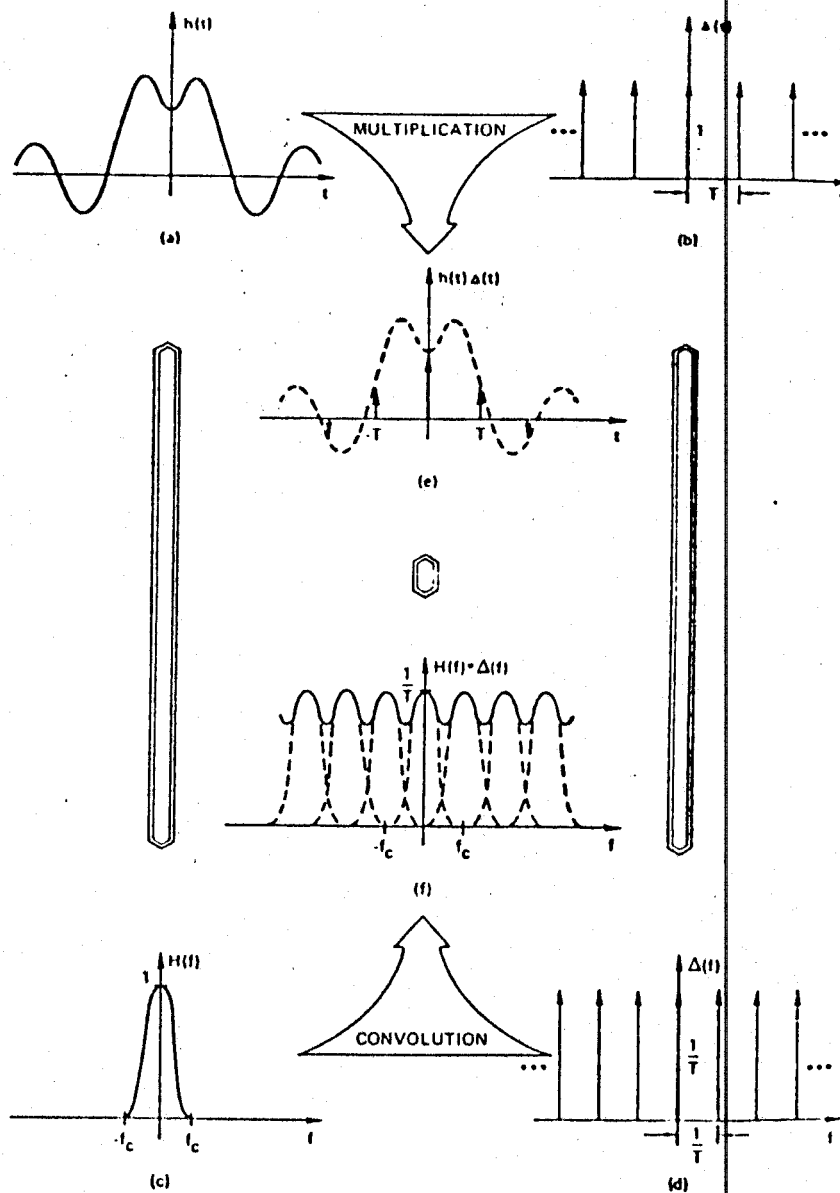
$$f_s = 2 f_c \quad (2-18)$$

dengan f_c adalah frekuensi tertinggi dari hasil transformasi Fourier sinyal kontinyu $h(t)$.

Secara sederhana dari ketentuan di atas dapat disimpulkan bahwa jika frekuensi sampling lebih kecil dari *Nyquist sampling rate* maka hasil penyamplingan tidak bisa menggambarkan sinyal yang disampling dengan benar.

Pada gambar 2.8 ditunjukkan terjadinya aliasing bila frekuensi sampling lebih kecil dari f_s .

14) Ibid, hal 85



GAMBAR 2.8 15)

TERJADINYA ALIASING PADA FUNGSI TERSAMPLING JIKA PERIODE SAMPLING TERLALU BESAR

- (a). FUNGSI YANG AKAN DISAMPLING
- (b). FUNGSI IMPULSE PENYAMPLING
- (c). HASIL TRANSFORMASI FOURIER FUNGSI (a)
- (d). HASIL TRANSFORMASI FOURIER FUNGSI (b)
- (e). HASIL PERKALIAN FUNGSI (a) DENGAN FUNGSI (b)
- (f). HASIL KONVOLUSI FUNGSI (c) DENGAN FUNGSI (d)

2.4.2 TEORI SAMPLING

Teori sampling menyatakan bahwa jika transformasi Fourier dari sebuah fungsi $h(t)$ sama dengan nol untuk semua frekuensi yang lebih besar dari *Nyquist sampling rate* (f_s) maka fungsi kontinyu dari $h(t)$ bisa didapatkan dari hasil penyamplingannya, dengan rumus: ¹⁶⁾

$$h(t) = T \sum_{n=-\infty}^{\infty} \left[h(nT) \frac{\sin 2\pi f_c (t-nT)}{\pi (t-nT)} \right] \quad (2-19)$$

dengan:

$h(t)$: fungsi kontinyu

$h(nT)$: fungsi tersampling

$T = 1/f_s$

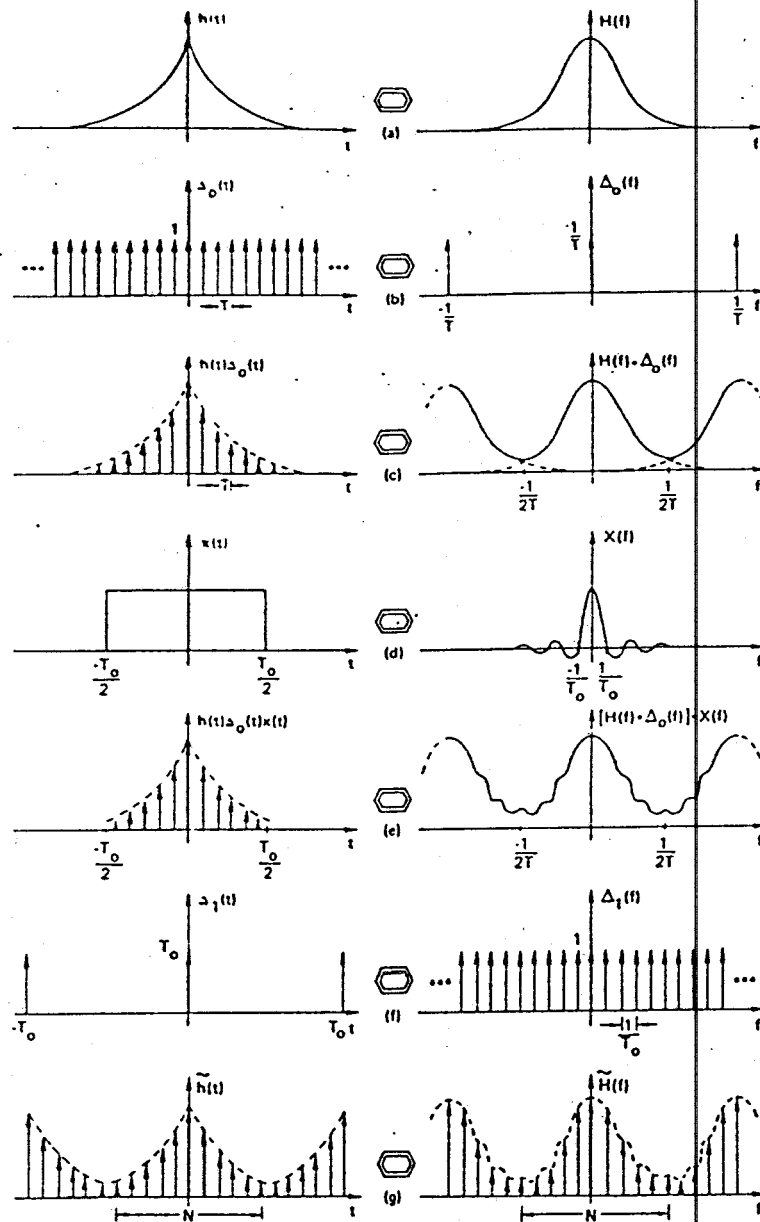
2.5. TRANSFORMASI FOURIER DISKRIT

Transformasi Fourier diskrit merupakan kasus khusus dari transformasi Fourier kontinyu. Melalui transformasi Fourier diskrit ini transformasi Fourier kontinyu bisa didapatkan dengan komputer digital.

2.5.1. PENJELASAN SECARA GRAFIS

Pada gambar 2.9 ditunjukkan proses pembentukan transformasi Fourier diskrit.

¹⁶⁾ Ibid, hal 83



GAMBAR 2.9 17)

- PENURUNAN TRANSFORMASI FOURIER DISKRIT SECARA GRAFIS
- (a). FUNGSI ASAL
 - (b). FUNGSI IMPULSE UNTUK PENYAMPLING DAERAH WAKTU
 - (c). FUNGSI TERSAMPLING
 - (d). FUNGSI KOTAK SEBAGAI PEMBATAS
 - (e). FUNGSI TERSAMPLING YANG DIBATASI
 - (f). FUNGSI IMPULSE UNTUK PENYAMPLING DAERAH FREKUENSI
 - (g). FUNGSI HASIL PROSES TRANSFORMASI FOURIER DISKRIT

17) Ibid, hal 92

Pada gambar tersebut ditunjukkan bahwa sinyal $h(t)$ dan $H(f)$ diubah sedemikian rupa sehingga kedua pasangan tersebut berbentuk sinyal tersampling yang periodik tetapi tetap menggambarkan bentuk sinyal aslinya.

Langkah-langkah perubahan itu adalah sebagai berikut:

1. Mula-mula sinyal $h(t)$ disampling dengan periode T . Pada gambar 2.9 (c) ditunjukkan timbulnya efek *aliasing* dari proses penyamplingan ini. Pasangan transformasi Fourier tersebut masih belum cocok untuk diterapkan pada komputer digital karena adanya proses sampling yang tak terhingga.
2. Untuk itu sinyal tersampling harus dibatasi sampai jumlah sample tertentu, misalkan disebut dengan N . Pembatasan itu dilakukan dengan cara mengalikan sinyal tersampling dengan sinyal kotak dengan lebar T_0 . Karena dalam sinyal kotak terdapat N sample dengan interval T maka didapat bahwa $N = T_0/T$. Pada gambar 2.9 (e) nampak timbul adanya ripple pada sinyal fungsi frekuensi sebagai efek dari pemotongan (*truncation*).
3. Hasil ini masih belum bisa diterapkan pada komputer digital karena sinyal fungsi frekuensi masih berupa sinyal kontinyu. Oleh karena itu sinyal fungsi frekuensi disampling dengan periode $1/T_0$.

agar didapat sinyal yang diskrit.

Hasil proses ini sudah dapat diterima dalam komputer digital karena baik sinyal fungsi waktu maupun sinyal fungsi frekuensi sudah mempunyai harga yang diskrit. Seperti diperlihatkan pada gambar 2.9 (g), baik sinyal fungsi frekuensi maupun sinyal fungsi waktu dibentuk melalui pendekatan dengan N sample. Juga kedua sinyal tersebut menjadi periodik dengan adanya penyamplingan pada sinyal fungsi waktu maupun pada sinyal fungsi frekuensi.

2.5.2. PENJELASAN TEORITIS

Secara teoritis pembentukan transformasi Fourier diskrit didasarkan pada penjelasannya secara grafis.

Pada langkah pertama sinyal $h(t)$ disampling dengan periode T . Hal ini bisa dituliskan secara matematis sebagai berikut: ¹⁸⁾

$$\begin{aligned} h(t)\Delta_0(t) &= h(t) \sum_{k=-\infty}^{\infty} [\delta(t-kT)] \\ &= \sum_{k=-\infty}^{\infty} [h(kT)\delta(t-kT)] \end{aligned} \quad (2-20)$$

dengan T : interval penyamplingan

$\Delta_0(t)$: fungsi impulse

Hasil dari perkalian ini ditunjukkan pada gambar 2.9 (c). Interval penyamplingan perlu dipilih sedemikian rupa agar efek *aliasing* dapat ditekan sekecil mungkin.

¹⁸⁾ Ibid, hal 94

Langkah berikutnya, sinyal tersampling dipotong dengan cara mengalikan sinyal tersebut dengan gelombang kotak. Gelombang kotak pada gambar 2.9 (d) memenuhi persamaan sebagai berikut:

$$x(t) = 1 \quad -T_0/2 < t < T_0/2$$

$$= 0 \quad \text{interval waktu yang lain}$$

dengan T_0 adalah lebar sinyal kotak tersebut. Jadi secara matematis langkah kedua dapat ditulis:

$$h(t)\Delta_0(t)x(t) = \left[\sum_{k=-\infty}^{\infty} [h(kT)\delta(t-kT)] \right] x(t)$$

$$= \sum_{k=0}^{N-1} h(kT)\delta(t-kT) \quad (2-21)$$

Langkah terakhir adalah penyamplingan sinyal fungsi frekuensi dengan interval sampling $1/T_0$. Misalkan $\Delta_1(f)$ adalah sinyal sampling fungsi frekuensi dengan interval $1/T_0$ maka transformasi Fourier inverse dari sinyal sampling tersebut juga berupa sinyal sampling tetapi dengan interval sampling sebesar T_0 dan dituliskan sebagai berikut:

$$\Delta_1(t) = T_0 \sum_{r=-\infty}^{\infty} \delta(t-rT_0)$$

Penyamplingan sinyal dalam daerah frekuensi sama dengan perkalian sinyal tersebut dengan sinyal sampling. Dan berdasarkan teori konvolusi frekuensi, di mana perkalian dalam daerah frekuensi sama dengan konvolusi dalam daerah

waktu, maka didapat hasil dari langkah terakhir ini sebagai berikut:

$$[h(t)\Delta_0(t)x(t)]*\Delta_1(t) = \left[\begin{array}{c} N-1 \\ \Sigma h(kT)\delta(t-kT) \\ k=0 \end{array} \right] * \left[T_0 \Sigma \delta(t-rT_0) \right] \quad (2-22)$$

Salah satu sifat fungsi impulse adalah: ¹⁹⁾

$$\delta(t-t_1)*\delta(t-t_2) = \delta(t-t_1-t_2) \quad (2-23)$$

Dengan menerapkan persamaan (2-23) pada persamaan (2-22) didapat:

$$h'(t) = T_0 \Sigma_{r=-\infty}^{\infty} \left[\begin{array}{c} N-1 \\ \Sigma h(kT)\delta(t-kT-rT_0) \\ k=0 \end{array} \right] \quad (2-24)$$

dimana $h'(t) = [h(t)\Delta_0(t)x(t)]*\Delta_1(t)$

$h'(t)$: pendekatan dari fungsi $h(t)$ dengan N sample
Persamaan (2-24) di atas menunjukkan adanya keperiodikan, yaitu dengan adanya pertambahan T_0 setiap pertambahan r .

Untuk mendapatkan sinyal diskrit pada daerah waktu perlu dilihat lagi penjelasan mengenai transformasi Fourier untuk fungsi periodik (sub bab 2.4.1). Jika persamaan (2-15) diterapkan pada persamaan (2-24) maka didapat:

$$H' \left(\frac{n}{T_0} \right) = \Sigma_{n=-\infty}^{\infty} \alpha_n \delta(f - nf_0) \quad f_0 = 1/T_0 \quad (2-25)$$

dengan

$$\alpha_n = \frac{1}{T_0} \int_{-T_0/2}^{+T_0/2} h'(t) e^{-j2\pi n t/T_0} dt \quad (2-26)$$

$$n = 0, \pm 1, \pm 2, \dots$$

¹⁹⁾ Ibid, hal 229

Dengan mensubstitusikan persamaan (2-24) ke persamaan (2-26) didapat:

$$\alpha_n = \frac{1}{T_0} \int_{-T_0/2}^{+T_0/2} \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} h(kT) \delta(t-kT-rT_0) e^{-j2\pi n t/T_0} dt$$

Karena integrasi hanya dilakukan untuk satu periode maka diambil satu periode saja untuk $r=0$.

$$\begin{aligned} \alpha_n &= \int_{-T_0/2}^{+T_0/2} \sum_{k=0}^{N-1} h(kT) \delta(t-kT) e^{-j2\pi n t/T_0} dt \\ &= \sum_{k=0}^{N-1} h(kT) \int_{-T_0/2}^{+T_0/2} \delta(t-kT) e^{-j2\pi n t/T_0} dt \end{aligned}$$

Salah satu sifat fungsi impulse adalah: 20)

$$\int_{-\infty}^{+\infty} \delta(t-t_0) \phi(t) dt = \phi(t_0)$$

Dengan menerapkan persamaan itu dan mengganti T_0/T dengan N didapat:

$$\alpha_n = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi k n/N} dt \quad n = 0, \pm 1, \pm 2, \dots \quad (2-27)$$

dan didapat transformasi Fourier dari persamaan (2-24)

$$H' \left(\frac{n}{NT} \right) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} h(kT) e^{-j2\pi k n/N} \delta(f - n f_0)$$

Karena fungsi-fungsi di depannya tidak mengandung f maka fungsi impulse $\delta(f - n f_0)$ dapat dieliminasi sehingga didapatkan:

$$H' \left(\frac{n}{NT} \right) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} h(kT) e^{-j2\pi k n/N} \quad (2-28)$$

Persamaan di atas masih belum menunjukkan adanya keperio-

20) Ibid, hal 228

dikan. Misalkan harga $n = r$ maka

$$H' \left(\frac{r}{NT} \right) = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kr/N}$$

Sekarang jika dimasukkan $n = r + N$ didapat

$$H' \left(\frac{r+N}{NT} \right) = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi k(r+N)/N}$$

$$\begin{aligned} e^{-j2\pi k(r+N)/N} &= e^{-j2\pi kr/N} / e^{-j2\pi k} \\ &= e^{-j2\pi kr/N} \end{aligned}$$

Karena $e^{-j2\pi k} = \cos(2\pi k) - j\sin(2\pi k) = 1$ untuk harga k integer maka untuk $n=r+N$ didapat:

$$\begin{aligned} H' \left(\frac{r+N}{NT} \right) &= \sum_{k=0}^{N-1} h(kT) e^{-j2\pi kr/N} \\ &= H' \left(\frac{r}{NT} \right) \end{aligned}$$

Dari penurunan tersebut dapat disimpulkan bahwa $H'(n/NT)$ periodik setiap N sample dan dapat ditulis:

$$H' \left(\frac{n}{NT} \right) = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi nk/N} \quad n = 0, 1, 2, \dots, N-1$$

(2-29)

Persamaan (2-29) di atas merupakan transformasi Fourier persamaan (2-24) secara lengkap yang dapat diproses pada komputer digital.

Notasi $H'(n/NT)$ digunakan untuk menunjukkan bahwa transformasi Fourier diskrit menghasilkan pendekatan untuk

transformasi Fourier kontinyu. Secara umum, transformasi Fourier diskrit ditulis sebagai:

$$G\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} g(kT) e^{-j2\pi nk/N} \quad n = 0, 1, 2, \dots, N-1 \quad (2-30)$$

karena transformasi Fourier dari fungsi tersampling $g(kT)$ identik dengan $G(n/NT)$.

2.5.3. TRANSFORMASI FOURIER INVERSE DISKRIT

Transformasi Fourier inverse diskrit dirumuskan sebagai: 21)

$$g(kT) = \frac{1}{N} \sum_{n=0}^{N-1} G\left(\frac{n}{NT}\right) e^{j2\pi nk/N} \quad k=0, 1, \dots, N-1 \quad (2-31)$$

Rumus tersebut bisa dibuktikan dengan cara memasukkannya langsung ke persamaan (2-30).

$$\begin{aligned} G\left(\frac{n}{NT}\right) &= \sum_{k=0}^{N-1} \left[\frac{1}{N} \sum_{r=0}^{N-1} G\left(\frac{r}{NT}\right) e^{j2\pi rk/N} \right] e^{-j2\pi nk/N} \\ &= \frac{1}{N} \sum_{r=0}^{N-1} G\left(\frac{r}{NT}\right) \left[\sum_{k=0}^{N-1} e^{j2\pi rk/N} e^{-j2\pi nk/N} \right] \\ &= G\left(\frac{n}{NT}\right) \quad n=0, 1, \dots, N-1 \end{aligned}$$

21) Ibid, hal 98

2.5.4. PENGARUH SAMPLING DAN PEMOTONGAN PADA TRANSFORMASI FOURIER DISKRIT

Dalam pengolahan sinyal dari kontinyu menjadi diskrit ada dua proses yang menimbulkan perubahan pada bentuk sinyal. Kedua proses itu adalah proses *sampling* dan proses *pemotongan (truncation)*. Efek dari proses *sampling* adalah timbulnya *aliasing* sedangkan efek dari proses *pemotongan* adalah timbulnya *ripple* pada hasil transformasi Fourier.

Dalam proses *sampling* yang perlu diperhatikan adalah frekuensi tertinggi sinyal yang akan ditransformasi.

Efek *aliasing* tidak akan timbul bila frekuensi *sampling* lebih besar dari 2 kali frekuensi tertinggi sinyal yang akan disampling. Dalam penerapannya, sinyal input belum tentu diketahui frekuensinya sehingga frekuensi *sampling* dibuat sebesar mungkin. Jika frekuensi *sampling* besar maka jumlah *sampling* juga bertambah. Penambahan jumlah *sampling* akan memperbaiki ketelitian pengolahan tetapi juga memperpanjang waktu pengolahan data.

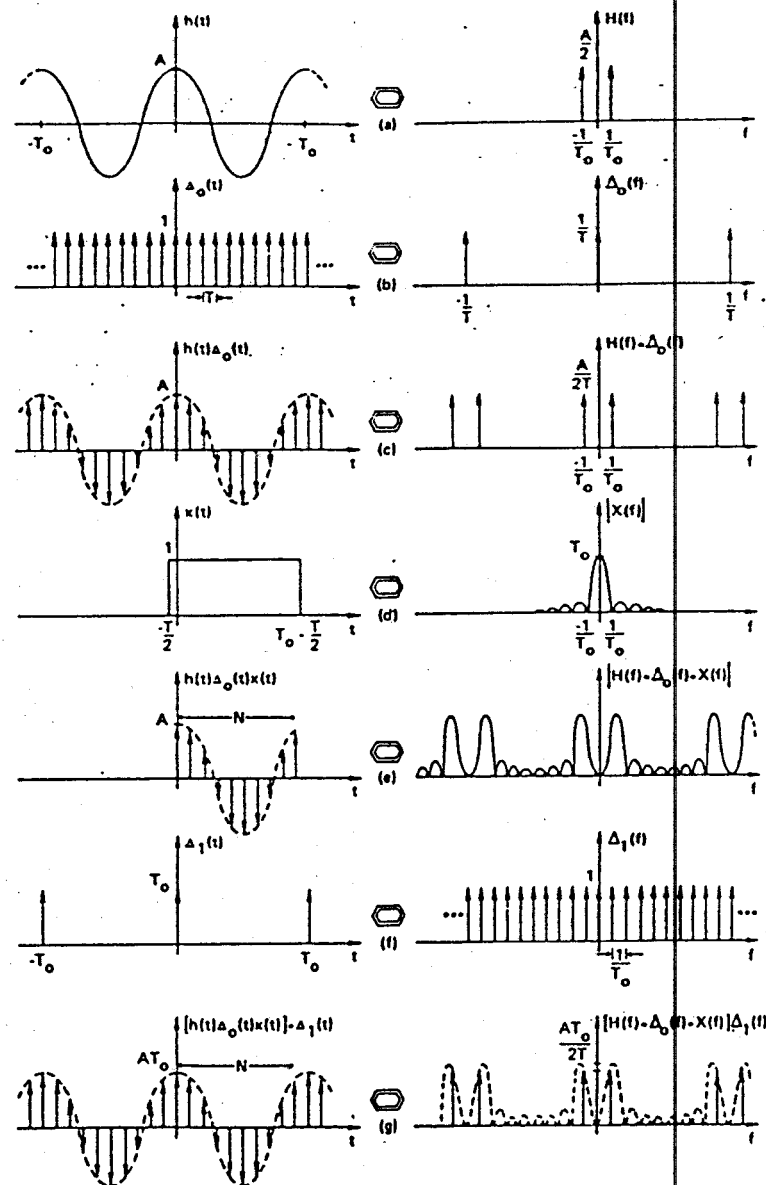
Ada tiga kasus berbeda yang berkenaan dengan pemotongan sinyal yaitu $h(t)$ (fungsi waktu) periodik dan interval pemotongan merupakan kelipatan dari periode $h(t)$, $h(t)$ periodik dan interval pemotongan tidak sama dengan kelipatan periode $h(t)$, $h(t)$ tidak periodik.

2.5.4.1. Pemotongan untuk sinyal periodik

Lebar pulsa sama dengan kelipatan periode sinyal input

Pada gambar 2.10 ditunjukkan proses pemotongan pada sinyal periodik dengan interval yang sama dengan periode sinyal input.

Pada gambar 2.10 (a) ditunjukkan sinyal sinus dan transformasi Fouriernya. Amplitudo transformasi Fouriernya adalah setengah dari amplitudo sinus. Setelah disampling dengan sinyal impulse maka amplitudo berubah menjadi $1/T$ amplitudo semula. Pada gambar 2.10 (c) nampak bahwa hasil transformasi Fourier tidak menunjukkan aliasing sebab frekuensi sampling cukup besar. Kemudian sinyal tersampling dipotong tepat satu periode dengan cara mengalikan-nya dengan sinyal kotak yang intervalnya tepat sama dengan periode sinyal input. Transformasi Fourier dari fungsi kotak seperti pada gambar 2.10 (d) adalah $T_0 \sin(f)/f$. Konvolusi fungsi ini dengan fungsi impulse pada gambar 2.10 (d) menghasilkan fungsi seperti pada gambar 2.10 (e). Pada gambar ini tampak adanya distorsi. Distorsi ini akan hilang bila sinyal pada gambar 2.10 (e) dikalikan dengan sinyal sampling dengan interval $1/T_0$ karena sinyal pada gambar 2.10 (e) bernilai nol untuk setiap kelipatan $1/T_0$ kecuali pada frekuensi $1/T_0$. Sinyal impulse pada gambar 2.10 (g) ini mempunyai amplitudo sebesar $AT_0/(2T)$.



GAMBAR 2.10 22)

TRANSFORMASI FOURIER DISKRIT DENGAN INTERVAL PEMOTONGAN
SAMA DENGAN PERIODE SINYAL INPUT

- (a). FUNGSI ASAL PERIODIK
- (b). FUNGSI IMPULSE SEBAGAI PENYAMPLING PADA DAERAH WAKTU
- (c). FUNGSI TERSAMPLING PERIODIK
- (d). FUNGSI KOTAK DENGAN LEBAR SAMA DENGAN T_0
- (e). FUNGSI TERSAMPLING TEPAT SATU PERIODE
- (f). FUNGSI IMPULSE PENYAMPLING DAERAH FREKUENSI
- (g). FUNGSI HASIL IDFT SAMA DENGAN FUNGSI ASAL TERSAMPLING

Jadi untuk mendapatkan amplitudo sebenarnya dari hasil transformasi Fourier diskrit maka amplitudo sinyal tersebut harus dikalikan dengan T/T_0 . Dalam penerapannya, faktor pengali $1/T_0$ bisa dihilangkan dengan mengalikan lebih dulu tiap-tiap sample dengan T_0 sebelum dilakukan pengolahan data. Kemudian setelah pengolahan data selesai barulah perkalian dengan faktor pengali T dilakukan. Dengan adanya faktor pengali T ini maka persamaan (2-30) berubah menjadi:

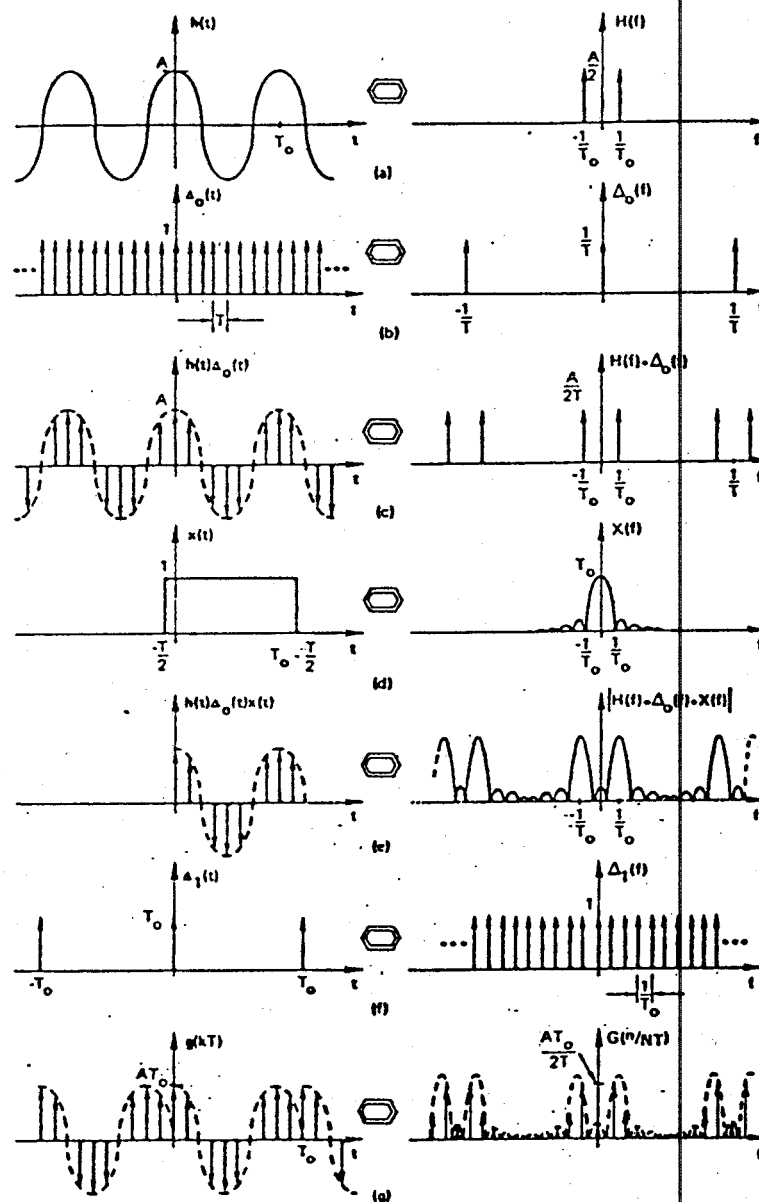
$$G\left(\frac{n}{NT}\right) = T \sum_{k=0}^{N-1} g(kT) e^{-j2\pi kr/N} \quad (2-32)$$

Kasus ini merupakan satu-satunya kasus di mana hasil transformasi Fourier diskrit sama dengan hasil transformasi Fourier kontinyu dengan skala konstan. Syarat kasus ini adalah:

1. Sinyal fungsi waktu $h(t)$ harus periodik.
2. Frekuensi sampling minimum dua kali frekuensi maksimum dari $h(t)$.
3. Sinyal fungsi waktu $h(t)$ harus dibatasi.
4. Lebar interval fungsi kotak harus kelipatan periode sinyal $h(t)$.

Lebar pulsa tidak sama dengan kelipatan periode sinyal input

Jika lebar pulsa tidak sama dengan kelipatan periode fungsi input maka hasil transformasi Fourier diskrit



GAMBAR 2.11 23)

TRANSFORMASI FOURIER DISKRIT DENGAN INTERVAL PEMOTONGAN
TIDAK SAMA DENGAN PERIODE SINYAL INPUT

- (a). FUNGSI ASAL PERIODIK
- (b). FUNGSI IMPULSE SEBAGAI PENYAMPLING PADA DAERAH WAKTU
- (c). FUNGSI TERSAMPLING PERIODIK
- (d). FUNGSI KOTAK DENGAN LEBAR TIDAK SAMA DENGAN T_0
- (e). FUNGSI TERSAMPLING TIDAK TEPAT SATU PERIODE
- (f). FUNGSI IMPULSE PENYAMPLING DAERAH FREKUENSI
- (g). FUNGSI HASIL IDFT LAIN DENGAN FUNGSI ASAL TERSAMPLING

berbeda dengan hasil transformasi Fourier kontinyu. Pada gambar 2.11 diperlihatkan perbedaan itu.

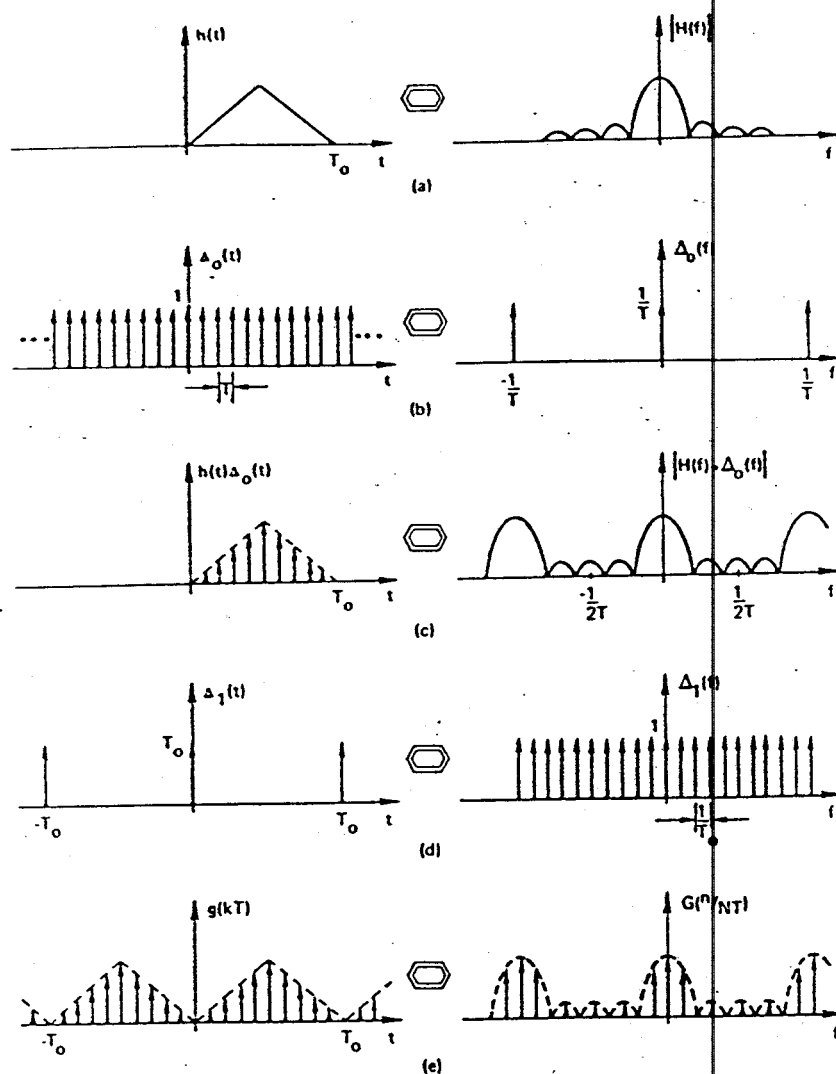
Perbedaan dengan gambar 2.10 terletak pada gambar 2.11 (d) yaitu lebar interval fungsi kotak tidak sama dengan kelipatan periode fungsi sinus gambar 2.11 (a). Akibatnya hasil transformasi Fourier diskrit memiliki komponen pada frekuensi kelipatan $1/T_0$ selain $\pm 1/T_0$.

Jika hasil transformasi Fourier pada gambar 2.11 (g) dikembalikan ke fungsi waktu dengan transformasi Fourier inverse diskrit maka akan dihasilkan fungsi sinus yang diskontinyu akibat adanya komponen frekuensi tambahan. Efek ini disebut dengan *leakage* dan timbul karena proses pemotongan.

2.5.4.2. Pemotongan untuk sinyal yang nonperiodik

Untuk bentuk gelombang yang umum maka hasil transformasi Fouriernya tetap akan berupa sinyal yang periodik. Permasalahan yang dihadapi tetap sama yaitu adanya efek dari sampling yaitu *aliasing* dan efek dari pemotongan yaitu *leakage*. Seperti pada kasus sebelumnya, efek aliasing bisa direduksi dengan memperbesar frekuensi sampling.

Pada gambar 2.12 berikut ditunjukkan proses transformasi Fourier diskrit untuk bentuk gelombang yang tidak periodik dan hasil transformasi Fourier-nya yang periodik.



GAMBAR 2.12 24)

TRANSFORMASI FOURIER DISKRIT UNTUK SINYAL INPUT
YANG TIDAK PERIODIK

- (a). FUNGSI ASAL TIDAK PERIODIK
- (b). FUNGSI IMPULSE SEBAGAI PENYAMPLING PADA DAERAH WAKTU
- (c). FUNGSI TERSAMPLING
- (d). FUNGSI IMPULSE PENYAMPLING DAERAH FREKUENSI
- (e). FUNGSI HASIL IDFT MENJADI PERIODIK

2.6. ALGORITMA FAST FOURIER TRANSFORM

Fast Fourier Transform merupakan algoritma untuk menghitung transformasi Fourier diskrit dengan mereduksi banyaknya perkalian sehingga transformasi Fourier diskrit bisa diperoleh dengan lebih cepat.

Hubungan transformasi Fourier diskrit adalah sebagai berikut:

$$G\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} g(kT)e^{-j2\pi nk/N} \quad n = 0, 1, 2, \dots, N-1$$

Untuk memudahkan penulisan maka kT ditulis sebagai k dan n/NT ditulis n sehingga persamaan di atas dapat ditulis:²⁵⁾

$$X(n) = \sum_{k=0}^{N-1} x(k)w^{nk} \quad n = 0, 1, 2, \dots, N-1 \quad (2-33)$$

dengan $W = \text{twiddle factor} = e^{-j2\pi/N}$

dan $N = \text{banyaknya data sampling.}$

Sebelum membahas lebih lanjut mengenai algoritma FFT maka lebih dulu dijelaskan tentang sifat-sifat *twiddle factor*.

Twiddle factor (W) mempunyai dua sifat penting yaitu:²⁶⁾

$$1. w^{nk} = -w^{(nk-N/2)} \quad (2-34)$$

Bukti:

$$w^{nk} = e^{-j2\pi nk/N} = \cos(2\pi nk/N) - j\sin(2\pi nk/N)$$

$$w^{(nk-N/2)} = e^{-j2\pi(nk-N/2)/N} = e^{-j(2\pi nk/N - \pi)}$$

$$= \cos(2\pi nk/N - \pi) - j\sin(2\pi nk/N - \pi)$$

$$= -\cos(2\pi nk/N) + j\sin(2\pi nk/N) = -w^{nk}$$

25) Ibid, hal 148

26) Ibid, hal 149

$$2. \quad w^{nk} = w^{nk \bmod(N)} \quad (2-35)$$

dengan $[nk \bmod(N)]$ adalah sisa hasil bagi nk dengan N .

Bukti:

Dari persamaan (2-34) didapat bahwa $w^{nk} = w^{(nk-N)}$. Jika pengurangan ini dilakukan terus menerus maka akan didapat persamaan (2-35).

Misalkan diambil $N=4$ data sampling maka persamaan (2-33) dapat dijabarkan sebagai berikut:

$$X(0) = x_0(0)w^0 + x_0(1)w^0 + x_0(2)w^0 + x_0(3)w^0$$

$$X(1) = x_0(0)w^0 + x_0(1)w^1 + x_0(2)w^2 + x_0(3)w^3$$

$$X(2) = x_0(0)w^0 + x_0(1)w^2 + x_0(2)w^4 + x_0(3)w^6$$

$$X(3) = x_0(0)w^0 + x_0(1)w^3 + x_0(2)w^6 + x_0(3)w^9$$

Secara ringkas keempat persamaan itu bisa ditulis dalam bentuk matrix sebagai berikut:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2-36)$$

atau secara ringkas ditulis

$$X(n) = w^{nk} x_0(k) \quad (2-37)$$

Dengan adanya hubungan $w^{nk} = w^{nk \bmod(N)}$ matriks (2-36) dapat ditulis menjadi:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & w^0 & w^0 \\ 1 & w^1 & w^2 & w^3 \\ 1 & w^2 & w^0 & w^1 \\ 1 & w^3 & w^2 & w^1 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2-38)$$

Melalui matriks itu dapat ditunjukkan bahwa banyaknya perkalian untuk memperoleh DFT dengan jumlah sample N adalah N^2 dan banyaknya penjumlahan adalah $N(N-1)$.

2.6.1. PENGEMBANGAN TEORITIS ALGORITMA FFT BASIS 2

Dalam sub bab ini akan diterangkan bagaimana pengembangan secara teoritis algoritma FFT yang paling sederhana yaitu FFT basis 2. Yang dimaksud dengan basis disini adalah konstanta dasar jumlah sample (N). Dalam algoritma FFT basis 2, jumlah sample harus merupakan hasil eksponensial bilangan integer dari 2. Misalkan bilangan integer itu disebut r maka $N = 2^r$.

Berikut ini akan diturunkan algoritma FFT basis 2 dengan jumlah sample (N) = 4. Untuk $N=4$ maka $k = 0, 1, 2, 3$ dan $n = 0, 1, 2, 3$. Mula-mula bilangan tersebut harus dinyatakan dalam bilangan biner 2 bit sehingga didapat:

$k = (k_1, k_0) = 00, 01, 10, 11$ dan $n = (n_1, n_2) = 00, 01, 10, 11$
dengan $k = 2k_1 + k_0$ dan $n = 2n_1 + n_0$ dimana k_0, k_1, n_0, n_1 hanya bisa berharga 1 dan 0. Dengan notasi seperti di atas maka

$$2. w^{nk} = w^{nk \bmod(N)} \quad (2-35)$$

dengan $[nk \bmod(N)]$ adalah sisa hasil bagi nk dengan N .

Bukti:

Dari persamaan (2-34) didapat bahwa $w^{nk} = w^{(nk-N)}$. Jika pengurangan ini dilakukan terus menerus maka akan didapat persamaan (2-35).

Misalkan diambil $N=4$ data sampling maka persamaan (2-33) dapat dijabarkan sebagai berikut:

$$X(0) = x_0(0)w^0 + x_0(1)w^0 + x_0(2)w^0 + x_0(3)w^0$$

$$X(1) = x_0(0)w^0 + x_0(1)w^1 + x_0(2)w^2 + x_0(3)w^3$$

$$X(2) = x_0(0)w^0 + x_0(1)w^2 + x_0(2)w^4 + x_0(3)w^6$$

$$X(3) = x_0(0)w^0 + x_0(1)w^3 + x_0(2)w^6 + x_0(3)w^9$$

Secara ringkas keempat persamaan itu bisa ditulis dalam bentuk matrix sebagai berikut:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2-36)$$

atau secara ringkas ditulis

$$X(n) = w^{nk} x_0(k) \quad (2-37)$$

Dengan adanya hubungan $w^{nk} = w^{nk \bmod(N)}$ matriks (2-36) dapat ditulis menjadi:

persamaan (2-34) dapat ditulis sebagai berikut (N=4): ²⁷⁾

$$X(n_1, n_0) = \sum_{k_0=0}^1 \sum_{k_1=0}^1 x_0(k_1, k_0) w^{(2n_1+n_0)(2k_1+k_0)} \quad (2-39)$$

Faktor $w^{(2n_1+n_0)(2k_1+k_0)}$ dapat ditulis sebagai berikut:

$$\begin{aligned} w^{(2n_1+n_0)(2k_1+k_0)} &= w^{(4n_1k_1+2n_0k_1+(2n_1+n_0)k_0)} \\ &= [w^{4n_1k_1}] w^{2n_0k_1} w^{(2n_1+n_0)k_0} \\ &= w^{2n_0k_1} w^{(2n_1+n_0)k_0} \end{aligned} \quad (2-40)$$

Faktor yang didalam kurung siku berharga 1 karena:

$$w^{4n_1k_1} = [w^4]^{n_1k_1} = [e^{-j2\pi 4/4}]^{n_1k_1} = [1]^{n_1k_1} = 1$$

Persamaan (2-39) dapat ditulis menjadi:

$$X(n_1, n_0) = \sum_{k_0=0}^1 \left[\sum_{k_1=0}^1 x_0(k_1, k_0) w^{2n_0k_1} \right] w^{(2n_1+n_0)k_0} \quad (2-41)$$

Pengembangan selanjutnya adalah pemecahan $X(n_1, n_0)$ menjadi dua faktor yaitu faktor yang di dalam kurung siku dan di luarnya. Faktor pertama yang di dalam kurung siku dapat dituliskan sebagai berikut:

$$x_1(n_1, n_0) = \sum_{k_1=0}^1 x_0(k_1, k_0) w^{2n_0k_1} \quad (2-42)$$

Persamaan di atas dapat dijabarkan menjadi:

$$\begin{aligned} x_1(0, 0) &= x_0(0, 0) + x_0(1, 0)w^0 \\ x_1(0, 1) &= x_0(0, 1) + x_0(1, 1)w^0 \\ x_1(1, 0) &= x_0(0, 0) + x_0(1, 0)w^2 \\ x_1(1, 1) &= x_0(0, 1) + x_0(1, 1)w^2 \end{aligned} \quad (2-43)$$

Persamaan (2-43) dapat dituliskan dalam bentuk matriks:

²⁷⁾ Ibid, hal 172

$$\begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^0 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{bmatrix} \begin{bmatrix} x_0(0,0) \\ x_0(0,1) \\ x_0(1,0) \\ x_0(1,1) \end{bmatrix} \quad (2-44)$$

Faktor yang kedua dituliskan sebagai berikut:

$$x_2(n_0, n_1) = \sum_{k_0=0}^{(2n_1+n_0)k_0} x_1(n_0, k_0) w^{k_0} \quad (2-45)$$

dan ditulis dalam bentuk matriks sebagai berikut:

$$\begin{bmatrix} x_2(0,0) \\ x_2(0,1) \\ x_2(1,0) \\ x_2(1,1) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w^1 \\ 0 & 0 & 1 & w^3 \end{bmatrix} \begin{bmatrix} x_1(0,0) \\ x_1(0,1) \\ x_1(1,0) \\ x_1(1,1) \end{bmatrix} \quad (2-46)$$

Persamaan (2-44) dan (2-46) digabungkan menjadi:

$$\begin{bmatrix} x_2(0,0) \\ x_2(0,1) \\ x_2(1,0) \\ x_2(1,1) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w^1 \\ 0 & 0 & 1 & w^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^0 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{bmatrix} \begin{bmatrix} x_0(0,0) \\ x_0(0,1) \\ x_0(1,0) \\ x_0(1,1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & w^0 & w^0 & w^0 \\ 1 & w^2 & w^0 & w^2 \\ 1 & w^1 & w^2 & w^3 \\ 1 & w^3 & w^2 & w^5 \end{bmatrix} \begin{bmatrix} x_0(0,0) \\ x_0(0,1) \\ x_0(1,0) \\ x_0(1,1) \end{bmatrix} \quad (2-47)$$

Persamaan (2-47) akan sama dengan persamaan (2-38) jika

letak bit-bit n_0 dan n_1 lebih dulu dibalik (*scrambled*) sehingga didapat:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} x_2(0,0) \\ x_2(1,0) \\ x_2(0,1) \\ x_2(1,1) \end{bmatrix} \quad (2-48)$$

Sekarang akan dilihat lagi banyaknya perkalian. Mula-mula dihitung banyaknya perkalian pada persamaan (2-44).

$$\begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & w^0 & 0 \\ 0 & 1 & 0 & w^0 \\ 1 & 0 & w^2 & 0 \\ 0 & 1 & 0 & w^2 \end{bmatrix} \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix}$$

Pada baris pertama terdapat satu perkalian dan satu penjumlahan.

$$x_1(0) = x_0(0) + w^0 x_0(2)$$

Pada baris ketiga hanya terdapat satu penjumlahan karena faktor $w^2 x_0(2)$ sama dengan $-w^0 x_0(2)$ dan sudah dihitung pada baris pertama.

$$x_1(2) = x_0(0) + w^2 x_0(2) = x_0(0) - w^0 x_0(2)$$

Dengan cara yang sama didapat bahwa $x_1(1)$ dan $x_1(3)$ diperoleh dengan dua penjumlahan dan satu perkalian. Karena itu $x_1(0)$ sampai dengan $x_1(3)$ dapat diperoleh dengan dua perkalian dan empat penjumlahan. Selanjutnya

dihitung banyaknya perkalian pada persamaan (2-46).

$$\begin{bmatrix} x_2(0) \\ x_2(1) \\ x_2(2) \\ x_2(3) \end{bmatrix} = \begin{bmatrix} 1 & w^0 & 0 & 0 \\ 1 & w^2 & 0 & 0 \\ 0 & 0 & 1 & w^1 \\ 0 & 0 & 1 & w^3 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix}$$

Baris pertama dihitung melalui satu penjumlahan dan satu perkalian.

$$x_2(0) = x_1(0) + w^0 x_1(1)$$

Baris kedua dihitung melalui satu penjumlahan saja karena $w^2 = -w^0$ sehingga $x_2(1) = x_1(0) - w^0 x_1(1)$ dan faktor $w^0 x_1(1)$ sudah dihitung pada baris pertama.

Dengan cara yang sama didapat baris ketiga melalui satu penjumlahan dan satu perkalian sedang baris keempat didapat melalui satu penjumlahan saja. Jadi banyaknya perkalian semuanya adalah 4 dan penjumlahannya 8. Bandingkan dengan penghitungan DFT secara langsung yang memerlukan perkalian sebanyak 16 dan penjumlahan sebanyak 12.

Secara umum untuk jumlah sample sebanyak $N = 2^r$ maka perhitungan $X(n)$ dilakukan dengan memfaktorkan matriks $N \times N$ menjadi r matriks. Pada masing-masing faktor matriks terdapat $N/2$ perkalian dan N penjumlahan sehingga untuk menghitung $X(n)$ diperlukan $Nr/2$ perkalian dan Nr penjumlahan.

2.6.2. PENURUNAN ALGORITMA COOLEY-TUKEY UNTUK $N = 2^r$

Penjelasan di atas merupakan penerapan dari algoritma Cooley-Tukey untuk $r = 2$ ($N = 4$). Secara umum algoritma Cooley Tukey diturunkan sebagai berikut. ²⁸⁾

Jika $N = 2^r$ maka n dan k dapat dinyatakan dalam bentuk biner:

$$n = 2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0$$

$$k = 2^{r-1}k_{r-1} + 2^{r-2}k_{r-2} + \dots + k_0$$

Dengan bentuk tersebut persamaan (2-33) dapat ditulis:

$$X(n_{r-1}, n_{r-2}, \dots, n_0) = \sum_{k_0=0}^1 \sum_{k_1=0}^1 \dots \sum_{k_{r-1}=0}^1 X(k_{r-1}, k_{r-2}, \dots, k_0) W^p \quad (2-49)$$

dengan

$$p = (2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) (2^{r-1}k_{r-1} + 2^{r-2}k_{r-2} + \dots + k_0)$$

W^p dapat ditulis sebagai berikut:

$$\begin{aligned} W^p &= W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) (2^{r-1}k_{r-1} + 2^{r-2}k_{r-2} + \dots + k_0)} \\ &= W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) (2^{r-1}k_{r-1})} \\ &\quad \times W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) (2^{r-2}k_{r-2})} \\ &\quad \times \dots \times W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) k_0} \end{aligned} \quad (2-50)$$

Faktor pertama dari W^p :

$$\begin{aligned} W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0) (2^{r-1}k_{r-1})} &= [W^{2^r(2^{r-2}n_{r-1}k_{r-1})} \quad W^{2^r(2^{r-3}n_{r-2}k_{r-1})} \\ &\quad \times \dots \times W^{2^r n_1 k_{r-1}} \quad W^{2^r n_0 k_{r-1}}] W \end{aligned}$$

²⁸⁾ Ibid, hal 176 s.d. 177

$$= W^{2^{r-1}n_0k_{r-1}} \quad (2-51)$$

Faktor yang berada di dalam kurung siku berharga satu karena $2^r = N$ dan $w^{aN} = w^0 = 1$ bila a bilangan integer.

Dengan cara yang sama didapat faktor kedua:

$$W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0)(2^{r-2}k_{r-2})} = W^{(2n_1 + n_0)2^{r-1}k_{r-1}} \quad (2-52)$$

Proses di atas diulang terus sampai tidak ada lagi faktor yang mengandung w^{2^r} . Dengan hasil itu maka persamaan (2-49) dapat ditulis:

$$X(n_{r-1}, n_{r-2}, \dots, n_0) = \sum_{k_0=0}^1 \sum_{k_1=0}^1 \dots \sum_{k_{r-1}=0}^1 x_0(k_{r-1}, k_{r-2}, \dots, k_0) \\ \times W^{2^{r-1}n_0k_{r-1}} \times W^{(2n_1+n_0)2^{r-1}k_{r-1}} \\ \times \dots \times W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0)k_0} \quad (2-53)$$

Persamaan (2-53) ini merupakan rumus umum untuk algoritma FFT Cooley-Tukey basis 2 yang bila dijabarkan didapat:

$$x_1(n_0, k_{r-2}, \dots, k_0) = \sum_{k_{r-1}=0}^1 x_0(k_{r-1}, k_{r-2}, \dots, k_0) W^{2^{r-1}(n_0k_{r-1})} \\ x_2(n_0, n_1, k_{r-3}, \dots, k_0) = \sum_{k_{r-2}=0}^1 x_1(n_0, k_{r-2}, \dots, k_0) W^{(2n_1+n_0)2^{r-2}k_{r-2}} \\ \vdots \\ x_r(n_0, n_1, \dots, n_{r-1}) = \sum_{k_0=0}^1 x_{r-1}(n_0, n_1, \dots, k_0) W^{(2^{r-1}n_{r-1} + 2^{r-2}n_{r-2} + \dots + n_0)k_0} \\ X(n_{r-1}, n_{r-2}, \dots, n_0) = x_r(n_0, n_1, \dots, n_{r-1}) \quad (2-54)$$

Untuk jumlah sample sebesar $N = 2^r$ maka akan terdapat r persamaan penjumlahan yang masing-masing terdiri dari N perkalian. Pada N persamaan ini tiap persamaan mengandung 2 perkalian kompleks tetapi salah satu perkalian selalu merupakan perkalian dengan satu sehingga setiap persamaan penjumlahan terdiri dari N perkalian. Jadi banyaknya perkalian untuk seluruh persamaan penjumlahan adalah Nr . Tetapi dengan adanya sifat dari *twiddle factor* yang pertama yaitu persamaan (2-34) $w^p = -w^{p+N/2}$, maka jumlah perkalian bisa direduksi lagi menjadi $Nr/2$. Dengan cara yang sama didapat bahwa banyaknya penjumlahan adalah Nr .

2.6.3. BENTUK LAIN ALGORITMA FFT

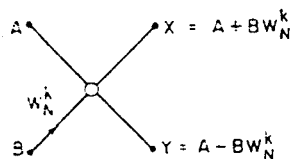
Algoritma FFT yang pertama kali dipublikasikan oleh Cooley dan Tukey kemudian dikembangkan lagi. Pengembangan pertama ialah dengan lebih dulu membalik (*scrambling*) data input sehingga hasil pengolahan mempunyai urutan yang benar. Algoritma ini dengan algoritma yang telah dijelaskan sebelumnya disebut dengan algoritma *FFT decimation in time*.

Selain algoritma *FFT decimation in time* juga terdapat algoritma lain yang disebut *decimation in frequency*. Pada algoritma ini operasi penjumlahan dilakukan lebih dulu. Seperti algoritma *FFT decimation in time* algoritma ini juga mempunyai dua bentuk yaitu bentuk di mana inputnya

berurutan dengan output yang terbalik urutannya dan bentuk di mana inputnya terbalik dengan output yang berurutan.

2.6.4. SIGNAL FLOW GRAPH

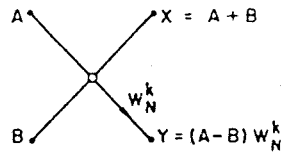
Persamaan (2-54) di atas dapat dinyatakan dalam bentuk gambar yang disebut dengan *Signal flow graph*. Dalam signal flow graph ini penjumlahan dinyatakan dalam bentuk node sedangkan perkalian dinyatakan dalam bentuk jalur. Bila pada jalur terdapat panah dengan notasi twiddle factor maka pangkal dari jalur itu harus lebih dulu dikalikan dengan twiddle factor tersebut sebelum dilakukan penjumlahan. Jika tidak ada panah maka perkaliannya dengan satu. Bentuk node dan jalur tersebut dinamakan *Butterfly*.



GAMBAR 2.13 29)

BUTTERFLY ALGORITMA FFT DECIMATION IN TIME

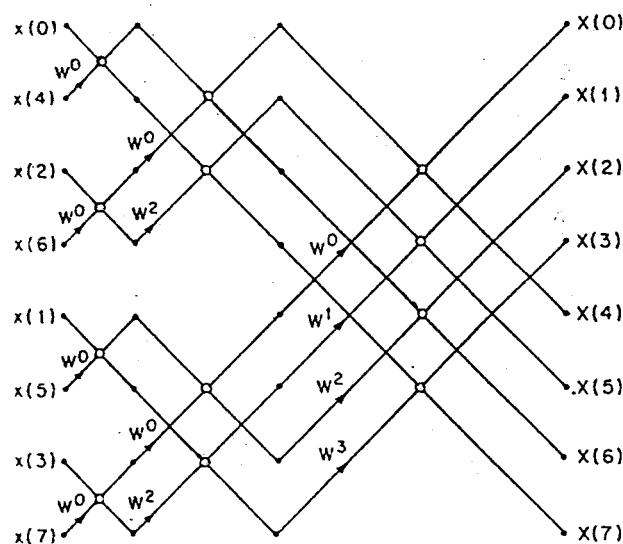
29) Rabiner, L.R. & Gold B., "Theory & Application of Digital Signal Processing", Prentice Hall, 19



GAMBAR 2.14 30)

BUTTERFLY ALGORITMA FFT DECIMATION IN FREQUENCY

Untuk menggambarkan signal flow graph ini mula-mula sample $x_0(k)$ dinyatakan sebagai kolom vertikal dari node yang terletak di ujung kiri. Di sebelah kanannya diletakkan kolom vertikal untuk $x_1(k)$. Demikian seterusnya sampai r kolom.

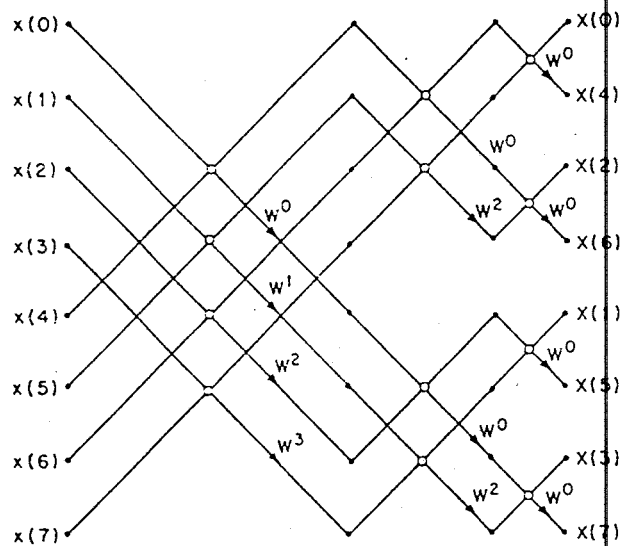


GAMBAR 2.15 31)

SIGNAL FLOW GRAPH FFT DECIMATION IN TIME UNTUK $N = 8$

30) Ibid, hal 371

31) Ibid, hal 362



GAMBAR 2.16 32)

SIGNAL FLOW GRAPH FFT DECIMATION IN FREQUENCY UNTUK $N = 8$

32) Ibid, hal 370

BAB III

DIGITAL SIGNAL PROCESSOR TMS32010 DAN INTERFACING-NYA PADA IBM PC/XT

3.1. PENDAHULUAN

Digital signal processor TMS32010 adalah 16/32 bit chip tunggal yang diperkenalkan pertama kali pada tahun 1983. Chip ini merupakan generasi pertama dari keluarga TMS320. Keluarga TMS320 ini dikembangkan lebih lanjut dengan munculnya chip-chip lain seperti TMS320C10, TMS320C15, TMS320E15, TMS320C17, TMS320E17 dan kemudian disusul dengan munculnya TMS32020, TMS320C20 dan TMS320C30 yang merupakan generasi kedua dan ketiga dari keluarga TMS320.

TMS32010 mempunyai beberapa kelengkapan dan kemampuan sebagai berikut: ³³⁾

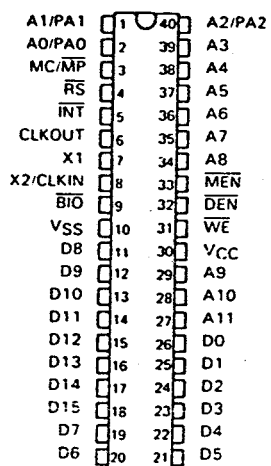
- Kecepatan pembacaan instruksi maksimum : 200 ns
- On-chip RAM : 144 word
- On-chip ROM : 1,5 K word
- Off-chip memory : 4 K word
- Data bus : 16 bit
- ALU/Accumulator : 32 bit
- I/O channel : 8 channel
- Tegangan supply : +5 V tunggal

³³⁾ Texas Instrument, "First-Generation TMS320 User's Guide", hal 1-5

- Teknologi pembuatan : NMOS
- Dilengkapi dengan 16 x 16 pengali paralel dengan hasil 32 bit
- Dilengkapi dengan 0 - 16 barrel shifter
- Dilengkapi dengan pembangkit clock

3.2. PIN-PIN TMS32010 DAN PENJELASANNYA

Pin-pin chip TMS32010 ditunjukkan pada gambar 3.1. berikut ini.



GAMBAR 3.1 ³⁴⁾

PIN-PIN TMS32010

Fungsi masing-masing pin tersebut adalah sebagai berikut:

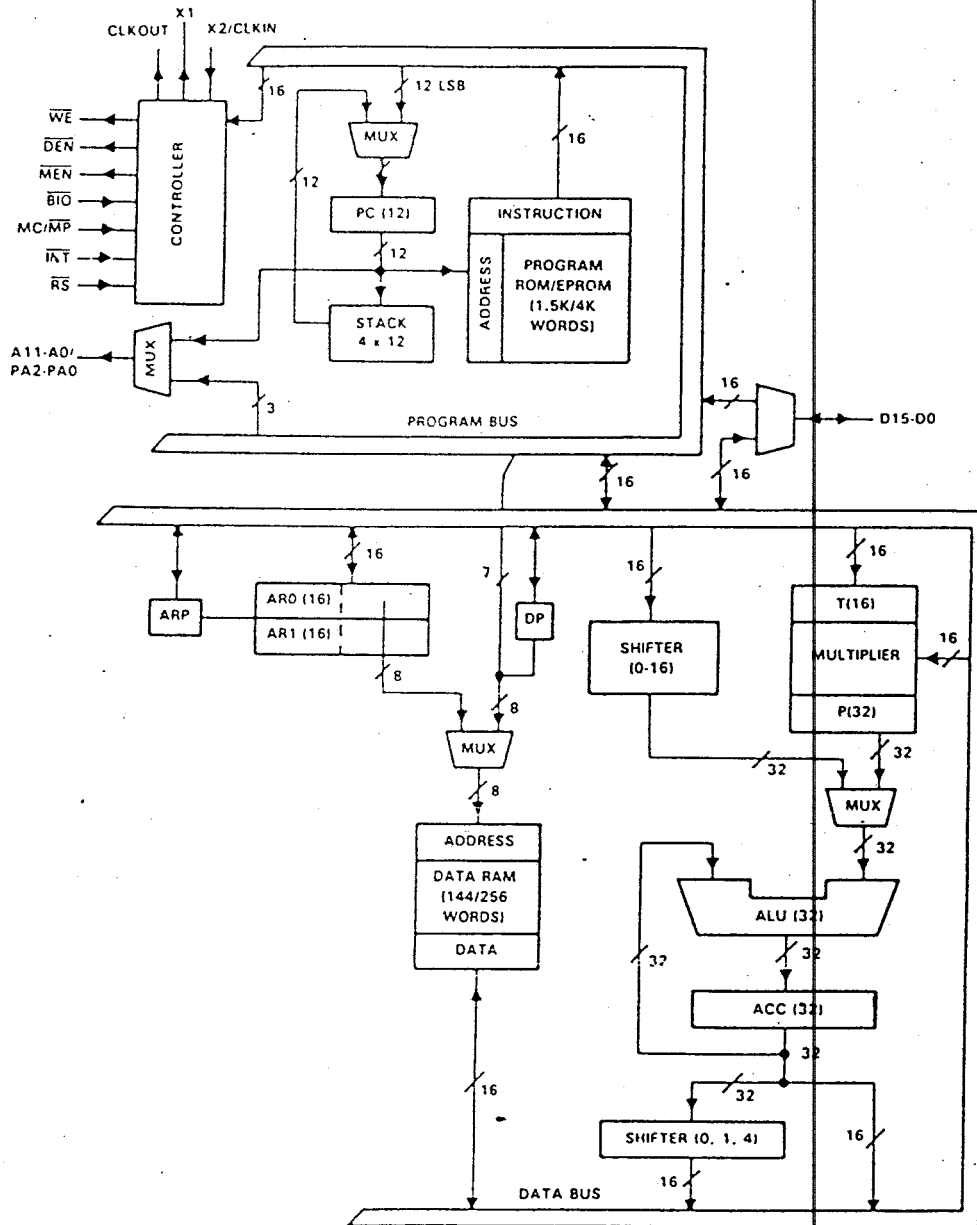
³⁴⁾ Ibid, hal 2-2

1. A_{11-A3} : Address bus. Pin output. Selalu aktif. Pada saat perintah IN atau OUT pin-pin ini selalu dalam keadaan low.
2. A_2/PA_2 : Berfungsi sebagai address bus A_2-A_0 pada
 A_1/PA_1 perintah selain IN dan OUT. Pada ins-
 A_0/PA_0 truksi IN atau OUT berfungsi sebagai address port PA_2-PA_0 .
3. D_{15-D0} : Data bus. Selalu dalam keadaan *high impedance* kecuali pada saat sinyal \overline{WE} low.
4. \overline{BIO} : Pin input yang digunakan sebagai polling dan berguna pada saat instruksi $BIOZ$.
5. \overline{DEN} : Pin output. Berguna sebagai data enable untuk peralatan input. Hanya aktif (low) pada saat perintah IN.
6. \overline{INT} : Pin input. Sinyal interrupt dibangkitkan jika ada trigger negative going edge.
7. MC/\overline{MP} : Pin input yang digunakan untuk menentukan mode kerja prosesor. Sinyal high pada pin ini menyebabkan prosesor bekerja sebagai mikrokomputer sedangkan sinyal low akan menyebabkan prosesor bekerja sebagai mikroprosesor.
8. \overline{MEN} : Pin output. \overline{MEN} selalu low untuk setiap cycle kecuali pada saat \overline{WE} aktif atau \overline{DE} aktif.
9. \overline{RS} : Pin input yang berfungsi untuk mereset

- peralatan.
10. \overline{WE} : Pin output. Dalam keadaan aktif menandakan bahwa data akan dikeluarkan dari prosesor.
 11. Vcc : Pin input untuk power supply 5 V.
 12. Vss : Pin input untuk ground.
 13. X2/CLKIN : Pin input yang dipergunakan sebagai input sinyal clock external (CLKIN) atau sebagai input clock internal (X2) yang dihubungkan pada kristal.
 14. X1 : Pin output yang merupakan output kristal pada oscillator internal.
 15. CLKOUT : Pin output yang berfungsi sebagai clock sistem. Frekuensinya sama dengan seperempat kali frekuensi kristal dengan *duty cycle* 50%.

3.3. BLOK DIAGRAM TMS32010

Pada gambar 3.2 ditunjukkan blok diagram prosesor TMS32010.



GAMBAR 3.2 35)
 BLOK DIAGRAM TMS32010

35) Ibid, hal 3-5

Fungsi tiap blok pada gambar 3.2 dijelaskan pada bagian berikut ini.

1. Auxiliary Register : Register 16-bit yang digunakan (AR0 dan AR1) sebagai penunjuk alamat data pada memory atau sebagai pengatur jumlah loop.
2. Auxiliary Register : Bit status yang menunjukkan auxiliary register mana yang aktif. Pointer (ARP)
3. Data bus : Bus 16-bit untuk jalur data dari (D15-D0) RAM.
4. Data Memory Page : Bit status yang menunjukkan *halaman* data RAM. (DP)
6. Data RAM : 144 word on-chip RAM yang berisi data.
7. Address bus : Address bus yang digunakan untuk Eksternal menentukan alamat memory eksternal. Tiga bit terbawah (PA0-PA2) digunakan juga sebagai alamat I/O port. (A11-A0/PA2-PA0)
8. Program bus : 16-bit bus yang digunakan sebagai (P15-P0) jalur instruksi dari program memory.
9. Program Counter : 12-bit register yang digunakan (PC11-PC0) sebagai penunjuk alamat program memory.

10. Program ROM : 1,5 K word on-chip ROM.
11. Stack : 4 x 12 stack yang digunakan untuk menyimpan isi PC selama terjadinya interrupt atau subroutine.
12. Interrupt Flag (INTF) : Bit flag yang menandakan bahwa sebuah permintaan interrupt terjadi.
13. Interrupt Mode (INTM) : Bit status untuk menghalangi interrupt flag.
14. Overflow Flag (OV) : Bit flag yang menandakan adanya *overflow* pada operasi arimatika.
15. Overflow Mode (OVM) : Bit status yang menyatakan mode *overflow* pada operasi aritmatika.
16. Status Register (ST) : 16-bit status register yang berisi bit-bit status dan kontrol.
17. Accumulator (ACC) : Register 32 bit yang digunakan sebagai penyimpan output dari ALU.
18. Arithmetic Logic Unit (ALU) : 32 bit ALU dengan sistem 2'S complement dengan dua 32-bit input dan satu 32-bit output.
19. Multiplier (MULT) : 16 x 16 pengali paralel
20. P register (P) : Register 32-bit yang berisi hasil operasi perkalian
21. T register (T) : Register 16-bit yang berisi pengali pada operasi perkalian.

22. Shifter (0-16) : *ALU barrel shifter* yang menggeser data memory ke kiri 0-16 bit saat dimasukkan pada ALU.
23. Shifter (0, 1, 4) : *Accumulator paralel shifter* yang berfungsi untuk menggeser isi accumulator 0, 1 atau 4 bit ke kiri dan menyimpan 15 bit MSB dari accumulator ke data RAM.
24. Central Arithmetic Unit (CALU) : Gabungan dari ALU, multiplier, accumulator dan shifter.

TMS32010 memiliki ALU dan Accumulator 32-bit yang menunjang operasi aritmatika 2'S complement presisi ganda. ALU bekerja dengan memakai data 16-bit yang diambil dari RAM atau data yang dimasukkan secara langsung (*immediate*). ALU juga dapat melakukan operasi Boolean sehingga dapat dilakukan manipulasi bit. Accumulator berfungsi untuk menyimpan output dari ALU atau sering juga digunakan untuk input ALU. Accumulator dibagi menjadi dua bagian yaitu bit 31 sampai dengan 16 yang disebut *high order word* dan bit 15 sampai dengan 0 yang disebut *low order word*.

Multiplier melakukan operasi perkalian 2'S complement 16 x 16 bit dengan hasil 32 bit yang dilaksanakan dalam satu cycle. Multiplier terdiri dari tiga elemen yaitu register T, register P dan multiplier array. Register T, yang terdiri dari 16 bit merupakan salah satu pengali

sedangkan register P berisi hasil perkalian sebesar 32 bit. Perkalian dilakukan dengan mengalikan register T dengan pengali kedua dan hasilnya disimpan pada register P. Pengali kedua dapat berupa data RAM atau data yang dimasukkan secara langsung dari program (*immediate*).

Dua buah shifter disediakan untuk manipulasi data. ALU barrel shifter berfungsi untuk menggeser ke kiri 0 sampai 16 bit data memory yang akan dimasukkan ke ALU. Accumulator paralel shifter berfungsi untuk menggeser seluruh isi accumulator 0, 1 atau 4 bit ke kiri dan menyimpan *high-order word* dari accumulator pada data RAM.

TMS32010 menawarkan dua mode operasi yaitu mode mikrokomputer dan mode mikroprosesor. Mode ini diatur melalui pin MC/ $\overline{\text{MP}}$. Pada mode mikrokomputer disediakan 1,5 K word on-chip ROM yang dipergunakan untuk program memory. Selain itu dapat juga ditambahkan program memory eksternal sebesar 2,5 K word. Pada mode mikroprosesor program memory sebesar 4 K word seluruhnya eksternal. Baik pada mode mikroprosesor atau mikrokomputer disediakan on-chip data RAM sebesar 144 word.

TMS32010 menyediakan empat level stack yang diperlukan sebagai penyimpan sementara isi program counter pada saat terjadinya interrupt atau pemanggilan subroutine.

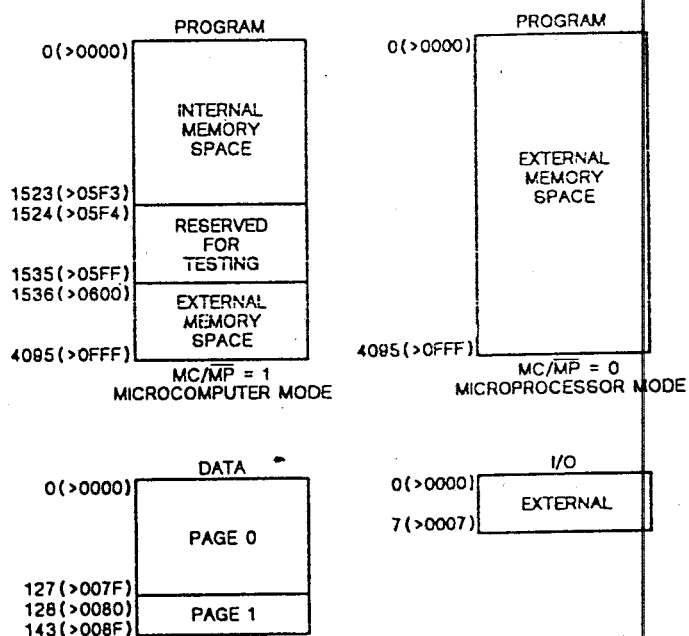
Untuk operasi input dan output TMS32010 menyediakan delapan channel dengan address yang dibentuk dari PA2, PA1 dan PA0. Operasi I/O ini dilaksanakan dalam dua cycle.

3.4. ORGANISASI MEMORY

Sistem memory pada TMS32010 dibedakan menjadi dua bagian yaitu program memory dan data memory. Program memory berisi instruksi program sedangkan data memory berisi data.

3.4.1. MEMORY MAP

TMS32010 menyediakan tiga alamat terpisah yang digunakan untuk program memory, data memory dan I/O.



GAMBAR 3.3 36)

MEMORY MAP TMS32010

Program memory mempunyai bentuk yang ditentukan oleh mode kerja TMS32010 yang ditentukan dari pin MC/MP. Pada mode mikrokomputer disediakan on-chip ROM sebesar 1,5 K word dan off-chip memory sekitar 2,5 K word. Dalam kedua mode disediakan RAM internal sebesar 144 word dan I/O sebanyak 8 channel. Pada gambar 3.3 diperlihatkan memory map TMS32010 dalam mode mikrokomputer dan mikroprosesor.

3.4.2. PROGRAM MEMORY

Telah disebutkan di atas bahwa bentuk program memory ditentukan oleh mode kerja prosesor. Dalam mode mikrokomputer disediakan on-chip ROM sebesar 1,5 K word untuk program memory dan 2,5 K word untuk eksternal memory. Perlu diperhatikan bahwa dalam mode ini alamat 1524 sampai dengan 1535 telah diisi oleh Texas Instrument, pembuat chip ini, dengan program untuk pengetesan. Dalam mode mikroprosesor tidak disediakan on-chip ROM tetapi disediakan eksternal memory sebesar 4 K word.

TMS32010 menyediakan 12 pin output yang digunakan untuk menunjukkan alamat 4 K word memory eksternal. Pin A11 sampai dengan A0 ini berisi harga program counter (PC) atau alamat I/O port. Ketika sebuah instruksi diambil dari memory eksternal, sinyal \overline{MEN} akan dibangkitkan. Sinyal \overline{MEN} selalu aktif dalam setiap cycle kecuali bila sinyal \overline{WE} atau \overline{DE} aktif yaitu pada saat perintah IN, OUT atau TBLW

dijalankan.

3.4.3. DATA MEMORY

TMS32010 menyediakan 144 word on-chip RAM yang digunakan sebagai data memory 16 bit. Jika kebutuhan data memory kurang besar maka data dapat disimpan dalam memory eksternal. Ada dua pasang instruksi yang dapat dipergunakan dalam pembacaan/penulisan data pada memory eksternal yaitu instruksi TBLR/TBLW dan IN/OUT. Instruksi TBLR (Table Read) berfungsi untuk memindahkan data dari program memory, baik pada on-chip ROM atau eksternal memory, ke on-chip data RAM eksternal. Sedangkan instruksi TBLW (Table Write) memindahkan data dari on-chip data RAM ke eksternal RAM. Kedua instruksi ditujukan untuk pemindahan data antar memory tanpa adanya penambahan komponen. Lain halnya dengan instruksi IN/OUT. Kedua instruksi ini sebenarnya ditujukan untuk pemindahan data I/O tetapi dengan sedikit penambahan hardware dapat juga dipakai untuk pemindahan data memory.

Penentuan alamat data memory dilakukan dengan dua cara. Cara pertama disebut *Direct addressing* yaitu dengan memasukkan alamat data memory sebesar 7 bit langsung melalui program. Untuk dapat mengakses 144 word data memory diperlukan 8 bit address sedangkan *direct addressing* hanya menyediakan 7 bit address sehingga diperlukan satu bit tambahan. Bit tambahan ini diperoleh dari bit

status Data Memory Page (DP). Data memory bagian bawah sebesar 128 word dapat diakses dengan mengisi DP dengan 0, sedangkan sisanya diakses dengan mengisi DP dengan 1.

Cara penentuan alamat yang kedua disebut dengan Indirect Addressing. Indirect addressing ini dilaksanakan dengan memakai auxiliary register (ARO atau AR1). Auxiliary register ini terdiri dari 16 bit dan dapat diisi melalui program. Indirect addressing dilakukan dengan memasukkan address data memory pada 8 LSB auxiliary register.

Auxiliary register yang bekerja dapat diatur melalui status bit Auxiliary Register Pointer (ARP).

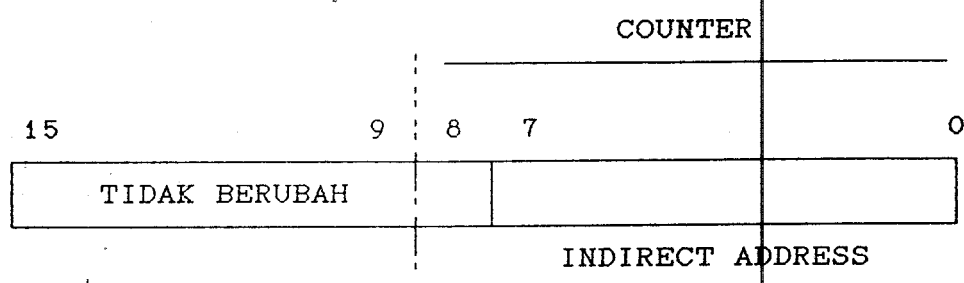
3.4.4. AUXILIARY REGISTER

TMS32010 menyediakan dua auxiliary register (ARO dan AR1) yang masing-masing terdiri dari 16 bit. Auxiliary register dipergunakan untuk tiga hal yaitu sebagai penunjuk alamat data memory dalam mode indirect addressing, sebagai penyimpan sementara dan sebagai pengontrol banyaknya loop.

Status bit ARP menunjukkan auxiliary register mana yang aktif. Jika ARP = 0 maka auxiliary register yang bekerja adalah ARO sedang jika ARP = 1 maka auxiliary register yang bekerja adalah AR1. ARP merupakan bagian dari register status sehingga dapat disimpan dalam memory dan dapat diubah melalui program. Auxiliary register dapat

disimpan dalam memory dengan perintah SAR (Store Auxiliary Register) dan diisi data memory dengan perintah LAR (Load Auxiliary Register).

Jika AR dipergunakan sebagai indirect addressing untuk menentukan alamat data memory internal maka hanya 8 bit terkecil yang digunakan. Hal ini dilakukan karena data memory internal besarnya hanya 144 word sehingga dengan 8 bit saja sudah cukup untuk menentukan semua alamat dalam data memory internal. Dalam mode indirect addressing ini harga dari AR bisa langsung dinaikkan/diturunkan setelah suatu instruksi dilaksanakan (*autoincrement/autodecrement*). Dengan cara ini AR bisa berfungsi sebagai counter. Jika digunakan sebagai counter maka 9 bit terkecil dari AR yang berubah-ubah. Bit 9 sampai dengan bit 15 tidak akan terpengaruh oleh adanya *autoincrement* atau *autodecrement*. Bit-bit ini berfungsi jika AR digunakan sebagai penyimpan sementara.



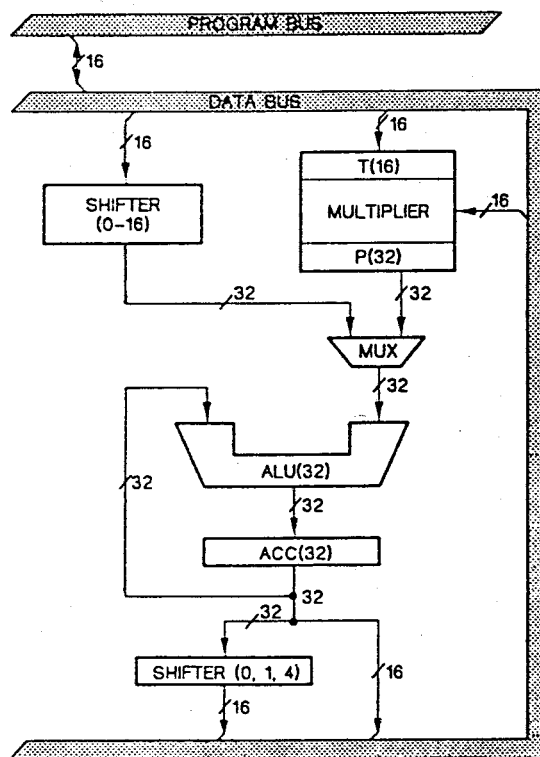
GAMBAR 3.4 ³⁷⁾

AUXILIARY REGISTER

³⁷⁾ Ibid, hal 3-15

3.5. CENTRAL ARITHMETIC LOGIC UNIT (CALU)

Central Arithmetic Logic Unit terdiri atas empat bagian yaitu pengali paralel 16 x 16 bit, Arithmetic Logic Unit (ALU) 32 bit, Accumulator (ACC) 32 bit dan dua buah penggeser (Shifter). Keempat komponen ini dihubungkan seperti pada gambar 3.5 di bawah ini.



GAMBAR 3.5 38)

CENTRAL ARITHMETIC LOGIC UNIT (CALU)

Dalam operasi-operasi aritmatika yang bukan perkalian, langkah-langkah berikut akan dilaksanakan.

1. Data diambil dari data bus.
2. Data dilewatkan pada *barrel shifter* di mana data dapat digeser ke kiri 0 sampai 16 bit tergantung dari instruksi yang dijalankan.
3. Data dilewatkan pada ALU di mana operasi aritmatika dijalankan dan kemudian hasilnya dimasukkan ke accumulator.
4. Data dalam accumulator dilewatkan pada *parallel left-shifter* di mana data dapat digeser 0, 1 atau 4 bit ke kiri.
5. Hasilnya disimpan dalam RAM internal. Karena ACC terdiri dari 32 bit sedangkan data bus 16 bit maka isi ACC bagian atas dan bagian bawah harus disimpan secara terpisah.

3.5.1. PENGGESER (SHIFTER)

Dua penggeser disediakan dalam TMS32010 untuk mempermudah manipulasi data. Penggeser pertama disebut *Barrel Shifter* yang berfungsi untuk menggeser data sebelum dimasukkan ke ALU. Sedangkan penggeser kedua disebut *Paralel Shifter* yang berfungsi untuk menggeser data dari accumulator sebelum dimasukkan ke data memory.

Barrel shifter mempunyai masukan 16 bit yang dihu-

bungkan pada data bus dan keluaran 32 bit yang dihubungkan pada ALU. Barrel shifter menggeser 0 sampai 16 bit ke kiri seluruh data yang dimasukkan melalui perintah LAC, SUB dan ADD. Shifter ini mengisi bagian kanan data yang digeser dengan 0 dan mengisi bagian kiri data yang digeser sesuai dengan isi MSB data. Jika MSB data tersebut berharga 0 maka bagian kiri data diisi dengan 0 dan sebaliknya jika MSB data berharga 1 maka bagian kiri data diisi seluruhnya dengan 1.

Paralel shifter dapat diaktifkan hanya dengan perintah SACH yang berfungsi untuk menyimpan data bagian atas Accumulator di data memory. Instruksi ini akan menyebabkan shifter diisi dengan 32 bit data pada ACC. Kemudian data digeser ke kiri 0, 1 atau 4 bit. Setelah data digeser, 16 MSB shifter disimpan di RAM. Dalam operasi ini harga dari accumulator tidak berubah.

3.5.2. ALU DAN ACCUMULATOR

ALU merupakan tempat operasi aritmatika dilakukan. ALU ini mempunyai masukan sebesar 16 bit dan keluaran sebesar 32 bit. Operasi yang dilakukan di ALU adalah penjumlahan, pengurangan dan operasi logika.

Dalam pelaksanaan operasi aritmatika accumulator merupakan salah satu operand sedangkan operand yang lain dapat berasal dari register P atau data memory.

Hasil dari operasi ALU sebesar 32 bit disimpan di ACC di mana operasi penggeseran dengan paralel shifter bisa dilakukan. Untuk menyimpan isi ACC dibutuhkan dua lokasi data memory yaitu masing-masing high-order word dan low-order word ACC.

Dalam pelaksanaan operasi logika, 16 LSB ALU akan berisi hasil operasi logika low-order word ACC dengan data memory sedangkan 16 MSB ALU akan berisi hasil operasi logika high-order word ACC dengan 0. Ada tiga macam operasi logika yang dapat dilakukan oleh ALU yaitu XOR, OR dan AND.

TABEL 3.1 ³⁹⁾

OPERASI LOGIKA PADA ALU

Fungsi	ACC bit 31-16	ACC bit 15-0
XOR	(0) XOR (ACC(31-16))	(dm) XOR (ACC(15-0))
OR	(0) OR (ACC(31-16))	(dm) OR (ACC(15-0))
AND	(0) AND (ACC(31-16))	(dm) AND (ACC(15-0))

dm = Data memory

Overflow pada ACC diatur melalui status bit yang disebut OVM (Overflow Mode). Jika OVM diisi dengan 1 dan terjadi overflow pada ACC maka ACC akan diisi dengan bilangan yang terbesar yaitu 7FFFFFFF untuk bilangan

³⁹⁾ Ibid, hal 3-20

positif dan 80000000 untuk bilangan negatif. Sedangkan jika OVM diisi dengan 0 dan terjadi overflow pada ACC maka nilai tersebut tidak diubah.

3.5.3. PENGALI, REGISTER T DAN REGISTER P

TMS32010 memiliki pengali 16 x 16 bit yang dibuat secara hardware sehingga mampu menghitung perkalian dua bilangan 16 bit dalam satu cycle dengan hasil 32 bit. Dalam operasi perkalian ada dua register yang ikut ambil bagian yaitu register T yang berfungsi sebagai salah satu operand pada operasi perkalian dan P register yang berfungsi untuk menyimpan hasil perkalian.

Langkah pertama dalam operasi perkalian adalah pengisian register T dengan perintah LT, LTA atau LTD. Langkah berikutnya adalah instruksi MPY atau MPYK. Jika instruksi MPY yang dipakai maka operand kedua dalam operasi perkalian adalah data memory dengan alamat seperti yang ditunjukkan pada program. Dengan perintah ini bisa dilakukan operasi perkalian 16 x 16 bit. Jika instruksi MPYK yang dipakai maka operand kedua sebesar 13 bit bisa dimasukkan secara langsung melalui program. Hasil dari perkalian ini dimasukkan ke register P. Perlu diperhatikan bahwa register P tidak dapat disimpan dalam data memory secara langsung melainkan harus terlebih dulu dipindahkan ke ACC dan kemudian disimpan di data memory. Jika terjadi

interrupt selama operasi perkalian maka interrupt tersebut akan ditangguhkan sampai satu instruksi sesudah instruksi MPY/MPYK selesai dilaksanakan. Untuk mencegah hilangnya data pada register P jika terjadi interrupt maka disarankan agar operasi perkalian diikuti dengan perintah untuk memindahkan hasil perkalian ke ACC.

3.6. SISTEM KONTROL

Sistem kontrol pada TMS32010 terdiri dari beberapa bagian yaitu *Program Counter (PC)*, *Stack*, *Sinyal Reset*, *Sinyal Interrupt* dan *Register Status*.

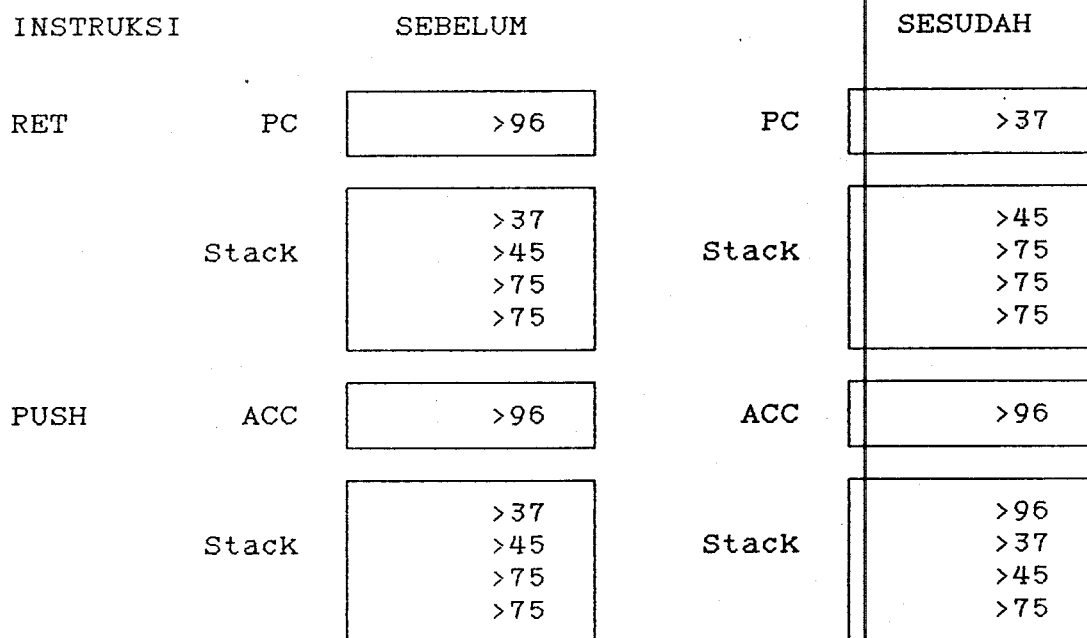
3.6.1. PROGRAM COUNTER DAN STACK

Program Counter (PC) adalah register 12 bit yang berisi alamat program memory instruksi berikutnya yang akan dieksekusi. TMS32010 membaca instruksi yang ditunjuk oleh PC kemudian menaikkan isi PC untuk menunjuk pada instruksi berikutnya.

Isi register PC dibuffer dan dikeluarkan melalui pin-pin address A11 - A0 baik pada mode mikrokomputer maupun pada mode mikroprosesor. Sembilan MSB dari PC digunakan khusus untuk address bus sedangkan tiga LSB (A2-A0) dimultipleks dengan address port I/O PA2-PA0

TMS32010 menyediakan 4 x 12 bit stack. Ketika terjadi interrupt atau pemanggilan subroutine, isi register PC disimpan dalam stack secara otomatis. Instruksi RET akan mengembalikan isi level teratas stack ke PC.

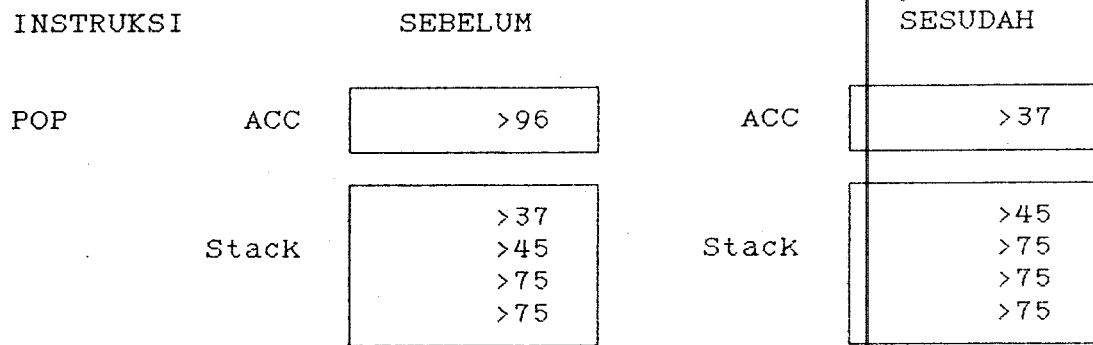
Melalui program isi accumulator juga dapat disimpan sementara dalam stack yaitu melalui instruksi PUSH dan POP. Instruksi PUSH berfungsi untuk menyimpan accumulator dalam level teratas stack sedangkan POP berfungsi untuk mengeluarkan kembali isi level teratas stack ke dalam ACC. Instruksi PUSH akan menyebabkan isi stack tergeser ke bawah dan menyebabkan level stack yang terbawah hilang. Sebaliknya POP akan menggeser isi stack ke atas dan menduplikasi level stack terbawah.



GAMBAR 3.6 40)

INSTRUKSI RET DAN PUSH

40) Ibid, hal 4-56 s.d. 4-57

GAMBAR 3.7 ⁴¹⁾

INSTRUKSI POP

3.6.2. RESET

Reset (RS) adalah suatu *non-maskable interrupt* yang membawa TMS32010 ke dalam keadaan tertentu. Sinyal reset ini akan menyebabkan prosesor menghentikan eksekusi program dan kembali pada keadaan inisial. Sinyal reset selalu diberikan pada saat prosesor dihidupkan (power up). Lama sinyal reset minimum adalah 5 clock cycle dan sinyal ini dapat diberikan selama waktu yang tak terdefinisikan.

Berikut ini adalah hal-hal yang terjadi pada saat reset diberikan.

1. Sinyal \overline{DE} , \overline{WE} dan \overline{MEN} high.
2. Data bus D15-D0 high impedance.
3. Register PC diset dengan 0 sehingga address bus selalu menunjuk ke alamat 0 setelah sinyal reset diberikan.
4. Interrupt dihalangi dan register flag interrupt direset semua dengan 0.

41) Ibid, hal 4-55

3. 6. 3. REGISTER STATUS

Register status terdiri dari lima bit status. Register status ini dapat diubah secara individual melalui program dan dapat juga disimpan dalam data memory.

Adapun lima bit status itu adalah sebagai berikut:

1. ARP : Auxiliary Register Pointer. Bit ini menentukan Auxiliary Register (AR) yang akan digunakan dalam indirect addressing. Jika ARP=0 maka AR0 yang dipakai sedangkan jika ARP=1 maka AR1 yang dipakai. Bit ini dapat diubah melalui perintah LARP, MAR dan LST.
2. DP : Data Memory Page Pointer. Bit ini menentukan halaman data memory yang akan diakses. DP digabung dengan 7 bit address dalam direct addressing untuk membentuk alamat absolut data memory. Jika DP=0 berarti data memory yang dituju terletak pada halaman 0 yaitu address 0 sampai dengan 127 sedangkan jika DP=1 berarti data memory yang dituju terletak pada halaman 1 yaitu address 128 sampai dengan 144.
3. INTM : Interrupt Mode. Bit ini menunjukkan mode dari interrupt. Jika bit ini berharga satu maka sinyal interrupt akan diabaikan sedangkan jika berharga nol maka prosesor akan menjalankan Interrupt Service Routine (ISR) jika terjadi

interrupt. Pada saat ISR dijalankan INTM akan diset menjadi satu secara otomatis. Bit ini dapat diubah melalui program dengan perintah DINT dan EINT.

4. OV : Overflow flag. Bila terjadi overflow pada accumulator maka bit ini akan diset menjadi 1 dan akan tetap berharga satu sampai dilaksanakannya perintah BV atau LST untuk mereset bit OV.
5. OVM : Overflow Mode Bit. Bit ini menentukan mode overflow prosesor. Jika OVM = 0 maka prosesor dijalankan dalam mode unsaturated. Artinya jika terjadi overflow pada ACC maka hasil tersebut tidak diubah. Jika OVM = 1 dan terjadi overflow maka ACC akan diisi dengan nilai yang paling positif (>7FFFFFFF) atau yang paling negatif (>80000000).

Perlu diperhatikan bahwa pada saat reset hanya bit INTM dan OV yang direset sama dengan nol. Bit ARP, DP dan OVM tidak diinisialisasi pada saat reset. Oleh karena itu ketiga bit itu perlu diinisialisasi melalui software.

Isi register status dapat disimpan pada data memory dengan instruksi SST. Lokasi masing-masing bit adalah sebagai berikut.

15	14	13	12 . . 9	8	7	1	0
OV	OVM	INTM	1 1 1 1	ARP	1 1 1 1 1 1 0	DP	

GAMBAR 3.8 ⁴²⁾

REGISTER STATUS DALAM DATA MEMORY

Instruksi untuk mengubah register status adalah LST. Melalui instruksi ini register status dapat diubah sesuai dengan isi data memory yang ditunjuk. Hal tersebut berlaku untuk semua bit status kecuali INTM. Bit ini hanya dapat diubah melalui perintah DINT atau EINT.

3.6.4. INTERRUPT

Interrupt pada TMS32010 dapat dibangkitkan dengan memberikan sinyal dengan level low atau pada saat sinyal berubah dari high ke low (*negative going edge*). Pengaruh sinyal interrupt ini dapat diatur melalui register status yaitu bit INTM, seperti yang telah dijelaskan di depan.

Pada TMS32010 sinyal interrupt membutuhkan tambahan hardware untuk men-*sinkronisasi* sinyal interrupt dengan clock sistem. Tambahan hardware dapat diberikan seperti pada gambar 3.9.

TMS32010 menangani adanya interrupt dengan melakukan hal-hal berikut ini secara berurutan.

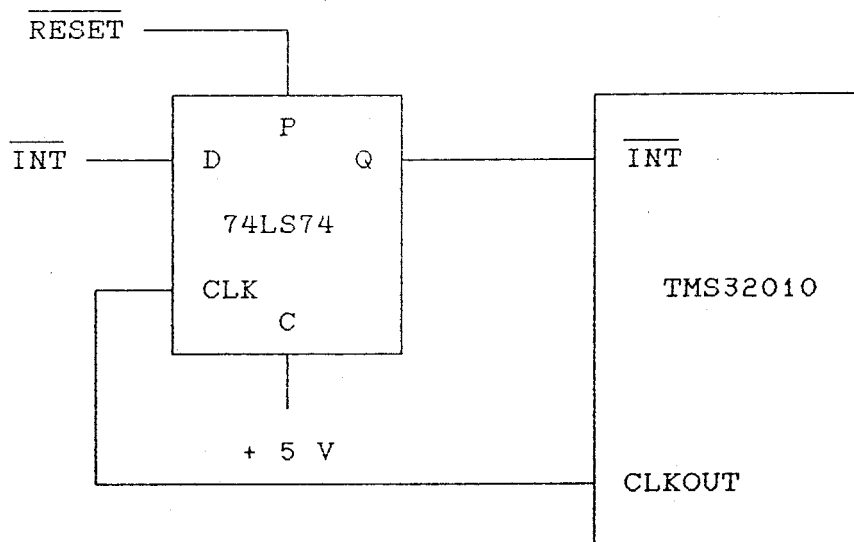
1. Bit INTF (Interrupt Flag) direset menjadi 0
2. Bit INTM (Interrupt Mode) diset 1 sehingga menghambat

42) Ibid, hal 3-26

operasi interrupt yang lain.

3. Isi PC dimasukkan pada level teratas stack.
4. PC diisi dengan address awal dari Interrupt Service Routine (ISR) yaitu 2.

Perlu diperhatikan bahwa ISR harus dimulai pada program memory dengan alamat >2. Pada akhir dari ISR harus ditambahkan instruksi EINT untuk mengembalikan bit INTM ke 0 sehingga TMS32010 dapat menangani interrupt lagi.



GAMBAR 3.9 ⁴³⁾

SINKRONISASI SINYAL INTERRUPT PADA TMS32010

Eksekusi ISR akan ditunda sampai:

1. Berakhirnya semua cycle untuk instruksi yang terdiri dari lebih dari satu cycle.
2. Diselesaikannya eksekusi satu instruksi sesudah instruksi MPY atau MPYK,
3. Adanya instruksi EINT bila sebelumnya interrupt diham-

⁴³⁾ Ibid, hal A-22

bat dengan instruksi DINT.

3.7. FUNGSI INPUT DAN OUTPUT

TMS32010 menyediakan beberapa fungsi input/output sehingga dapat dihubungkan dengan peralatan yang lain.

3.7.1. OPERASI INPUT/OUTPUT

TMS32010 memakai 3 LSB dari address bus (PA2-PA0) untuk menentukan alamat 8 channel input/output port. Selama operasi input/output, hanya ketiga bit itu yang aktif dan bit A11 - A3 akan berharga 0.

Operasi input/output dilakukan dengan memberikan perintah IN atau OUT. Perintah IN untuk operasi input dan OUT untuk operasi output.

Pada saat instruksi IN dilaksanakan maka sinyal $\overline{\text{DEN}}$ akan aktif yang menandakan TMS32010 siap menerima data dari peralatan luar. $\overline{\text{DEN}}$ aktif pada saat instruksi IN saja.

Pada saat instruksi OUT dilaksanakan, sinyal $\overline{\text{WE}}$ akan aktif, yang menandakan TMS32010 siap untuk mengirim data ke peralatan luar. $\overline{\text{WE}}$ aktif pada saat instruksi OUT dan TBLW.

Diagram waktu untuk IN dan OUT dapat dilihat pada gambar 3.10 dan 3.11.

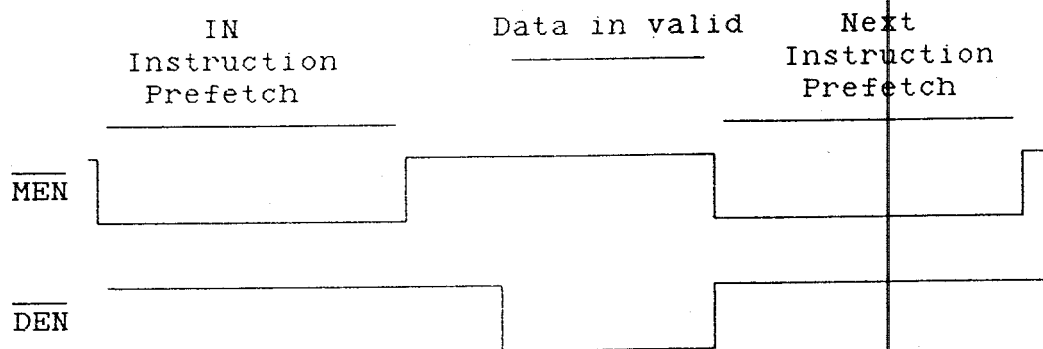
GAMBAR 3.10 ⁴⁴⁾

DIAGRAM WAKTU UNTUK INSTRUKSI IN

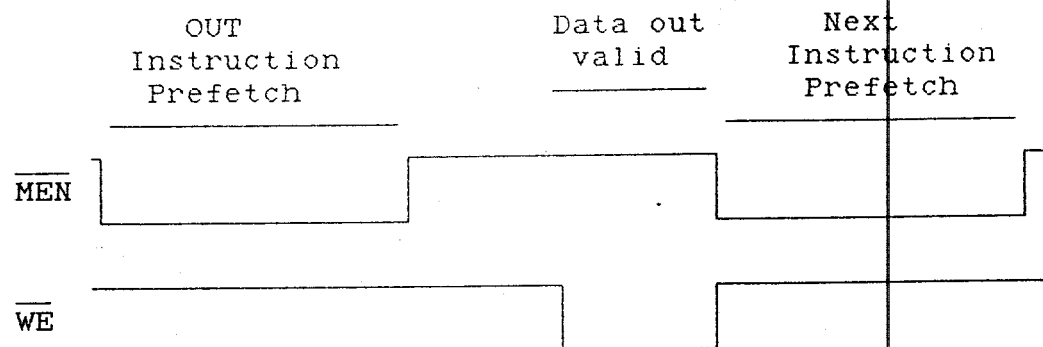
GAMBAR 3.11 ⁴⁵⁾

DIAGRAM WAKTU UNTUK INSTRUKSI OUT

3.7.2. OPERASI PADA EKSTERNAL MEMORY

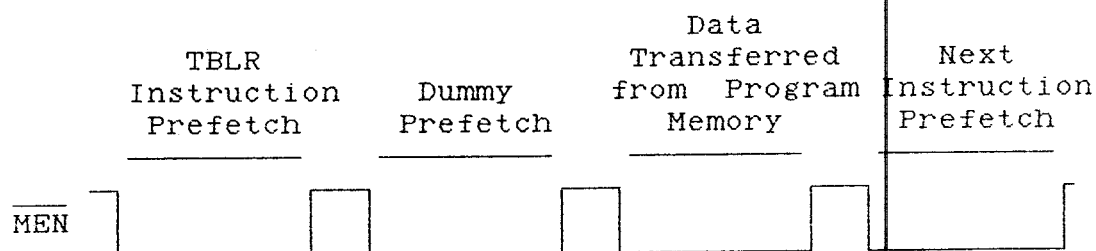
TMS32010 menyediakan dua instruksi untuk operasi pada eksternal memory. Instruksi TBLR (Table Read) digunakan untuk memindahkan word dari internal ROM atau eksternal ROM/RAM ke data memory (internal RAM). Sedangkan instruksi TBLW (Table Write) digunakan untuk memindahkan word dari data memory ke eksternal RAM.

Instruksi TBLR akan mengaktifkan sinyal \overline{MEN} sedangkan

44) Ibid, hal 3-29

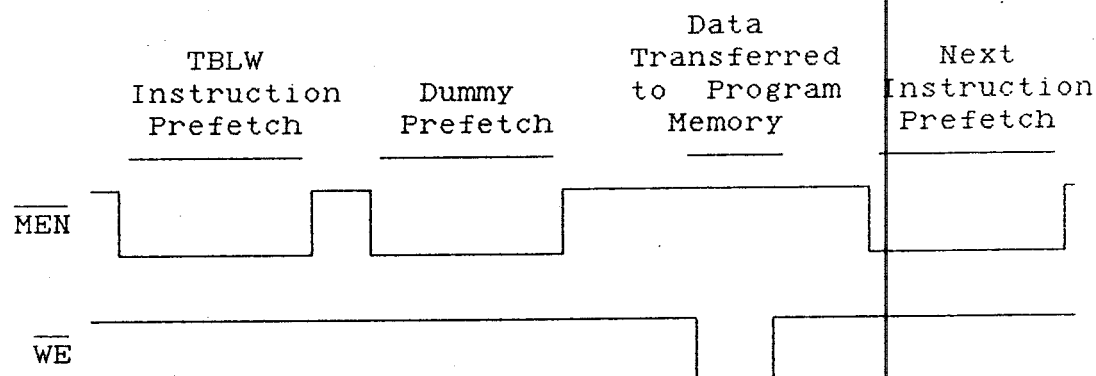
45) Ibid, hal 3-29

instruksi TBLW akan mengaktifkan sinyal WE. Pada gambar 3.12 dan 3.13 ditunjukkan diagram waktu dari instruksi TBLR dan TBLW.



GAMBAR 3.12 ⁴⁶⁾

DIAGRAM WAKTU INSTRUKSI TBLR



GAMBAR 3.13 ⁴⁷⁾

DIAGRAM WAKTU INSTRUKSI TBLW

Dalam desain sistem minimum TMS32010 perlu diperhatikan bahwa instruksi TBLW dan OUT mempunyai diagram waktu yang sama sehingga address >0 sampai dengan >7 lebih baik tidak dipakai sebagai RAM karena bisa menyebabkan terjadinya kesalahan penulisan data.

46) Ibid, hal 3-30

47) Ibid, hal 3-30

3. 8. BAHASA ASSEMBLY TMS32010

TMS32010 menyediakan 60 instruksi untuk programnya yang dibedakan menjadi 6 bagian yaitu:

1. Instruksi untuk Accumulator yang terdiri dari 18 instruksi.
2. Instruksi untuk Auxiliary Register dan Data Page Pointer yang terdiri dari 7 instruksi.
3. Instruksi untuk register T, register P dan perkalian yang terdiri dari 8 instruksi.
4. Instruksi Branch dan Call yang terdiri dari 13 instruksi.
5. Instruksi kontrol yang terdiri dari 9 instruksi.
6. Instruksi untuk operasi I/O dan Data Memory yang terdiri dari 5 instruksi.

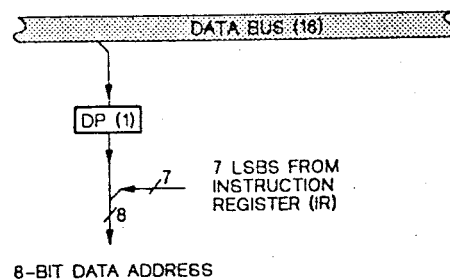
Untuk memahami software TMS32010 perlu diketahui lebih dulu mode penentuan alamat memory. TMS32010 menyediakan tiga mode pengalamatan memory yaitu:

1. *Mode Direct Addressing*
2. *Mode Indirect Addressing*
3. *Mode Immediate Addressing.*

3. 8. 1. MODE DIRECT ADDRESSING

Dalam mode Direct Addressing, instruksi mengandung 7

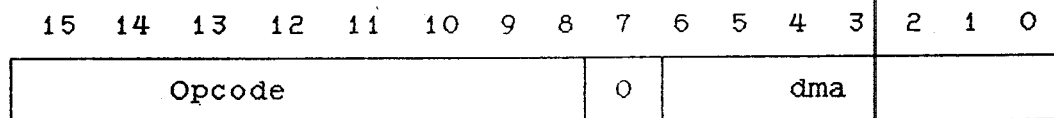
bit yang merupakan 7 LSB address Data Memory Address (dma). Agar dapat memanggil seluruh data memory maka ketujuh bit digabungkan dengan satu bit Data Page Pointer (DP) sehingga membentuk address 8 bit. Jadi data memory dibagi menjadi dua halaman sesuai dengan isi DP. Halaman pertama terdiri dari 128 word dan halaman kedua terdiri dari 16 word.



GAMBAR 3.14 48)

BLOK DIAGRAM MODE DIRECT ADDRESSING

Mode Direct Addressing mempunyai format seperti berikut ini. 49)



Bit 15 sampai dengan 8 berisi opcode instruksi. Bit 7 sampai dengan 0 yang menunjukkan bahwa mode penentuan alamat adalah Direct Addressing. Bit 6 sampai dengan 0 menunjukkan Data Memory Address yang diinginkan.

48) Ibid, hal 4-3

49) Ibid, hal 4-3

3.8.2. MODE INDIRECT ADDRESSING

Dalam mode Indirect Addressing, alamat data memory dibentuk dari 8 LSB Auxiliary Register (AR0 atau AR1). Dengan cara ini maka seluruh data memory dapat dibaca atau ditulis tanpa adanya tambahan bit. Auxiliary Register yang dipakai ditentukan dari bit status Auxiliary Register Pointer (ARP).

Keistimewaan mode Indirect Addressing terletak pada kemampuannya untuk menaikkan atau menurunkan isi Auxiliary Register secara otomatis setelah instruksi selesai dilaksanakan. Penulisan mode Indirect Addressing ini memakai simbol-simbol seperti berikut ini.

- * Isi AR(ARP) dipakai sebagai Data Memory Address
- *- Isi AR(ARP) dipakai sebagai Data Memory Address dan kemudian diturunkan setelah data memory tersebut diakses.
- *+ Isi AR(ARP) dipakai sebagai Data Memory Address dan kemudian dinaikkan setelah data memory tersebut diakses.

Mode indirect addressing mempunyai format sebagai berikut: 50)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								1	0	INC	DEC	NAR	0	0	ARP

Bit 15 sampai dengan 8 berisi Opcode dan bit 7 = 1 menyatakan bahwa mode penentuan alamat dma adalah indirect. Bit

50) Ibid, hal 4-4

6 sampai dengan 0 berisi bit control untuk mode indirect addressing. Bit 3 dan bit 0 mengatur pemakaian dari ARP. Isi bit 3 adalah NAR (New Auxiliary Register) yang menunjukkan apakah Auxiliary Register yang bekerja akan diubah. Jika isi bit 3 = 0 maka isi bit 0 dimasukkan pada ARP setelah eksekusi dilaksanakan. Jika bit 3 = 1 maka isi ARP tidak diubah. Bit 5 dan 4 menentukan apakah isi Auxiliary Register akan dinaikkan, diturunkan atau tidak diubah. Jika bit 5 berisi 1 maka isi auxiliary register dinaikkan 1 setelah instruksi dilaksanakan. Jika bit 4 berisi 1 maka isi auxiliary register diturunkan setelah instruksi dilaksanakan. Jika bit 5 dan bit 4 berisi 0 maka auxiliary register tidak akan berubah isinya. Bit 6, 2 dan 1 selalu berisi dengan 0.

3. 8. 3. MODE IMMEDIATE ADDRESSING

TMS32010 menyediakan lima instruksi yang bisa memakai mode immediate addressing. Banyaknya bit untuk lokasi operand berbeda untuk tiap-tiap instruksi.

LACK Mengisi accumulator dengan 8 bit konstan
 LARK Mengisi auxiliary register dengan 8 bit konstan
 LARP Mengisi auxiliary register pointer (ARP) 1 bit
 LDPK Mengisi data memory page pointer (DP) 1 bit
 MPYK Mengalikan register T dengan 13 bit konstan.

3. 8. 4. INSTRUKSI-INSTRUKSI TMS32010

Sebelum instruksi-instruksi TMS32010 dipelajari, lebih dulu akan diberikan simbol dan singkatan yang dipakai dalam penulisan bahasa assembly TMS32010. Berikut ini adalah simbol dan singkatan tersebut.

TABEL 3. 2 ⁵¹⁾

SIMBOL DAN SINGKATAN PADA PENULISAN ASSEMBLY TMS32010

SYMBOL	MEANING
A	Port address
ACC	Accumulator
ARn	Auxiliary Register n (AR0 and AR1) are predefined assembler symbols equal to 0 and 1, respectively.)
ARP	Auxiliary register pointer
B	Branch address
D	Data memory address field
DATn	Label assigned to data memory location n
dma	Data memory address
DP	Data page pointer
I	Addressing mode bit
INTM	Interrupt mode bit
K	Immediate operand field
>nn	Indicates nn is a hexadecimal number. (All others are assumed to be decimal values.)
OVM	Overflow (saturation) mode flag bit
P	Product register
PA	Port address (PA0 through PA7 are predefined assembler symbols equal to 0 through 7, respectively.)
PC	Program counter
pma	Program memory address
PRGn	Label assigned to program memory location n
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
T	Temporary register
TOS	Top of stack
X	3-bit accumulator left-shift field
→	Is assigned to
	An absolute value
< >	User-defined items
[]	Optional items
()	"Contents of"
{ }	Alternative items, one of which must be entered
< >	Angle brackets back-to-back indicate "not equal".
	Blanks or spaces must be entered where shown.

51) Ibid, hal 4-7

Berikut ini adalah kumpulan instruksi TMS32010 yang dibagi menjadi 6 bagian.

TABEL 3.3 52)

KUMPULAN INSTRUKSI TMS32010

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
ABS	Absolute value of accumulator	1	1	0111 1111	1000 1000
ADD	Add to accumulator with shift	1	1	0000 SSSS	1DDD DDDD
ADDH	Add to high accumulator	1	1	0110 0000	1DDD DDDD
ADDS	Add to low accumulator with sign-extension suppressed	1	1	0110 0001	1DDD DDDD
AND	AND with accumulator	1	1	0111 1001	1DDD DDDD
LAC	Load accumulator with shift	1	1	0010 SSSS	1DDD DDDD
LACK	Load accumulator immediate short	1	1	0111 1110	KKKK KKKK
OR	OR with accumulator	1	1	0111 1010	1DDD DDDD
SACH	Store high accumulator with shift	1	1	0101 1XXX	1DDD DDDD
SACL	Store low accumulator	1	1	0101 0000	1DDD DDDD
SUB	Subtract from accumulator with shift	1	1	0001 SSSS	1DDD DDDD
SUBC	Conditional subtract	1	1	0110 0100	1DDD DDDD
SUBH	Subtract from high accumulator	1	1	0110 0010	1DDD DDDD
SUBS	Subtract from low accumulator with sign-extension suppressed	1	1	0110 0011	1DDD DDDD
XOR	Exclusive-OR with low accumulator	1	1	0111 1000	1DDD DDDD
ZAC	Zero accumulator	1	1	0111 1111	1000 1001
ZALH	Zero low accumulator and load high accumulator	1	1	0110 0101	1DDD DDDD
ZALS	Zero accumulator and load low accumulator with sign-extension suppressed	1	1	0110 0110	1DDD DDDD
AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
LAR	Load auxiliary register	1	1	0011 100R	1DDD DDDD
LARK	Load auxiliary register immediate short	1	1	0111 000R	KKKK KKKK
LARP	Load auxiliary register pointer immediate	1	1	0110 1000	1000 000K
LDP	Load data memory page pointer	1	1	0110 1111	1DDD DDDD
LDPK	Load data memory page pointer immediate	1	1	0110 1110	0000 000K
MAR	Modify auxiliary register	1	1	0110 1000	1DDD DDDD
SAR	Store auxiliary register	1	1	0011 000R	1DDD DDDD
T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
APAC	Add P register to accumulator	1	1	0111 1111	1000 1111
LT	Load T register	1	1	0110 1010	1DDD DDDD
LTA	Load T register and accumulate previous product	1	1	0110 1100	1DDD DDDD
LTD	Load T register, accumulate previous product, and move data	1	1	0110 1011	1DDD DDDD
MPY	Multiply (with T register, store product in P register)	1	1	0110 1101	1DDD DDDD
MPYK	Multiply immediate	1	1	100K KKKK	KKKK KKKK
PAC	Load accumulator with P register	1	1	0111 1111	1000 1110
SPAC	Subtract P register from accumulator	1	1	0111 1111	1001 0000

52) Ibid, hal 4-9 s.d. 4-10

TABEL 3.3

KUMPULAN INSTRUKSI TMS32010 (Lanjutan)

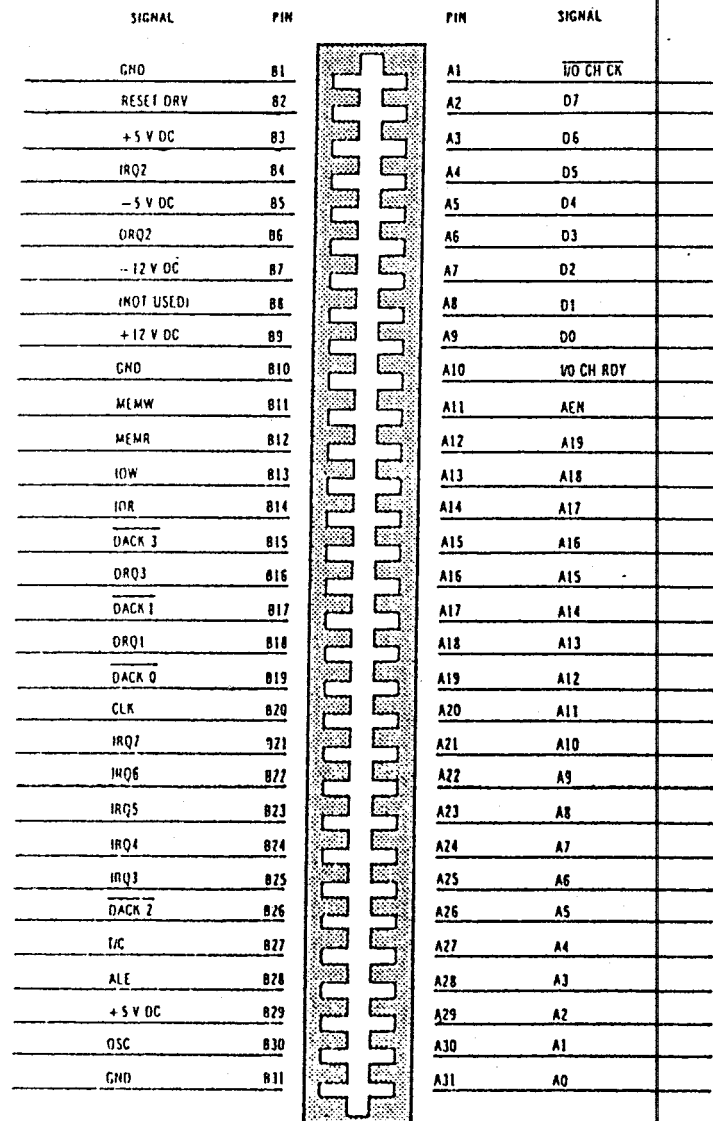
BRANCH/CALL INSTRUCTIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
B	Branch unconditionally	2	2	1111 1001	0000 0000
BANZ	Branch on auxiliary register not zero	2	2	1111 0100	0000 0000
BGEZ	Branch if accumulator ≥ 0	2	2	1111 1101	0000 0000
BGZ	Branch if accumulator > 0	2	2	1111 1100	0000 0000
BIOZ	Branch on I/O status = 0	2	2	1111 0110	0000 0000
BLEZ	Branch if accumulator ≤ 0	2	2	1111 1011	0000 0000
BLZ	Branch if accumulator < 0	2	2	1111 1010	0000 0000
BNZ	Branch if accumulator $\neq 0$	2	2	1111 1110	0000 0000
BV	Branch on overflow	2	2	1111 0101	0000 0000
BZ	Branch if accumulator = 0	2	2	1111 1111	0000 0000
CALA	Call subroutine indirect	2	1	0111 1111	1000 1100
CALL	Call subroutine	2	2	1111 1000	0000 0000
RET	Return from subroutine	2	1	0111 1111	1000 1101
CONTROL INSTRUCTIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
DINT	Disable interrupt	1	1	0111 1111	1000 0001
EINT	Enable interrupt	1	1	0111 1111	1000 0010
LST	Load status register from data memory	1	1	0111 1011	1DDD DDDD
NOP	No operation	1	1	0111 1111	1000 0000
POP	Pop top of stack to low accumulator	2	1	0111 1111	1001 1101
PUSH	Push low accumulator onto stack	2	1	0111 1111	1001 1100
ROVM	Reset overflow mode	1	1	0111 1111	1000 1010
SOVM	Set overflow mode	1	1	0111 1111	1000 1011
SST	Store status register	1	1	0111 1100	1DDD DDDD
I/O AND DATA MEMORY OPERATIONS					
Mnemonic and Description		Cycles	Words	16-Bit Opcode	
				MSB	LSB
DMOV	Data move in data memory	1	1	0110 1001	1DDD DDDD
IN	Input data from port	2	1	0100 0AAA	1DDD DDDD
OUT	Output data to port	2	1	0100 1AAA	1DDD DDDD
TBLR	Table read	3	1	0110 0111	1DDD DDDD
TBLW	Table write	3	1	0111 1101	1DDD DDDD

3. 9. TEORI PENUNJUANG UNTUK INTERFACING KE IBM PC/XT

Komputer IBM PC/XT mempunyai kemampuan untuk berhubungan dengan peralatan luar (peripheral). Hubungan ini dilaksanakan melalui delapan slot yang disediakan pada bagian belakang komputer IBM PC/XT. Kedelapan slot ini mempunyai konfigurasi pin-pin yang identik sehingga rangkaian interface dapat diletakkan pada sebarang slot yang tersedia. Tiap-tiap slot terdiri dari 62 pin yang terdiri dari pin-pin yang digunakan untuk operasi memori atau operasi input/output. Pada gambar 3.15 ditunjukkan konfigurasi slot dari IBM PC/XT.

Berikut ini adalah penjelasan tentang pin-pin yang digunakan dalam perencanaan alat ini.

- A0-A19 : Sinyal output yang berfungsi untuk menentukan alamat memori atau port input/output yang akan diakses IBM PC/XT.
- D0-D7 : Delapan jalur data dua arah yang digunakan untuk mengirim dan menerima data dari dan ke prosesor 8088.
- RESET : Sinyal reset dari sistem komputer IBM PC/XT. Pin ini akan aktif high pada saat IBM PC/XT direset.
- MEMR : Sinyal output yang aktif low pada saat IBM PC/XT membaca data pada memori.
- MEMW : Sinyal output yang aktif low pada saat IBM



GAMBAR 3.15 53)

KONFIGURASI PIN SLOT IBM PC/XT

PC/XT menulis data pada memori.

- IOW : Sinyal output aktif low yang menunjukkan bahwa IBM PC/XT telah siap mengirim data pada port I/O.
- AEN : Sinyal output yang aktif high selama IBM PC/XT melakukan operasi DMA (Direct Memory Access). Sinyal ini digunakan untuk men-disable operasi I/O selama operasi DMA berlangsung karena pada saat itu sinyal IOW atau IOR aktif sedangkan address bus menunjuk pada alamat memori.
- IRQ2 : Sinyal input yang digunakan untuk membangkitkan interrupt request pada 8088. Sinyal ini dihubungkan langsung ke IC 8259 Interrupt Controller dan akan membangkitkan sinyal interrupt bila ada perubahan level dari high ke low.

Selain sinyal-sinyal di atas, digunakan juga sinyal supply dan ground yaitu

- + 5 V DC Level tegangan ini tersedia pada dua dari pin-pin slot dan diregulasi pada tegangan $5 \pm 5\%$ volt DC.
- + 12 V DC Tersedia pada salah satu pin-pin slot dan diregulasi pada tegangan $12 \pm 5\%$ volt DC.
- 12 V DC Tersedia pada salah satu pin-pin slot dan diregulasi pada tegangan $-12 \pm 10\%$ volt DC.

GROUND Ground dari sistem IBM PC/XT ini disediakan pada tiga dari pin-pin slot.

3.9.1. MEMORY IBM PC/XT

Mikroprosesor 8088 menunjang pemakaian memory sebesar 1 M byte sesuai dengan address busnya yang terdiri dari 20 bit. Pada IBM PC/XT tidak seluruh memori ini dipakai. Pada gambar 3.16 diperlihatkan peta pemakaian memori pada IBM PC/XT. Pada gambar tersebut ditunjukkan adanya lokasi memori yang tidak dipakai pada IBM PC/XT. Lokasi memori ini bisa dipergunakan untuk pengembangan hardware Komputer IBM PC/XT.

TMS32010 mempunyai lokasi memory sebesar 4 K Word atau 8 K byte. Pada alat ini, seluruh memory tersebut dipakai. Karena memory ini dipergunakan bersama dengan IBM PC/XT maka harus dipilih lokasi memory pada IBM PC/XT sebesar 8 K byte yang tidak dipakai oleh IBM PC/XT.

Pada gambar 3.16 dapat dilihat bahwa address memory yang tidak terpakai pada IBM PC/XT di antaranya adalah address A4000H sampai A6000H. Address ini dipilih sebagai memory yang dipakai bersama dengan TMS32010.

Address A4000H sampai dengan A4FFFH digunakan sebagai lokasi program memory dari TMS32010 sedangkan address A5000H sampai dengan A5FFFH digunakan sebagai lokasi RAM bersama.

3.9.2. INPUT / OUTPUT PORT IBM PC/XT

Selain melakukan operasi memory, IBM PC/XT juga melakukan operasi I/O untuk mengatur peralatan penunjang pada sistem board IBM PC/XT dan untuk mengatur peralatan I/O adapter. Prosesor 8088 sebenarnya menunjang pemakaian alamat I/O sebesar 65536 address port tetapi pada IBM PC/XT hanya 10 bit terbawah yang dipakai sehingga address I/O maksimum yang bisa dipakai adalah 1024. Address I/O ini dibagi menjadi dua bagian yang masing-masing sebesar 512 address port. Port dengan alamat 0000H sampai dengan 01FFH dipakai untuk operasi I/O pada sistem board sedangkan alamat 0200H sampai dengan 03FFH dipakai untuk operasi I/O melalui slot IBM PC/XT.

Gambar 3.17 menunjukkan peta pemakaian I/O port pada sistem board IBM PC/XT. Melalui peta ini dapat diketahui alamat dari peralatan penunjang yang terdapat pada sistem board IBM PC/XT seperti 8237, 8255 dan lain-lain.

Gambar 3.18 menunjukkan peta pemakaian I/O port yang dapat diakses melalui slot IBM PC/XT. Sebagian dari address I/O port ini telah dipakai oleh beberapa adapter seperti misalnya Color Graphic Adapter, Disk Drive Adapter, Printer Adapter dan lain-lain. Adapter tersebut mempunyai alamat yang khusus sehingga alamat I/O port yang telah dipakai oleh adapter tersebut tidak dapat digunakan untuk adapter yang lain.

HEX ADDRESS	USES
0200H	(NOT USED)
0201H	GAME CONTROL ADAPTER
0202H	(NOT USED)
0203H - 0207H	(NOT USED)
0208H - 020FH	SECOND PRINTER PORT ADAPTER
0210H - 0217H	(NOT USED)
0218H - 021FH	SECOND SERIAL PORT ADAPTER CARD
0220H - 0227H	(NOT USED)
0228H - 022FH	PRINTER PORT ADAPTER CARD
0230H - 0237H	(NOT USED)
0238H - 023FH	MONOCHROME AND PRINTER ADAPTER
0240H - 0247H	(NOT USED)
0248H - 024FH	(NOT USED)
0250H - 0257H	(NOT USED)
0258H - 025FH	(NOT USED)
0260H - 0267H	(NOT USED)
0268H - 026FH	(NOT USED)
0270H - 0277H	(NOT USED)
0278H - 027FH	5 1/4 INCH DISKETTE DRIVE ADAPTER CARD
0280H - 028FH	SERIAL PORT ADAPTER CARD
0290H - 0297H	(NOT USED)
0298H - 029FH	(NOT USED)
02A0H - 02A7H	(NOT USED)
02A8H - 02AFH	(NOT USED)
02B0H - 02B7H	(NOT USED)
02B8H - 02BFH	(NOT USED)
02C0H - 02CFH	(NOT USED)
02D0H - 02DFH	COLORGRAPHICS ADAPTER
02E0H - 02EFH	(NOT USED)
02F0H - 02F7H	(NOT USED)
02F8H - 02FFH	(NOT USED)

NOTE: NEW FEATURES BY IBM AND OTHER MANUFACTURERS MAY USE SOME OF THE SPARE I/O ADDRESS DECODES

GAMBAR 3. 18 56)

PETA PEMAKAIAN I/O PORT PADA SLOT IBM PC/XT

3. 9. 3. OPERASI INTERRUPT IBM PC/XT

Mikroprosesor 8088 menyediakan dua jenis interrupt yaitu maskable interrupt dan nonmaskable interrupt (NMI). Maskable interrupt adalah interrupt yang dapat dihambat

56) Ibid, hal 129.

dengan software sedangkan NMI adalah interrupt yang tidak dapat dihambat dengan software.

Untuk mengatur maskable interrupt, pada IBM PC/XT dipakai IC 8259 yang berfungsi untuk mengembangkan sinyal interrupt 8088 menjadi 8 sinyal yang dapat diatur prioritasnya. Kedelapan level interrupt tersebut sudah diatur oleh IBM PC/XT sebagai berikut:

TABEL 3.4. 57)

LEVEL PRIORITAS INTERRUPT IBM PC/XT

LEVEL INTERRUPT	KEGUNAAN
IRQ0	Sistem timer output 8253-5 channel 0
IRQ1	Keyboard scan code interrupt
IRQ2	Belum digunakan
IRQ3	Asynchronous Communications (Secondary) SDLC Communications
IRQ4	Asynchronous Communications (Primary) SDLC Communications
IRQ5	Fixed Disk
IRQ6	Diskette
IRQ7	Paralel printer

57) IBM, "IBM Personal Computer XT Technical Reference", hal 1-8.

Dari level-level tersebut hanya IRQ2 yang belum dipakai. Oleh karena itu untuk komunikasi antara IBM PC/XT dan TMS32010 dipakai sinyal IRQ2. Sinyal ini akan menginterrupt IBM PC/XT jika terjadi perubahan level dari low ke high. Perubahan level ini akan di-latch oleh interrupt controller 8259 dan kemudian 8259 mengirimkan sinyal interrupt pada 8088 beserta dengan address Interrupt Service Routine.

B A B IV

PERENCANAAN HARDWARE DAN SOFTWARE

4. 1. PENDAHULUAN

Sebelum perencanaan alat dibahas secara rinci, lebih dulu diuraikan cara kerja secara umum dan blok diagram alat.

4. 1. 1. BLOK DIAGRAM

Implementasi algoritma FFT dengan prosesor TMS32010 diwujudkan dengan menggunakan sistem minimum TMS32010 yang di-interface-kan dengan IBM PC/XT. Prosesor TMS32010 berfungsi sebagai pengolah data berdasarkan algoritma Fast Fourier Transform sedangkan IBM PC/XT berfungsi untuk menampilkan data pada monitor.

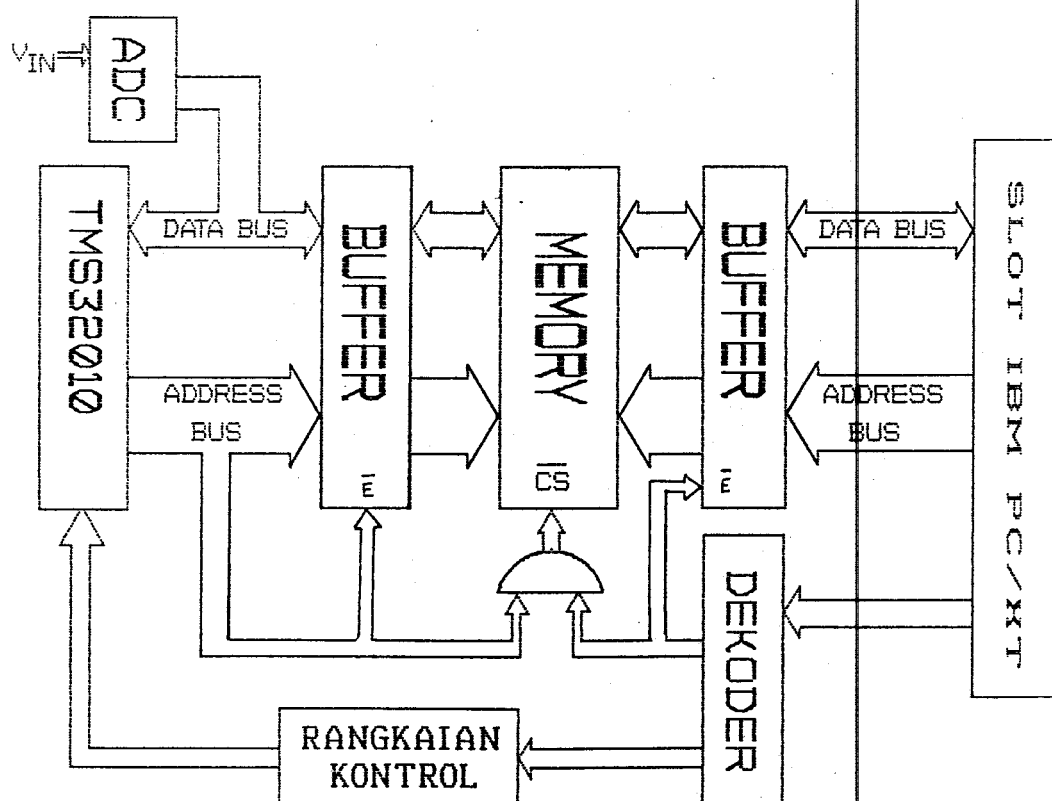
Dalam sistem minimum TMS32010 tersebut, terdapat 4 K word Random Access Memory (RAM) yang dibedakan menjadi dua. Dua kiloword yang pertama dipergunakan untuk program memory dan 2 K word berikutnya dipergunakan untuk data memory eksternal dari prosesor TMS32010.

Program memory pada sistem minimum ini disimpan dalam RAM sehingga dapat diprogram secara langsung melalui IBM PC/XT. Dengan demikian prosesor TMS32010 ini dapat

diaplikasikan untuk tujuan lain dengan mengubah programnya melalui IBM PC/XT.

Untuk pertukaran data, dipakai RAM bersama (Shared RAM) yang dapat diakses baik dari TMS32010 maupun dari IBM PC/XT. Dengan cara ini data hasil olahan TMS32010 dapat diakses secara langsung oleh IBM PC/XT.

Blok diagram seluruh rangkaian diperlihatkan pada gambar 4.1 berikut ini.



GAMBAR 4.1

BLOK DIAGRAM RANGKAIAN SECARA LENGKAP

4. 1. 2. CARA KERJA RANGKAIAN SECARA UMUM

Mula-mula program untuk menjalankan algoritma FFT pada TMS32010, yang telah ditulis pada IBM PC/XT, dipindahkan dari disk ke RAM bersama. Selama proses pemindahan program ini berlangsung, TMS32010 dalam kondisi RESET. Setelah proses ini selesai maka IBM PC/XT mengirim sinyal I/O untuk mengubah kondisi RESET itu.

Kemudian IBM PC/XT menjalankan program untuk menginisialisasi lama proses sampling dan banyaknya sample. Data ini kemudian disimpan pada RAM bersama. Kemudian IBM PC/XT mengirim sinyal BIO yang menandakan bahwa proses inisialisasi telah selesai. TMS32010 kemudian mengambil alih kontrol sistem dan mulai mengaktifkan ADC untuk memulai proses sampling. Hasil penyamplingan kemudian disimpan dalam RAM bersama. Setelah didapat jumlah sample yang sesuai, TMS32010 mulai melaksanakan manipulasi data berdasarkan algoritma FFT. Hasilnya ditempatkan pada RAM bersama. TMS32010 kemudian mengirimkan sinyal interrupt pada IBM PC/XT sebagai tanda bahwa pengolahan data telah selesai. Kemudian IBM PC/XT mengambil data dari RAM bersama untuk ditampilkan pada layar monitor. Setelah itu IBM PC/XT mengirimkan sinyal BIO sebagai tanda bahwa kerja IBM PC/XT telah selesai. TMS32010 memulai kembali proses pengambilan sample. Prosedur tersebut berlangsung terus menerus selama program belum dihentikan.

4.2. PERENCANAAN RANGKAIAN DEKODER DARI IBM PC/XT

Untuk membedakan memory atau I/O port mana yang bekerja diperlukan rangkaian untuk membangkitkan suatu sinyal yang akan aktif hanya bila address suatu memory atau I/O port tertentu dipanggil. Rangkaian tersebut disebut dengan rangkaian dekoder. Rangkaian dekoder untuk IBM PC/XT dibedakan menjadi dua yaitu dekoder untuk memory dan dekoder untuk I/O. Kedua sistem dekoder tersebut memakai sinyal dari 8088 yang berbeda. Pada dekoder memory dipakai sinyal $\overline{\text{MEMR}}$ dan $\overline{\text{MEMW}}$ yang berfungsi sebagai tanda apabila 8088 akan membaca dan menulis pada memory. Sedangkan dekoder I/O memakai sinyal $\overline{\text{IOR}}$ dan $\overline{\text{IOW}}$ untuk membaca dan menulis pada I/O port.

4.2.1. PERENCANAAN RANGKAIAN DEKODER MEMORY IBM PC/XT

Pada bab 3 telah ditunjukkan lokasi memory pada IBM PC/XT yang tidak di-dekode. Lokasi tersebut antara lain adalah A4000H sampai dengan A5FFFH. Lokasi memory sebesar 8 K byte ini dapat dipakai sebagai RAM bersama IBM PC/XT dan TMS32010. Memory yang akan dipakai pada alat ini adalah empat IC RAM HM6116 yang masing-masing berkapasitas 2 K byte sehingga dekoder memory direncanakan dengan interval 2 K byte.

Dalam perancangan dekoder memory ini perlu diperhatikan bahwa data bus TMS32010 16 bit sedangkan data bus IBM PC/XT 8 bit sehingga address dari IBM PC/XT harus dibedakan menjadi address genap dan address ganjil. Address genap digunakan untuk mengakses data D0 - D7 pada sistem minimum TMS32010 sedangkan address ganjil digunakan untuk mengakses data D8 - D15 pada sistem minimum TMS32010.

Pada gambar 4.2 ditunjukkan perbandingan address memory dilihat dari IBM PC/XT dan dari TMS32010.

Address IBM PC/XT													Address TMS32010												
A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0-1	A0-0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	8 bit	8 bit	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	8 bit	8 bit	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1	0	8 bit	8 bit	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	1	8 bit	8 bit	0	0	0	0	0	0	0	0	0	0	1	1

GAMBAR 4.2

PERBANDINGAN ADDRESS MEMORY PADA IBM PC/XT DAN TMS32010

Untuk membedakan address genap dan ganjil dipakai Least Significant Bit address bus IBM PC/XT yaitu A0 yang dihubungkan pada pin A0 IC Dekoder 74LS138. IC Dekoder ini mempunyai 8 buah output yang disebut dengan YA0, YA1, YA2

dan seterusnya yang aktif pada kondisi low. Output YA0 digunakan untuk mengakses data D0 - D7 pada Program Memory TMS32010 sedangkan output YA1 digunakan untuk mengakses data D8 - D15. Dengan cara yang sama output YA2 dan YA3 mengakses data pada RAM bersama.

4. 2. 2. PERENCANAAN DEKODER I/O PORT IBM PC/XT

Untuk operasi I/O, IBM PC/XT mengeluarkan lima buah sinyal yaitu sinyal untuk mengaktifkan TMS32010, sinyal BIO dan sinyal INT untuk TMS32010, sinyal untuk mengembalikan level IRQ2 pada level high dan sinyal untuk mereset TMS32010 dengan software. Kelima sinyal tersebut bersifat output karena itu sinyal IOW harus dimasukkan dalam sistem dekoder.

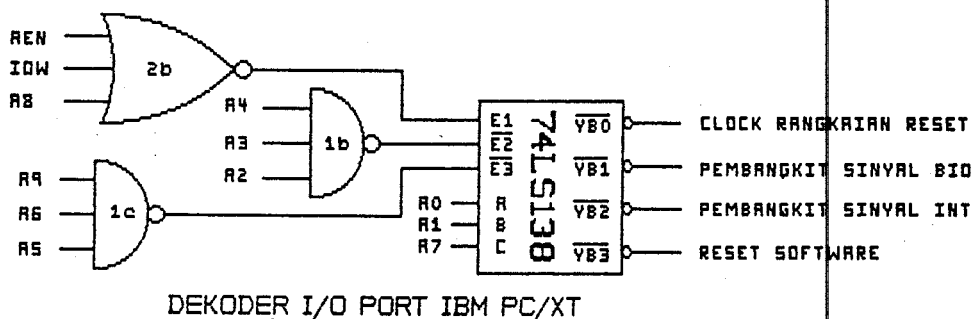
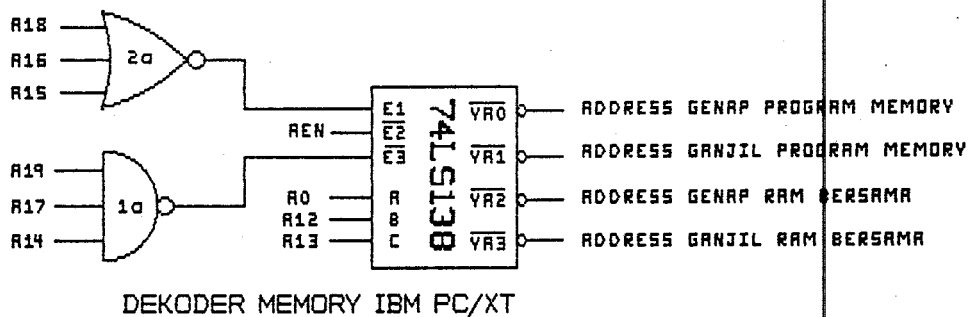
Pada bab 3 telah ditunjukkan lokasi I/O Port IBM PC/XT yang belum dipakai. Dari lokasi tersebut dipilih address:

- 27CH untuk mengaktifkan TMS32010.
- 27DH untuk membangkitkan sinyal BIO TMS32010.
- 27EH untuk membangkitkan sinyal INT TMS32010.
- 27FH untuk mereset TMS32010 dengan software.

Address 27CH juga digunakan untuk mengembalikan level IRQ2 pada level high. Hal ini bisa dilakukan karena sinyal untuk mengaktifkan TMS32010 hanya perlu dilakukan satu kali saja.

Untuk operasi I/O ini juga dipakai IC dekoder 74LS138 dengan output YB0 digunakan sebagai clock untuk mengaktifkan prosesor TMS32010, output YB1 digunakan untuk membangkitkan sinyal BIO, output YB2 digunakan untuk membangkitkan sinyal INT prosesor TMS32010 dan output YB3 digunakan untuk membangkitkan sinyal reset untuk TMS32010 dengan software. Output YB0 dipergunakan juga untuk mengembalikan level IRQ2 IBM PC/XT pada high level.

Pada gambar 4.3 berikut ini ditunjukkan sistem dekoder untuk memory dan operasi I/O.



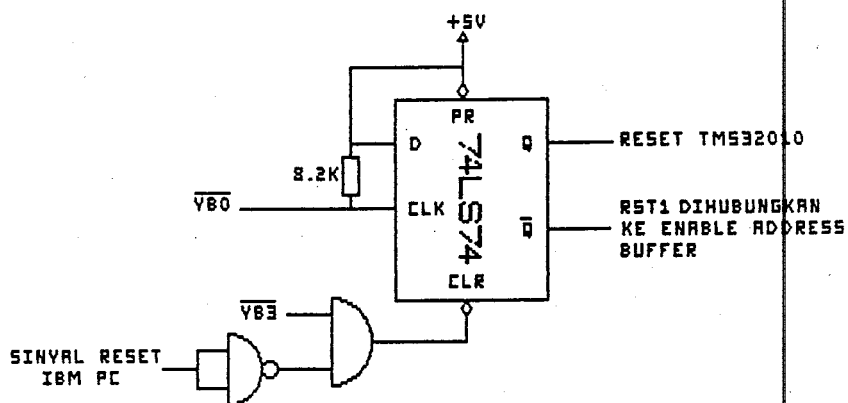
GAMBAR 4.3

RANGKAIAN DEKODER UNTUK MEMORY DAN I/O PORT

4. 2. 3. RANGKAIAN UNTUK MENGUBAH LEVEL RESET TMS32010

Sebelum program memory ditulis melalui IBM PC/XT, pin RESET TMS32010 dalam kondisi low (RESET). Setelah program memory selesai ditulis maka sinyal reset ini diubah menjadi high dengan software.

Pada bagian di atas telah dijelaskan bahwa untuk tujuan ini telah ditentukan address I/O 27CH. Output dekoder untuk address I/O ini adalah YB0 yang digunakan sebagai clock untuk mengubah RESET TMS32010 dari level low menjadi high.



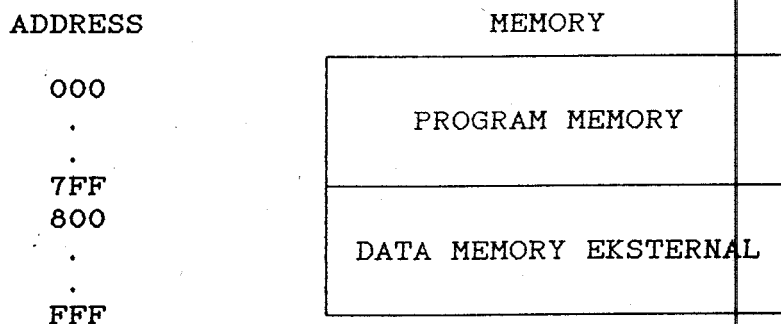
GAMBAR 4. 4

RANGKAIAN UNTUK MENGUBAH LEVEL RESET TMS32010

4. 3. PERENCANAAN RANGKAIAN BUFFER DAN MEMORY

4. 3. 1. PETA PEMAKAIAN MEMORY TMS32010

Sebelum perencanaan rangkaian buffer dan memory dibahas, maka lebih dulu harus diketahui pemakaian memory TMS32010. Memory TMS32010 dibagi menjadi dua bagian yang masing-masing berkapasitas 2 K word. Bagian yang pertama digunakan sebagai program memory yang berisi program TMS32010 dan data koefisien twiddle factor. Bagian yang kedua dipergunakan sebagai data memory eksternal yang berisi hasil pengolahan data.



GAMBAR 4. 5

PETA PEMAKAIAN MEMORY TMS32010

4. 3. 2. PERENCANAAN BUFFER

Memory pada minimum sistem TMS32010 dapat diakses dari IBM PC/XT dan dari TMS32010. Agar tidak terjadi konflik pada data bus atau address bus IBM PC/XT dan TMS32010 maka memory harus diberi buffer pada data bus dan

pada address bus.

Program memory TMS32010 ditulis melalui IBM PC/XT pada saat TMS32010 dalam kondisi RESET. Pada kondisi RESET ini data bus TMS32010 berada dalam keadaan impedansi tinggi sehingga data bus dari TMS32010 yang menuju program memory ini tidak perlu diberi buffer.

Address bus dari TMS32010 perlu diberi buffer sebab pada saat TMS32010 direset address busnya berada dalam level low. Untuk menghindari konflik, pada saat program memory TMS32010 ditulis melalui IBM PC/XT buffer ini harus dalam kondisi impedansi tinggi. Sesudah program memory selesai ditulis, RESET TMS32010 diubah levelnya menjadi high dan buffer ini di-enable. Jadi enable buffer address ini dihubungkan pada inverting sinyal reset TMS32010.

Data bus dari IBM PC/XT ke program memory dipisah menjadi dua bagian yaitu data bus untuk address D0-D7 dan data bus untuk D8-D15 dari TMS32010. Keenambelas bit data tersebut diakses sekaligus oleh TMS32010. Oleh karena itu agar D0-D7 tidak konflik dengan D8-D15 maka kedua data bus tersebut perlu dipisahkan melalui buffer. Enable data buffer tersebut dihubungkan dengan output dekoder memory IBM PC/XT sesuai dengan lokasi memory yang bersangkutan yaitu YA0 untuk data D0-D7 dan YA1 untuk data D8-D15.

Selama TMS32010 mengakses program memory, IBM PC/XT mengeluarkan address bus sendiri. Oleh karena itu agar

Kedua address bus tidak konflik maka address bus dari IBM PC/XT harus di-buffer. Buffer address ini akan aktif bila salah satu dari chip program memory diakses IBM PC/XT, karena itu enable buffer address ini dihubungkan pada output YAO AND YA1. Dengan alasan yang sama maka data bus dan address bus dari IBM PC/XT ke data memory eksternal (RAM bersama) perlu diberi buffer.

Data bus dari TMS32010 ke data memory eksternal perlu juga diberi buffer sebab saat data memory eksternal diakses oleh IBM PC/XT, TMS32010 harus tetap membaca program memory. Buffer data ini aktif hanya pada saat TMS32010 mengakses 2 K word memory bagian atas yaitu pada kondisi di mana A11 berada pada level '1'. Jadi enable buffer data ini dihubungkan pada inverting address A11. Dengan alasan yang sama maka buffer address dari TMS32010 ke data memory eksternal juga diberi buffer.

4.3.3. HUBUNGAN SINYAL PEMILIH KE MEMORY

Memory pada alat ini dapat diakses dari dua arah yaitu dari arah IBM PC/XT dan TMS32010. Oleh karena itu chip select memory juga dihubungkan dengan sinyal pemilih dari IBM PC/XT dan TMS32010. Sinyal pemilih dari IBM PC/XT berasal dari output dekoder memory. YAO untuk address genap program memory, YA1 untuk address ganjil program memory, YA2 untuk address genap data memory eksternal dan

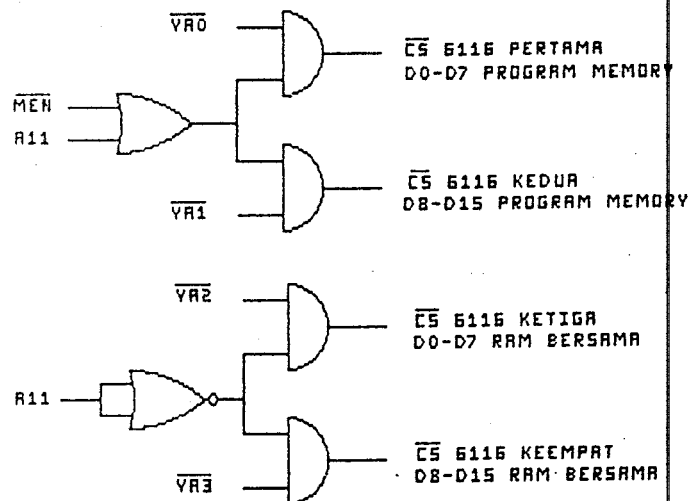
YA3 untuk address ganjil data memory eksternal.

Sinyal pemilih dari TMS32010 adalah bit A11 dari address bus. Jika A11 low maka TMS32010 akan mengakses address 000H sampai dengan 7FFH yang merupakan program memory sedangkan jika A11 high maka TMS32010 akan mengakses address 800H sampai dengan FFFH yang merupakan data memory eksternal (RAM bersama).

Selain itu perlu diperhatikan bahwa TMS32010 juga mengeluarkan sinyal untuk operasi I/O. Address yang dipakai pada operasi I/O ini adalah 000H sampai dengan 007H. Untuk mengetahui apakah suatu address merupakan address operasi memory atau address operasi I/O dipakai sinyal MEN. Sinyal MEN akan aktif (low) hanya pada operasi memory. Karena itu sinyal pemilih pada program memory bit A11 perlu di-OR-kan dengan MEN. Jadi TMS32010 akan mengakses program memory jika A11 dan MEN keduanya dalam level low dan melakukan operasi I/O jika A11 low dan MEN high.

Address 800H sampai dengan FFFH yang dipakai untuk data memory eksternal tidak dipakai dalam operasi I/O karena itu sinyal A11 bisa langsung dipakai sebagai sinyal pemilih setelah lebih dulu di-inverting.

Hubungan sinyal pemilih dari IBM PC/XT dan TMS32010 ke RAM ditunjukkan pada gambar 4.6 berikut ini.



GAMBAR 4.6

HUBUNGAN SINYAL PEMILIH PADA MEMORY

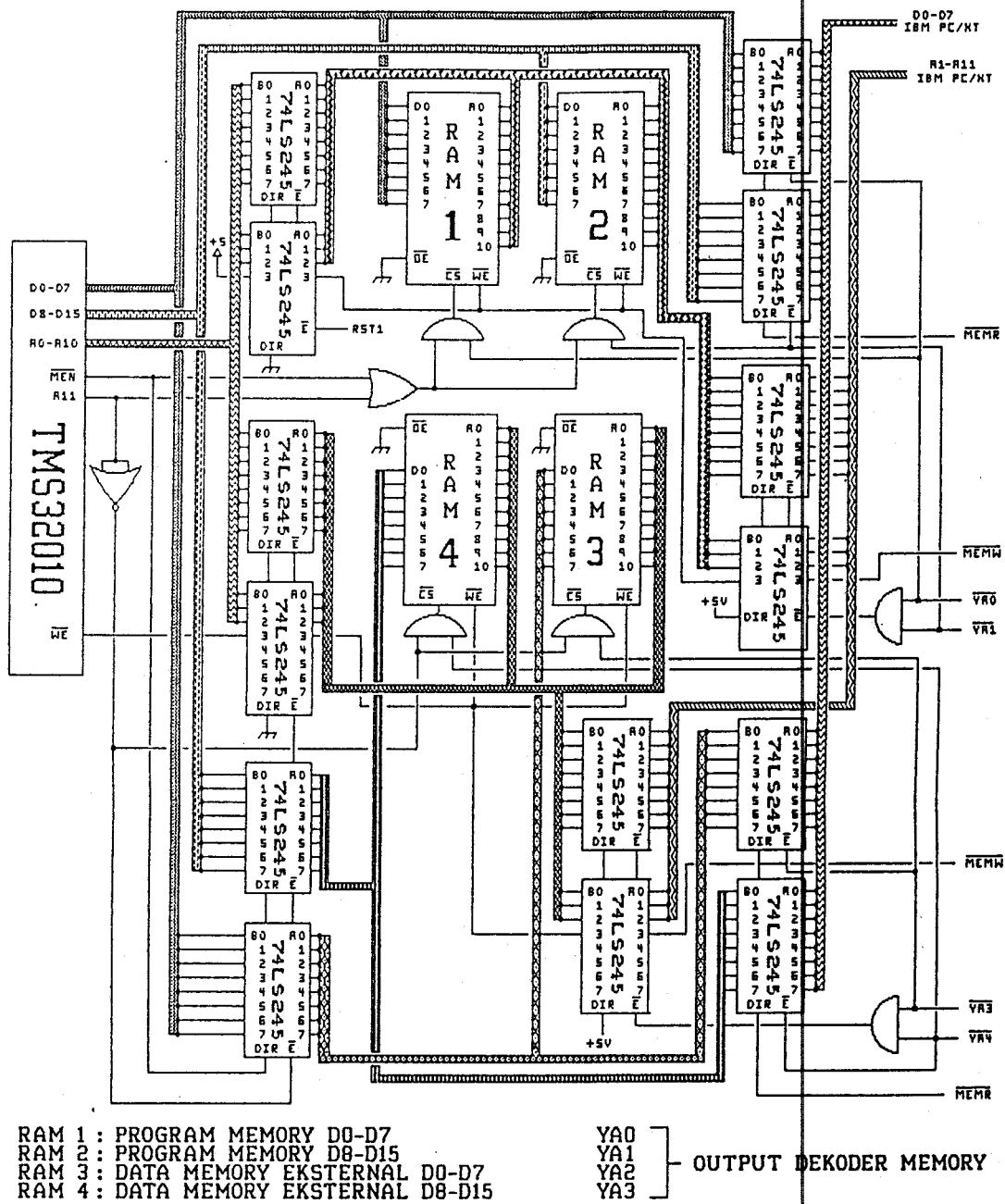
Rangkaian lengkap buffer dan memory ditunjukkan pada gambar 4.7.

4.3.4. PERHITUNGAN WAKTU AKSES MAKSIMUM MEMORY

Sebelum memory yang akan dipakai dipilih perlu lebih dulu dihitung waktu akses memory maksimum dari minimum sistem TMS32010. Waktu akses tersebut dapat dihitung dari rumus berikut: 58)

$$td1 + ta(A) + tsu(D) = tc(C)$$

58) Texas Instrument, op. cit., hal 6-2



GAMBAR 4. 7

RANGKAIAN LENGKAP BUFFER DAN MEMORY

dengan

t_{d1} = delay dari CLKOUT low ke address valid

$t_a(A)$ = waktu akses memory maksimum jika dihubungkan langsung ke TMS32010

$t_{su}(D)$ = waktu set-up dari data bus valid ke CLKOUT low.

$t_c(C)$ = cycle time sistem TMS32010 yang besarnya adalah 4 kali periode kristal yang dipakai.

Dari data TMS32010 diperoleh besarnya $t_{d1} = 50$ ns dan $t_{su}(D) = 50$ ns. Jika frekuensi kristal yang dipakai sebesar 12 MHz maka frekuensi clock sistem adalah seperempatnya yaitu 3 MHz dan periodenya ($t_c(C)$) adalah 333,33 ns. Dari keterangan di atas diperoleh access time maksimum yang diperbolehkan adalah :

$$\begin{aligned} t_a(A) &= t_c(C) - t_{d1} - t_{su}(D) \\ &= 333,33 - 50 - 50 \\ &= 233,33 \text{ ns} \end{aligned}$$

Pada perhitungan tersebut belum dimasukkan delay dari komponen buffer. Untuk mendapatkan access time memory, delay dari buffer data dan buffer address 74LS245 sebesar 18 ns perlu dikurangkan pada hasil di atas.

$$\begin{aligned} t_{acc} &= t_a(A) - 2 \times 18 \text{ ns} \\ &= 197,33 \text{ ns} \end{aligned}$$

Dari hasil tersebut dipilih komponen memory HM6116L-2 yang mempunyai access time sebesar 120 ns.

4. 4. PERENCANAAN RANGKAIAN UNTUK OPERASI I/O TMS32010

Selain melakukan operasi memory, TMS32010 juga melakukan beberapa operasi I/O. Ada dua komponen yang didukung oleh operasi I/O TMS32010. Kedua komponen tersebut adalah ADC0820 dan 74LS74 yang masing-masing digunakan dalam operasi ADC dan pembangkitan sinyal IRQ2 IBM PC/XT.

Pada operasi ADC diperlukan dua macam sinyal yaitu sinyal output untuk mengaktifkan ADC dan sinyal input untuk mengambil data dari ADC. Sedangkan pengiriman sinyal interrupt ke IBM PC/XT memerlukan sinyal output saja. Jadi operasi I/O yang dilakukan TMS32010 terdiri dari dua sinyal output dan satu sinyal input. Selain ketiga sinyal tersebut TMS32010 juga mengirimkan sinyal untuk menyalakan LED yang berguna untuk pengecekan.

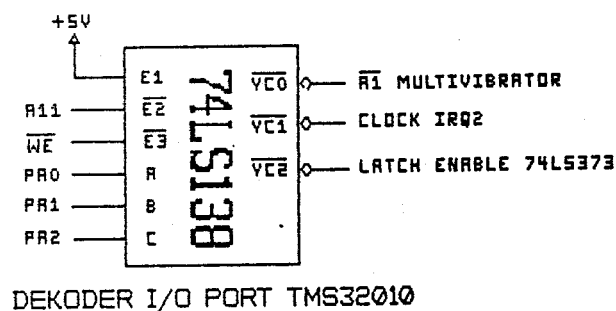
Pada operasi I/O, tiga bit terendah address bus menyatakan address dari port I/O. Sedangkan bit address bus yang lain berharga 0. Pada operasi input (IN), sinyal DE dari TMS32010 akan low sedang sinyal kontrol yang lain high. Pada operasi output (OUT) sinyal yang low adalah WE. Kedua sinyal ini dipakai sebagai indikator adanya operasi input atau output.

Harus diperhatikan bahwa pada eksekusi perintah TBLW sinyal WE juga akan berharga low, sedangkan sinyal kontrol lain high. Perintah TBLW ini berfungsi untuk menulis data ke eksternal memory (data memory) di mana harga A11 selalu

high, sedangkan pada perintah OUT harga A11 selalu low. Jadi kedua perintah tersebut dibedakan berdasarkan harga A11.

Untuk mengatur sinyal output mana yang bekerja maka dipakai rangkaian dekoder I/O. Rangkaian dekoder I/O ini akan aktif hanya bila sinyal A11 dan WE pada kondisi low. Karena itu enable dari rangkaian ini dihubungkan pada kedua pin tersebut, sedangkan address pemilih dihubungkan ke pin-pin I/O TMS32010 yaitu pin PA0, PA1 dan PA2.

Address yang dipakai pada operasi I/O ini adalah 000H untuk operasi ADC dan 001H untuk pengiriman sinyal interupt ke IBM PC/XT.



GAMBAR 4. 8

RANGKAIAN DEKODER UNTUK OPERASI OUTPUT TMS32010

TMS32010 melakukan hanya satu operasi input yaitu operasi pengambilan data dari ADC0820 karena itu tidak diperlukan rangkaian dekoder. Sinyal pengontrol langsung diambil dari sinyal DE.

4. 5. PERENCANAAN RANGKAIAN UNTUK SARANA KOMUNIKASI

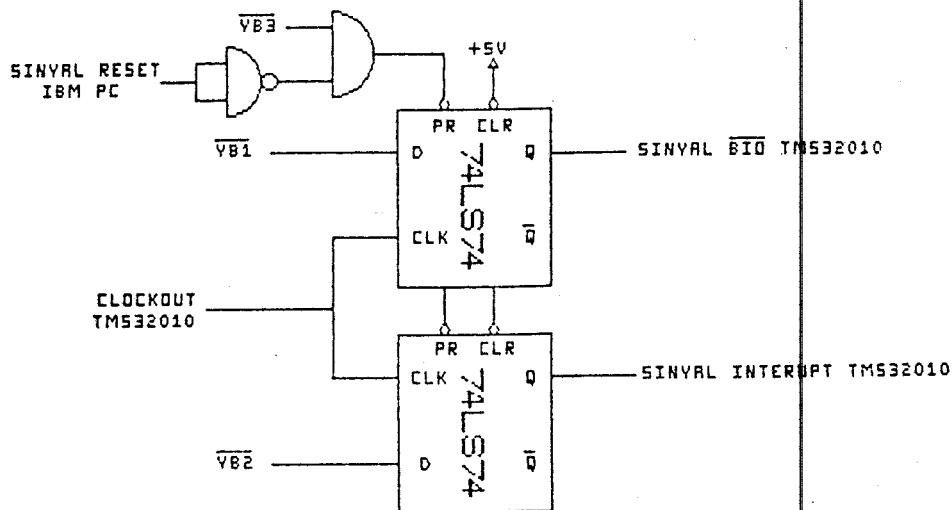
TMS32010 dan IBM PC/XT merupakan dua buah sistem yang bekerja secara bergantian. Agar kedua sistem tersebut dapat bekerja optimal, diperlukan sarana komunikasi. Komunikasi dilakukan dua arah, dari TMS32010 dan dari IBM PC/XT.

Komputer IBM PC/XT membangkitkan dua sinyal untuk sarana komunikasi dengan TMS32010. Kedua sinyal tersebut adalah sinyal BIO dan sinyal INT. Sedangkan prosesor TMS32010 membangkitkan satu sinyal untuk berkomunikasi yaitu sinyal IRQ2 (Interrupt Request level priority 2).

4. 5. 1 PERENCANAAN PEMBANGKIT SINYAL BIO DAN INT TMS32010

Sinyal BIO dan INT untuk TMS32010 diaktifkan melalui address port 27DH dan 27EH IBM PC/XT. Address port tersebut didekode untuk menghasilkan output YB1 dan YB2 (gambar 4. 3). Kedua pin tersebut dihubungkan pada D Flip-flop yang akan membangkitkan sinyal BIO dan INT untuk TMS32010.

Sistem pada IBM PC/XT mempunyai frekuensi clock yang berbeda dengan sistem pada TMS32010. Oleh karena itu untuk operasi I/O diperlukan rangkaian untuk menyelaraskan kedua sistem clock tersebut. Rangkaian sinkronisasi untuk sinyal BIO dan INT ditunjukkan pada gambar 4. 9.



GAMBAR 4.9 59)

RANGKAIAN SINKRONISASI SINYAL $\overline{\text{BIO}}$ DAN $\overline{\text{INT}}$

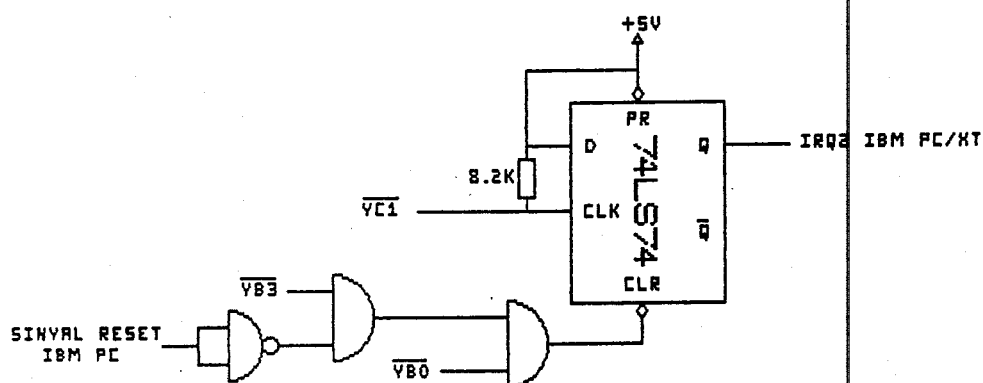
4.5.2. PERENCANAAN INTERRUPT UNTUK IBM PC/XT

Sinyal interrupt yang dibangkitkan TMS32010 berfungsi untuk sarana komunikasi dari TMS32010 ke IBM PC/XT. Output dari dekoder I/O port TMS32010 pin YC1 berfungsi sebagai clock dari IC D Flip-flop 74LS74. Sinyal ini akan mengubah kondisi level IRQ2 pada IBM PC/XT dari low ke high. Pada 4.2.4 dijelaskan bahwa adanya perubahan level dari low ke high akan menyebabkan IC 8259 mengirimkan sinyal interrupt ke 8088. Kemudian 8088 menjalankan suatu subroutine yang disebut Interrupt Service Routine (ISR).

Pada slot IBM PC/XT tidak disediakan sinyal yang bisa

59) Texas Instrument, op. cit., hal A-22

digunakan sebagai tanda bahwa ISR telah selesai dilakukan. Oleh karena itu dipakai I/O port dari IBM PC/XT address 27CH atau pin output dekoder I/O YB0 untuk mengembalikan IRQ2 pada level tegangan high. YB0 juga dipakai untuk mengaktifkan TMS32010. Pin YB0 ini tidak akan mempengaruhi kondisi RESET TMS32010 karena D Flip-flop pada rangkaian RESET TMS32010 dihubungkan langsung ke +5V (gambar 4.4) sehingga perubahan clock tidak akan mempengaruhi output.



GAMBAR 4. 10

RANGKAIAN PEMBANGKIT SINYAL INTERRUPT IBM PC/XT

4. 6. PERENCANAAN RANGKAIAN ANALOG TO DIGITAL CONVERTER

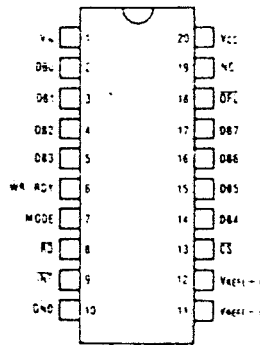
Dalam perencanaan alat ini diperlukan rangkaian Analog to Digital Converter (ADC) yang digunakan untuk mendapatkan sinyal digital dari sinyal input analog yang akan dicari transformasi Fourier-nya. Rangkaian ADC ini terdiri atas tiga bagian yaitu ADC0820 yang berfungsi sebagai ADC, dua buah IC 741 yang berfungsi untuk mengubah sinyal analog agar dapat diterima oleh ADC0820 dan monostable multivibrator 74121 yang berfungsi untuk menyelaraskan sinyal dari TMS32010 dengan ADC0820.

4. 6. 1. ANALOG TO DIGITAL CONVERTER ADC0820

ADC0820 adalah 8 bit Analog to Digital Converter yang menggunakan metoda halfflash sehingga mempunyai waktu konversi yang cepat. ADC ini memiliki rangkaian track and hold internal yang mampu mengkonversikan input dengan slew rate maksimum 100 mV/ μ s.

ADC ini dapat dioperasikan dalam dua mode yaitu Read mode dan Write-Read mode. Pada alat ini, ADC0820 dioperasikan pada mode Write-Read karena mempunyai waktu konversi yang kecil yaitu maksimum 1,5 μ s.

Pada gambar 4.11 berikut ini ditunjukkan konfigurasi pin-pin ADC 0820.



GAMBAR 4. 11

KONFIGURASI PIN-PIN ADC 0820

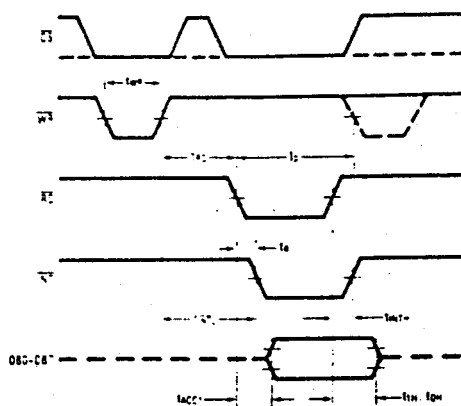
Pin-pin tersebut mempunyai fungsi sebagai berikut:

- Vin** : Masukan analog, $GND \leq Vin \leq Vcc$
- DB0-DB7** : Latch TRI-STATE data output
- WR/RDY** : Pada mode WR, jika CS low, konversi dimulai pada falling edge dari WR dan kurang lebih 800 ns dari rising edge sinyal WR, hasil konversi akan distrobe pada output latch selama sinyal RD tidak terjadi.
- MODE** : Input untuk menentukan mode operasi. Bila MODE low maka ADC0820 bekerja pada mode Read, jika MODE high ADC0820 bekerja pada mode Write-Read.
- RD** : Pada mode Write-Read, dengan CS low, TRI-STATE data output akan dilatch sesudah falling edge sinyal RD.

- INT : Pada mode Write-Read, INT akan low sebagai tanda bahwa konversi telah selesai dan data hasil konversi telah berada pada output latch. INT akan low sekitar 800 ns sesudah rising edge sinyal WR. INT direset oleh rising edge RD atau CS.
- GND : Ground
- Vref(-) : Tegangan referensi bawah, mempunyai jangkauan tegangan $GND \leq Vref(-) \leq Vref(+)$.
- Vref(+): Tegangan referensi atas, mempunyai jangkauan tegangan $Vref(-) \leq Vref(+)$ $\leq Vcc$.
- CS : CS harus low agar sinyal RD atau WR dapat diterima konverter.
- OFL : Bila input analog lebih besar dari Vref(+), OFL akan low pada akhir dari konversi.
- NC : Tidak dipakai
- Vcc : Tegangan supply + 5 V DC.

Pada rangkaian ADC, komponen ini dioperasikan pada mode Write-Read dengan CS selalu low. Proses konversi dimulai dengan memberikan sinyal WR low selama minimum 600 ns. Sekitar 800 ns kemudian sinyal INT akan low yang menandakan bahwa konversi telah selesai. Sinyal ini akan tetap low selama tidak ada rising edge dari RD atau CS. Kemudian dikirim sinyal RD untuk mengambil data dari ADC0820. Pada gambar 4.12 ditunjukkan Timing diagram

proses konversi ADC0820 untuk mode Write-Read.



GAMBAR 4. 12

DIAGRAM WAKTU KONVERSI ADC0820 MODE WRITE-READ

Pin Vref(-) dihubungkan ke GND sedangkan Vref(+) dihubungkan ke Vcc sehingga input ADC0820 mempunyai jangkauan tegangan dari 0 sampai 5 V.

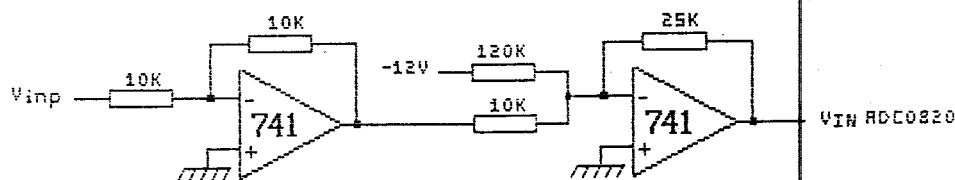
4. 6. 2. RANGKAIAN OP-AMP PENGUBAH TEGANGAN ANALOG

Jangkauan tegangan input ADC0820 dari 0 sampai 5 V sedangkan tegangan input analog bisa mempunyai harga negatif. Untuk mengatasi hal ini diperlukan rangkaian Op-Amp yang berfungsi untuk menaikkan tegangan negatif agar dapat diterima ADC0820.

Pada rangkaian ini direncanakan agar alat dapat menerima tegangan input dengan amplitudo 1 volt sehingga diperlukan rangkaian untuk mengubah sinyal dengan jangkauan -1 sampai +1 V menjadi 0 sampai 5 V. Dengan

demikian amplitudo maksimumnya menjadi 2,5 V.

Hal tersebut diwujudkan dengan dua buah Op-Amp 741 yang difungsikan sebagai penjumlahan inverting amplifier seperti ditunjukkan pada gambar 4.13.



GAMBAR 4.13

RANGKAIAN OP-AMP PENGUBAH TEGANGAN ANALOG

Op-Amp yang pertama berfungsi sebagai voltage follower, sedangkan Op-Amp yang kedua berfungsi untuk mengubah jangkauan tegangan input dari -1 sampai +1 menjadi 0 sampai 5 V.

$$V_{out1} = -V_{in}$$

$$\begin{aligned} V_{out2} &= -25 \left(\frac{V_{out1}}{10} - \frac{12}{120} \right) \\ &= 2,5 (V_{in} + 1) \end{aligned}$$

Untuk $V_{in} = -1$ V, $V_{out2} = 0$ V dan untuk $V_{in} = +1$ V,
 $V_{out} = +5$ V.

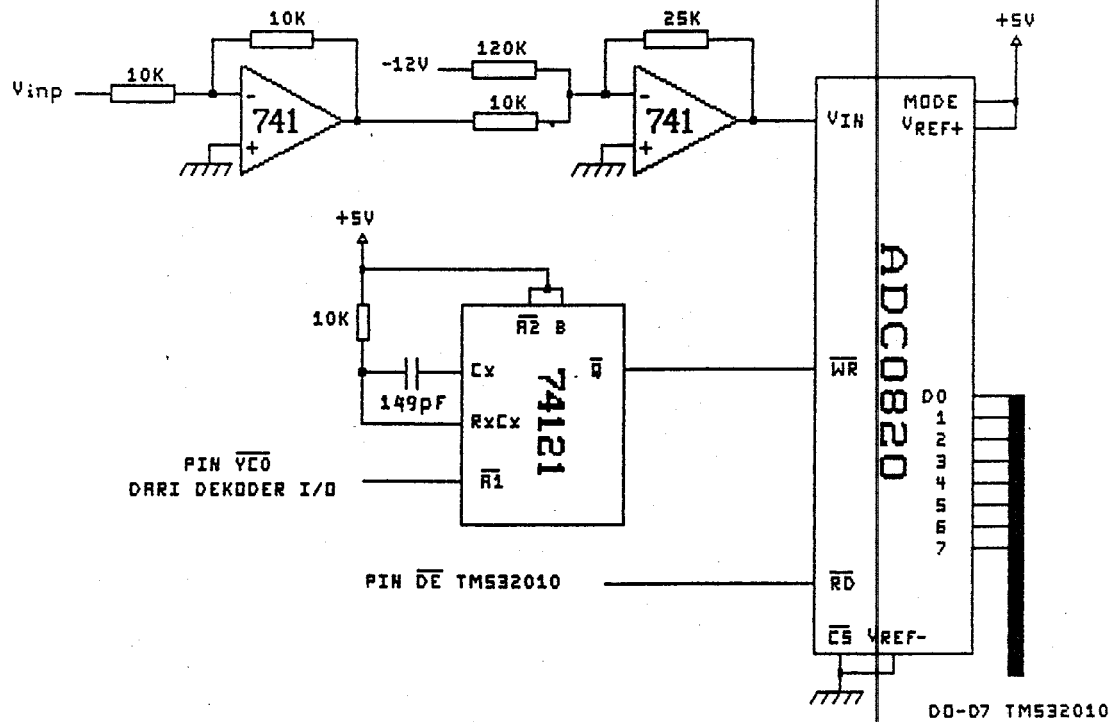
4. 6. 3. RANGKAIAN MONOSTABLE MULTIVIBRATOR

Kecepatan antara komponen Analog To Digital Converter dan prosesor TMS32010 berbeda sehingga diperlukan komponen tambahan untuk mengaturnya. Perbedaan kecepatan ini terletak pada lamanya sinyal WE. Pada saat TMS32010 mengeluarkan sinyal untuk mengaktifkan ADC maka sinyal WE akan berada dalam level low selama sekitar 300 ns sedangkan sinyal WE minimum yang dibutuhkan untuk ADC0820 sekitar 600 ns. Oleh karena itu ditambahkan komponen Monostable Multivibrator 74LS121 yang berfungsi untuk memperpanjang kondisi low sinyal WE TMS32010. Monostable multivibrator ini akan bekerja bila ditrigger oleh sinyal negative going edge dari output dekoder I/O TMS32010 pin YC0 (gambar 4.12). Pin YC0 ini merupakan hasil dekoding dari address port I/O 000H. Port ini merupakan port output yang berfungsi untuk mengaktifkan ADC.

Pada IC 74LS121 terdapat tiga pin kontrol yaitu A1, A2 dan B. IC ini akan bekerja jika pin B dan A2 diberi input high sedangkan pin A1 diberi negative going edge trigger. Karena itu pin B dan A2 dihubungkan pada Vcc dan pin A1 dihubungkan pada pin YC0.

Output 74LS121 diambil dari inverting output karena ADC0820 memerlukan sinyal WR low untuk mengaktifkannya. Lamanya sinyal output ini diatur sesuai dengan besarnya

resistor pada $RxCx$ dan kapasitor pada Cx yaitu sesuai dengan rumus $t = 0,69 RC$ ns dengan R adalah besarnya resistor dalam K Ohm, C adalah besarnya kapasitor dalam pF dan t adalah lamanya sinyal output dalam ns. Dengan $R = 10 K\Omega$ dan $C = 149$ pF maka didapat sinyal output monostable sebesar 1028 ns.



GAMBAR 4. 14
RANGKAIAN ANALOG TO DIGITAL CONVERTER

4. 7. PERENCANAAN PERANGKAT LUNAK (SOFTWARE)

Perangkat lunak pada alat tugas akhir ini dibedakan menjadi dua yaitu perangkat lunak untuk IBM PC/XT yang ditulis dalam bahasa tingkat tinggi TURBO PASCAL versi 4.0 dan perangkat lunak untuk prosesor TMS32010 yang ditulis dalam bahasa mesin TMS32010.

Perangkat lunak untuk IBM PC/XT digunakan untuk proses inisialisasi prosesor TMS32010 serta menampilkan data hasil olahan TMS32010 dalam bentuk gambar pada monitor.

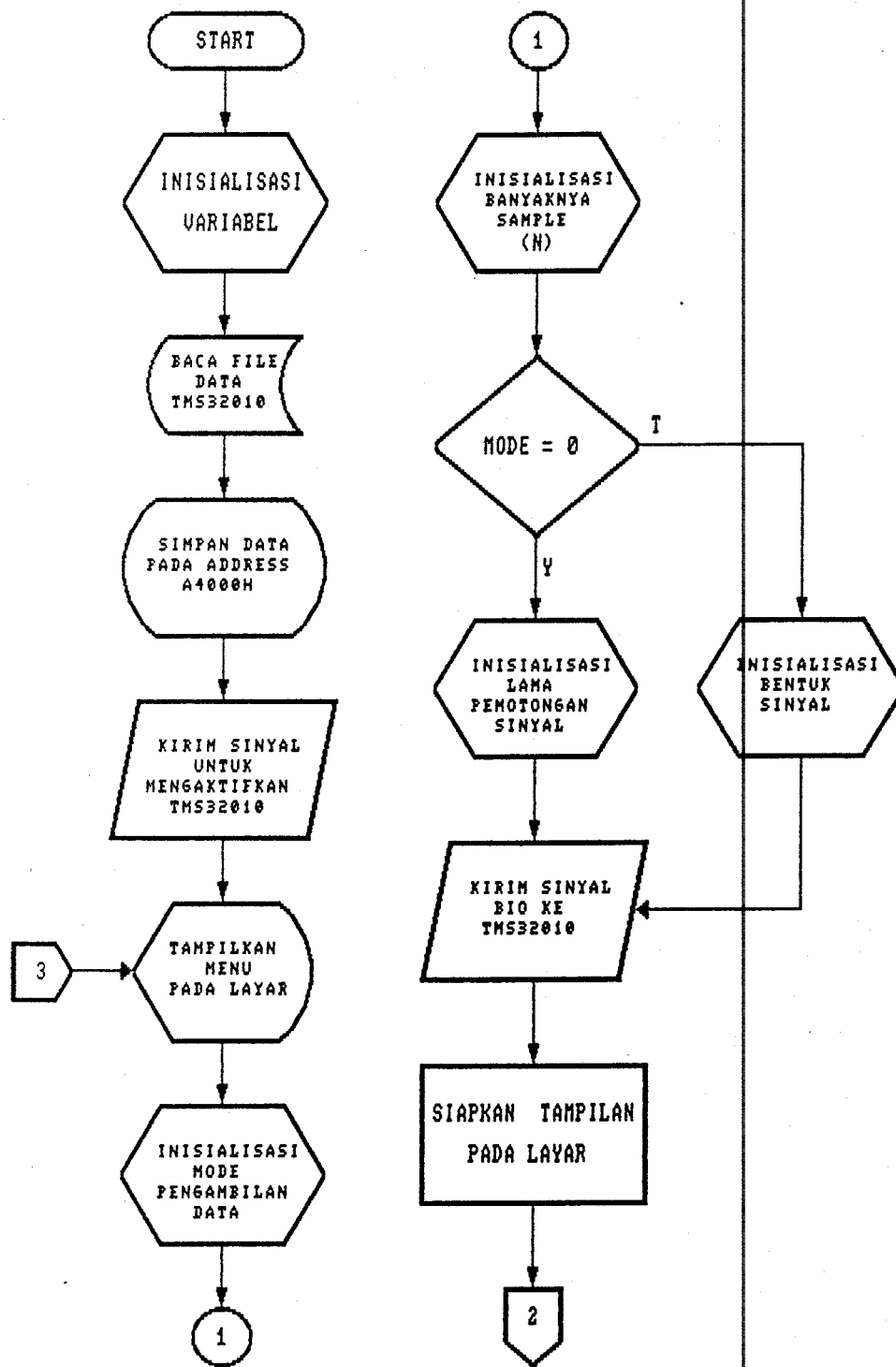
Perangkat lunak TMS32010 digunakan untuk mengolah data yang diinputkan pada ADC0820 berdasarkan algoritma FFT.

4. 7. 1. PERENCANAAN PERANGKAT LUNAK UNTUK IBM PC/XT

Perangkat lunak untuk IBM PC/XT dibagi menjadi 3 bagian yaitu:

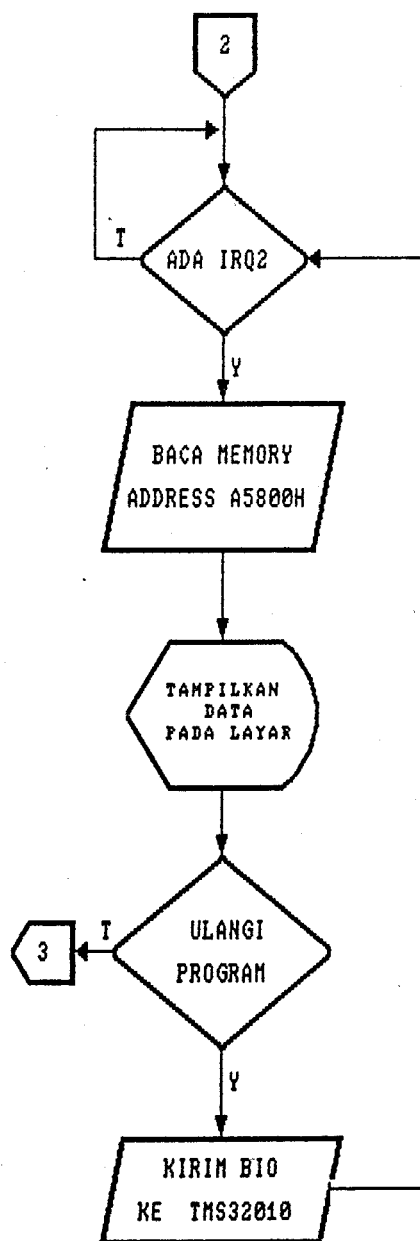
- Program untuk menulis program memory TMS32010.
- Program untuk inisialisasi prosesor TMS32010.
- Program untuk menampilkan data hasil olahan pada layar monitor.

Secara umum, software IBM PC/XT bekerja dengan flowchart sebagai berikut :



GAMBAR 4. 15

FLOWCHART PERANGKAT LUNAK IBM PC/XT



GAMBAR 4. 15

FLOWCHART PERANGKAT LUNAK IBM PC/XT (Lanjutan)

Dalam perencanaan software IBM PC/XT harus ada sinkronisasi antara IBM PC/XT dengan TMS32010. Sinkroni-

sasi ini dilakukan dengan sinyal BIO dan IRQ2. Pada saat IBM PC/XT mengambil data pada RAM bersama TMS32010 menunggu adanya sinyal BIO. Demikian juga pada saat TMS32010 menulis pada RAM bersama (data memory eksternal), IBM PC/XT menunggu adanya IRQ2. Sehingga tidak mungkin pada saat yang sama TMS32010 dan IBM PC/XT mengakses RAM bersama.

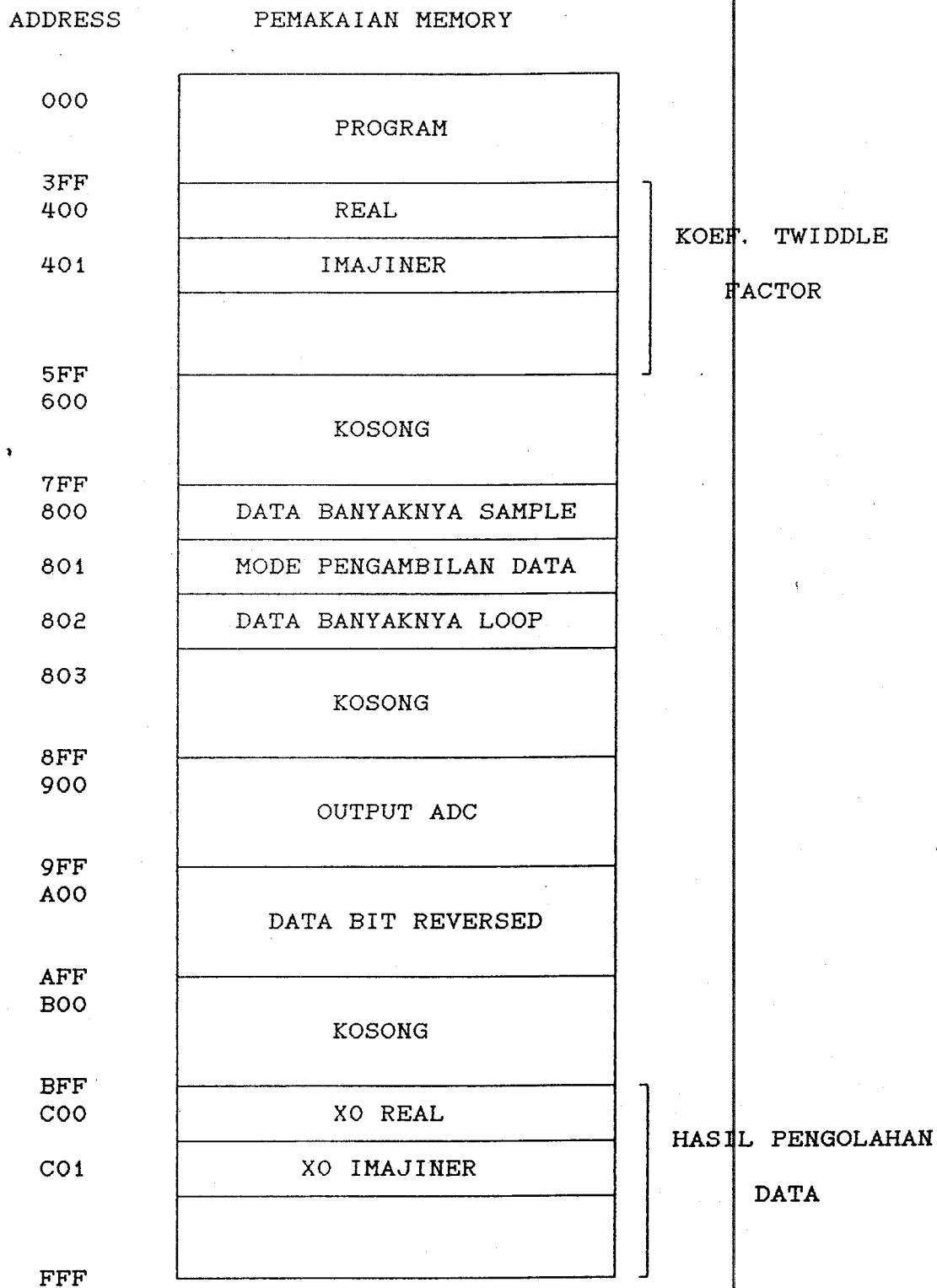
4. 7. 2. PERENCANAAN PERANGKAT LUNAK UNTUK TMS32010

Sebelum perencanaan perangkat lunak TMS32010 dilakukan, lebih dulu harus direncanakan peta pemakaian memory TMS32010. Gambar 4.16 menunjukkan pemakaian memory pada TMS32010.

Perangkat lunak TMS32010 dibagi menjadi tiga bagian yaitu :

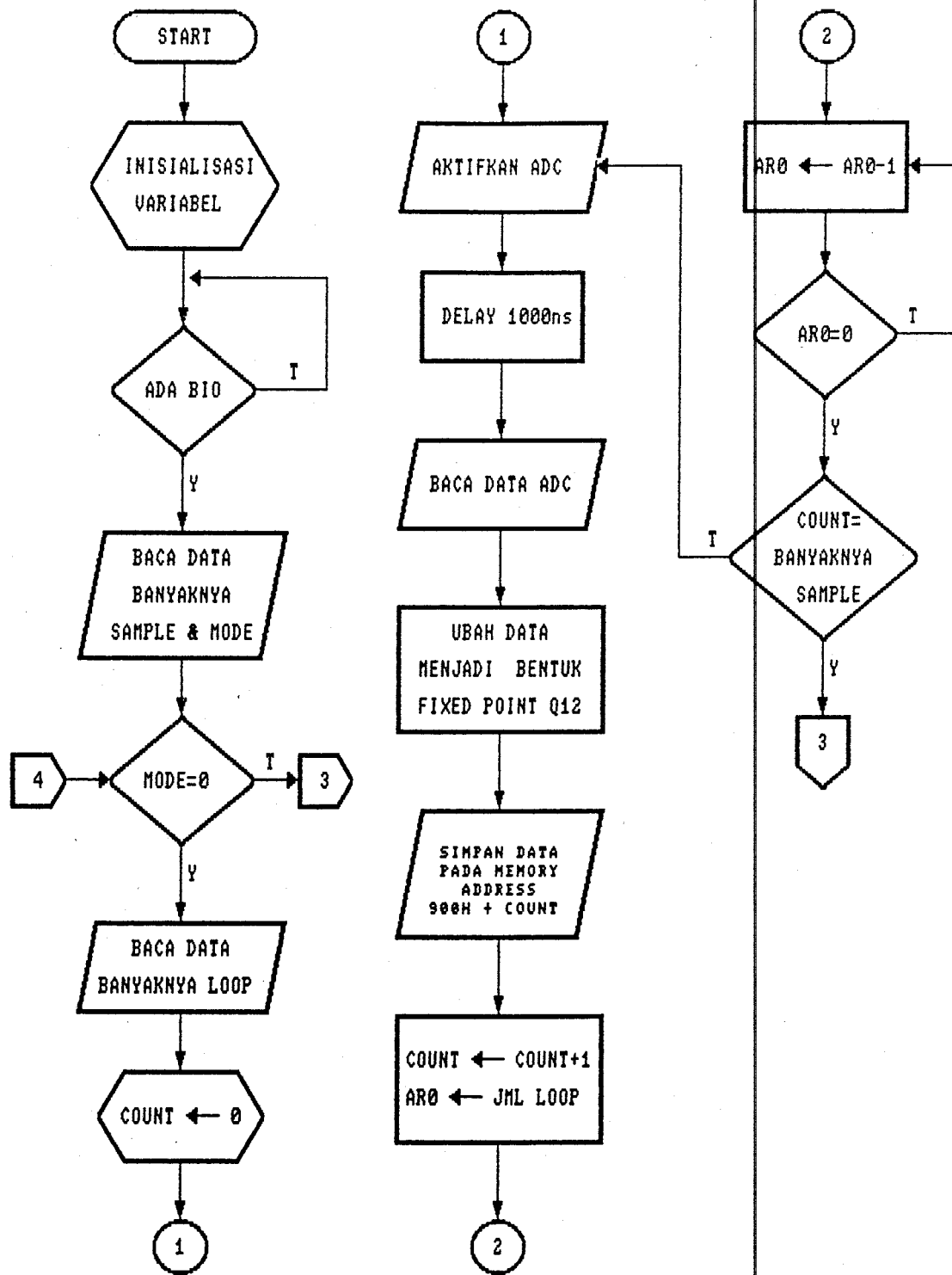
- program untuk mengambil sampel dari ADC0820 dan menyimpannya pada address awal 900H untuk mode pengambilan data dari ADC.
- program untuk mengambil data pada address awal 900H dan menyimpannya pada address C00H pada posisi bit reversed.
- program untuk mengolah data berdasarkan algoritma FFT dan menyimpan data kembali pada address awal C00H.

Program untuk TMS32010 secara umum mempunyai flow-chart seperti ditunjukkan pada gambar 4.17.



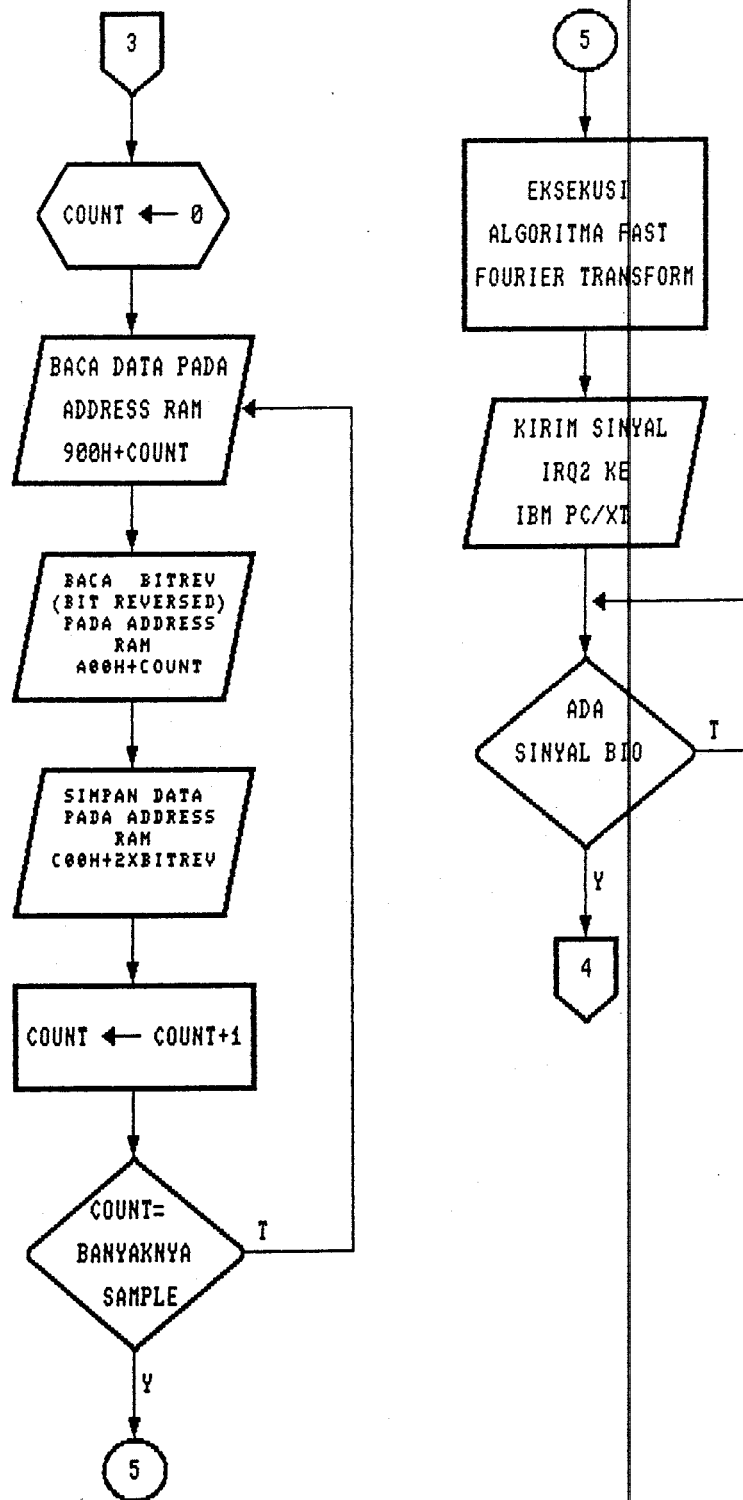
GAMBAR 4. 16

PETA PEMAKAIAN MEMORY TMS32010



GAMBAR 4.17

FLOWCHART PERANGKAT LUNAK TMS32010



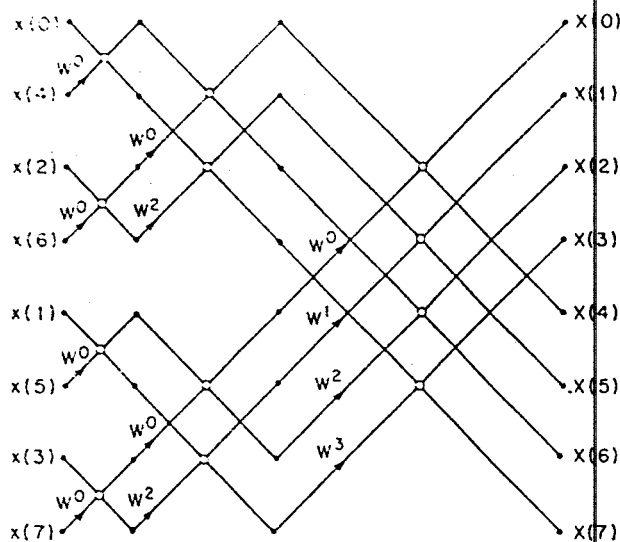
GAMBAR 4.17

FLOWCHART PERANGKAT LUNAK TMS32010 (Lanjutan)

4. 7. 3. PERENCANAAN PROGRAM BERDASARKAN ALGORITMA FFT

Pada gambar 4.18 ditunjukkan bagaimana algoritma FFT dilaksanakan untuk 8 point data. Melalui gambar ini akan dijelaskan bagaimana algoritma FFT diwujudkan.

Pada gambar tersebut ditunjukkan bahwa pada tiap tahap terdapat $N/2$ pasangan node. Data input untuk tiap tahap dipakai hanya pada tahap itu saja sehingga bisa dilakukan perhitungan di tempat (in place computation). Misalkan pasangan node $x_1(0)$ dan $x_1(4)$ dihitung dari data input $x_0(0)$ dan $x_0(4)$. Kemudian pasangan $x_2(0)$ dan $x_2(4)$ dihitung dari data $x_1(0)$ dan $x_1(4)$. Jadi lokasi memory data $x_0(0)$ dan $x_0(4)$ dapat diisi dengan $x_1(0)$ dan $x_1(4)$. Dengan cara ini pemakaian memory bisa dihemat.



GAMBAR 4. 18 60)

SIGNAL FLOW GRAPH FFT DECIMATION IN TIME UNTUK $N = 8$

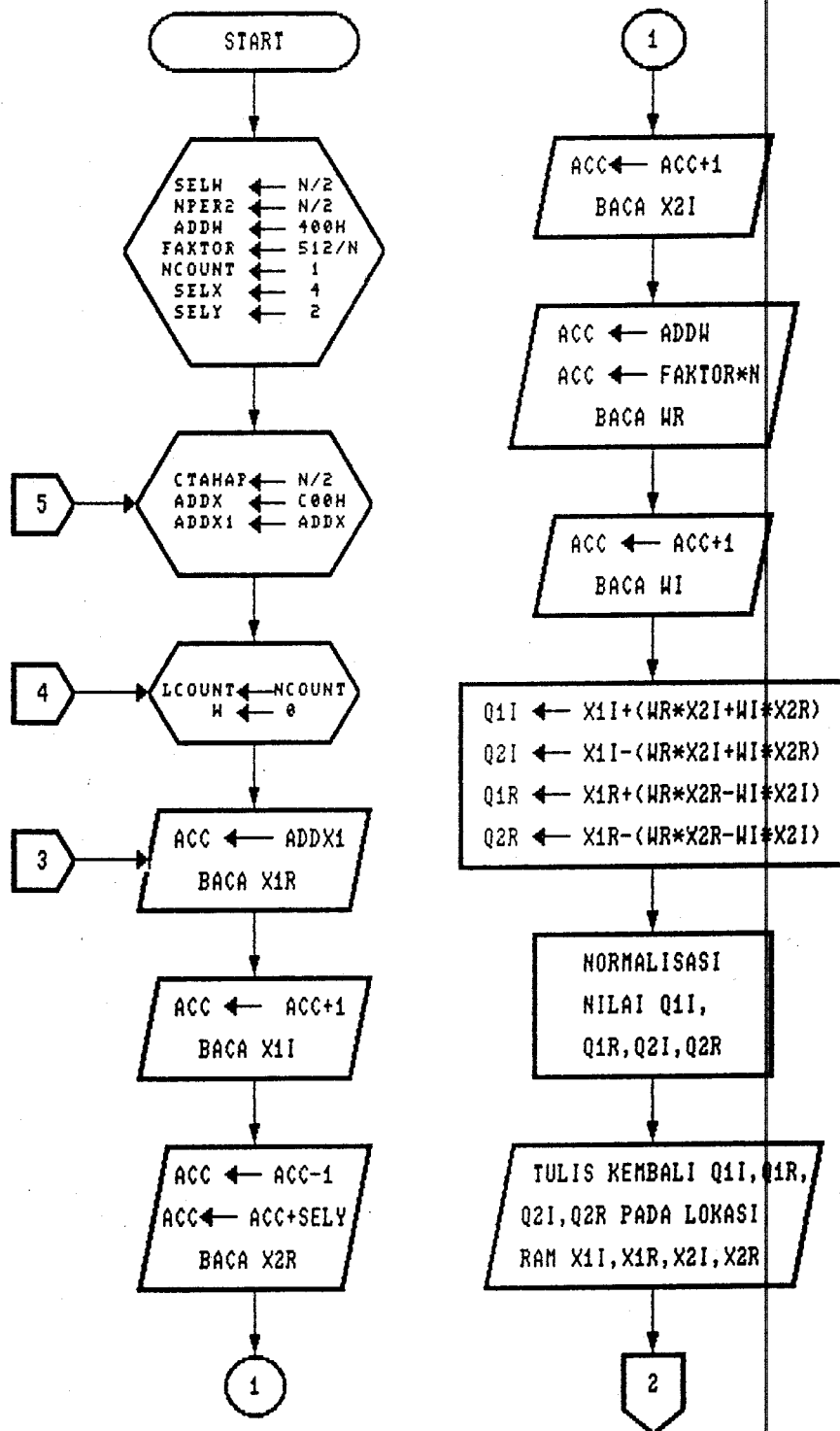
60) Rabiner, L.R. and Gold B., op. cit. hal 362

Jika diperhatikan, pada gambar 4.18 nampak bahwa jarak antar input pasangan node mempunyai pola tertentu. Pola ini tergantung pada tahap. Setiap input pada pasangan node mempunyai jarak yang sesuai dengan rumus $2^{(t-1)}$ dengan t adalah tahap. Misalkan pada tahap 2, pasangan node mempunyai jarak $2^{(2-1)} = 2$.

Hal lain yang perlu diperhatikan adalah input yang dilewati. Input yang dilewati adalah input yang sudah dipasangkan untuk membentuk pasangan node dengan input di atasnya. Pada saat iterasi menunjuk pada input ini, tidak dilakukan perhitungan. Banyaknya input yang terlewat ini mempunyai pola sesuai dengan rumus $2^{(t-1)}$, dengan t adalah tahap.

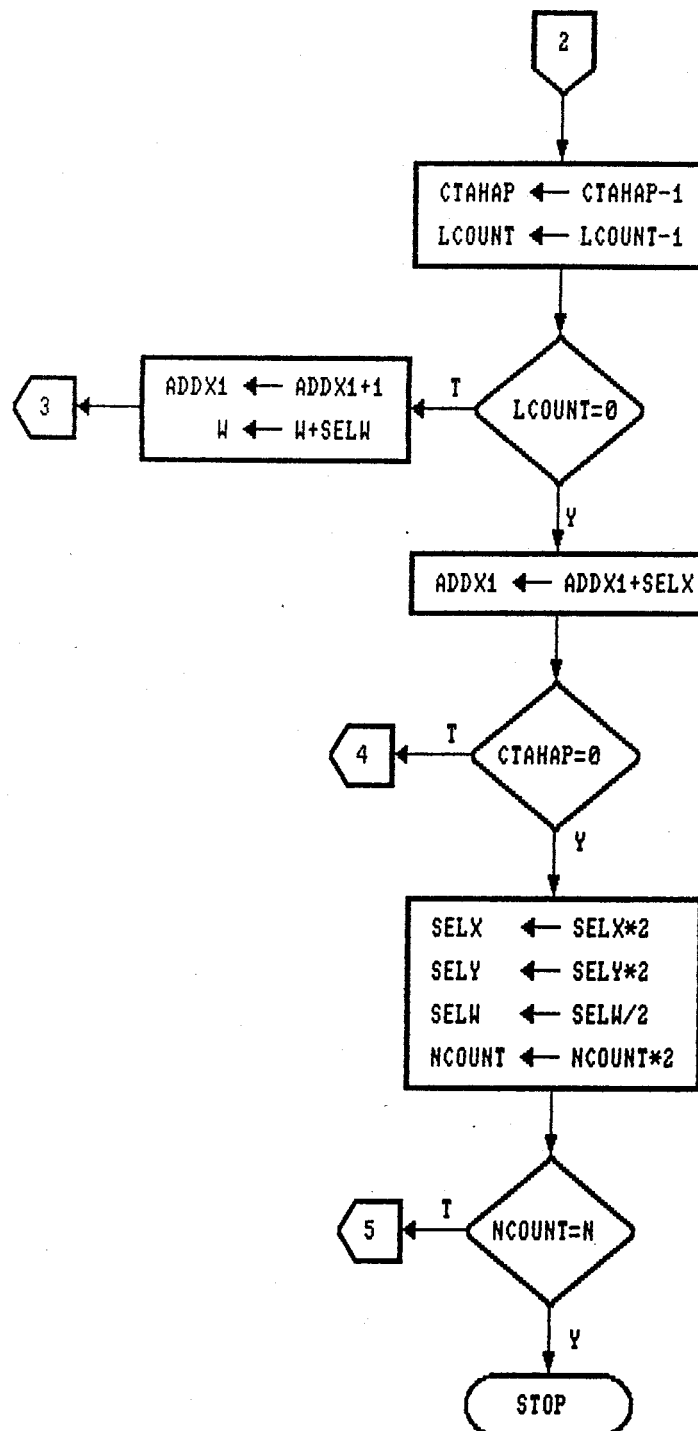
Banyaknya input yang dilewati ini juga menunjukkan berapa kali perhitungan dilakukan secara berurutan sebelum melakukan lompatan. Perhitungan yang dilakukan ini disebut operasi sejajar. Jadi operasi sejajar bisa dilakukan sampai sebanyak $2^{(t-1)}$ kali. Kemudian input dilompati $2^{(t-1)}$ dan operasi sejajar dimulai lagi. Sebagai contoh pada gambar 4.18, pada tahap 2 input yang dilewati = 2. Jadi operasi sejajar dilakukan dua kali sebelum melompati dua input.

Pada perhitungan juga diperlukan koefisien twiddle factor W_p . Faktor pemberat p yang dipakai selalu diinisialisasi dengan 0 dan kemudian dinaikkan dengan selang $N/2^t$ setiap kali operasi sejajar dilakukan. Faktor pembe-



GAMBAR 4. 19

FLOWCHART PENGOLAHAN DATA SESUAI DENGAN ALGORITMA FFT



GAMBAR 4.19 (Lanjutan)

FLOWCHART PENGOLAHAN DATA SESUAI DENGAN ALGORITMA FFT

rat dikembalikan lagi ke 0 setiap kali ada input yang dilompati.

Dari sifat-sifat di atas maka dapat ditentukan flowchart pengolahan data sesuai dengan algoritma FFT seperti ditunjukkan pada gambar 4.19.

BAB V

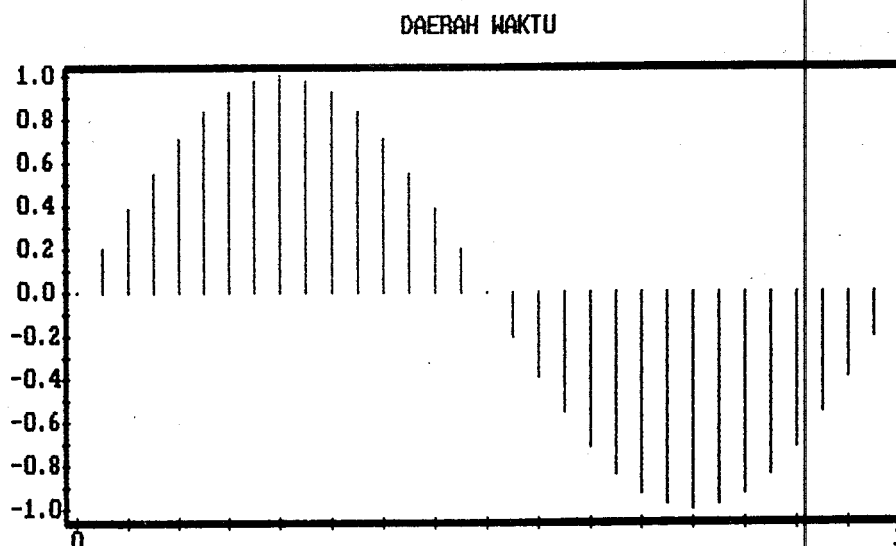
ANALISIS DATA DAN HASIL PENGUKURAN

5.1. ANALISIS FUNGSI-FUNGSI YANG TERSEDIA

Di bawah ini dianalisa fungsi-fungsi yang banyak digunakan yaitu fungsi sinus, kotak dan segitiga.

5.1.2. ANALISIS FUNGSI SINUS

Pada gambar 5.1 berikut ini ditunjukkan fungsi sinus yang disampling sebanyak 32 data. Ketigapuluhdua data ini digunakan sebagai data input Fast Fourier Transform.



GAMBAR 5.1
FUNGSI SINUS TERSAMPLING 32 DATA

Transformasi fourier kontinyu fungsi sinus dapat dihitung berdasarkan rumus (1-1) berikut ini:

$$Y(f) = \int_{-\infty}^{+\infty} y(t) e^{-j2\pi ft} dt$$

dengan:

$$y(t) = \sin(2\pi f_0 t)$$

$$f_0 = 1/T = 1$$

Jadi didapat

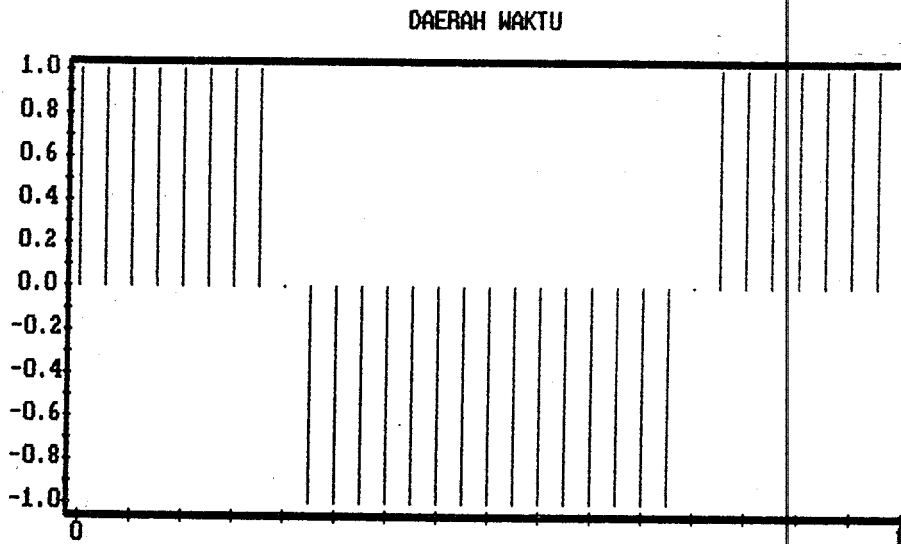
$$\begin{aligned} Y(f) &= \int_{-\infty}^{+\infty} \sin(2\pi f_0 t) e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{+\infty} \frac{1}{2} j (e^{-j2\pi t} - e^{j2\pi t}) e^{-j2\pi ft} dt \end{aligned}$$

$$\begin{aligned} Y(f) &= \frac{j}{2} \int_{-\infty}^{+\infty} (e^{j2\pi(f-1)t} - e^{j2\pi(f+1)t}) dt \\ &= \frac{j}{2} (\delta(f-1) - \delta(f+1)) \end{aligned} \quad (5-1)$$

Jadi transformasi Fourier dari fungsi sinus hanya ada pada frekuensi imajiner 1 dan -1 yang masing-masing amplitudonya 1/2 dan -1/2 kali amplitudo fungsi sinus pada daerah waktu.

5.1.2. ANALISIS FUNGSI KOTAK

Fungsi kotak yang disampling 32 data ditunjukkan pada gambar 5.2 berikut ini.



GAMBAR 5.2

FUNGSI KOTAK YANG TERSAMPLING 32 DATA

Hasil sampling fungsi tersebut merupakan input Transformasi Fourier. Dari gambar 5.2 dapat dilihat bahwa fungsi kotak mempunyai harga 1 untuk $n = 0$ sampai dengan $n = N/4 - 1$ dan $n = 3N/4 + 1$ sampai dengan $n = N - 1$, 0 untuk $n = N/4$ dan $n = 3N/4$, dan -1 untuk $n = N/4 + 1$ sampai dengan $3N/4 - 1$, dengan N adalah jumlah sample.

Fungsi kotak kontinyu yang periodik dapat dihasilkan dari proses berikut ini:

- Mula-mula fungsi kotak dibuat pada daerah $-T/4$ sampai dengan $T/4$.
- Konvolusikan fungsi tersebut dengan fungsi impulse yang periodenya T .
- Hilangkan sinyal DC dengan menggeser fungsi tersebut ke bawah sebesar 1.

Dengan mengetahui proses pembuatannya, transformasi fourier fungsi kotak dapat dicari sebagai berikut:

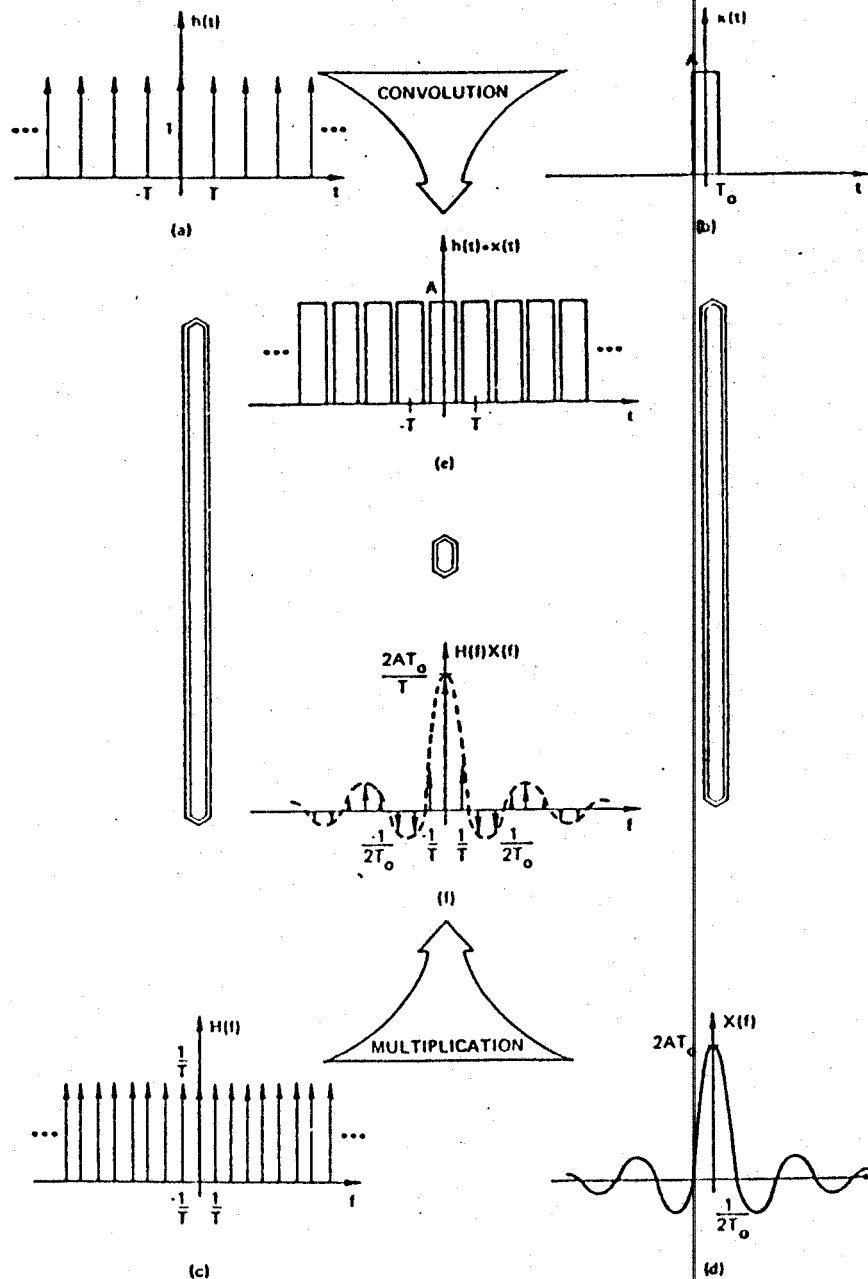
- Cari transformasi fourier fungsi kotak pada daerah waktu $-T/4$ sampai dengan $T/4$.
- Kalikan hasilnya dengan transformasi fourier fungsi impulse yang periodenya T .
- Hilangkan sinyal pada frekuensi = 0

Pada gambar 5.3 ditunjukkan bagaimana proses pembuatan fungsi kotak periodik dan juga transformasi fouriernya.

Berikut ini akan dihitung transformasi fourier dari fungsi kotak tunggal.

$$H(f) = \int_{-\infty}^{+\infty} h(t) e^{-j2\pi ft} dt$$

$$= \int_{-T/4}^{T/4} (2) e^{-j2\pi ft} dt$$



GAMBAR 5.3 61)

PENGHITUNGAN TRANSFORMASI FOURIER FUNGSI KOTAK PERIODIK SECARA GRAFIS

- (a). FUNGSI IMPULSE DENGAN PERIODE T
- (b). FUNGSI KOTAK TUNGGAL
- (c). HASIL TRANSFORMASI FOURIER FUNGSI IMPULSE
- (d). HASIL TRANSFORMASI FOURIER FUNGSI KOTAK TUNGGAL
- (e). FUNGSI KOTAK PERIODIK
- (f). HASIL TRANSFORMASI FOURIER FUNGSI KOTAK PERIODIK

61) Brigham, O.E., opcit, hal 62.

Fungsi kotak tersebut merupakan fungsi genap sehingga persamaan di atas dapat disederhanakan menjadi:

$$\begin{aligned}
 H(f) &= 2 \int_{-T/4}^{T/4} \cos(2\pi ft) dt \\
 &= \frac{2}{2\pi f} \sin(2\pi ft) \Big|_{-T/4}^{T/4} \\
 &= \frac{4}{2\pi f} \sin(2\pi fT/4)
 \end{aligned}$$

Hasil di atas merupakan transformasi Fourier untuk satu fungsi kotak. Untuk mendapatkan fungsi kotak yang periodik, hasil tersebut harus dikalikan dengan transformasi fourier fungsi impulse yang besarnya :

$$\begin{aligned}
 X(f) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f-n/T) \\
 H(f)X(f) &= \sum_{n=-\infty}^{\infty} \frac{\sin(2\pi fT/4)}{(2\pi fT/4)} \delta(f-n/T)
 \end{aligned}$$

Dengan memakai salah satu sifat fungsi impulse yaitu:

$$f(t) \delta(t-t_0) = f(t_0)$$

maka persamaan di atas dapat disederhanakan menjadi:

$$H(f)X(f) = \sum_{n=-\infty}^{\infty} \frac{\sin(2\pi n/4)}{(2\pi n/4)}$$

Persamaan tersebut merupakan transformasi fourier dari fungsi kotak periodik yang masih mengandung sinyal DC. Untuk menghilangkan sinyal DC, persamaan tersebut harus

dikurangi dengan frekuensi DC ($n=0$) dari persamaan tersebut sehingga didapat:

$$Y(f) = \sum_{n=-\infty}^{-1} \frac{\sin(2\pi n/4)}{(2\pi n/4)} + \sum_{n=1}^{\infty} \frac{\sin(2\pi n/4)}{(2\pi n/4)} \quad (5-2)$$

Dari rumus tersebut dapat dihitung transformasi fourier untuk fungsi kotak periodik untuk sample ke n .

Untuk :

$$n=1 \quad Y(f) = 0.63662$$

$$n=2 \quad Y(f) = 0$$

$$n=3 \quad Y(f) = -0.2122$$

$$n=4 \quad Y(f) = 0$$

dan seterusnya.

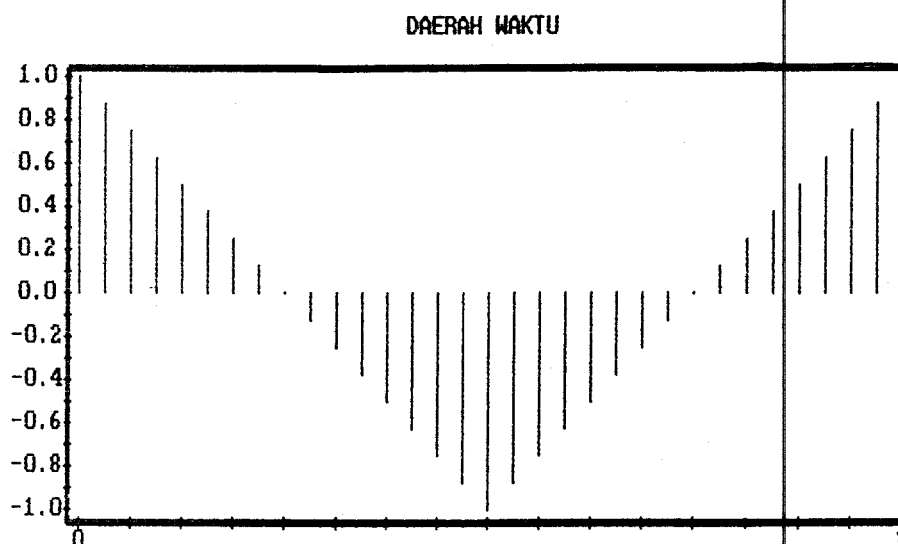
5.1.3. ANALISIS FUNGSI SEGITIGA

Pada gambar 5.4 di bawah ini ditunjukkan fungsi segitiga yang disampling sebanyak 32 data.

Fungsi segitiga tersebut mempunyai rumus sebagai berikut:

$$y(t) = \begin{cases} -4t + 1, & 0 \leq t < T/2 \\ 4t + 1, & T/2 < t < T \end{cases}$$

Transformasi Fourier Kontinyu untuk fungsi segitiga dihitung dengan metode yang sama seperti penghitungan transformasi fourier fungsi kotak yaitu:



GAMBAR 5.4

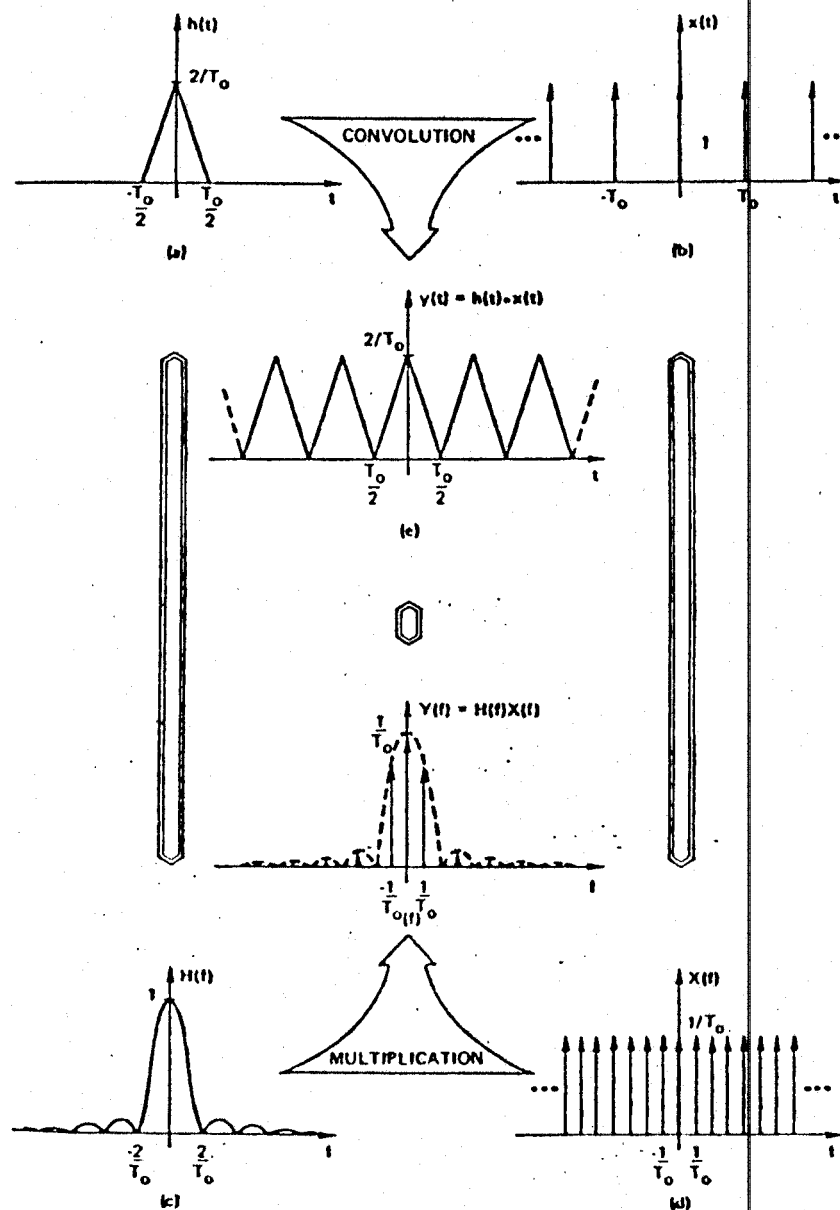
FUNGSI SEGITIGA TERSAMPLING 32 DATA

- Mula-mula dicari dulu transformasi fourier fungsi segitiga pada daerah waktu $0 < t < T$.
- Kalikan hasilnya dengan transformasi fourier fungsi impulse yang periodenya T .

Di sini penghilangan sinyal DC tidak diperlukan karena rumus di atas sudah merupakan fungsi segitiga tanpa sinyal DC.

Pada gambar 5.5 ditunjukkan bagaimana transformasi fourier fungsi segitiga diperoleh.

Transformasi fourier dari fungsi segitiga dicari sebagai berikut:



GAMBAR 5. 5 62)

PENGHITUNGAN TRANSFORMASI FOURIER FUNGSI SEGITIGA PERIODIK
 SECARA GRAFIS

- (a). FUNGSI SEGITIGA TUNGGAL
- (b). FUNGSI IMPULSE DENGAN PERIODE T_0
- (c). HASIL TRANSFORMASI FOURIER FUNGSI SEGITIGA
- (d). HASIL TRANSFORMASI FOURIER FUNGSI IMPULSE
- (e). FUNGSI SEGITIGA PERIODIK
- (f). HASIL TRANSFORMASI FOURIER FUNGSI SEGITIGA PERIODIK

$$\begin{aligned}
 H(f) &= \int_{-\infty}^{+\infty} h(t) e^{-j2\pi ft} dt \\
 &= \int_0^{T/2} (-4t+1) e^{-j2\pi ft} dt + \int_{T/2}^T (4t+1) e^{-j2\pi ft} dt
 \end{aligned}$$

Karena fungsi tersebut merupakan fungsi genap maka dapat disederhanakan menjadi:

$$= \int_0^{T/2} (-4t+1) \cos(2\pi ft) dt + \int_{T/2}^T (4t+1) \cos(2\pi ft) dt$$

Hasil integrasi persamaan di atas adalah:

$$\begin{aligned}
 H(f) &= -4 \left. \frac{t \sin(2\pi ft)}{2\pi f} - 4 \frac{\cos(2\pi ft)}{(2\pi f)^2} \right|_0^{T/2} + \\
 &\quad 4 \left. \frac{t \sin(2\pi ft)}{2\pi f} + 4 \frac{\cos(2\pi ft)}{(2\pi f)^2} \right|_{T/2}^T \\
 &= \frac{2T \sin(2\pi fT)}{\pi f} + \frac{\cos(2\pi fT)}{\pi^2 f^2} + \frac{1}{\pi^2 f^2} - \\
 &\quad \frac{2T \sin(\pi fT)}{\pi f} - \frac{2 \cos(\pi fT)}{\pi^2 f^2}
 \end{aligned}$$

Persamaan di atas merupakan transformasi fourier untuk fungsi segitiga tunggal. Untuk mendapatkan transformasi fourier fungsi segitiga yang periodik, persamaan tersebut harus dikalikan dengan transformasi

fourier fungsi impulse:

$$X(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f-n/T)$$

dan didapat:

$$\begin{aligned} H(f)X(f) &= \left[\frac{2T \sin(2\pi fT)}{\pi f} + \frac{\cos(2\pi fT)}{\pi^2 f^2} + \frac{1}{\pi^2 f^2} - \right. \\ &\quad \left. \frac{2T \sin(\pi fT)}{\pi f} - \frac{2 \cos(\pi fT)}{\pi^2 f^2} \right] \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(f-n/T) \\ &= \frac{2T \sin(2\pi n)}{\pi n} + \frac{T \cos(2\pi n)}{\pi^2 n^2} + \frac{T}{\pi^2 n^2} - \\ &\quad \frac{2T \sin(\pi n)}{\pi n} - \frac{2T \cos(\pi n)}{\pi^2 n^2} \end{aligned}$$

Karena n merupakan bilangan bulat maka suku yang mengandung fungsi sinus selalu berharga 0. Untuk n genap, maka suku yang mengandung fungsi cosinus juga berharga 0, sedangkan untuk n ganjil $\cos(2\pi n)$ selalu berharga 1 dan $\cos(\pi n)$ selalu berharga -1. Dengan memasukkan harga tersebut serta $T=1$ maka persamaan di atas dapat ditulis sebagai berikut:

$$\begin{aligned} Y(f) &= \frac{4}{\pi^2 n^2}, \quad n \text{ ganjil} \\ &= 0, \quad n \text{ genap} \end{aligned} \tag{5-3}$$

Dari rumus tersebut dapat dihitung transformasi fourier untuk fungsi segitiga periodik untuk sample ke n .

Untuk :

$$n=1 \quad Y(f) = 0.40528$$

$$n=2 \quad Y(f) = 0$$

$$n=3 \quad Y(f) = 0.045$$

$$n=4 \quad Y(f) = 0$$

dan seterusnya.

5.2. PENGUKURAN SPEKTRUM FREKUENSI FUNGSI YANG TERSEDIA

Dari hasil analisis pada sub bab 5.1. dapat diperkirakan hasil pengukuran yang dihasilkan oleh alat tugas akhir ini.

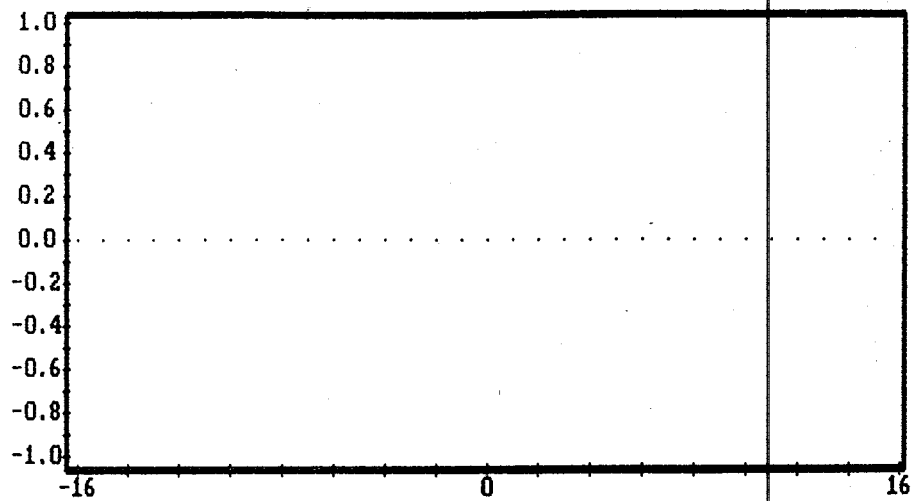
Spektrum frekuensi dibedakan menjadi dua yaitu spektrum frekuensi daerah real dan daerah imajiner. Kedua bagian tersebut ditampilkan pada halaman layar yang berbeda.

5.2.1. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI SINUS

Hasil analisis transformasi Fourier kontinyu fungsi sinus menghasilkan sinyal impulse pada frekuensi imajiner f_0 dan $-f_0$ yang masing-masing besarnya $A/2$ dan $-A/2$.

Pada gambar 5.6(a) berikut ini ditunjukkan hasil pengukuran spektrum frekuensi real fungsi sinus sedangkan gambar 5.6(b) ditunjukkan spektrum frekuensi imajiner.

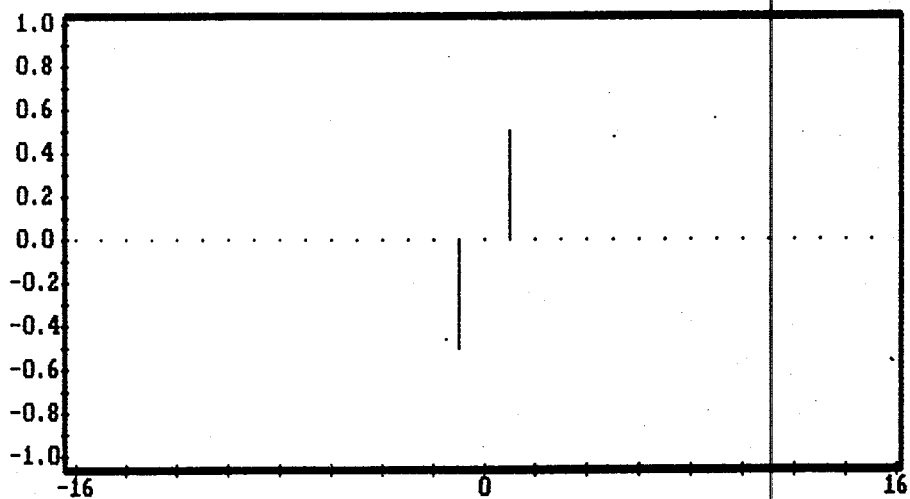
DAERAH FREKUENSI REAL



GAMBAR 5. 6 (a)

HASIL PENGUKURAN FREKUENSI REAL FUNGSI SINUS

DAERAH FREKUENSI IMAJINER



GAMBAR 5. 6 (b)

HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI SINUS

TABEL 5.1

PENGUKURAN TRANSFORMASI FOURIER FUNGSI SINUS 32 DATA

HASIL TRANSFORMASI FOURIER			
SAMPLE KE	AMPLITUDO SINYAL (V)		
	DAERAH WAKTU	FREKUENSI REAL	FREKUENSI IMAJINER
0	0.00000	0.00049	0.00000
1	0.19507	-0.00024	0.49951
2	0.39038	0.00024	0.00000
3	0.55542	0.00000	0.00000
4	0.70703	0.00000	0.00000
5	0.83130	-0.00024	0.00024
6	0.92383	-0.00049	0.00000
7	0.98071	0.00000	0.00000
8	1.00000	-0.00049	0.00000
9	0.98071	0.00000	-0.00024
10	0.92383	-0.00049	0.00000
11	0.83130	0.00000	-0.00049
12	0.70703	0.00000	0.00000
13	0.55542	0.00000	-0.00049
14	0.39038	0.00024	0.00000
15	0.19507	0.00000	-0.00024
16	0.00000	0.00049	0.00000
17	-0.19507	0.00000	0.00024
18	-0.38257	0.00024	0.00000
19	-0.55542	0.00000	0.00024
20	-0.70703	0.00000	0.00000
21	-0.83130	-0.00024	0.00024
22	-0.92383	-0.00049	0.00000
23	-0.98071	0.00000	0.00000
24	-1.00000	-0.00049	0.00000
25	-0.98071	0.00000	-0.00024
26	-0.92383	-0.00049	0.00000
27	-0.83130	0.00000	-0.00049
28	-0.70703	0.00000	0.00000
29	-0.55542	0.00000	-0.00049
30	-0.38257	0.00024	0.00000
31	-0.19507	-0.00024	-0.50000

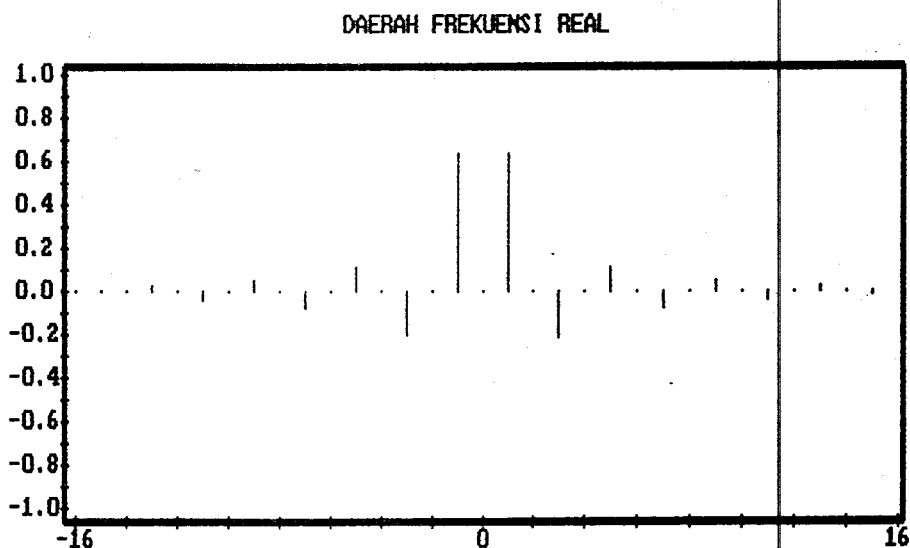
5.2.2. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI KOTAK

Transformasi fourier kontinyu fungsi kotak telah dianalisis pada sub bab 5.1. Didapat bahwa spektrum frekuensinya adalah:

$$Y(f) = \sum_{n=-\infty}^{\infty} \frac{-1 \sin(2\pi n/4)}{(2\pi n/4)} + \sum_{n=1}^{\infty} \frac{\omega \sin(2\pi n/4)}{(2\pi n/4)}$$

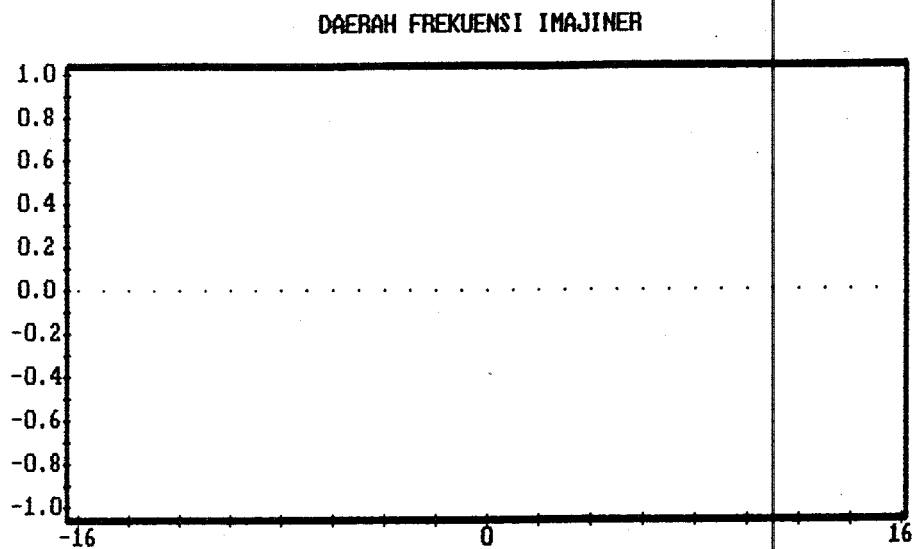
Pada gambar 5.7(a) ditunjukkan hasil pengukuran spektrum frekuensi real fungsi kotak dan gambar 5.7(b) ditunjukkan spektrum frekuensi imajineranya.

Data hasil pengukuran ditunjukkan pada tabel 5.2.



GAMBAR 5.7(a)

HASIL PENGUKURAN FREKUENSI REAL FUNGSI KOTAK



GAMBAR 5. 7 (b)

HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI KOTAK

5. 2. 3. HASIL PENGUKURAN SPEKTRUM FREKUENSI FUNGSI SEGITIGA

Transformasi Fourier Kontinyu fungsi segitiga telah dianalisis pada sub bab 5.1. Didapat bahwa spektrum frekuensinya adalah:

$$Y(f) = \frac{4}{\pi^2 n^2}, \quad n \text{ ganjil}$$

$$= 0, \quad n \text{ genap}$$

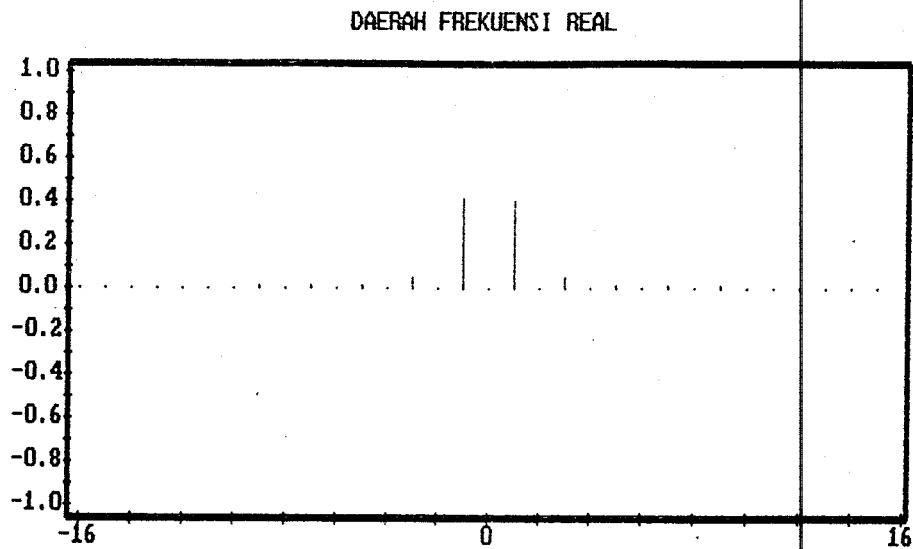
Pada gambar 5.8(a) ditunjukkan hasil pengukuran spektrum frekuensi real fungsi segitiga dan gambar 5.8(b) ditunjukkan spektrum frekuensi imajinernya.

Hasil pengukuran ditunjukkan pada tabel 5.3.

TABEL 5.2

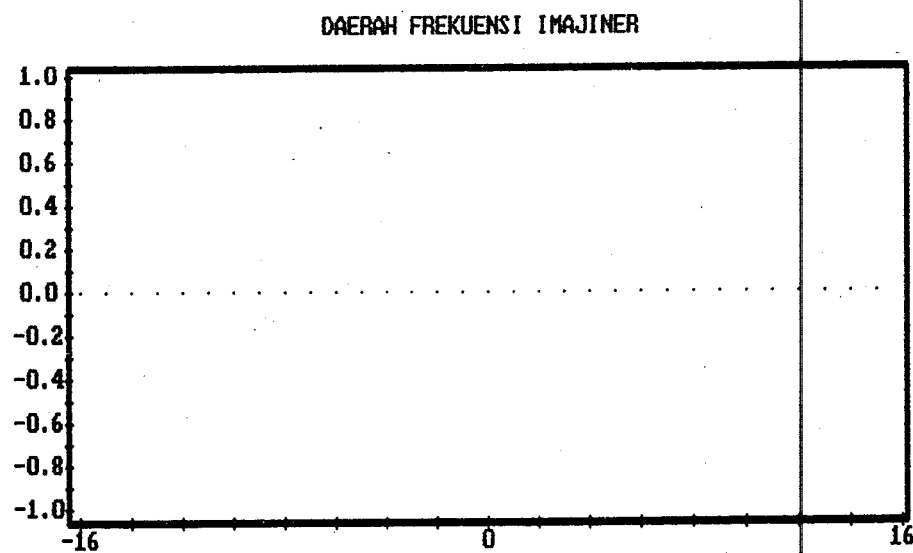
PENGUKURAN TRANSFORMASI FOURIER FUNGSI KOTAK 32 DATA

HASIL TRANSFORMASI FOURIER			
SAMPLE KE	AMPLITUDO SINYAL (V)		
	DAERAH WAKTU	FREKUENSI REAL	FREKUENSI IMAJINER
0	1.00000	0.00000	0.00000
1	1.00000	0.63428	-0.00049
2	1.00000	0.00000	0.00000
3	1.00000	-0.20630	-0.00024
4	1.00000	0.00000	0.00000
5	1.00000	0.11670	-0.00024
6	1.00000	0.00000	0.00000
7	1.00000	-0.07617	-0.00024
8	0.00000	0.00000	0.00000
9	-1.00000	0.05127	-0.00024
10	-1.00000	0.00000	0.00000
11	-1.00000	-0.03345	-0.00024
12	-1.00000	0.00000	0.00000
13	-1.00000	0.01880	0.00000
14	-1.00000	0.00000	0.00000
15	-1.00000	-0.00635	-0.00024
16	-1.00000	0.00000	0.00000
17	-1.00000	-0.00610	0.00000
18	-1.00000	0.00000	0.00000
19	-1.00000	0.01880	0.00000
20	-1.00000	0.00000	0.00000
21	-1.00000	-0.03345	0.00000
22	-1.00000	0.00000	0.00000
23	-1.00000	0.05127	0.00000
24	0.00000	0.00000	0.00000
25	1.00000	-0.07617	0.00000
26	1.00000	0.00000	0.00000
27	1.00000	0.11694	0.00000
28	1.00000	0.00000	0.00000
29	1.00000	-0.20605	0.00000
30	1.00000	0.00000	0.00000
31	1.00000	0.63428	0.00000



GAMBAR 5.8 (a)

HASIL PENGUKURAN FREKUENSI REAL FUNGSI SEGITIGA



GAMBAR 5.8 (b)

HASIL PENGUKURAN FREKUENSI IMAJINER FUNGSI SEGITIGA

TABEL 5.3

PENGUKURAN TRANSFORMASI FOURIER FUNGSI SEGITIGA 32 DATA

HASIL TRANSFORMASI FOURIER			
SAMPLE KE	AMPLITUDO SINYAL (V)		
	DAERAH WAKTU	FREKUENSI REAL	FREKUENSI IMAJINER
0	1.00000	0.00000	0.00000
1	0.87500	0.40625	-0.00024
2	0.75000	0.00000	0.00000
3	0.62500	0.04614	-0.00024
4	0.50000	0.00000	0.00000
5	0.37500	0.01733	-0.00024
6	0.25000	0.00000	0.00000
7	0.12500	0.00952	-0.00024
8	0.00000	0.00000	0.00000
9	-0.12500	0.00635	0.00000
10	-0.25000	0.00000	0.00000
11	-0.37500	0.00488	0.00000
12	-0.50000	0.00000	0.00000
13	-0.62500	0.00415	-0.00024
14	-0.75000	0.00000	0.00000
15	-0.87500	0.00391	-0.00024
16	-1.00000	0.00000	0.00000
17	-0.87500	0.00391	0.00000
18	-0.75000	0.00000	0.00000
19	-0.62500	0.00415	0.00000
20	-0.50000	0.00000	0.00000
21	-0.37500	0.00488	0.00000
22	-0.25000	0.00000	0.00000
23	-0.12500	0.00635	0.00000
24	0.00000	0.00000	0.00000
25	0.12500	0.00977	0.00000
26	0.25000	0.00000	0.00000
27	0.37500	0.01758	0.00000
28	0.50000	0.00000	0.00000
29	0.62500	0.04639	0.00000
30	0.75000	0.00000	0.00000
31	0.87500	0.40527	0.00000

BAB VI

KESIMPULAN

Dari hasil pembahasan dan pengukuran dapat diambil beberapa kesimpulan yaitu:

1. Pengolahan sinyal digital yang memerlukan banyak operasi perkalian seperti halnya Fast Fourier Transform memerlukan prosesor khusus yang memiliki hardware multiplier seperti halnya TMS32010. TMS32010 mengeksekusi operasi perkalian dengan satu clock saja sehingga waktu pengolahan data dapat direduksi menjadi jauh lebih singkat.
2. Ada beberapa hal yang membatasi kecepatan pengolahan sinyal pada alat ini yaitu kecepatan memory yang dipakai, kecepatan konversi Analog to Digital Converter serta kecepatan penampilan display pada IBM PC/XT. Faktor-faktor di atas menyebabkan pemakaian TMS32010 sebagai pengolah data tidak optimum.
3. Dalam pengoperasian peralatan tugas akhir ini diperlukan suatu komunikasi untuk mengatur prosesor mana yang bekerja. Hal ini diperlukan untuk menjaga agar kedua prosesor tersebut tidak mengalami konflik pada saat pengambilan data RAM bersama.
4. Spektrum frekuensi yang diperoleh dari data sinyal

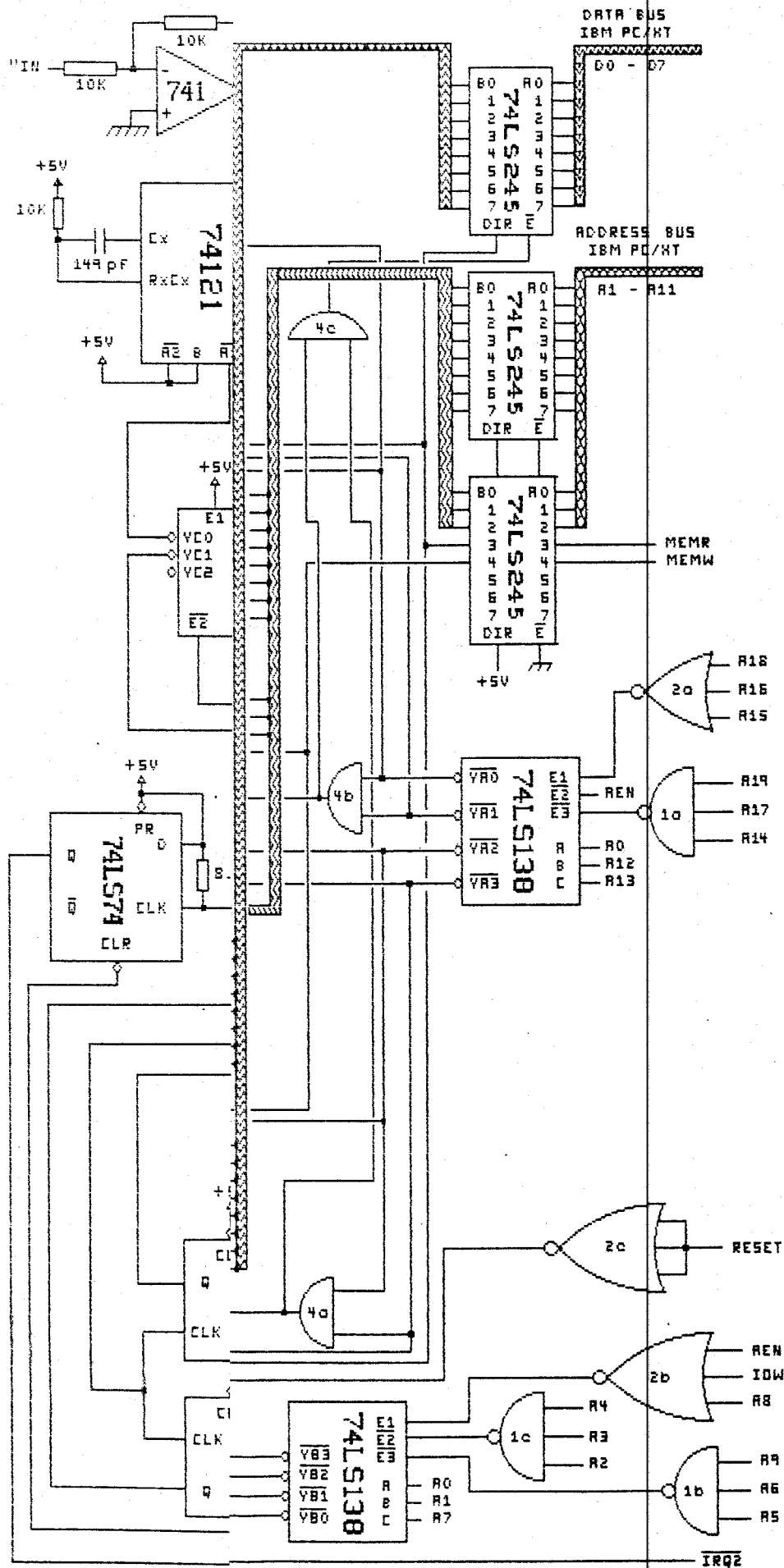
input sangat ditentukan oleh lama pemotongan sinyal. Oleh karena itu untuk mendapatkan spektrum frekuensi sinyal dengan baik, lama pemotongan sinyal diatur agar sebanding dengan periode sinyal input.

DAFTAR PUSTAKA

1. Alan V. Oppenheim & Schafer R.W., *Digital Signal Processing*, Prentice-Hall Inc., 1975.
2. E. Olan Brigham, *The Fast Fourier Transform*, Prentice-Hall Inc., 1974.
3. Erwin Kreyzig, *Advanced Engineering Mathematics*, Third Ed., John Wiley and Sons Inc., 1972.
4. Jogiyanto H.M., *Turbo Pascal versi 5.0*, Andi Offset, 1989.
5. Lawrence R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, 1976.
6. Lewis C. Eggebrecht, *Interfacing to The IBM Personal Computer*, Howard W. Sams & Co., 1987.
7. Louis Leithold, *The Calculus with Analytic Geometry*, Fourth Ed., Harper & Row, 1981.
8. Murray R. Spiegel, *Fourier Analysis*, Schaum's series McGraw-Hill, 1974.
9. Robert F. Coughlin & Driscoll F.F., *Penguat Operasional dan Rangkaian Terpadu Linier*, Erlangga, 1982.
10. Roman Kuc, *Introduction to Digital Signal Processing*, McGraw-Hill, 1988.
11. Ronald N. Bracewell, *The Fourier Transform and Its Applications*, Second Ed., McGraw-Hill, 1986.
12. Steven Holzner, *Advanced Assembly Language on the IBM*

PC, Prentice-Hall.

13. -----, *CMOS Data Book*, National, 1982.
14. -----, *First-Generation TMS320 User's Guide*, Texas Instrument, 1988.
15. -----, *IBM Personal Computer XT System*, International Business Machine.
16. -----, *Logic Data Book*, National, 1982.
17. -----, *Turbo Pascal version 4.0 Reference Guide*, Borland.



SLOT IBM PC/XT

TMS320 FIRST-GENERATION DIGITAL SIGNAL PROCESSORS

JANUARY 1987 - REVISED MARCH 1988

- 160-ns Instruction Cycle
- 144/256-Word On-Chip Data RAM
- 1.5K/4K-Word On-Chip Program ROM
- 4K-Word On-chip Program EPROM (TMS320E15/E17)
- EPROM Code Protection for Copyright Security
- 4K-Word Total External Memory at Full Speed
- 32-Bit ALU/Accumulator
- 16 x 16-Bit Multiplier with a 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- Dual-Channel Serial Port (TMS32011/C17/E17)
- 16-Bit Bidirectional Data Bus with 50-Mbps Transfer Rate
- Single 5-V Supply
- Packaging: 40-Pin DIP and 44-Pin PLCC
- Commercial and Military Versions Available
- NMOS Technology:
 - TMS32010 200-ns cycle time
 - TMS32010-14 280-ns cycle time
- CMOS Technology:
 - TMS320C10-25 160-ns cycle time
 - TMS320C10 200-ns cycle time
 - TMS320C10-14 280-ns cycle time
 - TMS320C15-25 160-ns cycle time
 - TMS320C15 200-ns cycle time
 - TMS320E15 (EPROM) . 200-ns cycle time
 - TMS320C17-25 160-ns cycle time
 - TMS320C17 200-ns cycle time
 - TMS320E17 (EPROM) . 200-ns cycle time

TMS32010, TMS320C10
PIN PACKAGE
(TOP VIEW)

A1/PA1	1	40	A2/PA2
AO/PA0	2	39	A3
MC/MP	3	38	A4
\overline{RS}	4	37	A5
\overline{INT}	5	36	A6
CLKOUT	6	35	A7
X1	7	34	A8
X2/CLKIN	8	33	MEN
\overline{BIO}	9	32	\overline{DEN}
VSS	10	31	\overline{WE}
D8	11	30	VCC
D9	12	29	A9
D10	13	28	A10
D11	14	27	A11
D12	15	26	D0
D13	16	25	D1
D14	17	24	D2
D15	18	23	D3
D7	19	22	D4
D6	20	21	D5

This data sheet provides complete design documentation for all the first-generation devices of the TMS320 family. This facilitates the selection of the devices best suited for user applications by providing all specifications and special features for each TMS320 member. This data sheet is divided into four major sections: architecture, electrical specifications (NMOS and CMOS), timing diagrams, and mechanical data. In each of these sections, generic information is presented first, followed by specific device information. An index is provided for quick reference to specific information about a device.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.


**TEXAS
INSTRUMENTS**

POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

Copyright © 1987, Texas Instruments Incorporated

TMS320 FIRST-GENERATION DEVICES

description

The TMS320 family of 16/32-bit single-chip digital signal processors combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly paralleled architecture and efficient instruction set provide speed and flexibility to produce a MOS microprocessor family capable of executing 6.4 MIPS (million instructions per second). The TMS320 family optimizes speed by implementing functions in hardware that other processors implement through microcode or software. This hardware-intensive approach provides the design engineer with processing power previously unavailable on a single chip.

The TMS320 family consists of two generations of digital signal processors. The first generation contains the TMS32010 and its spinoffs, as described in this data sheet. The TMS32020 and TMS320C25 are the second-generation processors, designed for higher performance. Many features are common among the TMS320 processors. Specific features are added in each processor to provide different cost/performance tradeoffs. Software compatibility is maintained throughout the family to protect the user's investment in architecture. Each processor has software and hardware tools to facilitate rapid design.

introduction

The TMS32010, the first NMOS digital signal processor in the TMS320 family, was introduced in 1983. Its powerful instruction set, inherent flexibility, high-speed number-crunching capabilities, and innovative architecture have made this high-performance, cost-effective processor the ideal solution to many telecommunications, computer, commercial, industrial, and military applications. Since that time, the TMS320C10, a low-power CMOS version of the industry-standard TMS32010, and other spinoff devices have been added to the first generation of the TMS320 family.

The TMS32010 microprocessor is available in two speed versions: TMS32010 (20 MHz) and TMS32010-14 (14 MHz). These devices are capable of executing a 16 x 16-bit multiply with a 32-bit result in a single instruction cycle. On-chip data RAM of 144 words and on-chip program ROM of 1.5K words are available. Full-speed execution of 4K words of off-chip program memory is also possible. The TMS32010-14 provides a low-cost alternative for DSP applications not requiring the maximum operating frequency of the TMS32010. The device provides a direct EPROM interface for cost-effective system development and modification. Both of these devices are pin-for-pin and object-code compatible with the TMS32010 and its development tools.

The TMS320C10 is object-code and pin-for-pin compatible with the TMS32010. It is processed in CMOS technology, achieving a power dissipation less than one-sixth that of the NMOS device. The lower power dissipation makes the TMS320C10 ideal for power-sensitive applications such as digital telephony and portable products. The TMS320C10-25, a 25-MHz version of the TMS320C10, has a 160-ns instruction cycle time and is well suited for high-performance DSP applications. The TMS320C10 is also available in a 280-ns version, the TMS320C10-14. This device provides a low-cost alternative for DSP applications not requiring the maximum operating frequency of the TMS320C10.

The TMS320C15 and TMS320E15 CMOS devices are object-code and pin-for-pin compatible with the TMS32010 and offer expanded on-chip RAM of 256 words and on-chip program ROM or EPROM of 4K words. These devices allow the capability of upgrading performance and reducing power, board space, and system cost without hardware redesign. The TMS320C15 is also available in a 160-ns version, the TMS320C15-25.

TMS320 FIRST-GENERATION DEVICES

introduction (continued)

The TMS320C17 and TMS320E17 also offer expanded on-chip RAM of 256 words and on-chip program ROM or EPROM of 4K words. These devices provide a dual-channel serial interface, on-chip μ -law/A-law companding hardware, and a serial port timer. In addition, a 16-bit coprocessor interface provides a direct communication channel to common 4/8-bit microcomputers (no glue logic required), and minimal logic interface to most common 16/32-bit microprocessors. The devices are object-code compatible with the TMS32010 and processed in CMOS technology. The TMS320C17 is also available in a 160-ns version, the TMS320C17-25.

Table 1 provides an overview of the first generation of TMS320 processors with comparisons of memory, I/O, cycle timing, power, package type, technology, and military support. For specific availability, contact the nearest TI sales office.

TABLE 1. TMS320 FIRST-GENERATION DEVICE OVERVIEW

DEVICE	MEMORY				I/O [†]			CYCLE TIME (ns)	TYP POWER (mW)	PACKAGE TYPE		
	ON-CHIP		EPROM	OFF-CHIP EXPANSION	SER	PAR	CPX			DIP	PLCC	
	RAM	ROM										
TMS32010 [‡]	(NMOS)	144	1.5K	—	4K	—	8 x 16	—	200	900	40	—
TMS32010-14	(NMOS)	144	1.5K	—	4K	—	8 x 16	—	280	900	40	—
TMS320C10-25	(CMOS)	144	1.5K	—	4K	—	8 x 16	—	160	200	40	44
TMS320C10 [§]	(CMOS)	144	1.5K	—	4K	—	8 x 16	—	200	165	40	44
TMS320C10-14	(CMOS)	144	1.5K	—	4K	—	8 x 16	—	280	140	40	44
TMS320C15-25	(CMOS)	256	4K	—	4K	—	8 x 16	—	160	250	40	44
TMS320C15 [§]	(CMOS)	256	4K	—	4K	—	8 x 16	—	200	225	40	44
TMS320E15 [§]	(CMOS)	256	—	4K	4K	—	8 x 16	—	200	275	40	—
TMS320C17-25	(CMOS)	256	4K	—	—	2	6 x 16	YES	160	275	40	44
TMS320C17	(CMOS)	256	4K	—	—	2	6 x 16	YES	200	250	40	44
TMS320E17	(CMOS)	256	—	4K	—	2	6 x 16	YES	200	275	40	—

[†] SER = serial; PAR = parallel; CPX = coprocessor interface.

[‡] Military version available.

[§] Military version planned; contact nearest TI sales office for availability.

TMS320 FIRST-GENERATION DEVICES

architecture

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

32-bit ALU/accumulator

The TMS320 first-generation devices contain a 32-bit ALU and accumulator for support of double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

16 x 16-bit parallel multiplier

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering.

data and program memory

Since the TMS320 devices use a Harvard architecture, data and program memory reside in two separate spaces. The first-generation devices have 144 or 256 words of on-chip data RAM and 1.5K or 4K words of on-chip program ROM. On-chip program EPROM of 4K words is provided on the TMS320E15/E17. The EPROM cell utilizes standard PROM programmers and is programmed identically to a 64K CMOS EPROM (TMS27C64).

program memory expansion

The first-generation devices are capable of executing up to 4K words of external memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality. The TMS320C17/E17 provides no memory expansion capability.

TMS320 FIRST-GENERATION DEVICES

microcomputer/microprocessor operating modes (TMS32010/C10/C15/E15)

The TMS32010/C10 and TMS320C15/E15 devices offer two modes of operation defined by the state of the MC/MP pin: the microcomputer mode (MC/MP = 1) or the microprocessor mode (MC/MP = 0). In the microcomputer mode, on-chip ROM is mapped into the memory space with up to 4K words of external memory available. In the microprocessor mode, all 4K words of memory are external.

interrupts and subroutines

The TMS320 first-generation devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The interrupts used in these devices are maskable.

input/output

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the three LSBs on the address lines. In addition, a polling input for bit test and jump operations (BIO) and an interrupt pin (INT) have been incorporated for multitasking.

serial port (TMS320C17/E17)

Two of the I/O ports on the TMS320C17/E17 are dedicated to the serial port and companding hardware. I/O port 0 is dedicated to control register 0, which controls the serial port, interrupts, and companding hardware. I/O port 1 accesses control register 1, as well as both serial port channels, and the companding hardware. The six remaining I/O ports are available for external parallel interfaces.

The dual-channel serial port is capable of full-duplex serial communication and offers direct interface to combo-codecs. Receive and transmit registers that operate with 8-bit data samples are I/O-mapped. Either internal or external framing signals for serial data transfers are selected through the system control register. The serial port clock provides the bit timing for transfers with the serial port, and may be either an input or output. A framing pulse signal provides framing pulses for combo-codec circuits, an 8-kHz sample clock for voice-band systems, or a timer for control applications.

companding hardware (TMS320C17/E17)

On-chip hardware enables the TMS320C17/E17 to compand (COMpress/exPAND) data in either μ -law or A-law format. The companding logic operation is configured via the system control register. Data may be companded in either a serial mode for operation on serial port data (converting between linear and logarithmic PCM) or a parallel mode for computation inside the device. The TMS320C17/E17 allows the hardware companding logic to operate with either sign-magnitude or two's-complement numbers.

coprocessor port (TMS320C17/E17)

The coprocessor port on the TMS320C17/E17 provides a direct connection to most 4/8-bit microcomputers and 16/32-bit microprocessors. The port is accessed through I/O port 5 using IN and OUT instructions. The coprocessor interface allows the device to act as a peripheral (slave) microcomputer to a microprocessor, or as a master to a peripheral microcomputer. In the microcomputer mode, the 16 data lines are used for the 6 parallel 16-bit I/O ports. In the coprocessor mode, the 16-bit parallel port is reconfigured to operate as a 16-bit latched bus interface. For peripheral transfer, an 8-bit or 16-bit length of the coprocessor port can be selected.

TEXAS
INSTRUMENTS

POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

TMS320 FIRST-GENERATION DEVICES

instruction set

A comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. All of the first-generation devices are object-code compatible and use the same 60 instructions. The instruction set consists primarily of single-cycle single-word instructions, permitting execution rates of more than six million instructions per second. Only infrequently used branch and I/O instructions are multicycle. Instructions that shift data as part of an arithmetic operation execute in a single cycle and are useful for scaling data in parallel with other operations.

Three main addressing modes are available with the instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the 1-bit data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words, and the second page contains up to 128 words.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented and the ARP changed in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

Immediate instructions derive data from part of the instruction word rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 2 lists the symbols and abbreviations used in Table 3, the instruction set summary. Table 3 contains a short description and the opcode for each TMS320 first-generation instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 2. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TMS320 FIRST-GENERATION
DEVICES

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY

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	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	← S →		1	← D →								
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	0	1	← D →					
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	1	← D →						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	1	← D →						
LAC	Load accumulator with shift	1	1	0	0	1	0	← S →		1	← D →								
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	← K →							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	1	← D →						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	← X →		1	← D →							
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	1	← D →						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	← S →		1	← D →								
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	0	1	0	0	1	← D →					
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	1	← D →						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	1	← D →						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	1	← D →						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	1	← D →						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	1	← D →						
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	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	1	← D →						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	← K →							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	← D →						
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	1	← D →						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	← D →						
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	1	← D →						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	1	← D →						

TEXAS
INSTRUMENTS

POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

**TMS320 FIRST-GENERATION
DEVICES**

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONTINUED)

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	0
B	Branch unconditional	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				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
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
BGTZ	Branch if accumulator > 0	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
BIOZ	Branch on $\overline{BO} = 0$	2	2	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				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
				0	0	0	0	← BRANCH ADDRESS →				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
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
BV	Branch on overflow	2	2	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				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
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
CALL	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALLL	Call subroutine immediate	2	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →				0	0	0	0	0	0	0	0
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	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	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	1	← D →						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	1	← D →						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	1	← D →						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	1	← D →						
MPYK	Multiply T register with immediate operand, store product in P register	1	1	1	0	0	← K →												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TMS320 FIRST-GENERATION
DEVICES

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONCLUDED)

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	0	
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	1	1	← D →						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	1	1	← D →						
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	0	
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	1	1	← D →						
IN	Input data from port	2	1	0	1	0	0	0	← PA →	1	1	1	1	← D →						
OUT	Output data to port	2	1	0	1	0	0	1	← PA →	1	1	1	1	← D →						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	1	1	← D →						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	0	1	1	1	1	← D →						

development support

Texas Instruments offers an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems such as the XDS/22. Table 4 lists the development support products for the first-generation TMS320 devices.

System development begins with the use of the Evaluation Module (EVM) or Emulator (XDS). These hardware tools allow the designer to evaluate the processor's performance, benchmark time-critical code, and determine the feasibility of using a TMS320 device to implement a specific algorithm.

Software and hardware can be developed in parallel by using the macro assembler/linker and simulator for software development and the XDS for hardware development. The assembler/linker translates the system's assembly source program into an object module that can be executed by the simulator, XDS, or EVM. The XDS provides realtime in-circuit emulation and is a powerful tool for debugging and integrating software and hardware modules.

Additional support for the TMS320 products consists of extensive documentation and three-day DSP design workshops offered by the TI Regional Technology Centers (RTCs). The workshops provide hands-on experience with the TMS320 development tools. Refer to the *TMS320 Family Development Support Reference Guide* for further information about TMS320 development support products and DSP workshops. When technical questions arise regarding the TMS320, contact the Texas Instruments TMS320 DSP Hotline, (713) 274-2320.

TMS320 FIRST-GENERATION
NMOS DEVICES

NMOS DEVICE ELECTRICAL SPECIFICATIONS

This section contains all the electrical specifications for the TMS320 NMOS first-generation devices. Refer to the top corner for the specific device.

absolute maximum ratings over specified temperature range (unless otherwise noted)[†]

Supply voltage range, V_{CC}^{\ddagger}	-0.3 V to 7 V
Input voltage range	-0.3 V to 15 V
Output voltage range	-0.3 V to 15 V
Continuous power dissipation	1.5 W
Air temperature range above operating device	0°C to 70°C
Storage temperature range	-55°C to +150°C

[†]Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only, and functional operation of the device or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

[‡]All voltage values are with respect to V_{SS} .

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.75	5	5.25	V
V_{SS}	Supply voltage		0		V
V_{IH}	High-level input voltage	All inputs except CLKIN		2	V
		CLKIN		2.3	
V_{IL}	Low-level input voltage (all inputs)			0.8	V
I_{OH}	High-level output current (all outputs)			-300	μ A
I_{OL}	Low-level output current (all outputs)			2	mA
T_A	Operating free-air temperature		0	70	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP [†]	MAX	UNIT	
V_{OH}	High-level output voltage	$I_{OH} = \text{MAX}$		2.4	3		V	
V_{OL}	Low-level output voltage	$I_{OL} = \text{MAX}$			0.3	0.5	V	
I_{OZ}	Off-state output current	$V_{CC} = \text{MAX}$	$V_O = 2.4 \text{ V}$		20		μ A	
			$V_O = 0.4 \text{ V}$		-20			
I_I	Input current	$V_I = V_{SS} \text{ to } V_{CC}$	All inputs except CLKIN		± 20		μ A	
			CLKIN		± 50			
I_{CC}^{\ddagger}	Supply current	$V_{CC} = \text{MAX}$	$T_A = 0^\circ\text{C}$		180	275	mA	
			$T_A = 70^\circ\text{C}$			235 [§]		
C_i	Input capacitance	Data bus	$f = 1 \text{ MHz, All other pins } 0 \text{ V}$		25 [§]		pF	
		All others			15 [§]			
C_o	Output capacitance	Data bus				25 [§]		pF
		All others				10 [§]		

[†]All typical values except for I_{CC} are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

[‡] I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.

[§]Value derived from characterization data and not tested.



POST OFFICE BOX 1443 • HOUSTON TEXAS 77001

TMS320 FIRST-GENERATION
NMOS DEVICES

PARAMETER MEASUREMENT INFORMATION

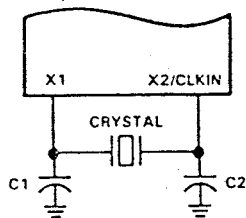


FIGURE 1. INTERNAL CLOCK OPTION

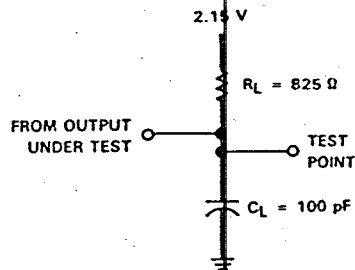


FIGURE 2. TEST LOAD CIRCUIT

input synchronization requirements

For systems using asynchronous inputs to the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ pins on the TMS32010, the external hardware shown in the diagrams below is recommended to ensure proper execution of interrupts and the BIOZ instruction. This hardware synchronizes the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ input signals with the rising edge of CLKOUT on the TMS32010. The pulse width required for these input signals is $t_{c(C)}$, which is one TMS32010 clock cycle, plus sufficient setup time for the flip-flop (dependent upon the flip-flop used). Note that these input synchronization requirements apply only to NMOS versions of the TMS32010 and not to other members of the TMS320 family.

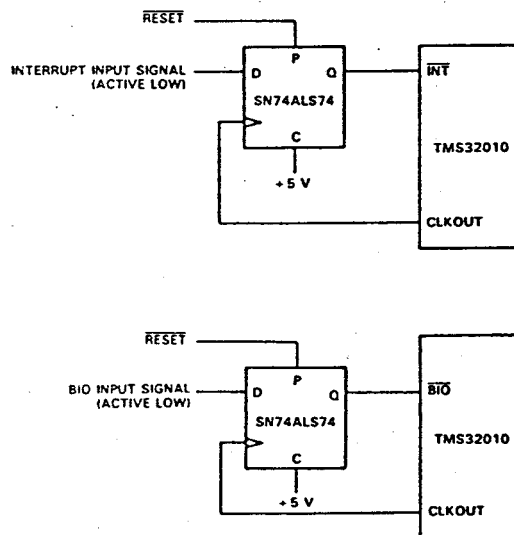


FIGURE 3. ASYNCHRONOUS INPUT SYNCHRONIZATION CIRCUITS

CLOCK CHARACTERISTICS AND TIMING

The TMS32010 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	TMS32010			TMS32010-14			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Crystal frequency f_x	0°C to 70°C	6.7		20.5	6.7		14.4	MHz
C1, C2	0°C to 70°C		10			10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	TMS32010			TMS32010-14			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(CLK)}$ CLKOUT cycle time [†]	$R_L = 825 \Omega$ $C_L = 100 \text{ pF}$ See Figure 2	195.12	200		277.78			ns
$t_r(CLK)$ CLKOUT rise time			10			10		ns
$t_f(CLK)$ CLKOUT fall time				8			8	ns
$t_w(CL)$ Pulse duration, CLKOUT low				92			131	ns
$t_w(CH)$ Pulse duration, CLKOUT high				90			129	ns
$t_d(MCC)$ Delay time CLKINT to CLKOUT [‡]			25 [‡]		60 [‡]	25 [‡]		60 [‡]

[†] $t_{c(CLK)}$ is the cycle time of CLKOUT, i.e., $4 \cdot t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).

[‡] Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS32010			TMS32010-14			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_c(MC)$ Master clock cycle time		46.78	50	150	69.5		150	ns
$t_r(MC)$ Rise time master clock input			5 [†]	10 [†]		5 [†]	10 [†]	ns
$t_f(MC)$ Fall time master clock input			5 [†]	10 [†]		5 [†]	10 [†]	ns
$t_w(MCP)$ Pulse duration master clock		0.475 $t_{c(MC)}$ [†]		0.525 $t_{c(MC)}$ [†]	0.475 $t_{c(MC)}$ [†]		0.525 $t_{c(MC)}$ [†]	ns
$t_w(MCL)$ Pulse duration master clock low, $t_c(MC) = 50 \text{ ns}$			20 [†]			20 [†]		ns
$t_w(MCH)$ Pulse duration master clock high, $t_c(MC) = 50 \text{ ns}$			20 [†]			20 [†]		ns

[†] Values derived from characterization data and not tested.

TMS32010
TMS32010-14

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	TMS32010			TMS32010-14			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t_{d1} Delay time CLKOUT ₁ to address bus valid	$R_L = 825 \Omega$, $C_L = 100 \text{ pF}$, See Figure 2	10 [†]		50	10 [†]		50	ns
t_{d2} Delay time CLKOUT ₁ to $\overline{\text{MEN}}_1$		$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} + 15$	$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} + 15$	ns
t_{d3} Delay time CLKOUT ₁ to $\overline{\text{MEN}}_1$		-10 [†]		15	-10 [†]		15	ns
t_{d4} Delay time CLKOUT ₁ to $\overline{\text{DEN}}_1$		$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} + 15$	$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} + 15$	ns
t_{d5} Delay time CLKOUT ₁ to $\overline{\text{DEN}}_1$		-10 [†]		15	-10 [†]		15	ns
t_{d6} Delay time CLKOUT ₁ to $\overline{\text{WE}}_1$		$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} - 15$	$\frac{1}{2} t_{c(C)} - 5^{\dagger}$		$\frac{1}{2} t_{c(C)} + 15$	ns
t_{d7} Delay time CLKOUT ₁ to $\overline{\text{WE}}_1$		-10 [†]		15	-10 [†]		15	ns
t_{d8} Delay time CLKOUT ₁ to data bus OUT valid				$\frac{1}{2} t_{c(C)} - 65$			$\frac{1}{2} t_{c(C)} - 65$	ns
t_{d9} Time after CLKOUT ₁ that data bus starts to be driven				$\frac{1}{2} t_{c(C)} - 5^{\dagger}$			$\frac{1}{2} t_{c(C)} - 5^{\dagger}$	ns
t_{d10} Time after CLKOUT ₁ that data bus stops being driven				$\frac{1}{2} t_{c(C)} + 30^{\dagger}$			$\frac{1}{2} t_{c(C)} + 30^{\dagger}$	ns
t_v Data bus OUT valid after CLKOUT ₁				$\frac{1}{2} t_{c(C)} - 10$			$\frac{1}{2} t_{c(C)} - 10$	ns
$t_{h(A-WMD)}$ Address hold time after $\overline{\text{WE}}_1$, $\overline{\text{MEN}}_1$ or $\overline{\text{DEN}}_1$ (see Note 1)				0			0	ns
$t_{\text{setA-MD}}$ Address bus setup time prior to $\overline{\text{MEN}}_1$ or $\overline{\text{DEN}}_1$				$\frac{1}{2} t_{c(C)} - 45$			$\frac{1}{2} t_{c(C)} - 45$	ns

[†] Values derived from characterization data and not tested.
 NOTE 1: Address bus will be valid upon $\overline{\text{WE}}_1$, $\overline{\text{DEN}}_1$, or $\overline{\text{MEN}}_1$.

timing requirements over recommended operating conditions

PARAMETER	TEST CONDITIONS	TMS32010			TMS32010-14			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
t_{setD} Setup time data bus valid prior to CLKOUT ₁	$R_L = 825 \Omega$	50			50			ns
$t_{\text{h(D)}}$ Hold time data bus held valid after CLKOUT ₁ (see Note 2)	$C_L = 100 \text{ pF}$, See Figure 2	0			0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{MEN}}_1$ or $\overline{\text{DEN}}_1$ preceding CLKOUT₁.

RESET (\overline{RS}) TIMING

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11} Delay time $\overline{DEN1}$, $\overline{WE1}$, and $\overline{MEN1}$ from \overline{RS}	$R_L = 825 \Omega$, $C_L = 100 \text{ pF}$, See Figure 2			$4t_{c(C)} + 50^1$	ns
$t_{dis(R)}$ Data bus disable time after \overline{RS}				$4t_{c(C)} + 50^1$	ns

¹Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

	TMS32010			TMS32010-14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(R)}$ Reset \overline{RS} setup time prior to CLKOUT (see Note 3)	50			50			ns
$t_w(R)$ \overline{RS} pulse duration	$5t_{c(C)}$			$5t_{c(C)}$			ns

NOTE 3: \overline{RS} can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT (\overline{INT}) TIMING

timing requirements over recommended operating conditions

	TMS32010			TMS32010-14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_f(\overline{INT})$ Fall time \overline{INT}			15			15	ns
$t_w(\overline{INT})$ Pulse duration \overline{INT}	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su}(\overline{INT})$ Setup time \overline{INT} before CLKOUT ¹	50			50			ns

I/O (\overline{BIO}) TIMING

timing requirements over recommended operating conditions

	TMS32010			TMS32010-14			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_f(\overline{BIO})$ Fall time \overline{BIO}			15			15	ns
$t_w(\overline{BIO})$ Pulse duration \overline{BIO}	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su}(\overline{BIO})$ Setup time \overline{BIO} before CLKOUT ¹	50			50			ns

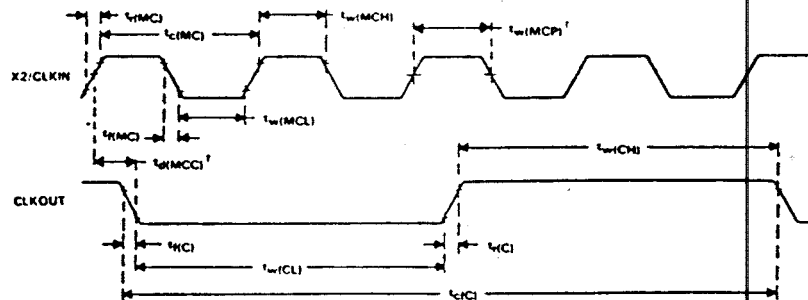
TMS320 FIRST-GENERATION DEVICES

TIMING DIAGRAMS

This section contains all the timing diagrams for the TMS320 first-generation devices. Refer to the top corner for the specific device.

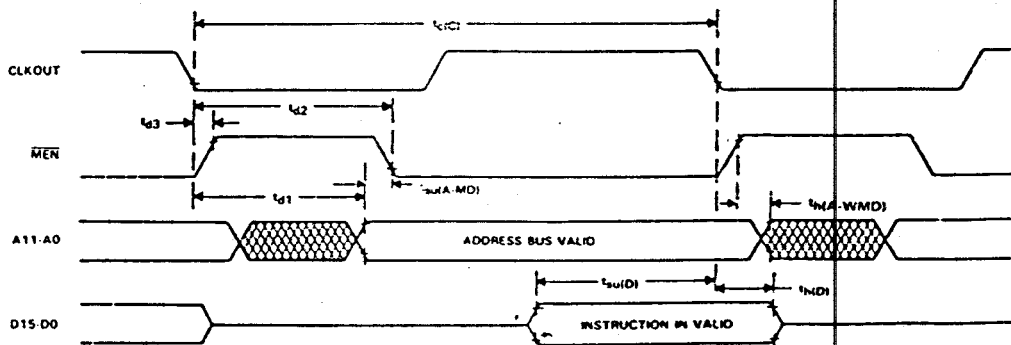
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing



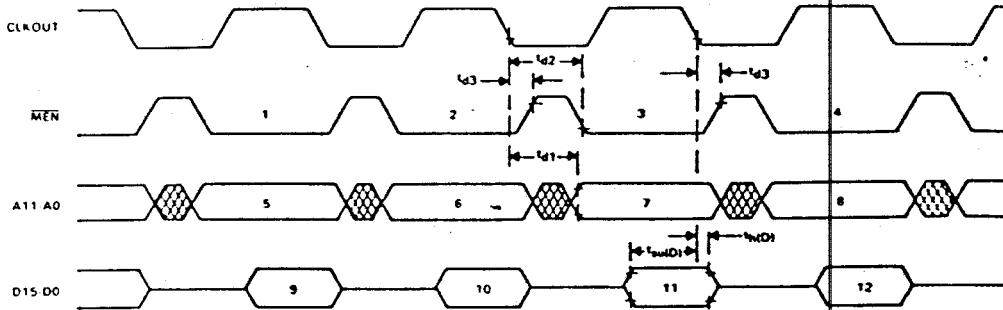
$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

memory read timing



**TMS320 FIRST-GENERATION
DEVICES**

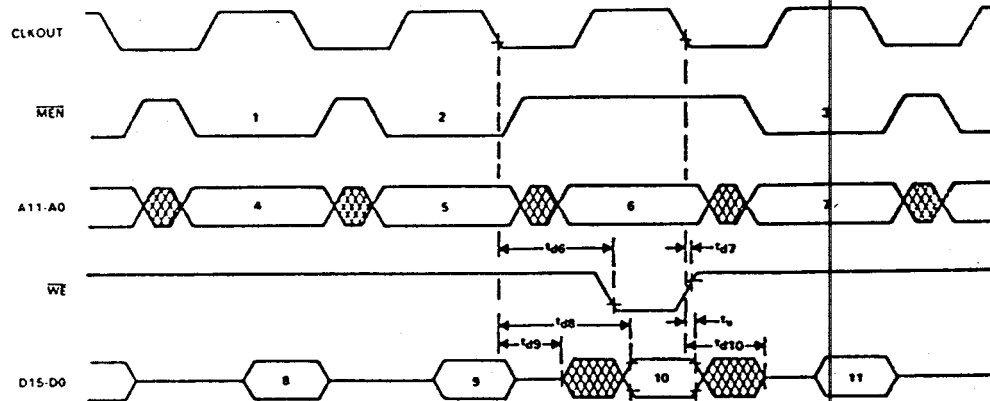
TBLR instruction timing



LEGEND:

- | | |
|------------------------------|--------------------------|
| 1. TBLR INSTRUCTION PREFETCH | 7. ADDRESS BUS VALID |
| 2. DUMMY PREFETCH | 8. ADDRESS BUS VALID |
| 3. DATA FETCH | 9. INSTRUCTION IN VALID |
| 4. NEXT INSTRUCTION PREFETCH | 10. INSTRUCTION IN VALID |
| 5. ADDRESS BUS VALID | 11. DATA IN VALID |
| 6. ADDRESS BUS VALID | 12. INSTRUCTION IN VALID |

TBLW instruction timing



LEGEND:

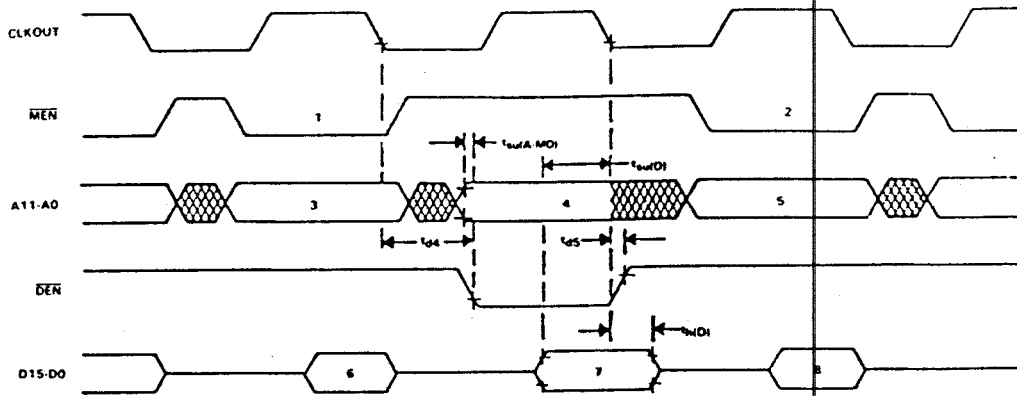
- | | |
|------------------------------|--------------------------|
| 1. TBLW INSTRUCTION PREFETCH | 7. ADDRESS BUS VALID |
| 2. DUMMY PREFETCH | 8. INSTRUCTION IN VALID |
| 3. NEXT INSTRUCTION PREFETCH | 9. INSTRUCTION IN VALID |
| 4. ADDRESS BUS VALID | 10. DATA OUT VALID |
| 5. ADDRESS BUS VALID | 11. INSTRUCTION IN VALID |
| 6. ADDRESS BUS VALID | |

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

TMS320 FIRST-GENERATION DEVICES

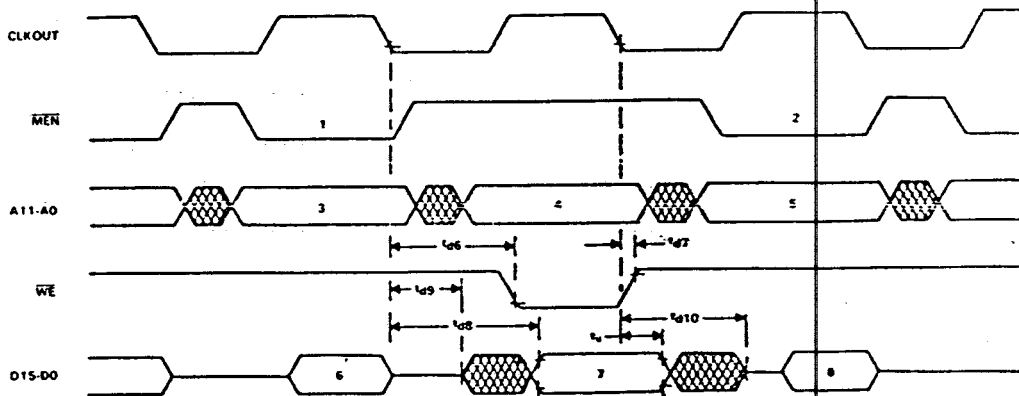
IN instruction timing



LEGEND:

- | | |
|------------------------------|-------------------------|
| 1. IN INSTRUCTION PREFETCH | 5. ADDRESS BUS VALID |
| 2. NEXT INSTRUCTION PREFETCH | 6. INSTRUCTION IN VALID |
| 3. ADDRESS BUS VALID | 7. DATA IN VALID |
| 4. PERIPHERAL ADDRESS VALID | 8. INSTRUCTION IN VALID |

OUT instruction timing

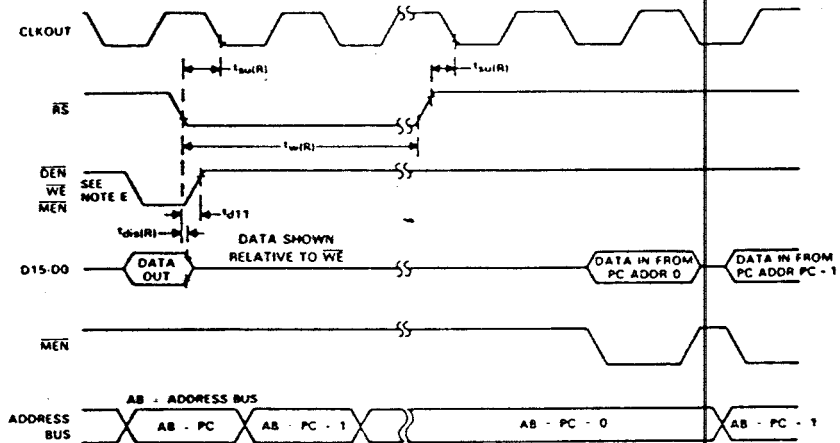


LEGEND:

- | | |
|------------------------------|-------------------------|
| 1. OUT INSTRUCTION PREFETCH | 5. ADDRESS BUS VALID |
| 2. NEXT INSTRUCTION PREFETCH | 6. INSTRUCTION IN VALID |
| 3. ADDRESS BUS VALID | 7. DATA IN VALID |
| 4. PERIPHERAL ADDRESS VALID | 8. INSTRUCTION IN VALID |

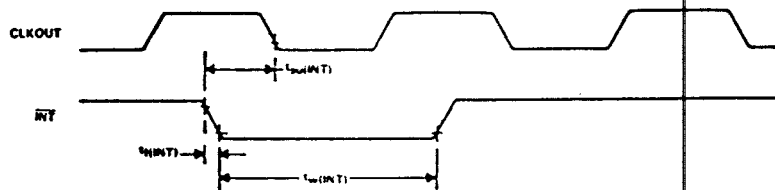
TMS32010, TMS32010-14
TMS320C10, TMS320C10-25, TMS320C10-14
TMS320C15/E15, TMS320C15-25

reset timing

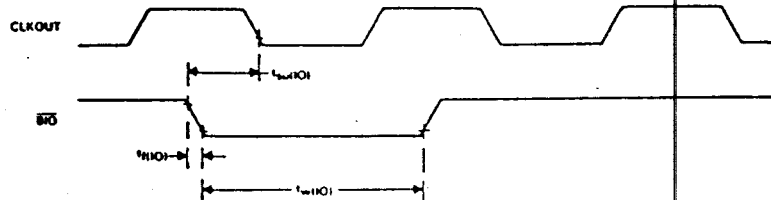


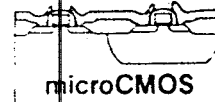
- NOTES: A. \overline{RS} forces \overline{DEN} , \overline{WE} , and \overline{MEN} high and places data bus D0 through D15 in a high-impedance state. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from \overline{RS} .
- B. \overline{RS} must be maintained for a minimum of five clock cycles.
- C. Resumption of normal program will commence after one complete CLK cycle from \overline{RS} .
- D. Due to the synchronizing action on \overline{RS} , time to execute the function can vary dependent upon when \overline{RS} or \overline{RS} occur in the CLK cycle.
- E. Diagram shown is for definition purpose only. \overline{DEN} , \overline{WE} , and \overline{MEN} are mutually exclusive.
- F. During a write cycle, \overline{RS} may produce an invalid write address.

interrupt timing



BIO timing





ADC0820 8-Bit High Speed μ P Compatible A/D Converter with Track/Hold Function

General Description

By using a half-flash conversion technique, the 8-bit ADC0820 CMOS A/D offers a 1.5 μ s conversion time and dissipates only 75 mW of power. The half-flash technique consists of 32 comparators, a most significant 4-bit ADC and a least significant 4-bit ADC.

The input to the ADC0820 is tracked and held by the input sampling circuitry eliminating the need for an external sample-and-hold for signals moving at less than 100 mV/ μ s.

For ease of interface to microprocessors, the ADC0820 has been designed to appear as a memory location or I/O port without the need for external interfacing logic.

Key Specifications

- Resolution 8 Bits
- Conversion Time 2.5 μ s Max (RD Mode)
1.5 μ s Max (WR-RD Mode)
- Input signals with slew rate of 100 mV/ μ s converted without external sample-and-hold to 8 bits
- Low Power 75 mW Max
- Total Unadjusted Error $\pm 1/2$ LSB and ± 1 LSB

Features

- Built-in track-and-hold function
- No missing codes
- No external clocking
- Single supply—5 V_{DD}
- Easy interface to all microprocessors, or operates stand-alone
- Latched TRI-STATE[®] output
- Logic inputs and outputs meet both MOS and T2L voltage level specifications
- Operates ratiometrically or with any reference value equal to or less than V_{DD}
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Overflow output available for cascading
- 0.3" standard width 20-pin DIP

Connection Diagram

Functional Diagram

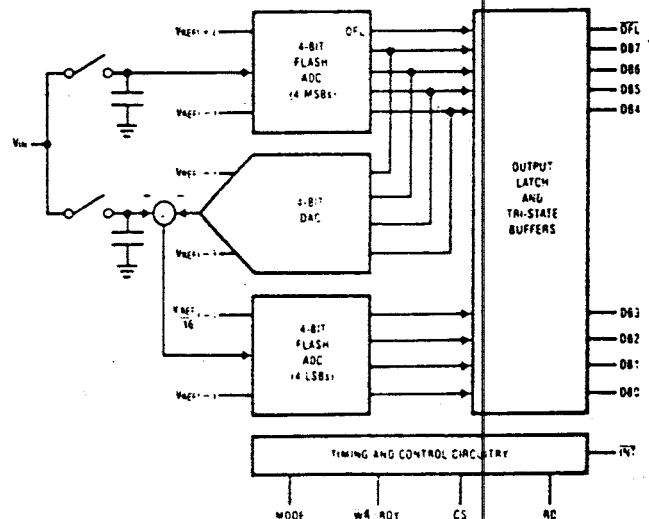
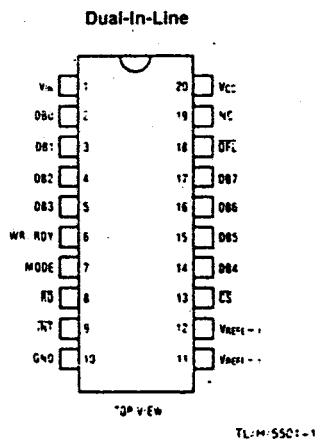


FIGURE 1

ADC0820

Absolute Maximum Ratings (Notes 1 & 2)

Supply Voltage (V_{CC})	10V
Logic Control Inputs	-0.2V to $V_{CC} + 0.2V$
Voltage at Other Inputs and Output	-0.2V to $V_{CC} + 0.2V$
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A = 25^\circ\text{C}$	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 & 2)

Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$ -55°C $\leq T_A \leq$ +125°C
ADC0820BD, ADC0820CD	-40°C $\leq T_A \leq$ +85°C
ADC0820BCD, ADC0820CCD	0°C $\leq T_A \leq$ 70°C
ADC0820BCN, ADC0820CCN	4.5V to 8V
V_{CC} Range	

Converter Characteristics The following specifications apply for RD mode (pin 7 = 0, $V_{CC} = 5V$, $V_{REF(+)} = 5V$, and $V_{REF(-)} = GND$ unless otherwise specified. **Boldface limits apply from T_{MIN} to T_{MAX}** ; all other limits $T_A = T_i = 25^\circ\text{C}$.

Parameter	Conditions	ADC0820BD, ADC0820CD ADC0820BCD, ADC0820CCD			ADC0820BCN, ADC0820CCN			Limit Units
		Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	
Resolution			8			8		Bits
Total Unadjusted Error (Note 3)	ADC0820BD, BCD ADC0820BCN ADC0820CD, CCD ADC0820CCN		$\pm 1/2$ ± 1			$\pm 1/2$ ± 1		LSB LSB LSB LSB
Minimum Reference Resistance		2.3	1.25		2.3	1.4		k Ω
Maximum Reference Resistance		2.3	6		2.3	5.3		k Ω
Maximum $V_{REF(-)}$ Input Voltage			V_{CC}			V_{CC}		V
Minimum $V_{REF(-)}$ Input Voltage			GND			GND		V
Minimum $V_{REF(+)}$ Input Voltage			$V_{REF(-)}$			$V_{REF(-)}$		V
Maximum $V_{REF(-)}$ Input Voltage			$V_{REF(-)}$			$V_{REF(+)}$		V
Maximum V_{IN} Input Voltage			$V_{CC} - 0.1$			$V_{CC} + 0.1$		V
Minimum V_{IN} Input Voltage			GND - 0.1			GND - 0.1		V
Maximum Analog Input Leakage Current	$\overline{CS} = V_{CC}$ $V_{IN} = V_{CC}$ $V_{IN} = GND$		3 -3			0.3 -0.3		μA μA
Power Supply Sensitivity	$V_{CC} = 5V \pm 5\%$	$\pm 1/16$	$\pm 1/4$		$\pm 1/16$	$\pm 1/4$		LSB

Ordering Information

Temperature Range	0°C to +70°C	-40°C to +85°C	-55°C to +125°C
Error	$\pm 1/2$ LSB Unadjusted	ADC0820BCN	ADC0820BCD
	± 1 LSB Unadjusted	ADC0820CCN	ADC0820CCD
Package Outline	N20A—Molded DIP	D20A—Cavity DIP	D20A—Cavity DIP

ADC0820

DC Electrical Characteristics The following specifications apply for $V_{CC} = 5V$, unless otherwise specified. Boldface limits apply from T_{MIN} to T_{MAX} ; all other limits $T_A = T_J = 25^\circ C$.

Parameter	Conditions	ADC0820BD, ADC0820CD ADC0820BCD, ADC0820CCD			ADC0820BCN, ADC0820CCN			Limit Units	
		Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)		
$V_{IN(1)}$, Logical "1" Input Voltage	$V_{CC} = 5.25V$ CS, WR, RD Mode		2.0			2.0	2.0	V	
			3.5			3.5	3.5	V	
$V_{IN(0)}$, Logical "0" Input Voltage	$V_{CC} = 4.75V$ CS, WR, RD Mode		0.8			0.8	0.8	V	
			1.5			1.5	1.5	V	
$I_{IN(1)}$, Logical "1" Input Current	$V_{IN(1)} = 5V$; CS, RD	0.005	1		0.005		1	μA	
	$V_{IN(1)} = 5V$; WR	0.1	3		0.1	0.3	3	μA	
	$V_{IN(1)} = 5V$; Mode	50	200		50	170	200	μA	
$I_{IN(0)}$, Logical "0" Input Current	$V_{IN(0)} = 0V$; CS, RD, WR, Mode	-0.005	-1		-0.005		-1	μA	
$V_{OUT(1)}$, Logical "1" Output Voltage	$V_{CC} = 4.75V$, $I_{OUT} = -360 \mu A$; DB0-DB7, OFL, INT $V_{CC} = 4.75V$, $I_{OUT} = -10 \mu A$; DB0-DB7, OFL, INT		2.4			2.8	2.4	V	
			4.5			4.6	4.5	V	
$V_{OUT(0)}$, Logical "0" Output Voltage	$V_{CC} = 4.75V$, $I_{OUT} = 1.6 mA$; DB0-DB7, OFL, INT, RDY		0.4			0.34	0.4	V	
I_{OUT} , TRI-STATE Output Current	$V_{OUT} = 5V$; DB0-DB7, RDY	0.1	3		0.1	0.3	3	μA	
	$V_{OUT} = 0V$; DB0-DB7, RDY	-0.1	-3		-0.1	-0.3	-3	μA	
I_{SOURCE} , Output Source Current	$V_{OUT} = 0V$; DB0-DB7, OFL, INT		-12	-6		-12	-7.2	-6	mA
			-9	-4.5		-9	-5.3	-4.5	mA
I_{SINK} , Output Sink Current	$V_{OUT} = 5V$; DB0-DB7, OFL, INT, RDY	14	7		14	6.4	7	mA	
I_{CC} , Supply Current	CS = WR = RD = 0	7.5	15		7.5	13	15	mA	

AC Electrical Characteristics The following specifications apply for $V_{CC} = 5V$, $t_r = t_f = 20 ns$, $V_{REF(+)} = 5V$, $V_{REF(-)} = 0V$ and $T_A = 25^\circ C$ unless otherwise specified.

Parameter	Conditions	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Units
t_{CRD} , Conversion Time for RD Mode	Pin 7 = 0, Figure 2	1.6		2.5	μs
t_{ACC0} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = 0, Figure 2	$t_{CRD} + 20$		$t_{CRD} + 50$	ns
t_{CWR-RD} , Conversion Time for WR-RD Mode	Pin 7 = V_{CC} ; $t_{WR} = 600 ns$, $t_{RD} = 600 ns$; Figures 3a and 3b			1.52	μs
t_{WR} , Write Time	Min	Pin 7 = V_{CC} ; Figures 3a and 3b		600	ns
	Max	(Note 4) See Graph	50		μs
t_{RD} , Read Time	Min	Pin 7 = V_{CC} ; Figures 3a and 3b (Note 4) See Graph		600	ns
t_{ACC1} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = V_{CC} , $t_{RD} < t_r$; Figure 3a $C_L = 15 pF$	190		280	ns
	$C_L = 100 pF$	210		320	ns
t_{ACC2} , Access Time (Delay from Falling Edge of RD to Output Valid)	Pin 7 = V_{CC} , $t_{RD} > t_r$; Figure 3b $C_L = 15 pF$	70		120	ns
	$C_L = 100 pF$	90		150	ns

ADC0820

AC Electrical Characteristics (Continued) The following specifications apply for $V_{CC} = 5V$, $t_r = t_f = 20 ns$, $V_{REF(+)} = 6V$, $V_{REF(-)} = 0V$ and $T_A = 25^\circ C$ unless otherwise specified.

Parameter	Conditions	Typ (Note 6)	Tested Limit (Note 7)	Design Limit (Note 8)	Units
t_i , Internal Comparison Time	Pin 7 = V_{CC} ; Figures 3b and 4 $C_L = 50 pF$	800		1300	ns
t_{1H} , t_{0H} , TRI-STATE Control (Delay from Rising Edge of \overline{RD} to Hi-Z State)	$R_L = 1k$, $C_L = 10 pF$	100		200	ns
t_{INTL} , Delay from Rising Edge of \overline{WR} to Falling Edge of \overline{INT}	Pin 7 = V_{CC} , $C_L = 50 pF$ $t_{RD} > t_i$; Figure 3b $t_{RD} < t_i$; Figure 3a	$t_{RD} + 200$		t_i $t_{RD} + 290$	ns ns
t_{INTH} , Delay from Rising Edge of \overline{RD} to Rising Edge of \overline{INT}	Figures 2, 3a and 3b $C_L = 50 pF$	125		225	ns
t_{INTWA} , Delay from Rising Edge of \overline{WR} to Rising Edge of \overline{INT}	Figure 4, $C_L = 50 pF$	175		270	ns
t_{RDV} , Delay from \overline{CS} to \overline{RDY}	Figure 2, $C_L = 50 pF$, Pin 7 = 0	50		100	ns
t_{D} , Delay from \overline{INT} to Output Valid	Figure 4	20		50	ns
t_{a} , Delay from \overline{RD} to \overline{INT}	Pin 7 = V_{CC} , $t_{RD} < t_i$ Figure 3a	200		290	ns
t_{a} , Delay from End of Conversion to Next Conversion	Figures 2, 3a, 3b and 4 (Note 4) See Graph			500	ns
Slew Rate, Tracking		0.1			V/ μs
C_{IN} , Analog Input Capacitance		45			pF
C_{OUT} , Logic Output Capacitance		5			pF
C_{IN} , Logic Input Capacitance		5			pF

Note 1: Absolute Maximum Ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: Total unadjusted error includes offset, full-scale, and linearity errors.

Note 4: Accuracy may degrade if t_{1WA} or t_{RD} is shorter than the minimum value specified. See Accuracy vs t_{1WA} and Accuracy vs t_{RD} graphs.

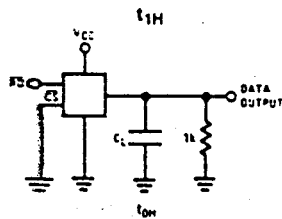
Note 5: The voltage at these pins should never go higher than V_{CC} nor lower than GND.

Note 6: Typicals are at $25^\circ C$ and represent most likely parametric norm.

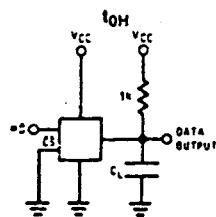
Note 7: Guaranteed and 100% production tested.

Note 8: Guaranteed, but not 100% production tested. These limits are not used to calculate outgoing quality levels.

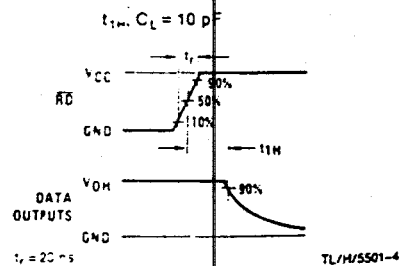
TRI-STATE Test Circuits and Waveforms



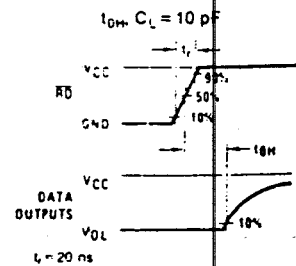
TL/H5501-3



TL/H5501-5



TL/H5501-4



Timing Diagrams

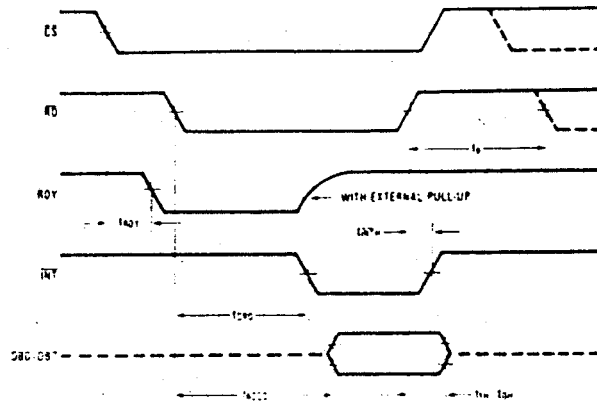


FIGURE 2. RD Mode (Pin 7 is Low)

TL/M/5501-7

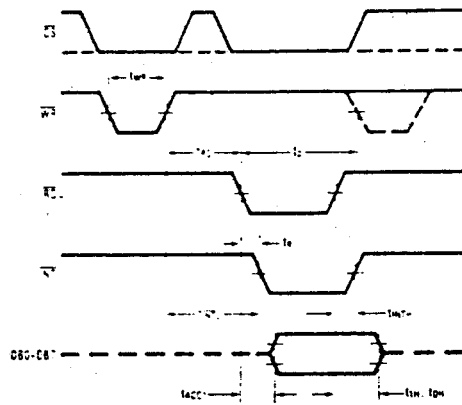


FIGURE 3a. WR-RD Mode (Pin 7 is High and $t_{RD} < t_I$)

TL/M/5501-8

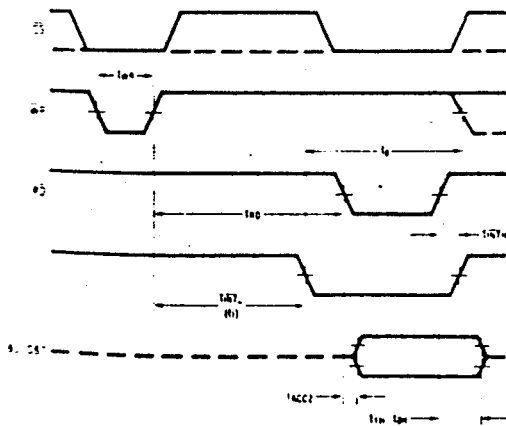


FIGURE 3b. WR-RD Mode (Pin 7 is High and $t_{RD} > t_I$)

TL/M/5501-9

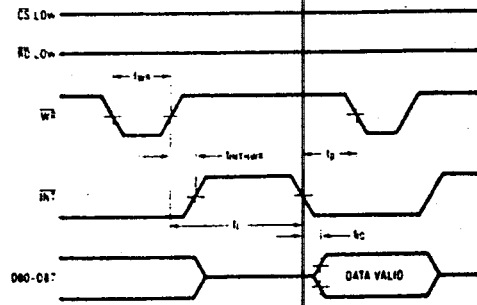
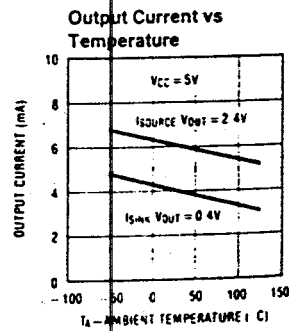
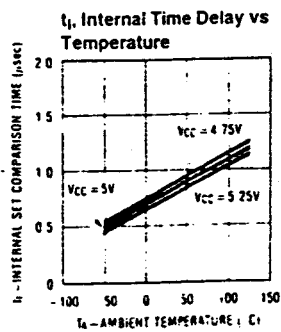
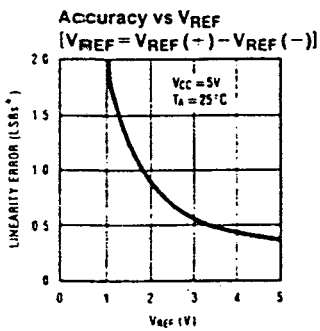
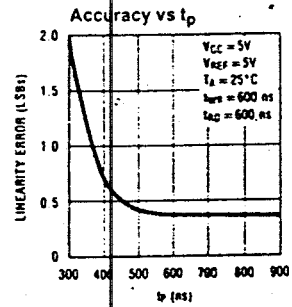
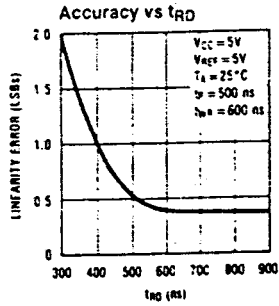
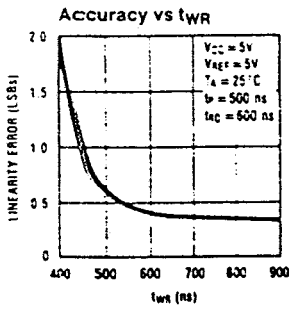
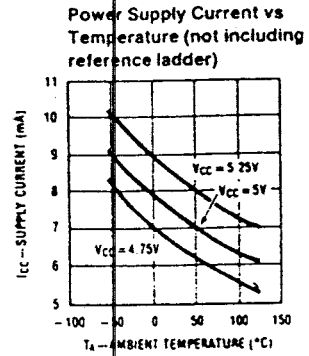
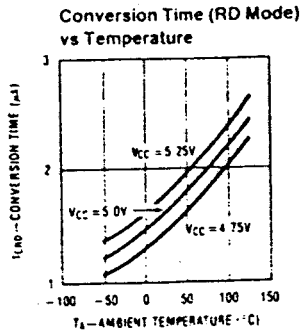
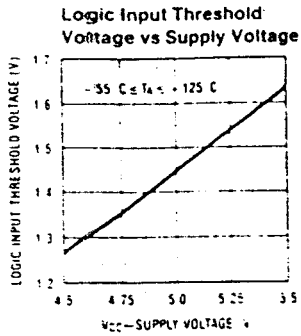


FIGURE 4. WR-RD Mode (Pin 7 is High) Stand-Alone Operation

TL/M/5501-10

Typical Performance Characteristics



$$1\text{ LSB} = \frac{V_{REF}}{256}$$

Description of Pin Functions

Pin Name	Function
1 V_{IN}	Analog input; range = $GND \leq V_{IN} \leq V_{CC}$
2 DB0	TRI-STATE data output—bit 0 (LSB)
3 DB1	TRI-STATE data output—bit 1
4 DB2	TRI-STATE data output—bit 2
5 DB3	TRI-STATE data output—bit 3
6 \overline{WR}/RDY	<p>WR-RD Mode</p> <p>WR: With \overline{CS} low, the conversion is started on the falling edge of \overline{WR}. Approximately 800 ns (the preset internal time out, t_i) after the \overline{WR} rising edge, the result of the conversion will be strobed into the output latch, provided that \overline{RD} does not occur prior to this time out (see Figures 3a and 3b).</p> <p>RD Mode</p> <p>RDY: This is an open drain output (no internal pull-up device). RDY will go low after the falling edge of \overline{CS}; RDY will go TRI-STATE when the result of the conversion is strobed into the output latch. It is used to simplify the interface to a micro-processor system (see Figure 2).</p>
7 Mode	<p>Mode: Mode selection input—it is internally tied to GND through a 50 μA current source.</p> <p>RD Mode: When mode is low</p> <p>WR-RD Mode: When mode is high</p>
8 \overline{RD}	<p>WR-RD Mode</p> <p>With \overline{CS} low, the TRI-STATE data outputs (DB0-DB7) will be activated when \overline{RD} goes low (see Figure 4). \overline{RD} can also be used to increase the speed of the converter by reading data prior to the preset internal time out (t_i, ~800 ns). If this is done, the data result transferred to output latch is latched after the falling edge of the \overline{RD} (see Figures 3a and 3b).</p> <p>RD Mode</p> <p>With \overline{CS} low, the conversion will start with \overline{RD} going low, also \overline{RD} will enable the TRI-STATE data outputs at the completion of the conversion. RDY going TRI-STATE and \overline{INT} going low indicates the completion of the conversion (see Figure 2).</p>

1.0 Functional Description

1.1 GENERAL OPERATION

The ADC0820 uses two 4-bit flash A/D converters to make an 8-bit measurement (Figure 1). Each flash ADC is made up of 15 comparators which compare the unknown input to a reference ladder to get a 4-bit result. To take a full 8-bit reading, one flash conversion is done to provide the 4 most significant data bits (via the MS flash ADC). Driven by the 4 MSBs, an internal DAC recreates an analog approximation of the input voltage. This analog signal is then subtracted from the input, and the difference voltage is converted by a second 4-bit flash ADC (the LS ADC), providing the 4 least significant bits of the output data word.

Pin Name	Function
9 \overline{INT}	<p>WR-RD Mode</p> <p>\overline{INT} going low indicates that the conversion is completed and the data result is in the output latch. \overline{INT} will go low, ~800 ns (the preset internal time out, t_i) after the rising edge of \overline{WR} (see Figure 3b); or \overline{INT} will go low after the falling edge of \overline{RD}, if \overline{RD} goes low prior to the 800 ns time out (see Figure 3a). \overline{INT} is reset by the rising edge of \overline{RD} or \overline{CS} (see Figures 3a and 3b).</p> <p>RD Mode</p> <p>\overline{INT} going low indicates that the conversion is completed and the data result is in the output latch. \overline{INT} is reset by the rising edge of \overline{RD} or \overline{CS} (see Figure 2).</p>
10 GND	Ground
11 $V_{REF(-)}$	The bottom of resistor ladder, voltage range: $GND \leq V_{REF(-)} \leq V_{REF(+)}$ (Note 5)
12 $V_{REF(+)}$	The top of resistor ladder, voltage range: $V_{REF(-)} \leq V_{REF(+)} \leq V_{CC}$ (Note 5)
13 \overline{CS}	\overline{CS} must be low in order for the \overline{RD} or \overline{WR} to be recognized by the converter.
14 DB4	TRI-STATE data output—bit 4
15 DB5	TRI-STATE data output—bit 5
16 DB6	TRI-STATE data output—bit 6
17 DB7	TRI-STATE data output—bit 7 (MSB)
18 \overline{OFL}	Overflow output—If the analog input is higher than the $V_{REF(+)}$, \overline{OFL} will be low at the end of conversion. It can be used to cascade 2 or more devices to have more resolution (9, 10-bit).
19 NC	No connection
20 V_{CC}	Power supply voltage

The internal DAC is actually a subsection of the MS flash converter. This is accomplished by using the same resistor ladder for the A/D as well as for generating the DAC signal. The DAC output is actually the tap on the resistor ladder which most closely approximates the analog input. In addition, the "sampled-data" comparators used in the ADC0820 provide the ability to compare the magnitudes of several analog signals simultaneously, without using input summing amplifiers. This is especially useful in the LS flash ADC, where the signal to be converted is an analog difference.

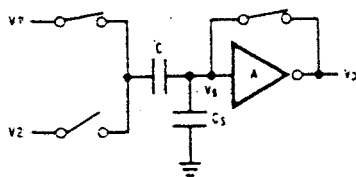
1.2 THE SAMPLED-DATA COMPARATOR

Each comparator in the ADC0820 consists of a CMOS inverter with a capacitively coupled input (Figure 5). Analog switches connect the two comparator inputs to the input capacitor (C) and also connect the inverter's input and output. This device in effect now has one differential input pair. A comparison requires two cycles, one for zeroing the comparator, and another for making the comparison.

In the first cycle, one input switch and the inverter's feedback switch (Figure 5a) are closed. In this interval, C is charged to the connected input (V1) less the inverter's bias voltage (VB, approximately 1.2V). In the second cycle (Figure 5b), these two switches are opened and the other (V2) input's switch is closed. The input capacitor now subtracts its stored voltage from the second input and the difference is amplified by the inverter's open loop gain. The inverter's input (VB') becomes

$$V_B' = (V_1 - V_2) \frac{C}{C - C_S}$$

and the output will go high or low depending on the sign of $V_B' - V_B$.



TL/H/5501-12

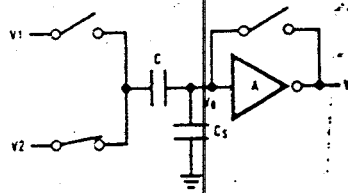
- $V_C = V_B$
- V on $C = V_1 - V_B$
- $C_S =$ stray input node capacitor
- $V_B =$ inverter input bias voltage

FIGURE 5a. Zeroing Phase

The actual circuitry used in the ADC0820 is a simple important expansion of the basic comparator described above. By adding a second capacitor and another switches to the input (Figure 6), the scheme can be adapted to make dual differential comparisons. In this circuit, feedback switch and one input switch on each capacitor (switches) are closed in the zeroing cycle. A comparison is then made by connecting the second input on each capacitor and opening all of the other switches (S switches). The change in voltage at the inverter's input, as a result of change in charge on each input capacitor, will now depend on both input signal differences.

1.3 ARCHITECTURE

In the ADC0820, one bank of 5 comparators is used in each 4-bit flash A/D converter (Figure 7). The MSB significant flash ADC also has one additional comparator to detect input overrange. These two sets of comparators operate alternately, with one group in its zeroing cycle while the other is comparing.

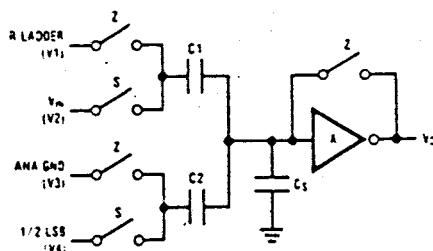


TL/H/5501-13

- $V_B' - V_B = (V_2 - V_1) \frac{C}{C + C_S}$
- $V_{O'} = \frac{-A}{C - C_S} [CV_2 - CV_1]$
- $V_{O'}$ is dependent on $V_2 - V_1$

FIGURE 5b. Compare Phase

FIGURE 5. Sampled-Data Comparator



TL/H/5501-14

$$V_C = \frac{-A}{C_1 - C_2 - C_S} [C_1(V_2 - V_1) - C_2(V_4 - V_3)]$$

$$= \frac{-A}{C_1 - C_2 - C_S} [\Delta C_1 + \Delta C_2]$$

FIGURE 6. ADC0820 Comparator (from MS Flash ADC)

Detailed Block Diagram

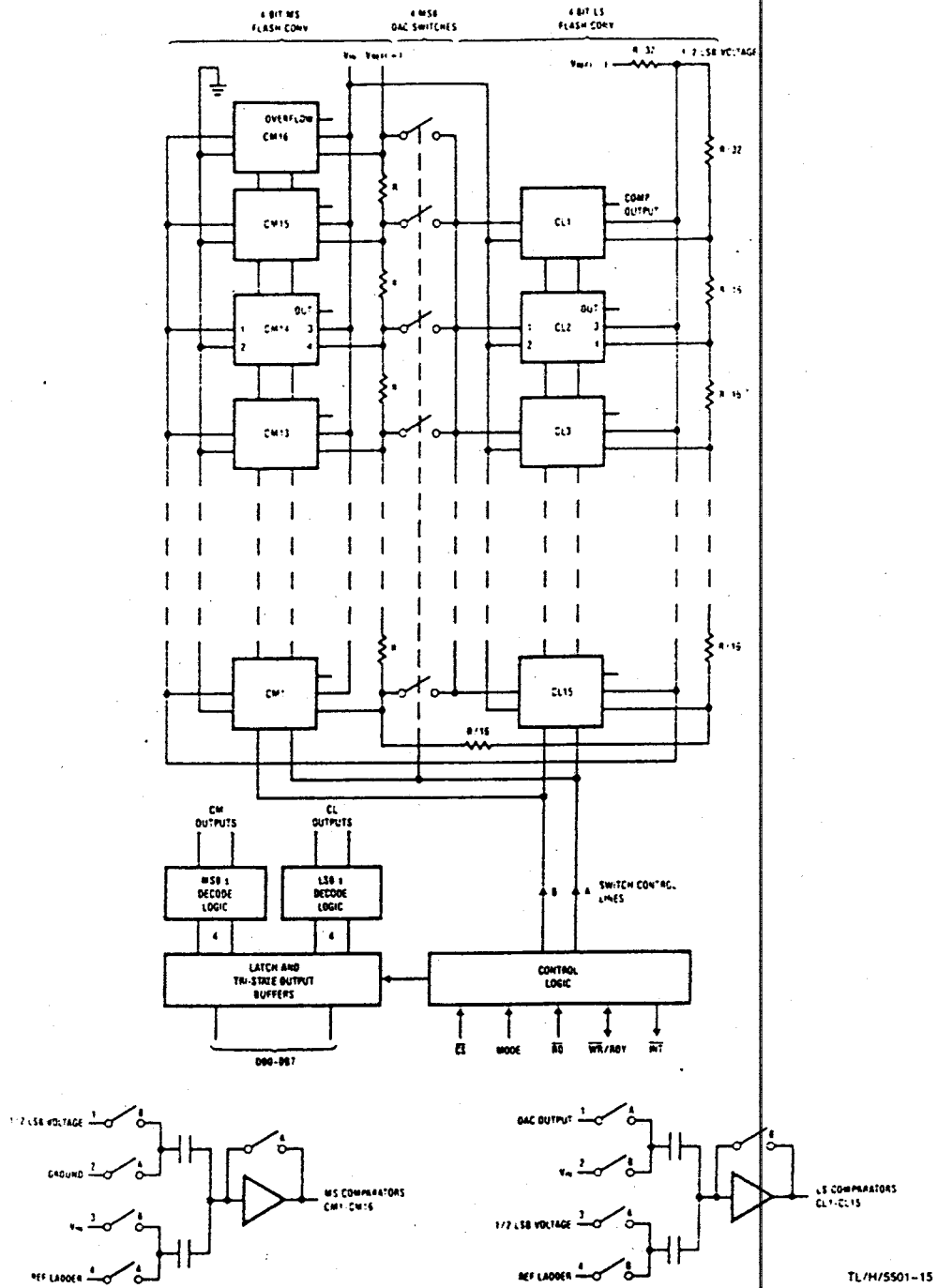


FIGURE 7

When a typical conversion is started, the \overline{WR} line is brought low. At this instant the MS comparators go from zeroing to comparison mode (Figure 5). When \overline{WR} is returned high after at least 600 ns, the output from the first set of comparators (the first flash) is decoded and latched. At this point the two 4-bit converters change modes and the LS (least significant) flash ADC enters its compare cycle. No less than 600 ns later, the \overline{RD} line may be pulled low to latch the lower 4 data bits and finish the 8-bit conversion. When \overline{RD} goes low, the flash A/Ds change state once again in preparation for the next conversion.

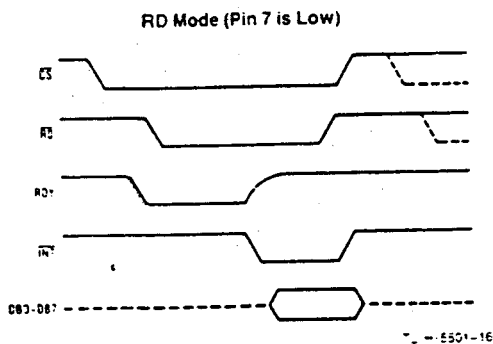
Figure 8 also outlines how the converter's interface timing relates to its analog input (V_{IN}). In WR-RD mode, V_{IN} is measured while \overline{WR} is low. In RD mode, sampling occurs during the first 800 ns of \overline{RD} . Because of the input connections to the ADC0820's LS and MS comparators, the converter has the ability to sample V_{IN} at one instant (Section 2.4), despite the fact that two separate 4-bit conversions are being done. More specifically, when \overline{WR} is low the MS flash is in compare mode (connected to V_{IN}), and the LS flash is in zero mode (also connected to V_{IN}). Therefore both flash ADCs sample V_{IN} at the same time.

1.4 DIGITAL INTERFACE

The ADC0520 has two basic interface modes which are selected by strapping the MODE pin high or low.

RD Mode

With the MODE pin grounded, the converter is set to Read mode. In this configuration, a complete conversion is done by pulling \overline{RD} low until output data appears. An \overline{INT} line is provided which goes low at the end of the conversion as well as a RDY output which can be used to signal a processor that the converter is busy or can also serve as a system Transfer Acknowledge signal.



When in RD mode, the comparator phases are internally triggered. At the falling edge of \overline{RD} , the MS flash converter goes from zero to compare mode and the LS ADC's comparators enter their zero cycle. After 800 ns, data from the MS flash is latched and the LS flash ADC enters compare mode. Following another 800 ns, the lower 4 bits are recovered.

WR then RD Mode

With the MODE pin tied high, the A/D will be set up for the WR-RD mode. Here, a conversion is started with the \overline{WR} input; however, there are two options for reading the output data which relate to interface timing. If an interrupt driven scheme is desired, the user can wait for \overline{INT} to go low before reading the conversion result (Figure 5). \overline{INT} will typically go low 800 ns after \overline{WR} 's rising edge. However, if a shorter conversion time is desired, the processor need not wait for \overline{INT} and can exercise a read after only 600 ns (Figure A). If this is done, \overline{INT} will immediately go low and data will appear at the outputs.

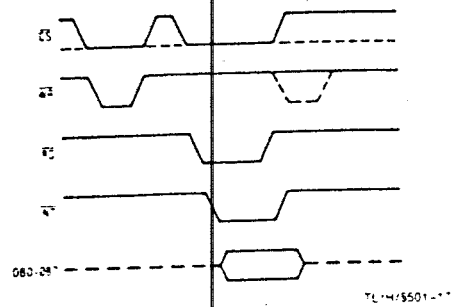


FIGURE A. WR-RD Mode (Pin 7 is High and $t_{RD} < t_I$)

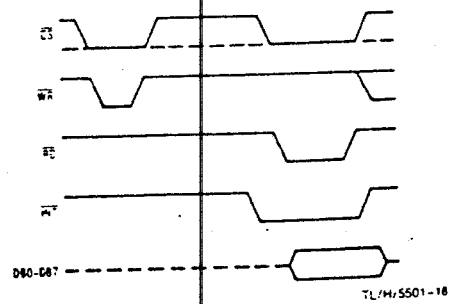
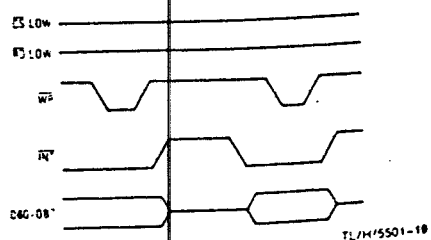


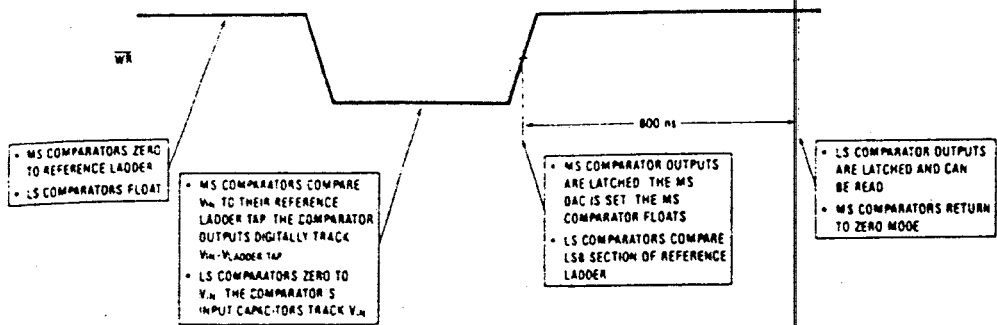
FIGURE B. WR-RD Mode (Pin 7 is High and $t_{RD} > t_I$)

Stand-Alone

For stand-alone operation in WR-RD mode, \overline{CS} and \overline{RD} can be tied low and a conversion can be started with \overline{WR} . Data will be valid approximately 800 ns following \overline{WR} 's rising edge.

WR-RD Mode (Pin 7 is High) Stand-Alone Operation





Note: MS means most significant.
LS means least significant.

FIGURE 8. Operating Sequence (WR-RD Mode)

TLH/5501-20

OTHER INTERFACE CONSIDERATIONS

In order to maintain conversion accuracy, \overline{WR} has a maximum width spec of 50 μ s. When the MS flash ADC's sampled-data comparators (Section 1.2) are in comparison mode (\overline{WR} is low), the input capacitors (C, Figure 5) must hold their charge. Switch leakage and inverter bias current can cause errors if the comparator is left in this phase for too long.

Since the MS flash ADC enters its zeroing phase at the end of a conversion (Section 1.3), a new conversion cannot be started until this phase is complete. The minimum spec for this time (tp, Figures 2, 3a, 3b, and 4) is 500 ns.

2.0 Analog Considerations

2.1 REFERENCE AND INPUT

The two V_{REF} inputs of the ADC0820 are fully differential and define the zero to full-scale input range of the A to D converter. This allows the designer to easily vary the span of the analog input since this range will be equivalent to the voltage difference between $V_{IN}(+)$ and $V_{IN}(-)$. By reducing $V_{REF}(V_{REF} = V_{REF}(+) - V_{REF}(-))$ to less than 5V, the sensitivity of the converter can be increased (i.e., if $V_{REF} = 2V$ then 1 LSB = 7.8 mV). The input/reference arrangement also facilitates ratiometric operation and in many cases the V_{REF} power supply can be used for transducer power as well as the V_{REF} source.

This reference flexibility lets the input span not only be varied but also offset from zero. The voltage at $V_{REF}(-)$ sets the input level which produces a digital output of all zeroes. Though V_{IN} is not itself differential, the reference design affords nearly differential-input capability for most measurement applications. Figure 9 shows some of the configurations that are possible.

2.2 INPUT CURRENT

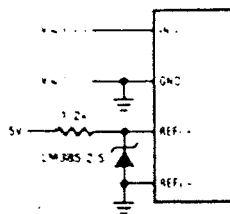
Due to the unique conversion techniques employed by the ADC0820, the analog input behaves somewhat differently than in conventional devices. The A/D's sampled-data comparators take varying amounts of input current depending on which cycle the conversion is in.

The equivalent input circuit of the ADC0820 is shown in Figure 10a. When a conversion starts (\overline{WR} low, WR-RD mode), all input switches close, connecting V_{IN} to thirty-one 1 pF capacitors. Although the two 4-bit flash circuits are not both in their compare cycle at the same time, V_{IN} still sees all input capacitors at once. This is because the MS flash converter is connected to the input during its compare interval and the LS flash is connected to the input during its zeroing phase (Section 1.3). In other words, the LS ADC uses V_{IN} as its zero-phase input.

The input capacitors must charge to the input voltage through the on-resistance of the analog switches (about 5 k Ω to 10 k Ω). In addition, about 12 pF of input stray capacitance must also be charged. For large source resistances, the analog input can be modeled as an RC network as shown in Figure 10b. As R_S increases, it will take longer for the input capacitance to charge.

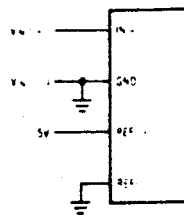
In RD mode, the input switches are closed for approximately 800 ns at the start of the conversion. In WR-RD mode, the time that the switches are closed to allow this charging is the time that \overline{WR} is low. Since other factors force this time to be at least 600 ns, input time constants of 100 ns can be accommodated without special consideration. Typical total input capacitance values of 45 pF allow R_S to be 1.5 k Ω without lengthening \overline{WR} to give V_{IN} more time to settle.

external Reference 2.5V Full-Scale



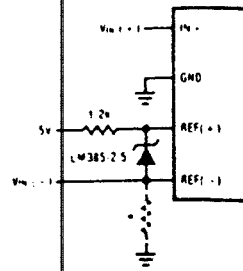
TL/H/5501-21

Power Supply as Reference



TL/H/5501-22

Input Not Referred to GND



* Current path must still exist from V_{IN-} to ground

TL/H/5501-2

FIGURE 9. Analog Input Options

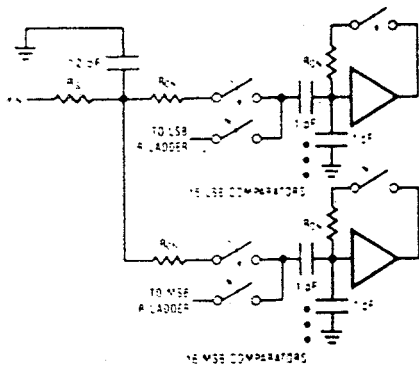
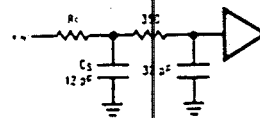


FIGURE 10a

TL/H/5501-24



TL/H/5501-2

FIGURE 10b

2.3 INPUT FILTERING

It should be made clear that transients in the analog input signal, caused by charging current flowing into V_{IN} , will not degrade the A/D's performance in most cases. In effect the ADC0820 does not "look" at the input when these transients occur. The comparators' outputs are not latched while \overline{WR} is low, so at least 600 ns will be provided to charge the ADC's input capacitance. It is therefore not necessary to filter out these transients by putting an external cap on the V_{IN} terminal.

2.4 INHERENT SAMPLE-HOLD

Another benefit of the ADC0820's input mechanism is its ability to measure a variety of high speed signals without the help of an external sample-and-hold. In a conventional SAR type converter, regardless of its speed, the input must remain at least $1/2$ LSB stable throughout the conversion process if full accuracy is to be maintained. Consequently, for many high speed signals, this signal must be externally sampled, and held stationary during the conversion.

Sampled-data comparators, by nature of their input switching, already accomplish this function to a large degree (Section 1.2). Although the conversion time for the ADC0820 is 1.5 μ s, the time through which V_{IN} must be $1/2$ LSB stable is much smaller. Since the MS flash ADC uses V_{IN} as its "compare" input and the LS ADC uses V_{IN} as its "zero" input, the ADC0820 only "samples" V_{IN} when \overline{WR} is low (Sections 1.3 and 2.2). Even though the two flashes are not done simultaneously, the analog signal is measured at one instant. The value of V_{IN} approximately 100 ns after the rising edge of \overline{WR} (100 ns due to internal logic prop delay) will be the measured value.

Input signals with slew rates typically below 100 mV/ μ s can be converted without error. However, because of the input time constants, and charge injection through the opened comparator input switches, faster signals may cause errors. Still, the ADC0820's loss in accuracy for a given increase in signal slope is far less than what would be witnessed in a conventional successive approximation device. An SAR type converter with a conversion time as fast as 1 μ s would still not be able to measure a 5V 1 kHz sine wave without the aid of an external sample-and-hold. The ADC0820, with no such help, can typically measure 5V, 7 kHz waveforms.

USULAN TUGAS AKHIR

1. JUDUL TUGAS AKHIR : IMPLEMENTASI ALGORITMA FAST FOURIER TRANSFORM DENGAN MENGGUNAKAN TMS 32010

2. RUANG LINGKUP :
 - Mikroprosesor
 - Elektronika Mikro
 - Bahasa Assembly
 - Rangkaian dan Komponen Digital
 - Rangkaian Tak Linier Aktif

3. LATAR BELAKANG : Perkembangan bidang elektronika digital telah sedemikian pesatnya sehingga berbagai bidang telah mempergunakan peralatan elektronik digital sebagai alat bantu yang vital. Bidang itu antara lain adalah industri, kedokteran, telekomunikasi dan militer. Dalam pemrosesan secara digital pengaruh noise dapat diabaikan. Hal ini yang merupakan keunggulan utama dibanding pemrosesan secara analog. Salah satu pemrosesan sinyal digital itu adalah Transformasi Fourier yang mentransformasikan suatu si-

nyal dalam daerah frekuensi.

Dalam daerah frekuensi, sinyal dapat lebih mudah dianalisa atau diproses. Transformasi Fourier ini banyak dipakai dalam berbagai bidang seperti misalnya analisa spectrum dan sintesa gelombang.

Dalam sistem digital, untuk mendapatkan Transformasi Fourier digunakan cara perhitungan langsung yang dinamakan Diskrit Fourier Transform (DFT). Kelemahan cara ini adalah waktu eksekusinya lama. Algoritma Fast Fourier Transform merupakan pengembangan dari cara perhitungan langsung DFT. Fast Fourier Transform (FFT) mempunyai waktu eksekusi yang lebih singkat. Fast Fourier Transform bisa diterapkan pada mikroprosesor, baik yang general purpose processor maupun yang special purpose processor. Jika diterapkan pada general purpose processor diperlukan waktu eksekusi yang cukup lama terutama bila jumlah data cukup besar. De-

ngan special purpose processor bisa didapatkan waktu eksekusi yang singkat.

Salah satu special purpose processor itu adalah keluarga TMS 320. TMS 32010 merupakan salah satu Digital Signal Processor yang banyak dipakai dalam pemrosesan sinyal digital.

4. TUJUAN

- : - Merencanakan serta membuat implementasi FFT dengan TMS 32010.
- Mendapatkan spektrum sinyal analog dengan metoda FFT dan ditampilkan di monitor dengan bantuan IBM PC.

5. PENELAAHAN STUDI

: Untuk membuat suatu implementasi FFT pertama kali harus dipelajari tentang algoritma FFT itu sendiri. Kemudian perlu dipelajari juga prinsip kerja dan karakteristik dari prosesor TMS 32010. Selain itu bahasa assembly dari prosesor TMS 32010 juga memegang peranan yang tidak kalah pentingnya karena ada keterkaitan yang tidak dapat dipisahkan antara perangkat keras dan

lunak.

Untuk membentuk suatu peralatan tentu saja prosesor TMS 32010 tidak dapat berdiri sendiri maka dibutuhkan komponen penunjang yang lain. Karena itu analisa dilanjutkan dengan mempelajari karakteristik dari komponen penunjang itu.

Selanjutnya juga perlu dipelajari teknik - teknik interface dari TMS 32010 ke IBM PC dan juga perangkat lunak dari 8088. Hal itu perlu karena disini TMS 32010 hanya berfungsi sebagai sinyal prosesor saja dan untuk menampilkan hasil pengolahan data masih dipakai IBM PC.

Dengan bekal pengetahuan tentang hal-hal di atas maka dapat direncanakan implementasi FFT dengan menggunakan TMS 32010.

6. LANGKAH-LANGKAH : 1. Studi literatur tentang algoritma Fast Fourier Transform.
2. Mempelajari perangkat keras dan perangkat lunak prosesor TMS 32010 beserta komponen penunjangnya.

3. Studi literatur mengenai teknik interface pada IBM PC serta perangkat lunak yang digunakan untuk tampilan pada monitor.
4. Merencanakan perangkat keras dan perangkat lunak untuk FFT processor.
5. Membuat dan mencoba perangkat keras dan perangkat lunak.
6. Menarik kesimpulan serta menulis naskah tugas akhir.

7. RELEVANSI

1. Sebagai tambahan bahan bagi para peminat yang mempelajari pemrosesan sinyal digital pada umumnya serta Fast Fourier Transform khususnya, terutama dengan memakai prosesor TMS 32010.
2. Dengan alat ini bisa didapatkan Fourier Transform secara cepat sehingga diharapkan alat ini bisa diaplikasikan pada berbagai bidang seperti misalnya telekomunikasi, medika dan militer.

8. JADWAL KERJA

- : Seluruh kegiatan dijadwalkan dapat selesai dalam waktu 6 (enam)

bulan. Jadwal kerja itu adalah sebagai berikut :

LANGKAH KEGIATAN	BULAN KE					
	I	II	III	IV	V	VI
1	██████████					
2	██████████████████					
3		██████████████████				
4		██████████████████████████████				
5			██████████████████████████████			
6					██████████████████	