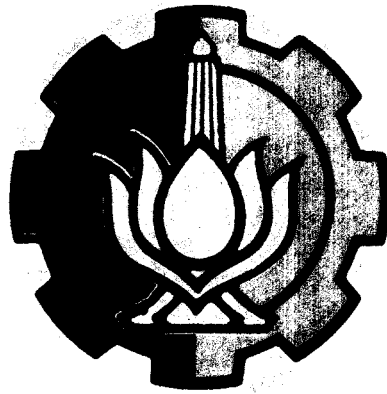


3928/ITS/H/91 ✓

**PERENCANAAN DAN PEMBUATAN
SINTESA SUARA UNTUK BAHASA INDONESIA
DENGAN IC UM 5100**



PSE
621.398
Set
P-1
1990

Oleh :

V. HENGKI SETIAWAN

2872200766

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

**PERENCANAAN DAN PEMBUATAN
SINTESA SUARA UNTUK BAHASA INDONESIA
DENGAN IC UM 5100**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Teknik Elektronika

Di

Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui
Dosen Pembimbing



(Dr. Ir. HANDAYANI TJ)

SURABAYA

AGUSTUS, 1990

ABSTRAK

UM 5100 adalah sebuah CMOS LSI pengolah suara yang diproduksi oleh United Microelectronics Corporation (UMC). IC ini merupakan sebuah IC sintesa suara jenis Natural Speech Analysis/Synthesis, yang memungkinkan pemakai merekam (record) dan mereproduksi (play) suara/kata yang dikehendaki tidak peduli dalam bahasa apapun.

Sinyal suara yang diterima oleh UM 5100, dengan menggunakan modulasi delta, akan diubah dalam kode-kode digital, kemudian disimpan di dalam memori. Pada saat play, kode-kode tersebut akan diubah kembali menjadi sinyal analog. Kemudian setelah melewati rangkaian low pass filter diperkuat oleh penguat audio sehingga dapat didengar kembali.

Dengan menginterfacekan dengan komputer, memungkinkan dilakukan pengontrolan rekam/play dengan perangkat lunak. Juga dapat dilakukan penyimpanan data suara ke dalam disket, konversi teks ke suara dan aplikasi lainnya. Sehingga keterbatasan perbendaharaan kata yang biasanya menyertai sintesa suara jenis Natural Speech Analysis/synthesis dapat diatasi.

KATA PENGANTAR

Puji syukur kami ucapkan pada Tuhan Yang Maha Kasih karena hanya dengan bimbingan dan anugerahNya penulis dapat menyelesaikan Tugas Akhir dengan sebaik-baiknya.

Tugas Akhir dengan bobot 6 SKS (Satuan Kredit Semester) disusun guna memenuhi sebagian persyaratan untuk memperoleh gelar Kesarjanaan pada Jurusan Teknik Elektro - Fakultas Teknologi Industri - ITS Surabaya.

Penyelesaian Tugas akhir ini penulis laksanakan berdasarkan pada teori-teori yang telah diperoleh selama kuliah, buku-buku literatur, bimbingan dosen, dan juga berkat dorongan dan bantuan banyak pihak.

Ucapan terima kasih yang setulusnya tak lupa kami sampaikan kepada :

- Ibu Dr. Ir. Handayani Tjandrasa, M.Sc, selaku dosen wali, dosen pembimbing, dan sekaligus sebagai Koordinator Bidang Studi Elektronika
- Bapak Ir. Syariffudin Mahmudsyah, M. Eng, selaku Ketua Jurusan Teknik Elektro FTI ITS
- Semua pihak yang telah banyak membantu sehingga Tugas Akhir ini dapat terselesaikan.

Akhir kata, semoga buku Tugas Akhir ini dapat bermanfaat bagi kita semua baik sebagai pengetahuan maupun sebagai dasar untuk dikembangkan lebih lanjut.

Surabaya, Juli 1990

Penulis

DAFTAR ISI

JUDUL	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
I. 1. LATAR BELAKANG	1
I. 2. PERMASALAHAN	1
I. 3. PEMBATAAN MASALAH	2
I. 4. METODOLOGI	3
I. 5. SISTEMATIKA PEMBAHASAN	3
BAB II TEORI PENUNJANG	5
II. 1. TEKNIK SINTESA SUARA ELEKTRONIK ...	5
II. 1. 1. Natural Speech Analysis/ Synthesis	5
(a). Time-domain Analysis/ Synthesis	7
i. Waveform Digitization	8
ii. Delta Modulation	9
iii. Differential PCM	11

iv. Adaptive Differential PCM	...	12
(b). Frequency-Domain Analysis/Synthesis	13
i. Linier Predictive Coding	13
ii. Formant Analysis/Synthesis	14
II.1.2. Artificial Speech Construction/Synthesis	16
II.2. UM 5100 - VOICE PROCESSOR	17
II.3. OPERATIONAL AMPLIFIER	19
II.3.1. Op Amp Sebagai Penguat	20
(a). Inverting Amplifier	..	20
(b). Non Inverting Amplifier		21
II.3.2. Filter Aktif	22
II.3.3. Comparator	24
II.4. IBM PC XT	24
II.4.1. Slot Pada IBM PC	24
II.4.2. Peta Pemakaian Memori	27
II.4.3. Diagram Waktu Sistem Bus	..	28
(a). I/O Port Read Bus Cycle		28
(b). I/O Port Write Bus Cycle		30
BAB III SINTESA SUARA DENGAN IC UM 5100	35
III.1. PENDAHULUAN	35

III. 2.	RANGKAIAN SINTESA SUARA	35
III. 2. 1.	Blok Diagram	35
III. 2. 2.	Penguat	37
III. 2. 3.	Comparator	38
III. 2. 4.	Low-Pass Filter	39
III. 2. 5.	Penguat Audio	41
III. 3.	RANGKAIAN INTERFACE	42
III. 3. 1.	Pendahuluan	42
III. 3. 2.	Dasar Perencanaan	42
III. 3. 3.	Decoder I	44
III. 3. 4.	Rangkaian Reset	45
III. 3. 5.	Rangkaian Play/Record	46
III. 3. 6.	Kontrol Buffer	47
III. 3. 7.	Decoder II	48
BAB IV	PERENCANAAN PERANGKAT LUNAK	50
IV. 1.	PENDAHULUAN	50
IV. 2.	INSTRUKSI-INSTRUKSI YANG DIPAKAI	50
IV. 2. 1.	Instruksi Port	50
IV. 2. 2.	Instruksi Mem, MemW, MemL	51
IV. 2. 3.	Instruksi-instruksi Untuk Pe- ngolahan File	52
IV. 3.	PROCEDURE MENU UTAMA	54
IV. 4.	PROCEDURE PLAY	55
IV. 5.	PROCEDURE REKAM	57
IV. 6.	PROCEDURE SIMPAN	58

IV. 7. PROCEDURE PANGGIL	60
IV. 8. PROCEDURE TEKS KE SUARA	62
BAB V UJI COBA DAN UJI UKUR	66
V. 1. UJI COBA	66
V. 2. UJI UKUR	71
V. 2. 1. Uji Ukur Penguat Microphone	72
V. 2. 2. Uji Ukur Comparator	75
V. 2. 3. Uji Ukur Low-Pass Filter	76
V. 2. 4. Uji Ukur Clock	77
V. 2. 5. Uji Ukur Gelombang Output	78
BAB VI PENUTUP	81
DAFTAR PUSTAKA	82
LAMPIRAN A RANGKAIAN SINTESA SUARA DAN INTERFACE	84
LAMPIRAN B PROGRAM PENGENDALI	85
LAMPIRAN C LEMBAR DATA	103
LAMPIRAN D USULAN TUGAS AKHIR	124

DAFTAR GAMBAR

2.1.	Blok Diagram Sistem Analisa/Sintesa	6
2.2.	Proses Analisa/Sintesa Pada Time-Domain	8
2.3.	Delta Modulasi	10
2.4.	Differential Pulse Code Modulation	11
2.5.	Linier Predictive Coding	14
2.6.	LPC Speech-synthesizer	15
2.7.	Pin Configurations UM 5100	17
2.8.	Inverting Amplifier	21
2.9.	Non Inverting Amplifier	21
2.10.	Low-Pass Filter 40 dB/dekade	23
2.11.	Comparator	25
2.12.	Slot IBM PC XT	26
2.13.	Memory Map IBM PC XT	29
2.14.	I/O Port Read Bus Cycle	31
2.15.	I/O Port Write Bus Cycle	33
3.1.	Blok Diagram Rangkaian Sintesa Suara	36
3.2.	Rangkaian Penguat I	37
3.3.	Rangkaian Penguat II	38
3.4.	Rangkaian Comparator	39
3.5.	Rangkaian Low-Pass Filter	40
3.6.	Rangkaian Penguat Audio	41
3.7.	Diagram Blok Rangkaian Interface	43

3. 8.	Hubungan Antara Interface, Memori Luar dan UM 5100	43
3. 9.	Rangkaian Decoder I	44
3. 10.	Rangkaian Reset	46
3. 11.	Rangkaian Play Record	47
3. 12.	Kontrol Buffer	47
3. 13.	Rangkaian Decoder II	48
4. 1.	Flowchart Procedure Menu Utama	56
4. 2.	Flowchart Procedure Play	57
4. 3.	Flowchart Procedure Rekam	58
4. 4.	Flowchart Procedure Simpan	59
4. 5.	Flowchart Procedure Panggil	61
4. 6.	Flowchart Procedure Konversi Dua Kata	65
5. 1.	Rangkaian Interface Terpasang Pada Mother Board	67
5. 2.	Menu - Utama Program Pengendali	67
5. 4.	Menu Teks ke Suara	70
5. 5.	PCB Rangkaian Sintesa Suara	72
5. 6.	PCB Rangkaian Interface	73
5. 7.	Bentuk Gelombang Input dan Output Penguat Microphone	74
5. 8.	Bentuk Gelombang Input dan Output Penguat Microphone Pada Saat Terjadi Pemotongan ...	74
5. 9.	Sinyal Output Comparator (Terminal (+) > Ter- minal (-))	76

5.10. Sinyal Output Comparator (Terminal (-) > Terminal (+))	77
5.11. Sinyal Input dan Sinyal Output Low-Pass Filter	78
5.12. Sinyal Clock	79
5.13. Sinyal Input dan Output Pada Frekwensi 1 KHz	80
5.14. Sinyal Input dan Output Pada Frekwensi 1500 Hz	80

DAFTAR TABEL

2.1. I/O Port Read Bus Cycle Timings	32
2.2. I/O Port Write Bus Cycle Timings	34

B A B I

P E N D A H U L U A N

I. 1 L A T A R B E L A K A N G

Perkembangan elektronika yang demikian pesat telah merambah segala sendi kehidupan. Hampir di setiap bidang, elektronika selalu hadir dalam bentuk yang berbeda-beda. Ini semua tak lepas dari kemajuan yang sangat berarti dalam bidang elektronika mikro dan rangkaian terpadu.

Begitu banyak IC untuk keperluan khusus diluncurkan di pasaran, sehingga peralatan yang dahulunya tampak begitu rumit karena harus dibangun dari berbagai komponen sekarang tampak begitu sederhana dan ringkas.

Demikian juga dalam bidang Sintesa Suara Elektronik, banyak IC pensintesa dan pemroses suara diciptakan untuk menjawab kebutuhan yang semakin meningkat. IC-IC ini tidak hanya diperlukan oleh dunia industri dan robotika, tetapi juga dibutuhkan oleh bidang-bidang lain. Berbagai macam chip, dengan teknik - teknik sintesa suara yang bermacam-macam pula, berlomba menarik minat konsumen. Votrax SC 01, DigitalKer DT-1050, UM 5100 merupakan contoh IC sintesa suara yang dikenal di Indonesia.

I. 2. P E R M A S A L A H A N

Pada dasarnya ada dua macam teknik yang digunakan dalam sintesa suara elektronik, yaitu :

- Natural Speech Analysis/synthesis, dan
- Artificial Speech Construction/synthesis

Teknik yang pertama menggunakan cara rekam/play-back suara, baik secara analog maupun digital. Sedang yang kedua dengan menyusun kembali suara-suara dasar (biasanya fonem), sehingga didapatkan suara yang diinginkan.

Natural speech analysis/synthesis menghasilkan suara yang lebih alami, namun memerlukan tempat penyimpanan data (memori) yang cukup besar dan perbendaharaan yang terbatas. Teknik Artificial speech construction/synthesis kelebihanannya adalah bahwa kata-kata yang dapat dihasilkan hampir tak terbatas walaupun dari segi kealamian suara masih kalah bila dibandingkan dengan Natural speech analysis/synthesis.

IC yang menggunakan teknik Artificial speech construction/synthesis biasanya dirancang untuk satu bahasa tertentu, kebanyakan bahasa Inggris. Kalau IC ini digunakan untuk sintesa dalam bahasa lain tentu saja hasilnya tidak sebaik bila digunakan untuk bahasa yang sesuai. Sehingga untuk keperluan sintesa suara dalam bahasa Indonesia dirasakan adanya keterbatasan.

1.3. PEMBATASAN MASALAH

Untuk mengatasi keterbatasan inilah, maka dalam Tugas Akhir ini direncanakan pembuatan sintesa suara untuk bahasa Indonesia. Sebagai komponen utama dipilih IC UM 5100, sebuah Voice Processor produksi UMC (United

Microelectronic Corporation). IC ini merupakan IC jenis Natural speech analysis/synthesis. Untuk mereduksi penggunaan memori, digunakan modulasi delta.

UM 5100 dipilih karena diharapkan dengan menggunakan IC jenis Natural speech analysis/synthesis akan didapatkan suara yang lebih alami. Untuk lebih meningkatkan kemampuan IC ini, rangkaian sintesa suara ini akan diinterfacekan ke Komputer IBM PC XT, sehingga keterbatasan pemakaian memori yang biasanya menyertai pemakaian IC jenis Natural speech analysis/synthesis dapat diatasi.

I. 4. METODOLOGI

Perencanaan alat yang dibuat untuk Tugas Akhir ini dilakukan dengan jalan mempelajari perangkat Keras maupun perangkat lunak. Pada perangkat Keras dipelajari antara lain: macam-macam teknik sintesa suara, teori Operational Amplifier, serta sistem pada IBM PC XT terutama mengenai teknik interface.

I. 5. SISTEMATIKA PEMBAHASAN

Sistematika pembahasan Tugas Akhir ini adalah sebagai berikut :

BAB I : Latar Belakang, permasalahan, pembatasan masalah, metodologi serta sistematika pembahasan.

BAB II : Menjelaskan tentang teori penunjang yang meliputi : teknik sintesa suara

elektronik, operational amplifier, IC UM 5100 serta mengenai teknik interface pada IBM PC XT.

BAB III : Perencanaan perangkat keras yang terdiri dari dua rangkaian yaitu : rangkaian sintesa suara dan rangkaian interface.

BAB IV : Perencanaan perangkat lunak, membahas mengenai perangkat lunak yang digunakan untuk menjalankan peralatan yang dibuat.

BAB V : Berisi hasil uji coba dan uji ukur dari peralatan yang dibuat.

BAB VI : Kesimpulan .

B A B II

TEORI PENUNJANG

II. 1. TEKNIK SINTESA SUARA ELEKTRONIK

Pada dasarnya ada dua teknik yang digunakan untuk mensintesa suara manusia secara elektronik, yaitu :¹⁾

1. Natural Speech Analysis/synthesis
2. Artificial Constructive/synthesis

Natural speech analysis/synthesis dapat dibedakan menjadi dua teknik : Analog dan Digital. Untuk yang analog, sebagai media perekam digunakan pita magnetik. Sedangkan untuk yang digital sinyal suara diubah dalam bentuk digital dan kemudian disimpan dalam memori (bisa RAM atau EPROM).

Pada teknik Artificial constructive/synthesis dikenal dua macam metode, yakni :

- Direct phoneme synthesis
- Text -to- speech phoneme synthesis

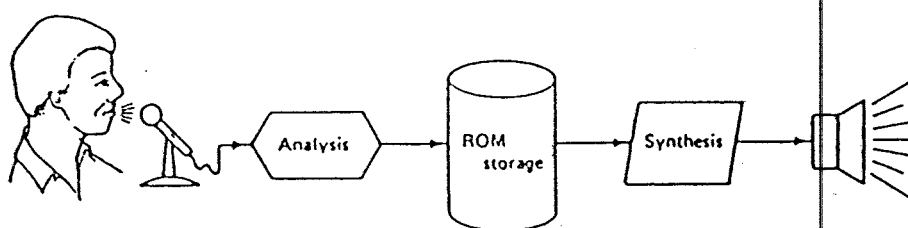
II. 1. 1. NATURAL SPEECH ANALYSIS/SYNTHESIS

Teknik ini dikatakan alami (natural) karena menggunakan metode rekam dan playback suara manusia. Metode rekam/play back analog menghasilkan suara yang lebih alami, tetapi sangat tidak praktis untuk penggunaan yang umum karena suara yang telah direkam harus diakses secara serial. Hampir tidak mungkin untuk merekam seluruh

1) A. C. Staugaard, Jr., 'Robotic And AI', New Jersey, Prentice Hall Inc, 1987, hal. 135

kata-kata yang diperlukan untuk sintesa suara yang dapat digunakan secara luas. Metode ini paling tepat digunakan untuk keperluan yang terbatas.

Metode rekam/play back digital selalu meliputi dua operasi : analisa dan sintesa. Gambar 2.1. menunjukkan blok diagramnya. Selama fase analisa, suara manusia dianalisa, diubah dalam bentuk digital, dan disimpan.



Gambar 2.1²⁾

Blok Diagram Sistem Analisa/Sintesa

Kemudian selama fase sintesa, suara yang telah dirubah dalam bentuk digital dipanggil kembali dari memori dan dikonversikan dalam bentuk analog untuk mendapatkan bentuk gelombang suara yang asli.

Metode sintesa/analisa digital lebih fleksibel dibandingkan dengan metode analog, karena kata-kata yang telah disimpan dapat diakses secara acak dari memori.

²⁾ Ibid, hal. 135.

Namun perbendaharaan katanya dibatasi oleh kapasitas memori. Karena itulah, beberapa teknik encoding yang berbeda digunakan untuk menganalisa dan memadatkan bentuk gelombang suara.

Salah satu jenis analisa/sintesa digital adalah Time-Domain Analysis/synthesis. Dengan analisa time-domain, gelombang suara diubah dalam bentuk digital pada time-domain. Dengan kata lain, gelombang analog secara periodik disampling dan dikonversikan ke dalam kode digital menggunakan A/D converter. Sampling suara yang telah disimpan kemudian dilewatkan D/A Converter untuk menghasilkan suara kembali selama operasi sintesa. Analisa sintesa time-domain juga disebut Waveform Digitization.

Jenis analisa/sintesa digital yang lain adalah Frequency-domain Analysis/synthesis. Dengan teknik ini, spektrum frekwensi dari gelombang analog dianalisa dan diberi kode. Teknik ini juga dilengkapi dengan parameter frekwensi selama tahap analisa untuk mengontrol generator dan filter frekwensi elektronik yang mereproduksi baik bunyi yang bersuara maupun tidak (voice and unvoice sound).

II. 1. 1. (a). Time-domain analysis/synthesis.

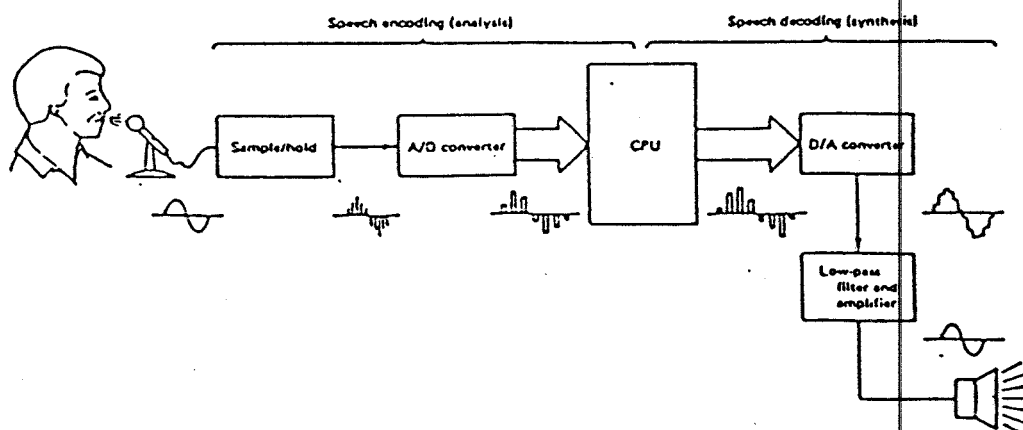
Time-domain analysis/synthesis (analisa/sintesa pada time-domain) mempunyai dua dasar yaitu :

- a. Encoding bentuk gelombang suara manusia untuk didigitisasi dan disimpan di dalam memori.
- b. Decoding suara yang telah didigitisasi ke dalam bentuk analog untuk didengarkan kembali.

Gambar 2.2. menerangkan proses sintesa/analisa pada time-domain.

II. 1. 1. (a). i. Wave Form Digitization

Salah satu contoh dari sintesa/analisa dari time-domain adalah Waveform Digitization atau Simple Pulse Code Modulation (SPCM).



Gambar 2.2³⁾

Proses analisa/sintesa pada time-domain

Waveform Digitization adalah suatu teknik sintesa suara yang menggunakan amplitudo gelombang suara sebagai

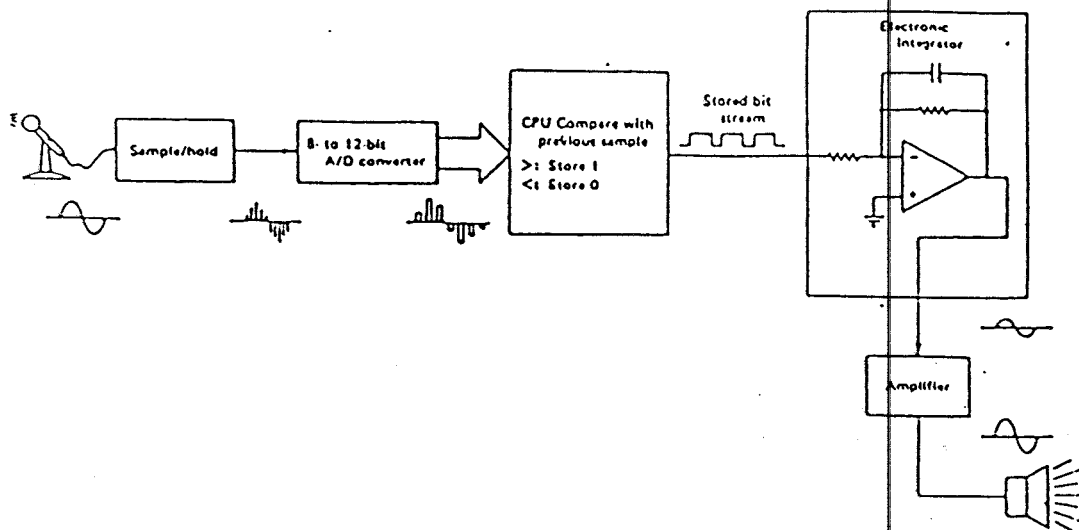
3) Ibid., hal 139.

variabel. Dalam teknik ini, karakteristik amplitudo dari suatu bentuk gelombang bunyi disimpan dalam memori dan bila diperlukan akan dihasilkan kembali. Salah satu kelemahan teknik ini adalah diperlukannya tempat penyimpanan data yang sangat besar. Karena alasan itulah, maka beberapa metode encoding pada time-domain diperkenalkan untuk mengurangi jumlah memori yang diperlukan, antara lain :

- a. Delta Modulation (DM)
- b. Differential Pulse Code Modulation (DPCM)
- c. Adaptive Differential Pulse Code Modulation (ADPCM)

II. 1. 1. (a). ii. Delta Modulation (DM)

Delta Modulation merupakan cara pengurangan data yang paling sederhana. Dengan DM, hanya satu bit saja yang disimpan untuk masing-masing sampel (sample) gelombang suara. Oleh A/D Converter, sampel ini diubah menjadi bentuk digital terlebih dahulu untuk kemudian dibandingkan dengan sampel sebelumnya. Jika sampel yang dibandingkan lebih besar dari sampel sebelumnya, maka oleh komputer akan diberi nilai 1. Sedangkan bila lebih kecil akan diberi nilai 0. Dengan demikian sinyal suara disimpan ke dalam memori sebagai sebuah urutan bit tunggal. Gambar 2.3 memperlihatkan metode dasar Delta Modulation.

Gambar 2. 3⁴⁾

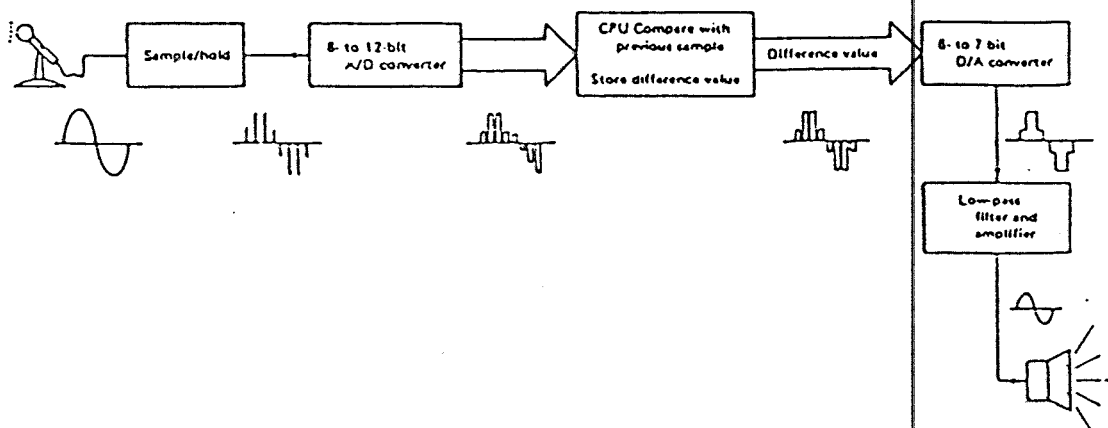
Delta Modulation

Karena hanya satu bit yang digunakan untuk mewakili masing-masing sampel, sampling rate nya harus sangat tinggi (± 32.000 sampel per detik) untuk dapat menangkap seluruh detail sinyal suara. Sehingga untuk 10 detik suara diperlukan 39 K bytes memori. Bila dibandingkan dengan SPCM 8 bit, hal ini berarti pengurangan data sebesar 40 %. Masalah yang ada pada DM adalah Compliance Error, yang terjadi jika gelombang suara berubah terlalu cepat. Kadang-kadang hal ini disebut juga sebagai Slope Overload , karena slope dari gelombang suara berubah terlalu cepat. Akibatnya data digital yang dihasilkan bukan merupakan data digital yang sebenarnya dan dapat

4) Ibid , hal. 142.

menimbulkan distorsi pada output. Slope overload hanya dapat diatasi dengan memperbesar sampling rate, tetapi ini berarti memperbesar pula data rate dan memori yang diperlukan.

II. 1. 1. (a). iii. Differential Pulse Code Modulation (DPCM)



Gambar 2. 4⁵⁾

Differential Pulse Code Modulation

DPCM menggunakan ide yang sama dengan DM. Hanya saja diperlukan beberapa bit untuk mewakili perbedaan antara dua sample yang dibandingkan, bukan satu seperti pada DM. Gambar 2.4 memperlihatkan rangkain DPCM.

Kadang-kadang perbedaan antara dua sampel diberi nilai dengan 6 atau 7 bit. Bit-bit ini terdiri dari 1 bit tanda ditambah 5 atau 6 bit untuk nilai beda. Bit tanda digunakan untuk menggambarkan slope dari gelombang input.

5) Ibid, hal. 143.

Sistem DPCM 7 bit dengan 6000 sampel per detik membutuhkan 51 K byte memori. Hal ini berarti terdapat penghematan 12 % bila dibandingkan sistem PCM 8 bit yang sederhana dan penghematan sebesar 25 % akan terjadi pada sistem DPCM 6 bit.

Seperti DM, pada DPCM juga terjadi Slope Overload. Bila amplitudo gelombang suara berubah terlalu drastis, 6 atau 7 bit saja tidaklah cukup untuk menggambarkan nilai beda. Pemecahannya hanya dengan meningkatkan jumlah bit atau meningkatkan sampling rate, di mana keduanya akan menambah jumlah memori yang diperlukan.

II. 1. 1. (a). iv. Adaptive Differential PCM (ADPCM)

ADPCM merupakan salah satu variasi dari DPCM yang mampu mengurangi masalah slope overload. Dengan teknik ini hanya diperlukan 3 atau 4 bit saja, sehingga memori yang dibutuhkan hanya 25 sampai 50 % dari memori yang dibutuhkan oleh SPCM.

ADPCM menggunakan cara yang cukup rumit untuk mendigitisasi gelombang suara. Gelombang suara disampel dengan dengan \pm 6000 sampel per detik oleh A/D converter 8 atau 10 bit. Kemudian komputer menghitung nilai beda, seperti juga pada DPCM. Hanya saja nilai ini akan diatur sedemikian rupa untuk mengkompensasi terhadap slope, menggunakan sebuah harga yang telah dihitung, yang disebut Faktor Kwantisasi (Quantization factor). Faktor kwantisasi mengatur nilai beda dengan dinamis

sesuai dengan slope dari gelombang input. Nilai beda yang telah diatur ini hanya memerlukan 3 atau 4 bit saja.

Disamping hanya memerlukan bit yang lebih sedikit, sampling rate dari sistem ADPCM juga dapat diturunkan, karena slope overload telah dapat dikurangi. Sampling rate serendah 4000 sampel per detik sudah memungkinkan. Dengan sampling rate sebesar itu dan kode 3 bit, data rate menjadi 3 bit x 4000 sampel per detik, atau 12.000 bps. Sehingga suara sepanjang 10 detik hanya memerlukan 15 K byte memori. Yang lebih umum dipakai adalah sampling rate 8000 dan 4 bit kode. Dengan sistem ini, untuk suara yang sama diperlukan memori sebesar 39 K byte.

II. 1. 1. (b). FREQUENCY-DOMAIN ANALYSIS/SYNTHESIS

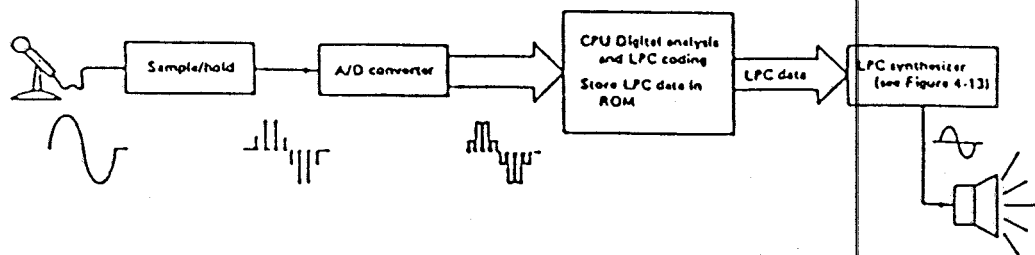
Pada dasarnya, ada 2 metode analisa/sintesa suara pada frequency-domain, yaitu :

- Linier Predictive Coding (LPC), dan
- Formant analysis/synthesis.

Dari keduanya, LPC lah yang menghasilkan suara yang lebih alami, walaupun lebih rumit dan kurang ekonomis.

II. 1. 1. (b). i. Linier Predictive Coding (LPC)

LPC dikenal pertama kali pada permulaan tahun 1970. Sistem LPC diperlihatkan pada Gambar 2.5. Langkah pertama pada LPC adalah mendigitisasi gelombang suara dengan A/D converter menggunakan SPCM. Dalam bentuk digital, gelombang dianalisa untuk mendapatkan frekwensi,

Gambar 2.5⁶⁾

Linier Predictive Coding

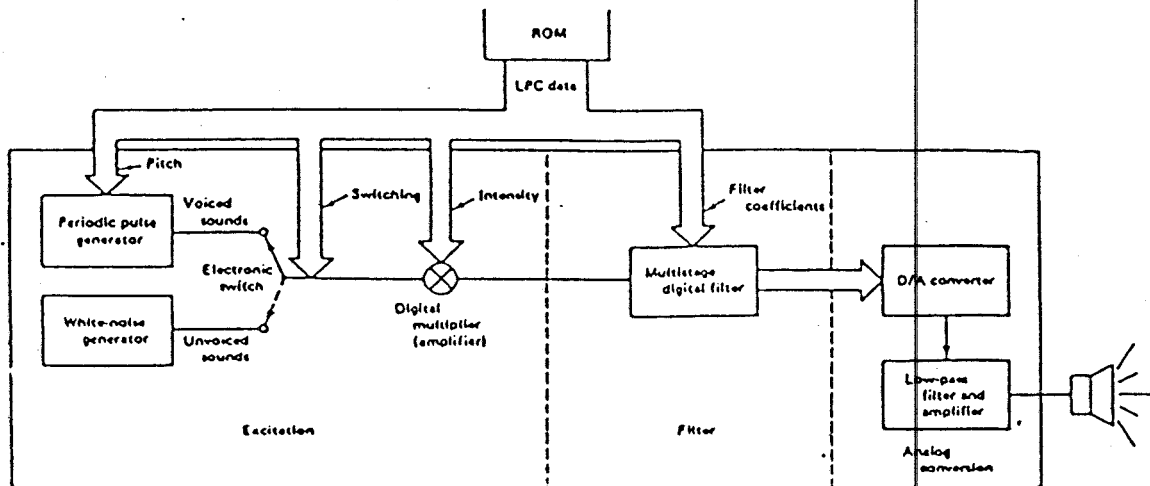
intensitas, dan variabel jenis alat ucap yang diperlukan untuk membentuk kembali gelombang suara secara matematis. Data suara yang telah diambil kemudian diberi kode dalam bentuk parameter persamaan linier yang berurutan, yang disebut kode-kode LPC.

Kode-kode inilah yang nantinya digunakan untuk mengontrol rangkaian synthesizer seperti yang tampak pada Gambar 2.6. Rangkaian yang dirancang sebagai sebuah model dari alat ucap manusia ini, terdiri dari 3 bagian utama : excitation source, multistage digital filter dan D/A converter.

II. 1. 1. (b). ii. Formant analysis/synthesis

Formant analysis/synthesis mempunyai kesamaan dengan LPC karena keduanya berdasarkan pada spektrum frekwensi dari suara alami. Dan menggunakan jenis rangkaian synthesizer yang sama pula. Seperti tersirat

6) Ibid, hal. 146.

Gambar 2.6⁷⁾

LPC Speech-synthesizer

dalam namanya, metode ini menghasilkan suara dengan membentuk kembali formant yang terdapat pada gelombang suara.

Bunyi bersuara terdiri dari beberapa frekwensi resonansi yang disebut formant. Ketika kita berbicara, frekwensi formant berubah secara konstan menghasilkan bunyi yang berbeda. Karakteristik frekwensi formant dari gelombang suara dapat diberi kode secara digital dan digunakan untuk mengontrol generator frekwensi dan filter dari synthesizer elektronik sehingga didapatkan gelombang suara yang asli.

7) Ibid, hal 147.

II. 1. 2. ARTIFICIAL SPEECH CONSTRUCTION/SYNTHESIS

Teknik ini juga disebut sebagai Phoneme Speech Synthesis. Dengan phoneme synthesis, seluruh variasi dari bunyi bersuara (allophones) diberi kode dan disimpan dalam memori. Kode-kode dari allophones ini biasanya dalam format LPC dan disimpan dalam sebuah tabel look up ROM (ROM look up table). Kode alamat, yang disebut kode phoneme, akan memberi alamat kepada ROM, sehingga ROM dapat menerjemahkannya ke dalam parameter LPC dengan menggunakan tabel look up. Parameter LPC ini kemudian diumpankan ke sebuah rangkaian synthesizer LPC untuk mendapatkan suara allophonic.

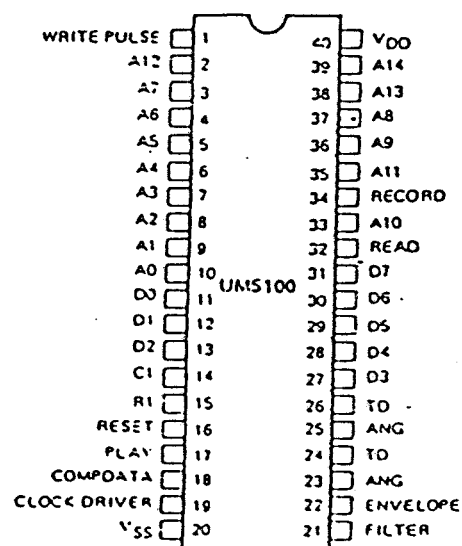
Sebuah phoneme speech synthesizer dapat digunakan dalam 2 cara : direct speech synthesis atau text-to-speech synthesis. Pada cara pertama, komputer diprogram untuk dapat menghasilkan urutan kode fonem yang akan diumpankan ke ROM synthesizer pada saat yang dikehendaki. Sebagai contoh, sebuah robot dapat diprogram untuk mengatakan " Low Voltage " jika baterenya perlu diisi kembali.

Penggunaan lain adalah untuk konversi dari teks ke suara. Seperti namanya, teks yang telah ditulis harus dapat diterjemahkan ke dalam kode fonem yang sesuai sehingga dapat dihasilkan suara. Salah satu aplikasi yang umum adalah sebagai alat bantu bagi tuna netra sehingga dapat membaca tanpa menggunakan huruf Braille.

II. 2. UM 5100 - VOICE PROCESSOR

UM 5100 adalah sebuah CMOS LSI pengolah suara produksi United Microelectronic Corporation. IC 40 pin ini termasuk jenis Natural Speech Analysis/synthesis. Lebih khusus lagi dapat digolongkan sebagai Time-domain Analysis/synthesis. Untuk mengurangi kebutuhan memori digunakan Modulasi Delta, yaitu Continuous Variabel Slope Delta Modulation.

IC UM 5100 mampu merekam dan mereproduksi suara dengan mengubah sinyal suara menjadi bentuk digital yang kemudian disimpan dalam Static ROM atau EPROM. Dengan 15 address line dan 8 data line, 32 K byte data dapat



Gambar 2. 7⁸⁾

Pin Configurations UM 5100

8) United Microelectronic Corporation, Data sheet IC UM 5100, Santa Clara, hal. 3-8.

diproses oleh IC ini. Gambar 2.7 menunjukkan pin-pin IC UM 5100.

Fungsi masing-masing pin adalah sebagai berikut :

- V_{DD}

Catu daya positif (3 - 6 V)

- V_{SS}

Ground

- A0 - A 14

Address bus

- D0 - D7

Data bus

- $\overline{\text{RECORD}}$

Input pin, active low. Dengan mentrigger pin ini, maka chip akan bekerja dalam mode speech analysis.

- $\overline{\text{WRITE PULSE}}$

Output pin, active low. Pin ini akan menghasilkan satu pulsa setiap 8 cycle clock, dan hanya aktif pada mode speech analysis.

- $\overline{\text{READ}}$

Output pin, active low. Pin ini merupakan signal kontrol bagi memori tambahan (external memory). Hanya aktif pada mode speech synthesis.

- $\overline{\text{PLAY}}$

Input pin, active low. Dengan mentrigger pin ini, maka chip akan bekerja dalam mode speech analysis.

- RESET

Input pin, active high. Jika diaktifkan seluruh internal counter diclearkan.

- ANG dan $\overline{\text{ANG}}$

Output sinyal suara analog dengan fase yang berlawanan.

- FILTER

Output pin. Bersama-sama dengan rangkaian integrator eksternal akan menghasilkan gelombang envelope.

- ENVELOPE

Input pin. Sinyal envelope akan masuk melalui pin ini untuk memodulasi amplitudo suara.

- TD dan $\overline{\text{TD}}$

Output pin. Output tambahan untuk sinyal frekwensi rendah.

- COMPDATA

Input pin. Mendeteksi Delta Slope yang dihasilkan dari perbandingan sinyal input dan feedbacknya.

- C, R

RC Oscillator pins,

$f = 40 \text{ KHz}$ untuk $R = 7,2 \text{ K}$ dan $C = 0,0047 \text{ uF}$.

- CLOCK DRIVER

Output pin penghasil tegangan negatif.

Keterangan lebih lengkap tentang IC ini dapat dilihat pada Lampiran.

II. 3. OPERATIONAL AMPLIFIER

Operational amplifier (Op Amp) merupakan IC yang

sangat populer dan paling banyak dipakai dalam berbagai keperluan. Penggunaan Op Amp antara lain adalah sebagai : penguat, integrator, comparator dan filter aktif.

II. 3. 1. OP AMP SEBAGAI PENGUAT

Salah satu pemakaian terpenting Op Amp adalah sebagai penguat. Penguat adalah suatu rangkaian yang menerima suatu sinyal pada input dan mengeluarkan sinyal yang tak berubah pada output dengan amplitudo yang lebih besar. Ada dua jenis penguat Op Amp dengan umpan balik negatif (Negative Feedback) yaitu :

a. Inverting Amplifier (Penguat Membalik)

b. Non Inverting Amplifier (Penguat Tak Membalik)

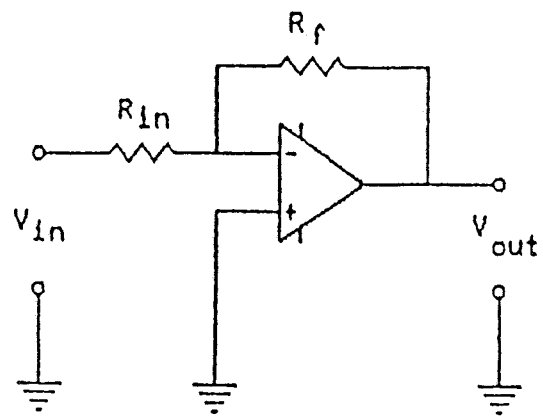
Untuk memudahkan analisa pada Op Amp dibuatlah suatu penyederhanaan, yaitu dengan menganggap bahwa tegangan antara terminal (+) dan terminal (-) berharga = 0 dan arus yang dialirkan oleh terminal (+) dan (-) diabaikan.

II. 3. 1. (a). Inverting Amplifier

Penguatan untuk gambar 2.8 didefinisikan sebagai berikut :

$$ACL = V_{out}/V_{in} = - R_f/R_{in} \dots \dots \dots (2.1)$$

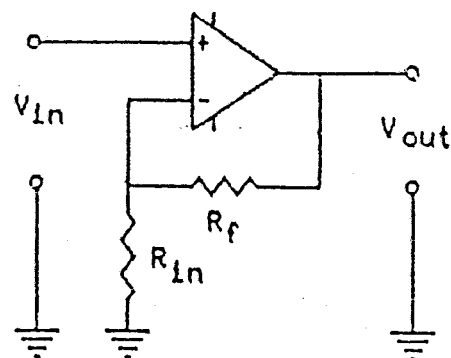
Tanda negatif pada rumus 2.1 menunjukkan bahwa sinyal output merupakan kebalikan dari sinyal input.



Gambar 2.8

Inverting Amplifier

II.3.1. (b). Non Inverting Amplifier



Gambar 2.9

Non Inverting Amplifier

Penguatan untuk Gambar 2.9 didefinisikan sebagai berikut :

$$A_{CL} = V_{out}/V_{in} = 1 + R_f/R_{in} \dots \dots \dots (2.2)$$

Dari rumus 2.2 tidak terlihat tanda negatif. Hal ini menunjukkan bahwa sinyal output sefase dengan sinyal

input, sehingga disebut Non Inverting Amplifier.

II. 3. 2. FILTER AKTIF

Filter adalah sebuah rangkaian yang dirancang agar melewati suatu pita frekwensi tertentu dan memperlemah semua sinyal di luar pita ini. Suatu filter dapat bersifat pasif atau aktif. Filter pasif tersusun dari komponen pasif seperti : resistor, kapasitor dan induktor. Filter aktif tersusun dari komponen aktif seperti : transistor atau Op Amp dan komponen pasif seperti : resistor dan kapasitor.

Ada 4 jenis filter, yaitu :

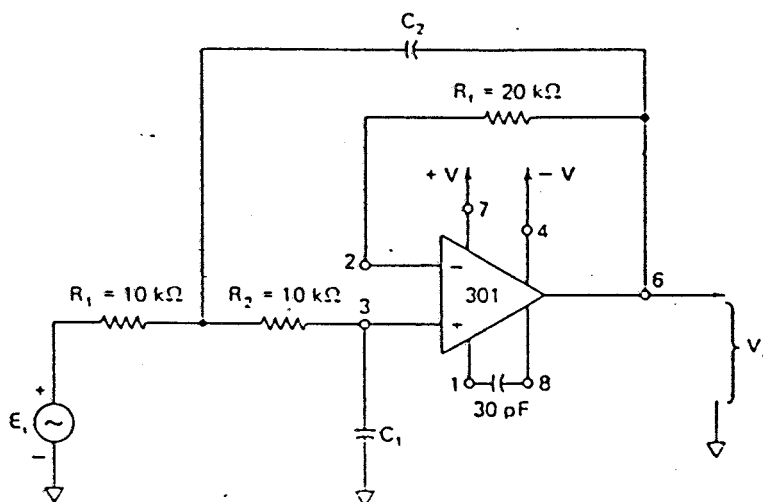
- Low - Pass Filter
- High - Pass Filter
- Band - Pass Filter
- Band - Reject Filter

Low - Pass Filter adalah sebuah rangkaian yang outputnya tetap pada frekwensi di bawah frekwensi cut-off, sedangkan untuk frekwensi di atas frekwensi cut-off sinyalnya diperlemah. High - Pass Filter merupakan kebalikan dari Low - Pass Filter, yaitu memperlemah sinyal di bawah frekwensi cut-off dan menghasilkan sinyal yang tetap di atas frekwensi cut-off. Band - Pass Filter melewati sinyal pada suatu pita frekwensi tertentu dan memperlemah semua sinyal di luar pita frekwensi tersebut. Band - Reject Filter merupakan kebalikan dari Band - Pass Filter, yaitu menolak sinyal pada pita frekwensi tertentu

dan melewatkan semua sinyal di luar pita tersebut.

Ada banyak jenis Low - Pass Filter, High - Pass Filter, Band - Pass Filter dan Band - Reject Filter, namun yang paling banyak digunakan adalah jenis Butterworth dan Chebyshev.

Dalam teori ini hanya akan dibahas prosedur perancangan High - Pass Filter orde dua (dengan landaian sebesar -40 dB/dekade) jenis Butterworth, karena filter inilah yang nantinya akan digunakan. Dengan landaian sebesar -40 dB/dekade maka penguatan pada filter ini akan turun sebesar 40 dB dengan naiknya ω sebesar 10 x. Gambar 2.10 menunjukkan rangkaian Low - Pass Filter yang



Gambar 2.10⁹⁾

Low - Pass Filter 40 dB/Dekade

9) R.F. Coughlin, F.P. Driscoll, H.W. Soemitro, 'Penguat Operational dan Rangkain Terpadu Lipier', Erlangga, 1985, hal. 267.

dimaksud. Harga-harga resistor dan kapasitor akan menentukan harga frekwensi cut-off filter tersebut.

Perencanaan suatu Low - Pass Filter dapat dilakukan dengan menggunakan prosedur sebagai berikut :

1. Tentukan frekwensi cut-off, ω_c atau f_c .
2. Buatlah harga $R_1 = R_2 = R$, dan pilih harga yang sesuai antara 10 KOhm dan 100 KOhm serta pilih $R_f = 2R$.
3. Hitung C_1 dengan rumus $C_1 = 0,707/(\omega_c \cdot R)$.
4. Pilih $C_2 = 2 \cdot C_1$.

Dengan menggunakan keempat prosedur tersebut, maka sebuah Low - Pass Filter dapat dirancang dengan lebih mudah dan lebih cepat.

II. 3. 3. COMPARATOR

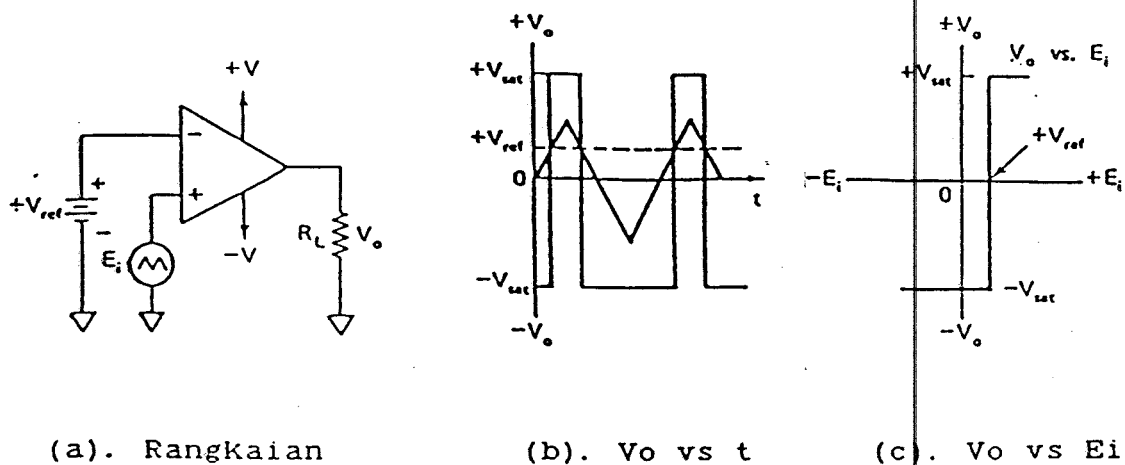
Sebuah comparator akan membandingkan tegangan sinyal input dengan tegangan acuan yang dipasangkan pada input yang lain. Gambar 2.11 memperlihatkan contoh pemakaian Op Amp sebagai comparator.

Bila $E_i > V_{ref}$, maka $V_o = +V_{sat}$; sedang bila $E_i < V_{ref}$, $V_o = -V_{sat}$. Apabila letak V_{ref} dan E_i dipertukarkan maka bila $E_i > V_{ref}$, $V_o = -V_{sat}$ sebaliknya bila $E_i < V_{ref}$, $V_o = +V_{sat}$.

II. 4. IBM PC XT

II. 4. 1. SLOT PADA IBM PC XT

IBM PC XT mempunyai 8 slot dengan konfigurasi pin yang sama. Semua slot dapat mengakses memori dan

Gambar 2.11⁽¹⁰⁾

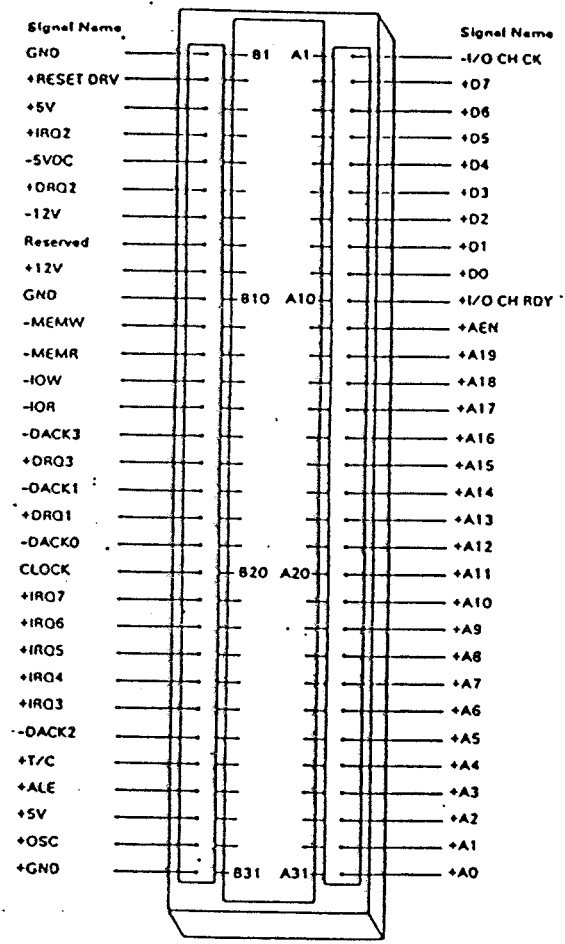
C o m p a r a t o r

input/output. Suatu interface dapat berkomunikasi dengan IBM PC XT melalui slot ini. Konfigurasi slot IBM PC XT yang merupakan konektor 62 terminal dapat dilihat pada Gambar 2.12. Untuk perencanaan sebuah interface tidak seluruh terminal digunakan.

Beberapa terminal yang biasanya digunakan untuk perencanaan sebuah interface antara lain adalah :

- AO-A19 : Saluran-saluran (lines) ini digunakan untuk mengaskes memori dan peralatan input/output dari suatu sistem. Kedua puluh saluran alamat (adres line) dapat mengakses memori sampai 1 Mb.

⁽¹⁰⁾ Ibid, hal. 20.



Gambar 2.12¹¹⁾
Slot IBM PC XT

11) A. Singh, V. A. Triebel, 'The 8088 Microprocessor', Singapore, PHI, 1989, hal. 418.

- DO-D7 : Saluran-saluran ini digunakan sebagai jalur data, baik oleh processor, memori atau peralatan input/output. DO merupakan Least Significant Bit (LSB), sedang D7 merupakan MSB (Most Significant Bit).
- $\overline{\text{IOR}}$: Sinyal ini digunakan untuk memerintahkan suatu peralatan input/output agar meletakkan datanya pada data bus. Sinyal $\overline{\text{IOR}}$ aktif pada logika rendah.
- $\overline{\text{IOW}}$: Sinyal ini digunakan untuk memerintahkan suatu peralatan input/output agar membaca data yang berada pada data bus. Sinyal $\overline{\text{IOW}}$ aktif pada logika rendah.
- $\overline{\text{MEMW}}$: Sinyal ini digunakan untuk menulis data dari data bus ke memori. Sinyal $\overline{\text{MEMW}}$ aktif pada logika rendah.
- $\overline{\text{MEMR}}$: Sinyal ini digunakan untuk membaca data dari memori. Sinyal $\overline{\text{MEMR}}$ aktif pada logika rendah.
- AEN : Singkatan dari Address Enable. Sinyal ini digunakan untuk membuat kontrol processor menjadi tidak aktif dan membuat DMA Controller mengambil alih operasi. Sinyal AEN aktif pada logika tinggi.

II. 4. 2. PETA PEMAKAIAN MEMORI

Dengan 20 saluran address, IBM PC XT dapat

mengakses sampai 1 Mb memori. Namun tidak semua alamat tersebut dapat digunakan untuk perencanaan interface, karena beberapa alamat sudah digunakan untuk berbagai peralatan, misalnya : RAM, Monitor dan Fixed Disk Control. Memory map dari IBM PC XT dapat dilihat selengkapnya pada Gambar 2.13.

II. 4. 3. DIAGRAM WAKTU SISTEM BUS

Suatu interface selalu melibatkan pemindahan data melalui sistem bus. Waktu yang diperlukan untuk pemindahan data dikenal dengan nama siklus bus (Bus Cycle). Pada IBM PC XT dikenal beberapa siklus pemindahan data. Pada teori ini hanya akan dibahas siklus yang berhubungan dengan interface yang direncanakan, yaitu Siklus Bus Pembacaan Port Input/Output (Input/Output Port Read Bus Cycle) dan Siklus Bus Penulisan Port Input/Output (Input/Output Port Write Bus Cycle).

II. 4. 3. (a). I/O PORT READ BUS CYCLE

Siklus ini terjadi pada saat processor mengeksekusi instruksi IN. Keseluruhan siklus ini memerlukan minimum 5 pulsa clock atau sekitar 1,05 mikrodetik. Selama siklus ini microprocessor mengirim alamat 16 bit (A0-A15), sedangkan 6 bit alamat terakhir tidak diaktifkan. Saat siklus T₁, ALE diaktifkan menunjukkan bahwa alamat A0-A15 adalah benar (valid). Pada siklus T₂, $\overline{\text{IOR}}$ diaktifkan

Start Address		Function
Decimal	Hex	
0	00000	64 to 256K Read/Write Memory on System Board
16K	04000	
32K	08000	
48K	0C000	
64K	10000	
80K	14000	
96K	18000	
112K	1C000	
128K	20000	
144K	24000	
160K	28000	
176K	2C000	
192K	30000	
208K	34000	
224K	38000	
240K	3C000	
256K	40000	Up to 384K Read/Write Memory in I/O Channel Up to 384K in I/O Channel
272K	44000	
288K	48000	
304K	4C000	
320K	50000	
336K	54000	
352K	58000	
368K	5C000	
384K	60000	
400K	64000	
416K	68000	
432K	6C000	
448K	70000	
464K	74000	
480K	78000	
496K	7C000	
512K	80000	
528K	84000	
544K	88000	
560K	8C000	
576K	90000	
592K	94000	
608K	98000	
624K	9C000	

Start Address		Function
Decimal	Hex	
640K	A0000	128K Reserved
656K	A4000	
672K	A8000	
688K	AC000	
704K	B0000	Monochrome
720K	B4000	
736K	B8000	Color/Graphics
752K	BC000	
768K	C0000	Fixed Disk Control
784K	C4000	
800K	C8000	192K Read Only Memory Expansion and Control
816K	CC000	
832K	D0000	
848K	D4000	
864K	D8000	
880K	DC000	
896K	E0000	Reserved
912K	E4000	
928K	E8000	
944K	EC000	
960K	F0000	48K Base System ROM
976K	F4000	
992K	F8000	
1008K	FC000	

Gambar 2.13^[2]

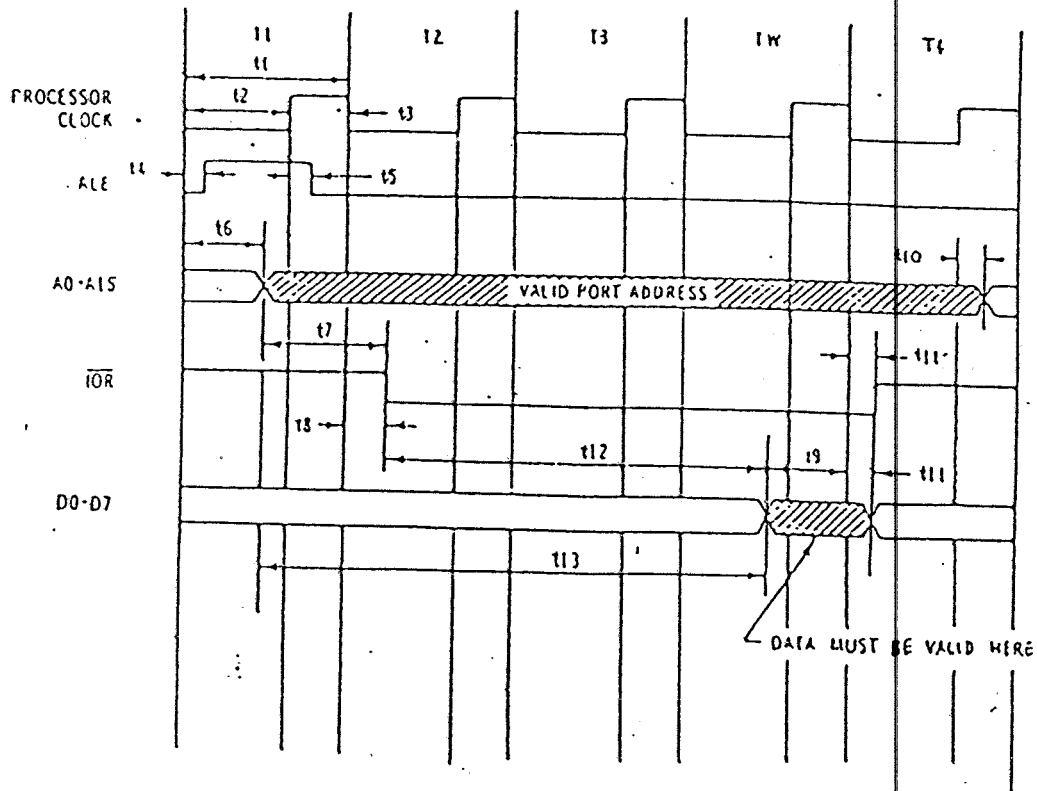
Memory Map IBM PC XT

[2] Ibid., hal. 383.

dan port siap mengirim data ke data bus. Pada permulaan T₄, processor mengambil data yang berada data bus dan selanjutnya sinyal $\overline{\text{IOR}}$ dinon-aktifkan. Siklus ini berakhir pada akhir T₄. Normalnya siklus ini hanya membutuhkan 4 pulsa clock, tetapi sebuah pulsa clock tambahan TW secara otomatis disisipkan pada setiap siklus ini. Diagram waktu I/O Port Read Bus Cycle ditunjukkan pada Gambar 2.14 dan pembagian waktu dalam satuan nanodetik siklus tersebut diperlihatkan oleh tabel 2.1.

II. 4. 3. (b). I/O PORT WRITE BUS CYCLE

Siklus ini terjadi pada saat processor mengeksekusi instruksi OUT. Siklus waktunya hampir sama dengan I/O Port Read Bus Cycle, perbedaannya hanya terletak pada siklus waktu T₂ yaitu saat processor mengeluarkan sinyal $\overline{\text{IOW}}$ dan data siap ditransfer ke salah satu alamat dari I/O port. Pada awal T₄, sinyal $\overline{\text{IOW}}$ dinon-aktifkan dan data yang benar masuk ke alamat I/O port. Sinyal TW secara otomatis juga disisipkan dalam siklus ini. Diagram waktu I/O Port Write Bus Cycle ditunjukkan pada Gambar 2.15 dan pembagian waktu siklus ini (dalam nanodetik) ditunjukkan pada Tabel 2.2.

Gambar 2. 14¹²⁾

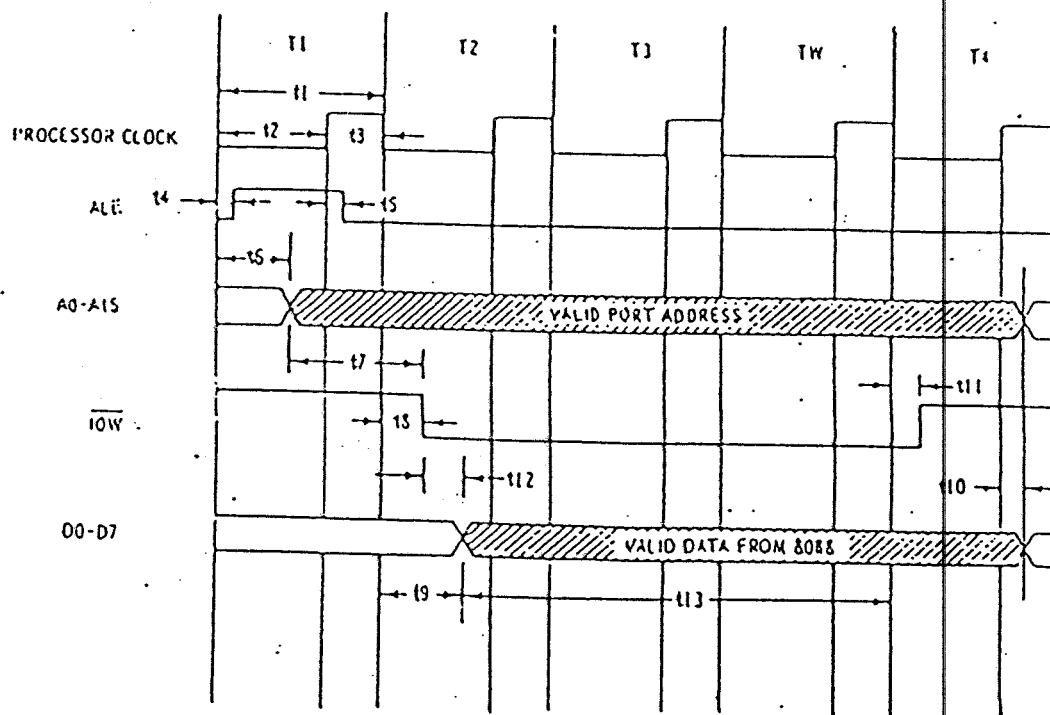
I/O Port Read Bus Cycle

12) L. C. Eggebrecht, 'Interfacing to the IBM PC', Indiana, Howard W. Sams & Co., Inc, 1983. hal. 64.

Tabel 2.1¹³⁾
I/O Port Read Bus Cycle Timings

S I M B O L	M A X	M I N
t1	—	209.5
t2	—	124.5
t3	—	71.8
t4	15	—
t5	15	—
t6	128	16
t7	—	91.5
t8	35	10
t9	—	42
t10	—	10
t11	35	10
t12	—	551.5
t13	—	668

13) Ibid.

Gambar 2. 15⁽¹⁴⁾

I/O Port Write Bus Cycle

(14) Ibid, hal. 65.

Tabel 2.2¹⁵⁾

I/O Port Write Bus Cycle Timings

S I M B O L	M A X	M I N
t1	—	209.5
t2	—	124.5
t3	—	71.8
t4	15	—
t5	15	—
t6	128	16
t7	—	91.5
t8	35	10
t9	122	14
t10	—	10
t11	35	10
t12	—	—
t13	—	506.5

15) Ibid.

B A B III

SINTESA SUARA DENGAN IC UM 5100

III. 1. PENDAHULUAN

Sintesa suara yang direncanakan terdiri dari dua rangkaian yaitu rangkaian sintesa suara dengan IC UM 5100 dan rangkaian interface. Dengan adanya interface, rangkaian sintesa suara dapat dikontrol oleh komputer yang dalam hal ini adalah komputer IBM PC XT. Selain itu interface memungkinkan dilakukannya perluasan pemakaian sintesa suara, misalnya : data suara dapat disimpan dalam floppy disk atau dapat juga dilakukan konversi dari teks ke suara.

III. 2. RANGKAIAN SINTESA SUARA

III. 2. 1. BLOK DIAGRAM

Diagram blok rangkaian sintesa suara yang menggunakan komponen utama IC UM 5100 terlihat pada Gambar 3.1. Rangkaian ini terdiri dari beberapa rangkaian, yaitu :

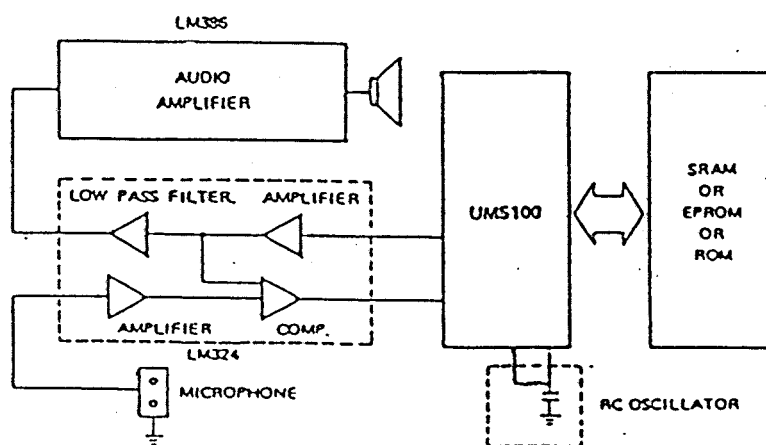
- a. Penguat (Amplifier)
- b. Comparator
- c. Low - Pass Filter
- d. Penguat Audio (Audio Amplifier)

yang menggunakan 2 buah IC Op Amp : LM 324 dan LM 386. Sebagai media perekam dapat digunakan Statis RAM, EPROM, atau ROM. Untuk rangkaian yang dibuat digunakan 2 buah Static RAM, yaitu RAM jenis 6116 yang mempunyai kapasitas

sebesar 2 KB.

Pada saat record, yaitu pada saat memasukkan data suara melalui microphone ke RAM, yang terjadi adalah : Sinyal suara yang telah dirubah menjadi sinyal listrik oleh microphone, diperkuat oleh Amplifier. Oleh Comparator sinyal ini dibandingkan dengan keluaran dari IC UM 5100. Hasil perbandingan inilah yang kemudian diolah oleh IC UM 5100 dan kemudian disimpan di dalam RAM.

Sebaliknya pada saat play, data dari RAM diambil oleh IC UM 5100 dan dirubah kembali menjadi sinyal analog. Oleh Low - Pass Filter komponen frekuensi tinggi dari sinyal ini dihilangkan sebelum akhirnya diumpankan ke speaker sehingga dapat didengar.



Gambar 3.1¹⁾

Blok Diagram Rangkaian Sintesa Suara

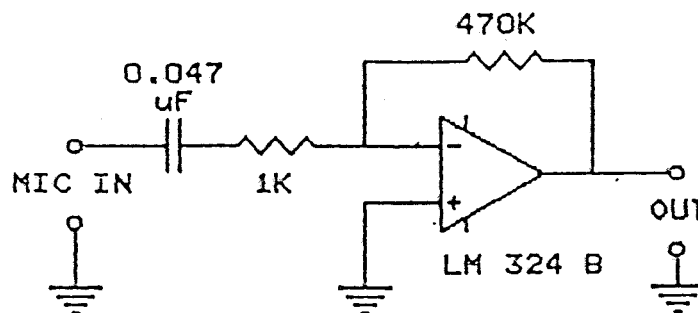
1) United Microelectronic Corporation, Data sheet IC UM 5100, Santa Clara, hal. 3-10.

III. 2. 2. PENGUAT

Ada 2 buah penguat yang digunakan. Penguat I yang menggunakan 1/4 LM 324 merupakan sebuah Inverting amplifier yang dipakai untuk memperkuat sinyal dari microphone. Rangkaian penguat I ditunjukkan pada Gambar 3.2. Dengan menggunakan rumus 2.1 (Bab II Teori Penunjang) dapat dihitung besar penguatannya, yaitu

$$\begin{aligned}ACL &= - R_f/R_{in} \\ &= - 470K/1K = 53,44 \text{ dB}\end{aligned}$$

Capacitor 0,047uF merupakan sebuah coupling capacitor.



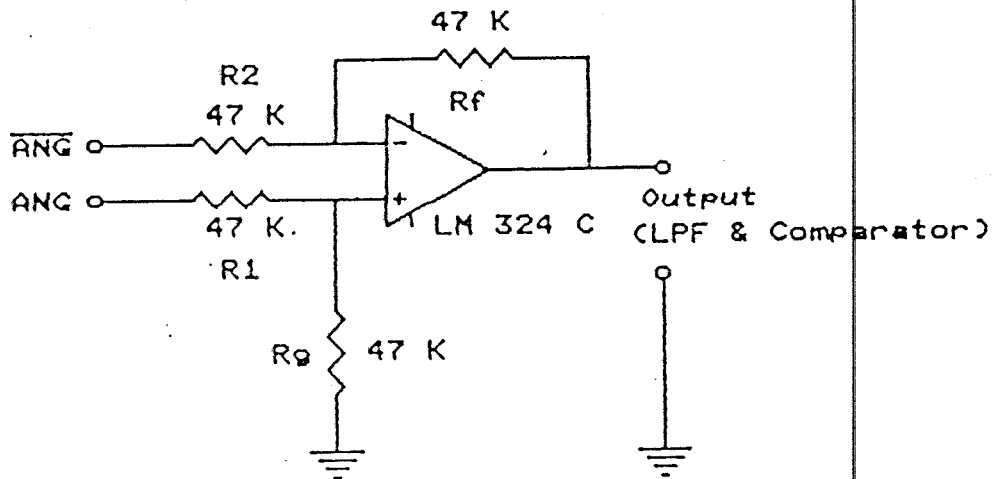
Gambar 3.2¹⁾

Rangkaian Penguat I

Sedangkan penguat II berfungsi untuk memperkuat output dari IC UM 5100 sebelum diumpankan ke Low - Pass Filter. Penguat ini merupakan sebuah Voltage Differential Amplifier dan juga menggunakan 1/4 LM 324. Gambar 3.3

2) Ibid, hal. 3-13.

menunjukkan rangkaian tersebut. Karena harga $R_1=R_2=R_f=R_g$ maka besar tegangan outputnya merupakan selisih antara tegangan input pada terminal (+) dan tegangan pada terminal (-).



Gambar 3. 3³⁾

Rangkaian Penguat II

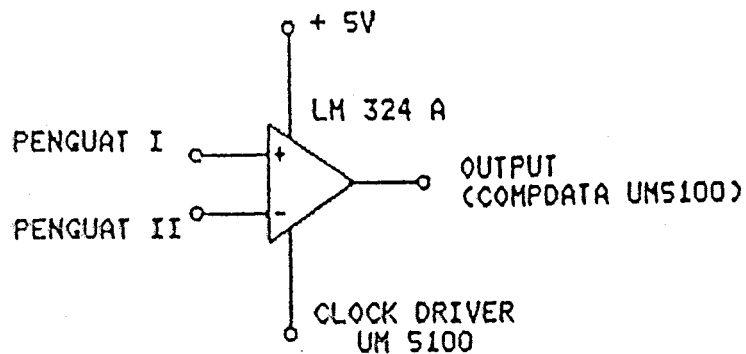
III. 2. 3. COMPARATOR

Comparator berfungsi untuk membandingkan sinyal input (output dari penguat I) dan sinyal umpan balik (ouput dari penguat II). Rangkaiannya terlihat pada Gambar 3. 4. Comparator ini menggunakan juga 1/4 LM 324.

Pada gambar di atas, bila input pada terminal (+) lebih besar dari input pada terminal (-), maka outputnya akan berlogika 1, sebaliknya bila input terminal (+) lebih kecil dari input pada terminal (-) akan dihasilkan

3) Ibid

output dengan logika 0.



Gambar 3. 4⁴⁾

Rangkaian Comparator

III. 2. 4. LOW - PASS FILTER

Low - pass filter (LPF) berfungsi melewatkan sinyal suara di bawah frekuensi cut-off dan meredam sinyal di atas frekuensi tersebut. LPF dipasang diantara penguat II dan penguat audio. Pemasangan LPF dimaksudkan untuk menghilangkan komponen frekuensi tinggi sinyal output UM 5100 dan melewatkan komponen frekuensi rendah yang merupakan sinyal suara.

Rangkaian lengkap dari LPF terlihat pada Gambar 3.5. Filter ini merupakan jenis Butterworth dengan landaian sebesar -40 dB/dekade. Filter ini dirancang dengan frekwensi cut off sebesar 3 Khz. Dengan

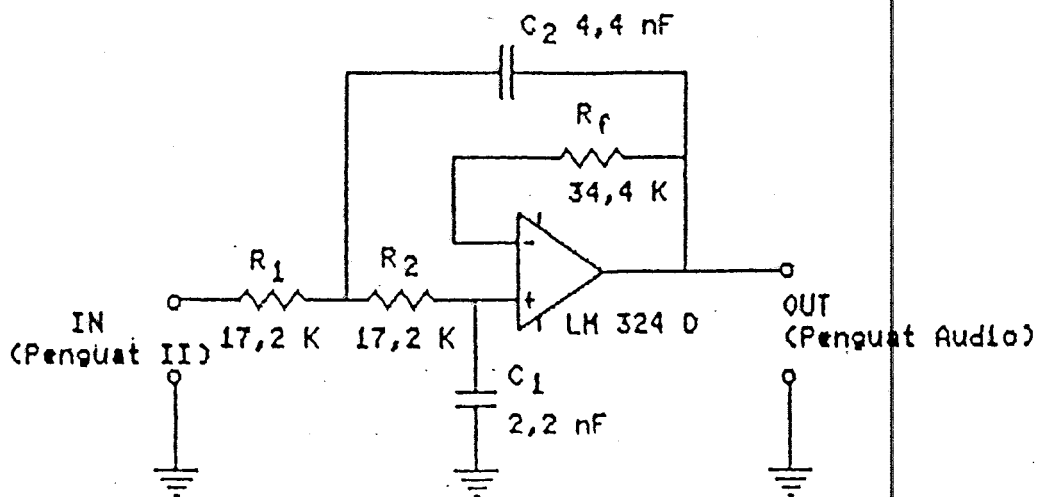
4) Ibid

menggunakan prosedur yang telah dibahas pada Bab II (Teori Penunjang) dapat dicari harga-harga kapasitor dan tahanan yang diperlukan.

Setelah beberapa kali dipilih harga R_1 untuk menentukan besarnya kapasitor, maka dengan mempertimbangkan persediaan di pasaran, akhirnya ditentukan harga $R_1 = R_2 = R = 17,2 \text{ K}$. Sehingga $R_f = 2R = 34,4 \text{ K}$. Sedang harga C_1 dapat dihitung dengan persamaan berikut :

$$\begin{aligned} C_1 &= 0,707 / (\omega_c \cdot R) \\ &= 0,707 / (2 \times 3,14 \times 3000 \times 17200) \\ &= 2,2 \text{ nF} \end{aligned}$$

Karena $C_2 = C_1$, maka $C_2 = 4,4 \text{ nF}$.

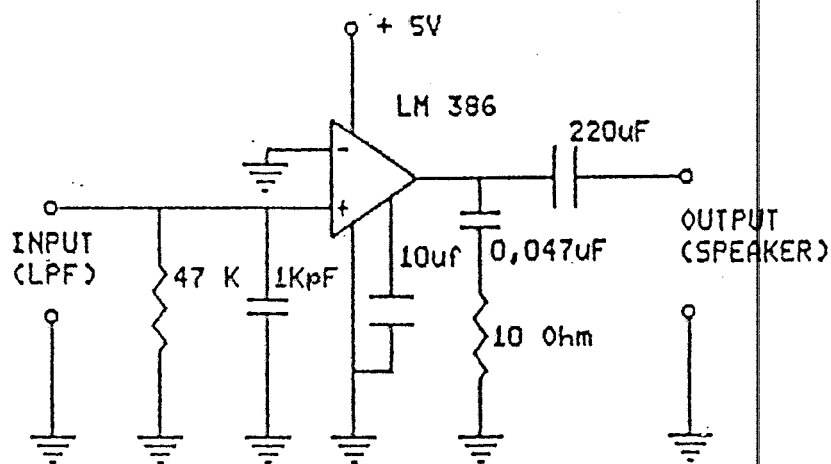


Gambar 3. 5.

Rangkaian Low - Pass Filter

III. 2. 5. PENGUAT AUDIO

Penguat audio berfungsi menguatkan sinyal suara output dari LPF sehingga dapat didengar melalui loudspeaker. Rangkaian ini menggunakan Op Amp jenis LM 386. Gambar 3.6 menunjukkan rangkaian tersebut.



Gambar 3.6⁵⁾

Rangkaian Penguat Audio

C 10 uF yang dipasang antara pin 7 dan ground merupakan bypass capacitor. Penguatan dari Penguat ini ditentukan oleh tahanan yang dipasang antara pin 1 dan 8, dengan hanya memasang kapasitor pada pin-pin tsb penguatannya adalah sebesar 46 dB.⁶⁾ Sedang kapasitor 0,047 uF yang diseri dengan tahanan 10 ohm berguna

5) Ibid.

6) Dennis Bohn, 'Audio Handbook', Santa Clara, National Semiconductor Corp., 1976, hal. 4-31.

untuk mengurangi bottom side fuzzies yaitu osilasi yang terjadi selama ayunan negatif pada arus beban yang tinggi.

III. 3. RANGKAIAN INTERFACE

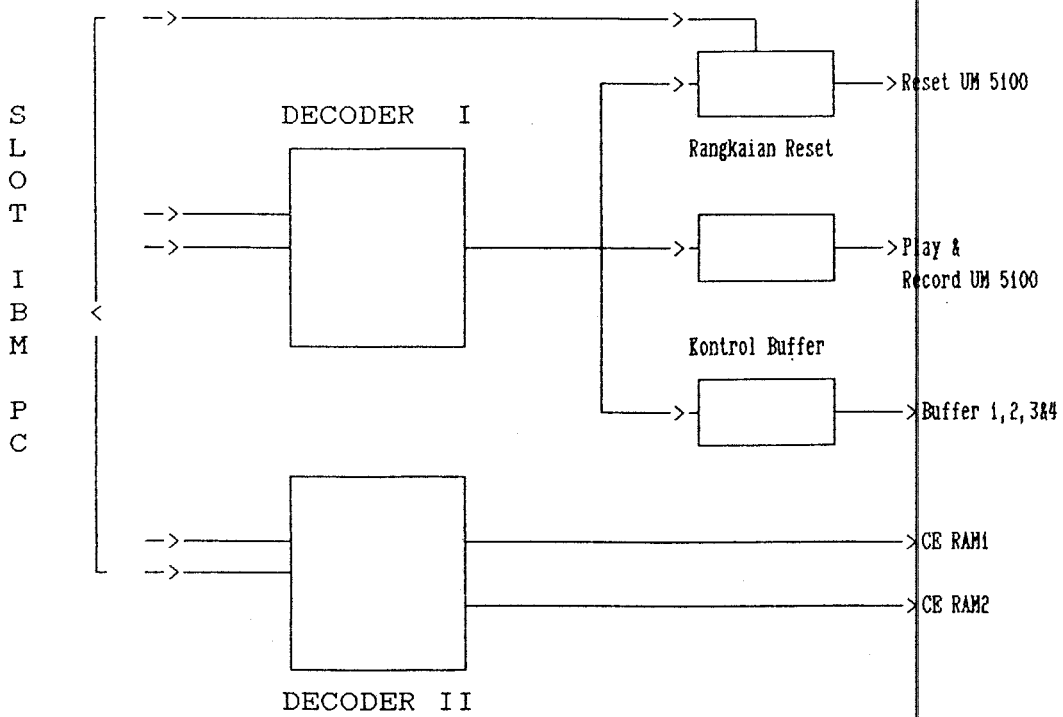
III. 3. 1. PENDAHULUAN

Rangkaian interface yang dibuat merupakan sebuah memory interface. Selain untuk mentransfer data dari memori rangkaian sintesa suara ke IBM PC XT, interface yang direncanakan dapat juga mengontrol play dan record rangkaian sintesa suara. Diagram blok dari rangkaian interface ditunjukkan pada Gambar 3.7 sedang hubungan antara interface, memori luar dan UM 5100 terlihat pada Gambar 3.8.

Pada saat sintesa suara dioperasikan dalam mode normal artinya tidak dihubungkan dengan memori komputer, maka buffer 1 dan 2 yang diaktifkan, sedang buffer 3 dan 4 dinonaktifkan. Sebaliknya bila akan dilakukan transfer data (pada saat menyimpan data di floppy disk, misalnya) maka buffer 3 dan 4 lah yang diaktifkan sedang buffer 1 dan 2 dinonaktifkan.

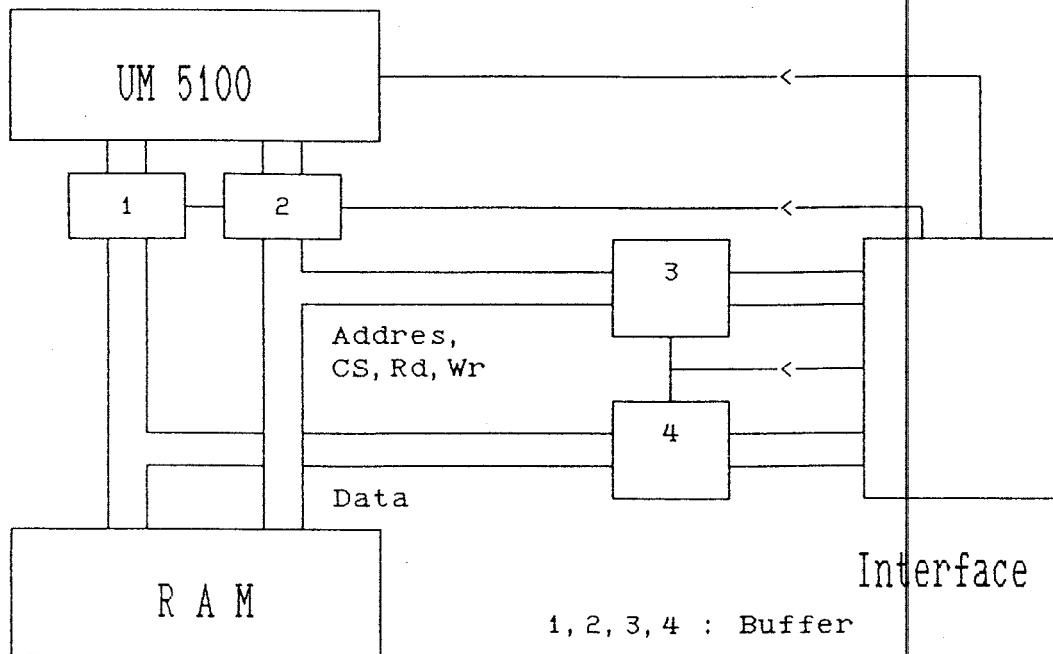
III. 3. 2. DASAR PERENCANAAN

Interface ini dirancang dengan memanfaatkan keadaan pada saat komputer mengeksekusi instruksi IN atau OUT. Seperti yang telah dijelaskan pada Bab I Teori Penunjang,



Gambar 3.7

Diagram Blok Rangkaian Interface



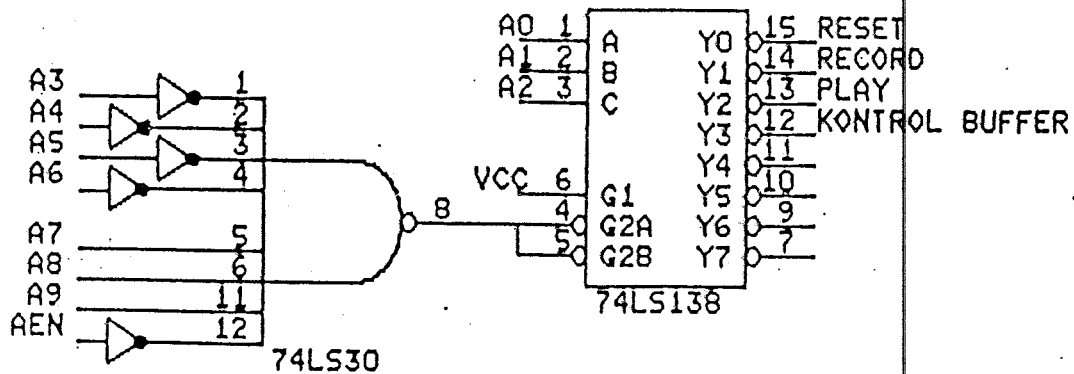
Gambar 3.8

Hubungan antara Interface, Memori Luar dan UM 5100

pada saat instruksi tersebut dieksekusi, microprocessor mengirim alamat 16 bit (A0 - A15) selama kurang lebih 800 nano detik. Keadaan inilah yang digunakan untuk mengaktifkan decoder I yang kemudian akan mengontrol rangkaian-rangkaian selanjutnya.

III. 3. 3. DECODER I

Decoder yang merupakan bagian terpenting dari interface ini terdiri dari IC 74LS138, 74LS30, dan 74LS04. Rangkaiannya terlihat pada Gambar 3.9. Decoder lah yang akan mengontrol play, record dan reset dari UM 5100 serta akan mengontrol pula rangkaian Kontrol Buffer.



Gambar 3.9

Rangkaian Decoder I

Decoder ini menggunakan alamat 380 - 383 yang

merupakan alamat I/O port yang tidak dipakai oleh Komputer. Susunannya adalah sebagai berikut :

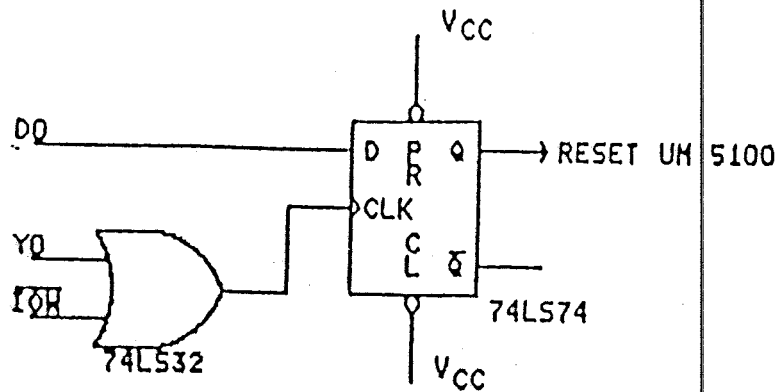
380 : Reset
381 : Record
382 : Play
383 : Kontrol Buffer

Untuk mengaktifkan alamat-alamat tersebut dapat digunakan instruksi IN/OUT (Bahasa Assembly) ke alamat yang sesuai, atau kalau dalam Pascal (yang dipakai oleh penulis), digunakan instruksi port, misalnya :

```
port[$380] := $2, atau  
$2 := port[$381]
```

III. 3. 4. RANGKAIAN RESET

Karena untuk mereset UM 5100 diperlukan lebar pulsa minimum 1 mikro detik, sedangkan dengan instruksi IN/OUT hanya didapatkan lebar pulsa \pm 800 nano detik, maka ditambahkan rangkaian reset seperti yang ditunjukkan pada Gambar 3.10. Pin D dari D Flip flop dihubungkan dengan DO dari komputer, hal ini dimaksudkan agar bisa diatur waktu set dan reset. Bila DO = 1 maka Q = 1 (pada keadaan ini UM 5100 akan reset), sebaliknya bila DO = 0 maka Q = 0. Clock dari D flip flop dihubungkan dengan YO dan $\overline{\text{IOW}}$ melalui OR gate, sehingga flip flop hanya aktif bila ada instruksi dengan alamat yang benar (380).



Gambar 3.10

Rangkaian Reset

Untuk reset instruksinya adalah sebagai berikut :

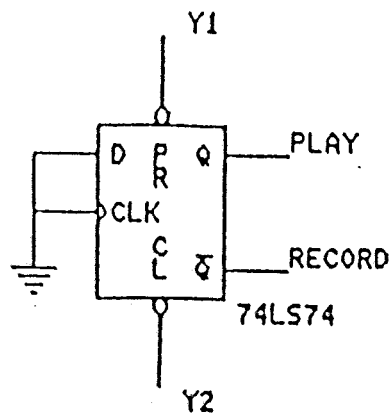
```
port[$380] := $1
```

sedang untuk set, instruksinya adalah :

```
port[$380] := $0
```

III. 3. 5. RANGKAIAN PLAY RECORD

Sama seperti rangkaian reset, rangkaian play record juga menggunakan sebuah D flip flop. Gambar 3.11 memperlihatkan rangkaian tersebut. Terlihat bahwa pin D dan CLK dihubungkan dengan ground, dengan demikian keadaan output Q hanya tergantung dari keadaan PR dan CL. Bila PR = 0, maka Q = 0, keadaan yang sama terjadi bila CL = 0. Dengan memberikan alamat port yang kita inginkan, kita dapat mengaktifkan play/record.

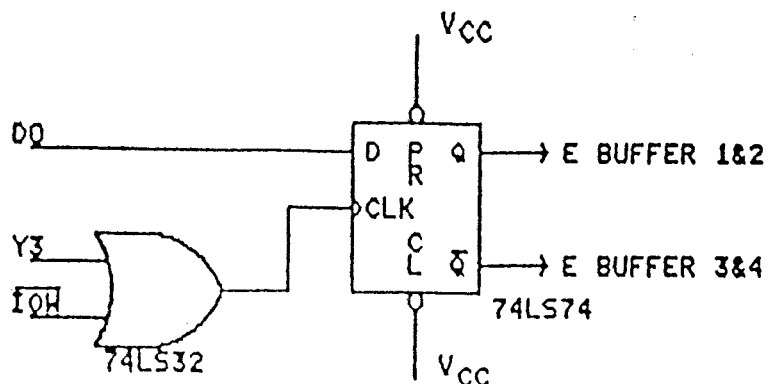


Gambar 3.11

Rangkaian Play Record

III. 3. 6. KONTROL BUFFER

Rangkaian yang digunakan untuk mengaktifkan buffer identik dengan rangkaian reset, hanya alamat yang dipilih berbeda. Untuk reset dipakai alamat 380, sedang untuk kontrol buffer dipakai alamat 383. Gambar 3.12 menunjukkan rangkaian kontrol buffer.



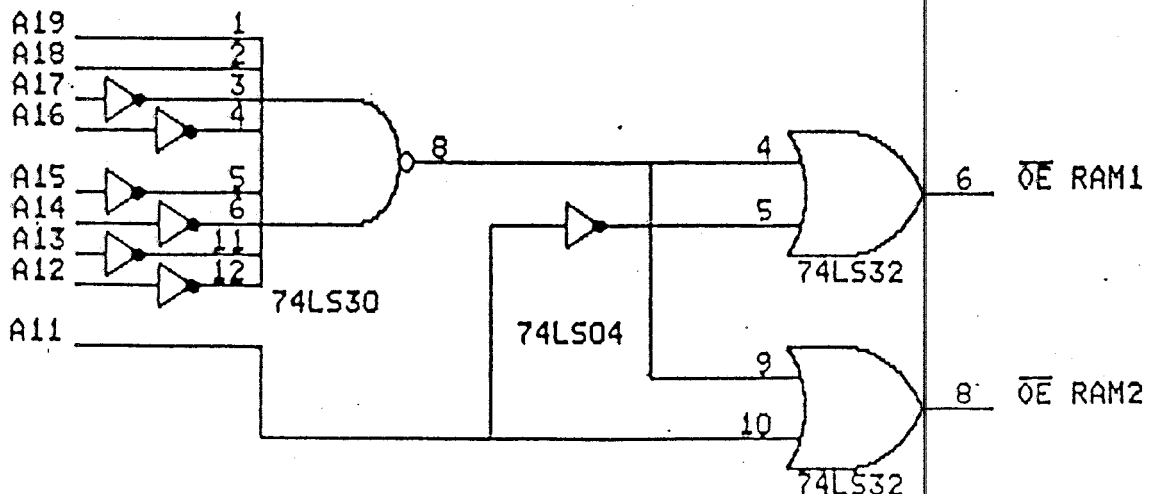
Gambar 3.12

Kontrol Buffer

III. 3. 7. DECODER II

Rangkaian decoder II tampak pada Gambar 3.13. Decoder II digunakan untuk mengaktifkan memori luar pada saat terjadi transfer data. Alamat awal yang dipakai adalah C0000 yang merupakan alamat memori yang tak terpakai. Karena digunakan 2 buah memori jenis 6116 (2 KB), maka alamat akhirnya adalah COFFF. Dalam bilangan biner dapat dituliskan sebagai berikut :

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
1	1	0	0	0	0	0	0	0	0	C0000
1	1	0	0	0	0	0	0	0	1	CO7FF
1	1	0	0	0	0	0	0	1	0	CO800
1	1	0	0	0	0	0	0	1	1	COFFF



Gambar 3. 13

Rangkaian Decoder II

Berdasarkan hal tersebut kita dapat mendisain decoder yang diperlukan untuk mengaktifkan kedua buah RAM, yaitu dengan memanfaatkan keadaan dari A_{11} . Pada 2K pertama (C000 - C7FF) $A_{11} = 0$ sehingga CS Memori I = 0 (Memori I aktif). Sedang pada 2K berikutnya karena $A_{11} = 1$, maka Memori II lah yang aktif. Decoder II dibentuk dari gerbang NOT (74LS04), NAND (74LS30) dan OR (74LS32).

B A B I V

PERENCANAAN PERANGKAT LUNAK

IV. 1. PENDAHULUAN

Pada bab ini akan dibahas perencanaan dari perangkat lunak yang digunakan untuk mengontrol rangkaian sintesa suara. Pembahasan antara lain meliputi instruksi-instruksi (dalam bahasa TURBO PASCAL release 5.5) yang banyak dipakai, flow chart dari masing-masing procedure dan penjelasan dari flow chart tersebut. Program lengkap dari masing-masing procedure akan disertakan pada lampiran.

Procedure yang akan dibahas antara lain :

- Procedure MENU - UTAMA
- Procedure PLAY
- Procedure REKAM
- Procedure SIMPAN
- Procedure PANGGIL dan
- Procedure TEKS KE SUARA

IV. 2. INSTRUKSI-INSTRUKSI YANG DIPAKAI

IV. 2. 1. Instruksi PORT

Instruksi ini digunakan untuk membaca dari dan menulis pada port dengan alamat tertentu. Instruksi ini akan banyak dipakai karena perangkat keras yang dirancang memanfaatkan port dari IBM PC.

Dalam Turbo Pascal release 5.5 dikenal dua macam instruksi Port, yaitu Port dan PortW. Alamat dari port

tergantung pada indeks yang diberikan. Jenis indeksnya adalah integer. Komponen dari Port adalah byte, sedangkan komponen dari PortW adalah word. Contoh penggunaan instruksi Port dan PortW adalah sebagai berikut :

```
Port[$380] := $2;
```

```
Data := PortW[$382];
```

Pada contoh pertama, \$2 akan dioutputkan pada port dengan alamat \$380, sedangkan pada contoh kedua, data yang ada pada port dengan alamat \$382 akan disimpan pada variabel Data.

IV. 2. 2. Instruksi Mem, MemW, dan MemL

Turbo Pascal menyediakan tiga macam instruksi untuk pengaksesan data langsung dari memori, yaitu : Mem, MemW, dan MemL. Komponen dari masing-masing instruksi adalah byte, word, dan longint. Instruksi ini memungkinkan pengaksesan data dari memori pada alamat tertentu. Alamat dari memori ditunjukkan oleh dua buah ekspresi integer jenis word yang dipisahkan oleh titik dua (:), ekspresi pertama menunjukkan segmen, sedang yang kedua menunjukkan offset dari alamat. Berikut adalah contoh penggunaan instruksi tersebut :

```
Mem[$0040:$0049] := 7;
```

```
Data := MemW[Seg(V):Ofs(V)];
```

```
MemLong := MemL[64:3*4];
```

Statemen pertama menyimpan nilai 7 dalam byte pada \$0040:\$0049. Contoh kedua memindahkan nilai word yang

disimpan pada 2 bytes pertama dari variabel V ke dalam variabel Data. Statemen Ketiga memindahkan nilai longint yang disimpan pada alamat \$0040:\$000C ke dalam variabel MemLong.

Pada program yang dipakai untuk mengontrol rangkaian sintesa suara ini, instruksi Mem digunakan pada saat terjadi transfer data dari RAM luar ke memori IBM atau sebaliknya.

IV. 2. 3. Instruksi-instruksi untuk pengolahan file

Karena nantinya data suara akan disimpan di disket, maka tentu saja akan banyak diperlukan instruksi untuk pengolahan file. Instruksi-instruksi yang dipakai adalah:

a. Assign

Fungsi : menempatkan nama dari file eksternal pada sebuah variabel file.

Bentuk : `Assign(var f; nama:string)`

Contoh : `var f : text;`

`begin`

`Assign(f, '');`

`Rewrite(f);`

`Writeln(f, 'standart output...');`

`Close(f);`

`end.`

b. Close

Fungsi : Menutup sebuah file yang terbuka.

Bentuk : Close(var f)

Contoh : var f : file

```
begin
```

```
    Assign(f, 'AUTOEXEC. BAT');
```

```
    Reset(f, 1);
```

```
    Writeln('File size = ', FileSize(f));
```

```
    Close(f);
```

```
end.
```

c. Reset

Fungsi : Membuka file yang sudah ada.

Bentuk : Reset(var f)

Contoh : var f, h : text;

```
begin
```

```
    Assign(f, FileName);
```

```
    Reset(f);
```

```
    Read(f, h);
```

```
    Close(f);
```

```
end;
```

d. Rewrite

Fungsi : Membuat dan membuka file baru.

Bentuk : Rewrite(var f)

Contoh : var f: text;

```
begin
```

```
    Assign(f, 'NEWFILE. $$$');
```

```
    Rewrite(f);
```

```
Write(f, 'Just created file with this text in  
it...');  
Close(f);  
end.
```

e. Read

Fungsi : Membaca sebuah komponen file dan menempatkan pada sebuah variabel.

Bentuk : Read(f, v1 [,v2,...,vn])

Contoh : Lihat Contoh pada Instruksi Reset

f. Write

Fungsi : Menulis sebuah variabel ke dalam sebuah komponen file.

Bentuk : Write(f, v1 [,v2,...,vn])

Contoh : Lihat Contoh pada Instruksi Rewrite

IV. 3. PROCEDURE MENU UTAMA

Procedure menu akan memberikan pilihan-pilihan yang ada pada pemakai, yaitu :

1. Play
2. Rekam
3. Simpan
4. Panggil
5. Teks ke Suara

Pilihan Play memungkinkan pemakai mem-playback suara yang tersimpan di dalam RAM. Pilihan Rekam digunakan apabila pemakai ingin memasukkan suara untuk selanjutnya disimpan

di dalam RAM. Dengan adanya pilihan Ketiga yaitu Simpan, data suara yang ada di RAM dapat disimpan ke dalam disket. Sebaliknya dengan Panggil, kita dapat memanggil kembali data suara yang disimpan dalam bentuk file di disket untuk kemudian di-playback .

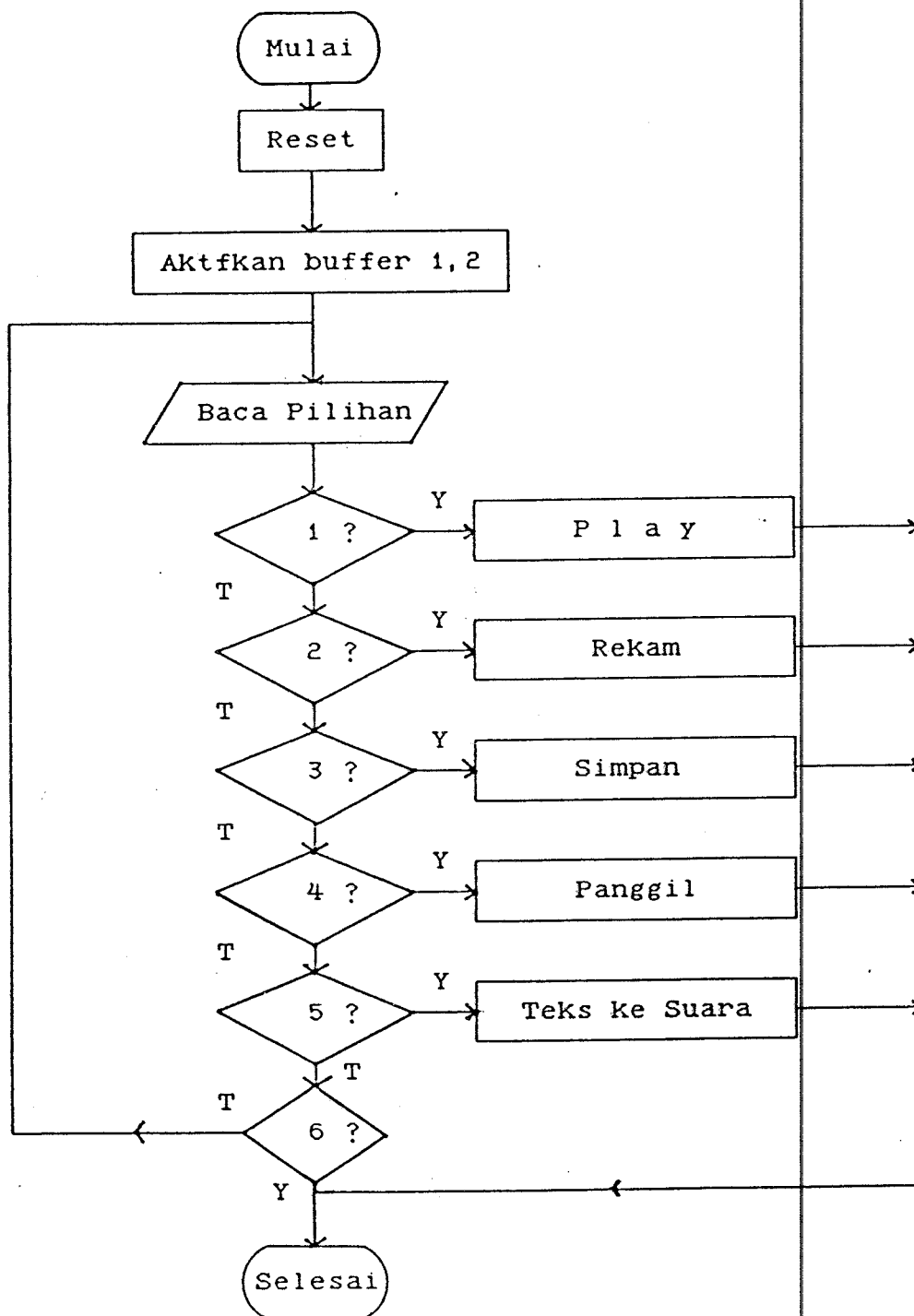
Pilihan kelima (Teks ke Suara), memungkinkan pemakai melakukan konversi dari teks ke suara. Tentu saja data suara (dalam hal ini satu kata) harus telah disimpan terlebih dahulu di disket.

Pada procedure menu ini juga ditempatkan instruksi untuk mereset UM 5100 yaitu `Port[$380] := $1`, dan instruksi untuk mengaktifkan Buffer 1,2 (buffer antara UM 5100 dan RAM) yaitu `Port[$383] := $0`. Flowchart dari procedure ini dapat dilihat pada Gambar 4.1.

IV. 4. PROCEDURE PLAY

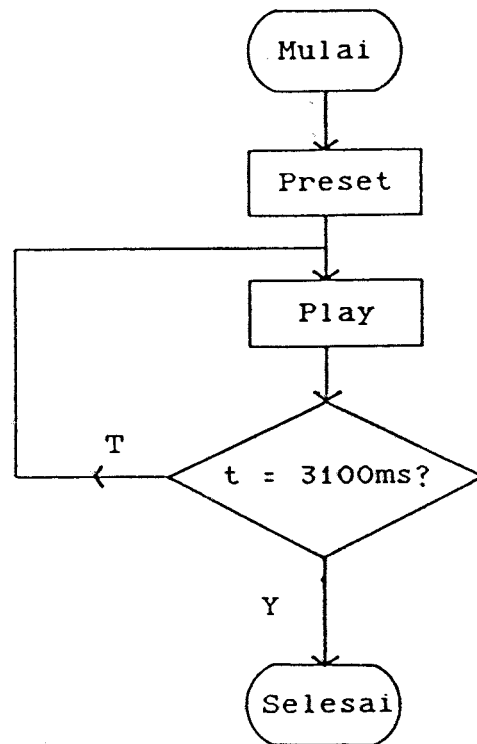
Procedure ini digunakan untuk mengaktifkan UM 5100 pada mode play, yaitu untuk memplay-back suara yang sudah tersimpan di RAM. Sebelum perintah play (`Port[$380] := $2`), UM 5100 akan dipreset terlebih dahulu dengan instruksi `Port[$380] := $0`.

Berdasarkan percobaan yang penulis lakukan, dengan menggunakan RAM sebesar 4 Kbyte dapat disimpan kata-kata sepanjang 3100 ms (3 kata). Karena itu pada procedure play akan diberikan instruksi play selama waktu tersebut. Gambar 4.2 memperlihatkan flowchart dari procedure ini.



Gambar 4.1

Flowchart Procedure Menu Utama



Gambar 4.2

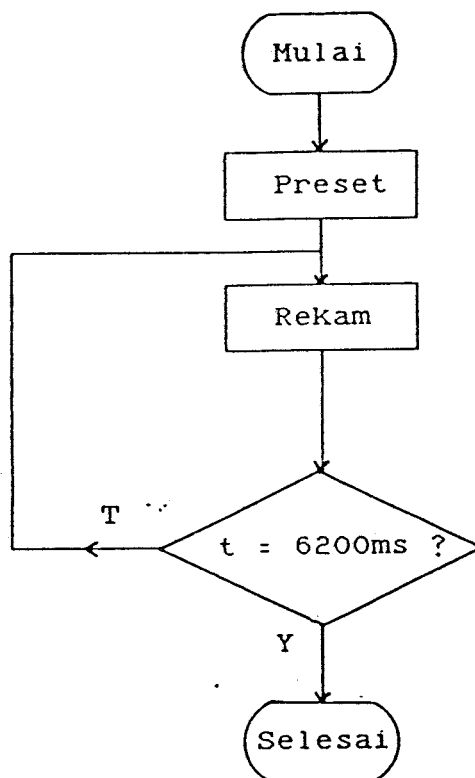
Flowcart Procedure Play

IV. 5. PROCEDURE REKAM

Bila kita ingin memasukkan data suara ke dalam RAM, maka untuk mengaktifkan UM 5100 kita harus memicu pin record dari IC tersebut. Karena adanya rangkaian interface, kita dapat melakukan hal itu secara perangkat lunak yaitu dengan menggunakan procedure Rekam. Sama seperti procedure play, sebelum diberikan intruksi rekam (`Port[$382] := $2`) rangkaian UM 5100 akan dipreset terlebih dahulu.

Pada mode record, setelah merekam data suara ke dalam RAM, UM 5100 akan langsung memplay-back suara yang disimpan, sehingga kita dapat langsung mengetahui

bagaimana suara yang dihasilkan, dan dapat melakukan perekaman ulang apabila dianggap kurang sempurna. Oleh karena itu delay untuk procedure Rekam diberikan dua kali dari procedure play (6200 ms). Flowchart procedure Rekam terdapat pada Gambar 4.3.

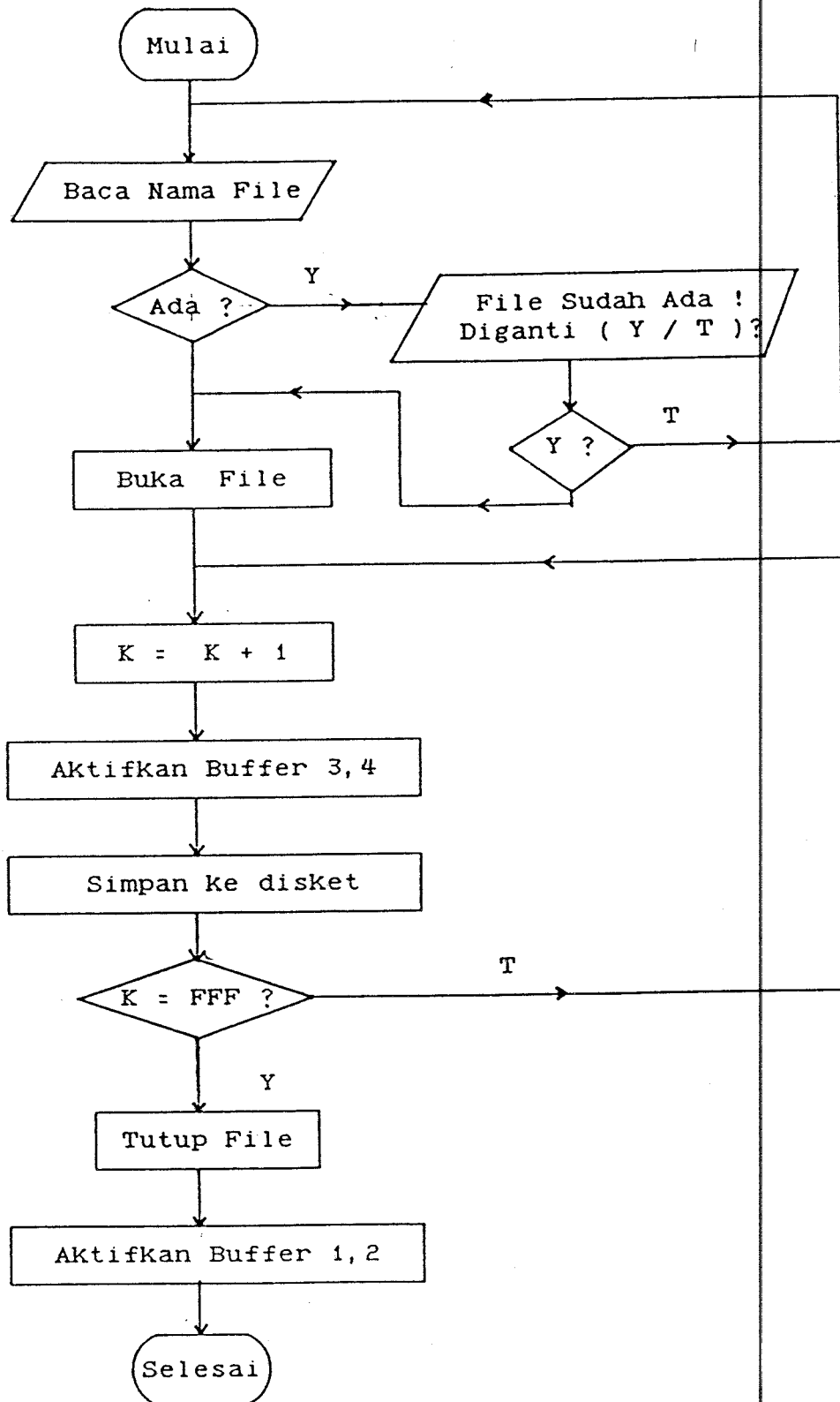


Gambar 4.3

Flowchart Procedure Rekam

IV.6. PROCEDURE SIMPAN

Procedure ini digunakan apabila data suara yang ada di RAM akan disimpan di disket. Untuk itu akan dipakai instruksi-instruksi pengolahan file dan pengaksesan langsung pada memori (instruksi Mem) yang terdapat pada



Gambar 4.4

Flowchart Procedure Simpan

Turbo Pascal 5. 5.

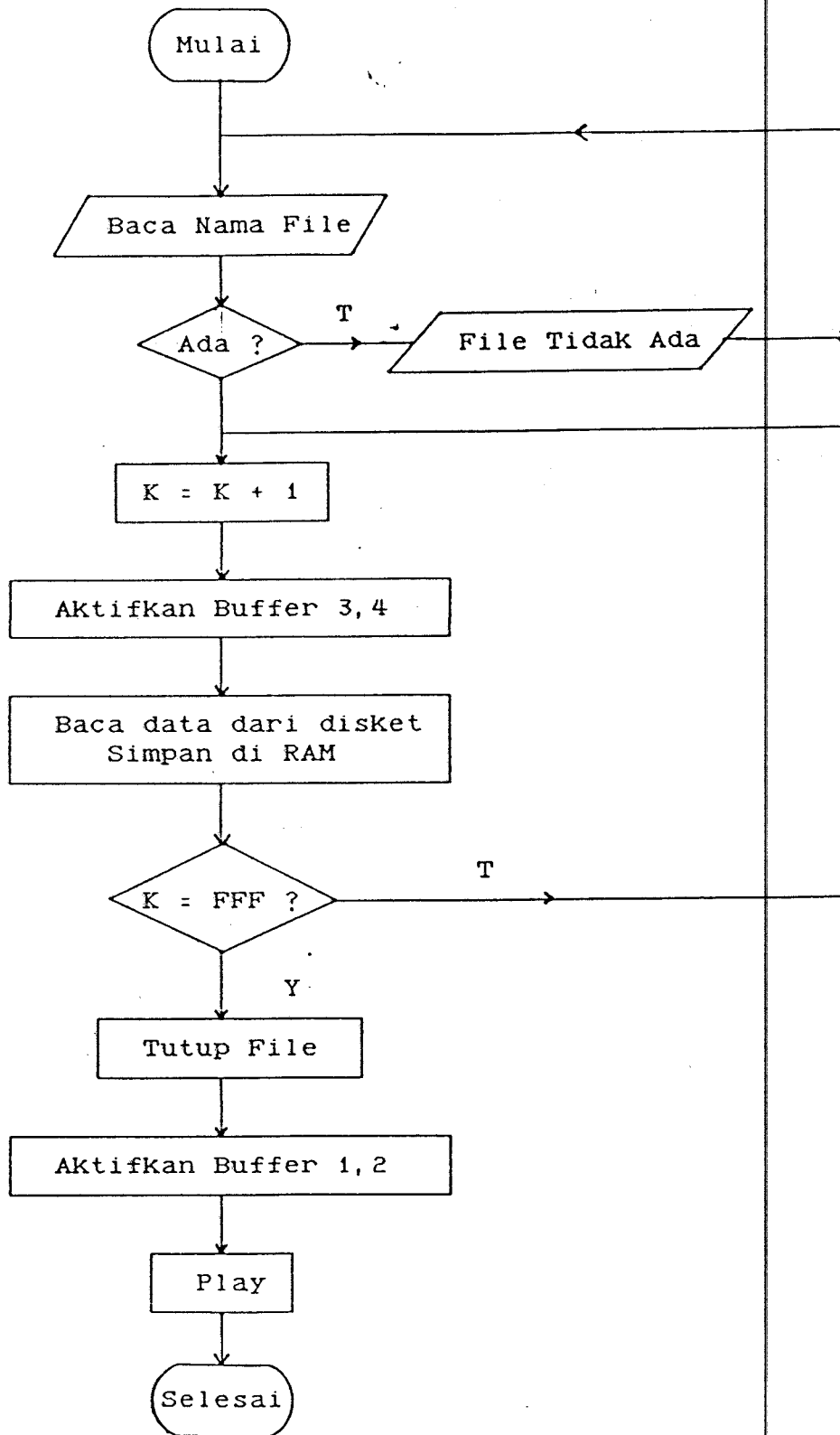
Untuk menyimpan di disket, ada dua tahapan yang dilalui, yaitu mentransfer data dari RAM rangkaian sintesa suara ke memori IBM dengan instruksi Mem dan kemudian memindahkan data dari memori IBM ke disket dengan perintah Write. Instruksi ini diulang sampai seluruh data (dari alamat C000:0000 sampai C000:0FFF) selesai disimpan. Gambar 4.4 menunjukkan flowchart dari procedure simpan.

Pada flowchart terlihat adanya instruksi yang akan memberitahukan jika nama file yang dituliskan sudah ada di directory. Bila ada, pemakai bisa memilih untuk mengganti isi file tersebut atau tidak. Bila memilih untuk mengganti isi file yang ada maka data yang baru langsung akan disimpan pada file tersebut. Jika pemakai tidak ingin mengganti maka akan ditanyakan kembali nama file yang lain.

Terlihat juga adanya instruksi untuk mengaktifkan buffer 3,4. Buffer inilah yang menghubungkan memori luar (RAM) dengan memori IBM (untuk lebih jelasnya lihat BAB III). Buffer ini hanya diaktifkan bila diperlukan transfer data antara RAM dengan memori IBM.

IV. 7. PROCEDURE PANGGIL

Procedure panggil diperlukan jika pemakai ingin mengambil data suara yang telah disimpan di disket untuk display-back. Gambar 4.5 memperlihatkan flowchart



Gambar 4.5

Flowchart Procedure Panggil

procedure ini . Pertama pemakai harus memasukkan nama file yang dikehendaki. Bila file ada di directory maka akan dilakukan transfer data dari disket ke memori IBM (dengan instruksi Read) lalu diteruskan ke RAM (dengan instruksi Mem). Jika file tidak ada, maka pemakai akan diminta untuk kembali memasukkan nama yang benar.

Setelah proses transfer selesai, maka dengan adanya instruksi Play, UM 5100 akan memplay-back suara yang sudah ada di RAM.

IV. 8. PROCEDURE TEKS KE SUARA

Pada procedure ini ada submenu yang dapat dipilih oleh pemakai, yaitu :

1. Rekam
2. Simpan
3. Pilihan Kata
4. Konversi

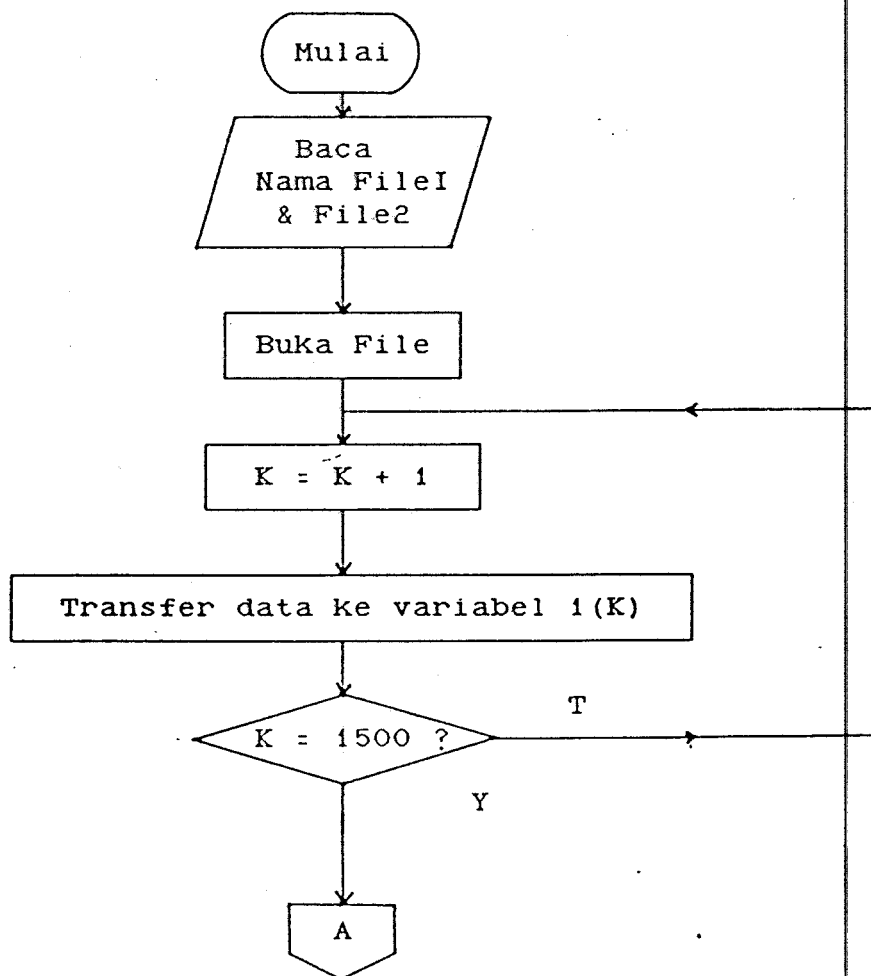
Sama seperti pilihan pada menu, Rekam digunakan untuk memasukkan data suara ke RAM, sedang Simpan digunakan untuk menyimpan data suara di dalam disket. Hanya saja dua pilihan tersebut disediakan untuk data yang hanya satu kata saja, karena nantinya Konversi (dari teks ke suara) akan dilakukan per kata. Pilihan Kata akan menunjukkan pada kita kata-kata yang ada di directory.

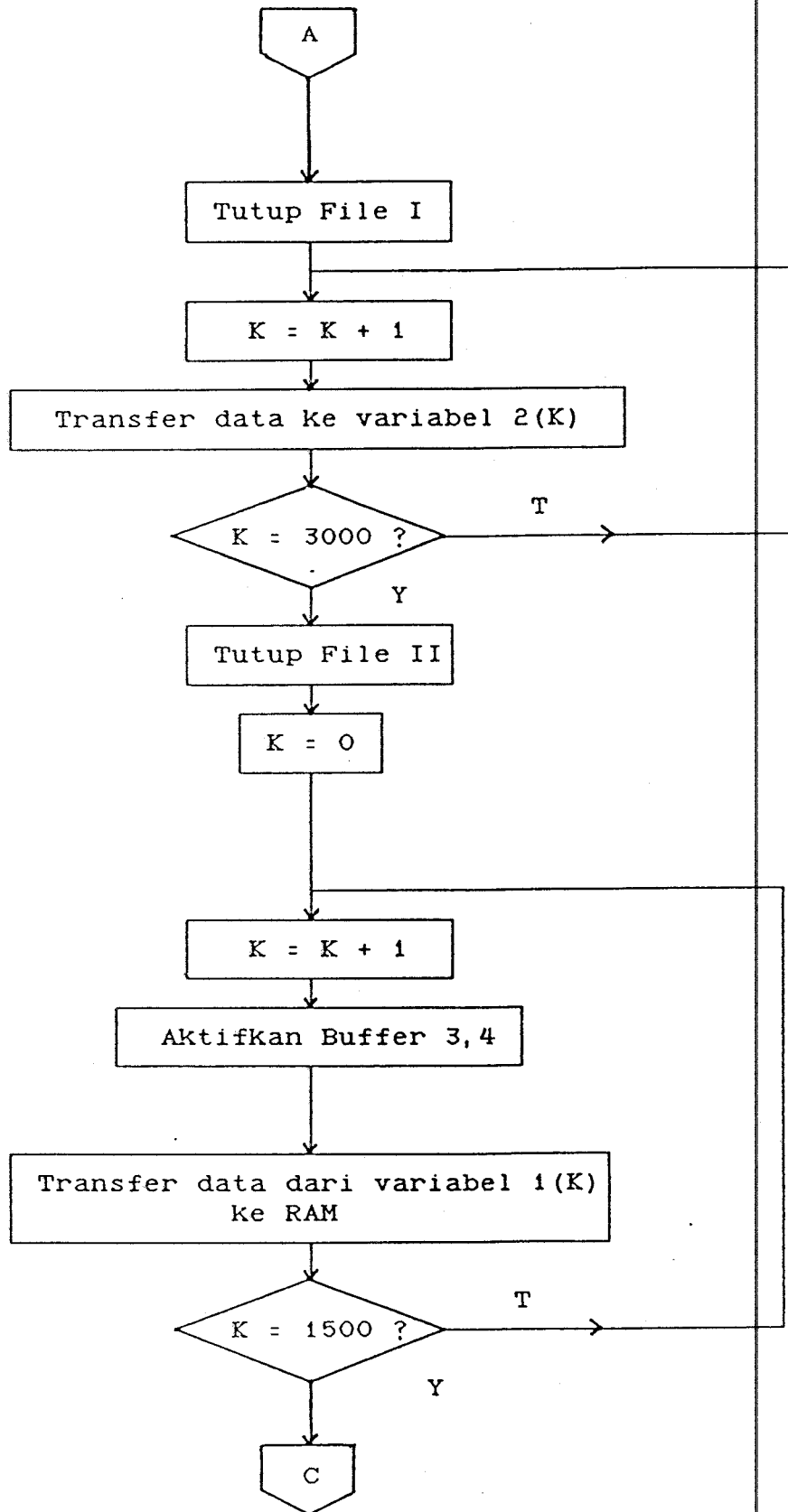
Bila kita memilih Konversi, maka akan ada pertanyaan yang harus kita jawab, yaitu mengenai jumlah kata yang akan dikonversi (1 s/d 4). Setelah itu pemakai

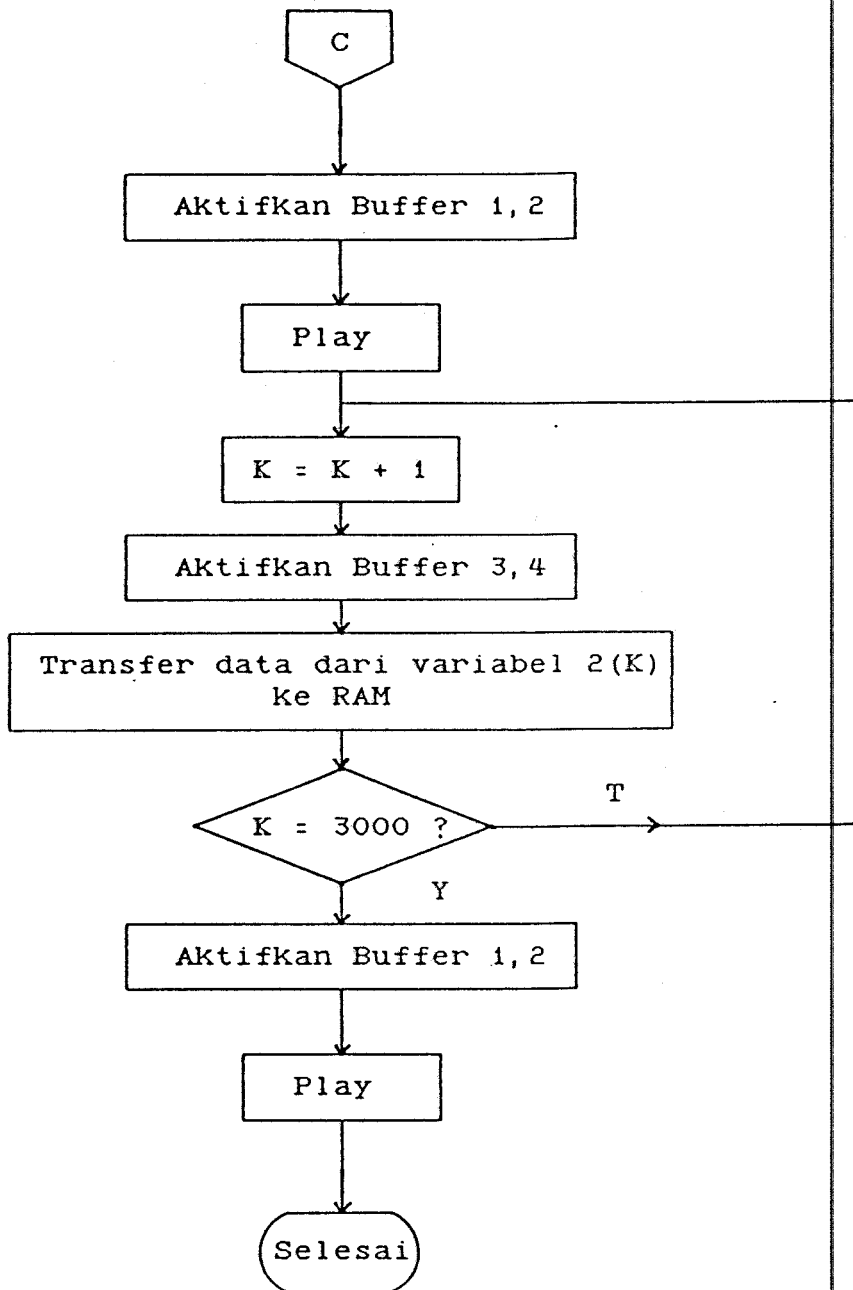
diminta untuk menuliskan kata-kata yang diinginkan.

Pada prinsipnya, procedure ini hampir sama dengan procedure panggil. Hanya saja file yang dipanggil bisa lebih dari satu. Gambar 4.6 menunjukkan contoh procedure konversi untuk dua kata.

Setelah file-file dibuka, maka data dari file pertama akan ditransfer ke variabel 1 dan data dari file kedua akan disimpan di variabel 2. Kemudian isi variabel 1 dipindahkan ke RAM dan langsung diplay-back, dan diikuti oleh kata kedua juga langsung diplay-back.







Gambar 4.6

Flowchart Procedure Konversi Dua Kata

B A B V

U J I C O B A D A N U J I U K U R

Pada bab ini akan dibahas hasil uji coba dan uji ukur dari rangkaian sintesa suara yang dibuat. Uji coba dimaksudkan untuk mengetahui apakah alat yang dibuat berfungsi sebagaimana yang direncanakan. Sedangkan yang dimaksud dengan uji ukur adalah pengujian yang dilakukan dengan menggunakan alat ukur sehingga bisa didapatkan bentuk gelombang output, besarnya penguatan, frekwensi cut off low pass filter dan lain-lain.

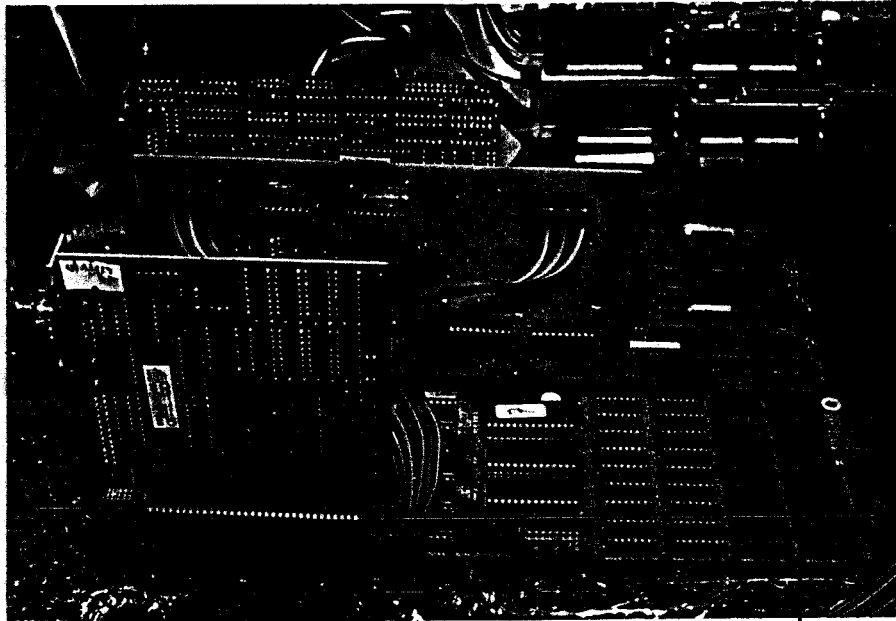
V. 1. U J I C O B A

Uji coba dilakukan dengan menggunakan IBM PC XT compatible. Pertama-tama rangkaian interface dipasang pada salah satu slot pada mother board (Gambar 5.1). Setelah komputer di-on-kan dan program pengendali di-load, maka tampaklah tampilan seperti pada Gambar 5.2. Kemudian rangkaian sintesa suara di-on-kan.

Tampak di layar monitor pilihan-pilihan yang ada, yaitu :

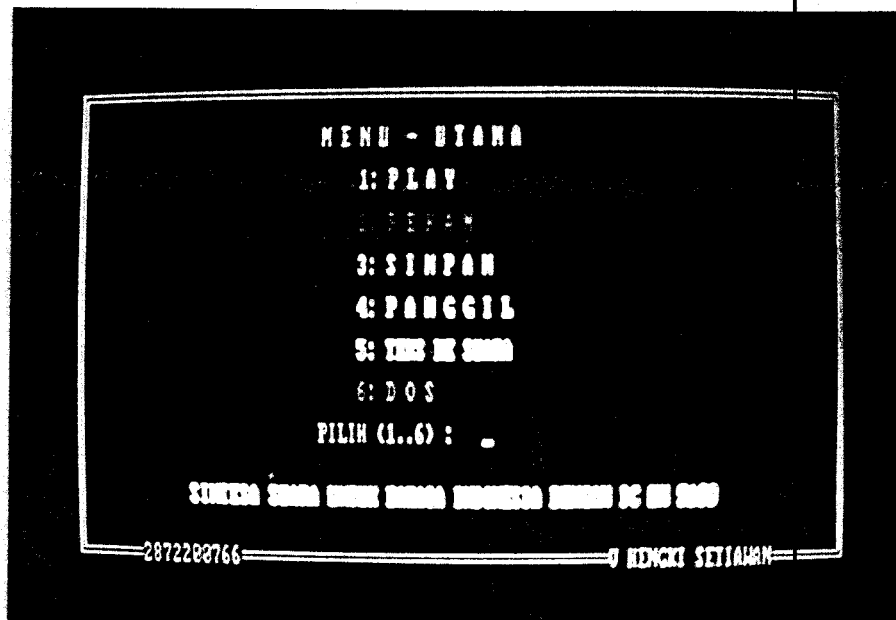
1. Play
2. Rekam
3. Simpan
4. Panggil
5. Teks ke Suara
6. DOS

Keenam pilihan yang ada akan diuji coba, apakah berjalan



Gambar 5.1

Rangkaian Interface Terpasang pada Mother Board



Gambar 5.2

Menu - Utama Program Pengendali

sebagaimana yang direncanakan.

Pilihan dilakukan dengan jalan menuliskan nomor yang dipilih dan diikuti penekanan tombol Enter. Pertama dicoba pilihan Play. Terdengar suara mendesis yang cukup keras dan dilayar monitor terlihat tulisan 'Play', lalu beberapa saat kemudian kembali ke menu utama. Suara tersebut terjadi karena tidak ada data suara di RAM. Untuk memasukkan data suara akan dicoba pilihan kedua.

Begitu tombol Enter ditekan pada layar monitor tampak tulisan 'Tekan Enter, dan mulailah berbicara'. Penulis mencoba memasukkan kata-kata 'satu dua tiga'. Tampaklah tulisan 'Rekam' di layar untuk beberapa saat lalu berganti tulisan 'Play' diikuti dengan terdengarnya 'satu dua tiga' sebelum akhirnya kembali ke menu utama.

Suara yang terdengar, walaupun masih terdapat distorsi, namun masih dapat dimengerti apa yang disintesa oleh UM 5100. Dari pengujian yang dilakukan jumlah kata maksimum adalah tiga kata.

Pengujian dilanjutkan dengan pilihan Simpan. Penulis menulis angka 3, lalu menekan Enter. Tampaklah pada layar tulisan 'Tulis Nama File : '. Penulis ingin menyimpan data suara 'satu dua tiga' yang sudah ada di RAM dengan nama Q123, maka dituliskan 'Q123' lalu menekan enter. Pada layar tampak tulisan 'Simpan' untuk beberapa saat, lalu munculah kembali menu utama. Ternyata pilihan Simpan dapat bekerja dengan baik.

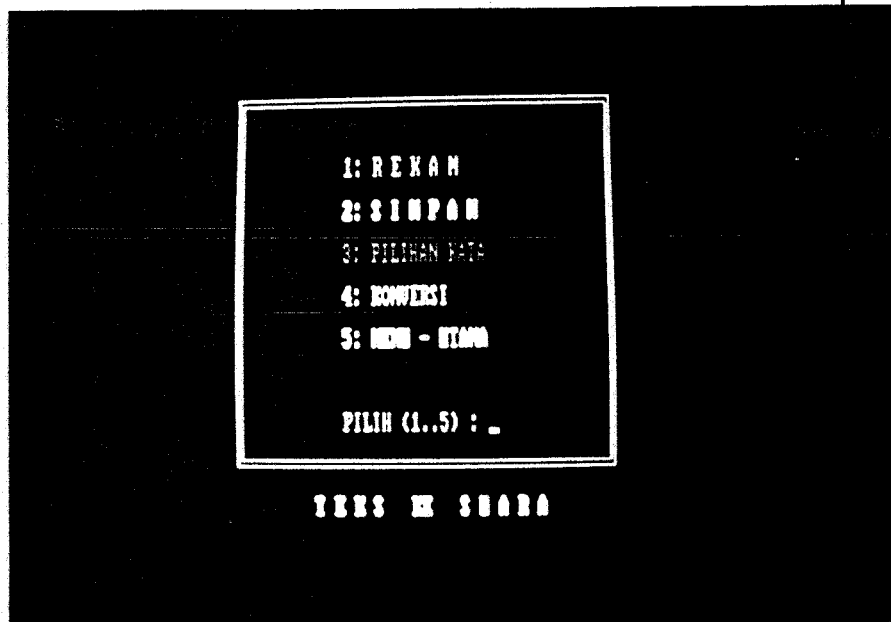
Karena pada pilihan simpan ada fasilitas untuk

melindungi file yang sudah ada penulis mencoba kembali menyimpan dengan nama Q123. Ternyata pada layar monitor muncul tulisan 'File Sudah Ada ! Diganti (Y/T)?'. Penulis menulis 'T', tampaklah kembali tulisan 'Tulis Nama File :'. Untuk menguji pilihan mengganti file yang sudah ada, penulis menuliskan kembali Q123 dan memilih 'Y' , ternyata data disimpan kembali, dan di layar tampak 'Simpan' lalu kembali ke menu utama.

Untuk menguji fungsi Panggil, penulis terlebih dahulu mengosongkan Ram dengan jalan, memilih Rekam tanpa memasukkan suara. Setelah itu baru memilih pilihan keempat. Pada layar tampil daftar file yang menyimpan data suara (dengan extension .QV), karena penulis baru menyimpan satu file, maka yang ada hanya Q123.QV. Pada bagian bawah dari layar tampaklah tulisan 'Tulis Nama File : ', penulis menuliskan Q123 lalu menekan Enter. Tampaklah tulisan 'Panggil' beberapa saat lalu berganti dengan 'Play' diikuti dengan terdengarnya 'satu dua tiga'. Dengan demikian sampai pilihan keempat sintesa suara dapat bekerja dengan baik.

Penulis melanjutkan dengan pilihan TEKS KE SUARA, tampak pada layar Menu seperti yang tampak pada Gambar 5.3. Pertama penulis mencoba pilihan 1 dan 2 dengan cara yang sama seperti pilihan 1 dan 2 pada Menu Utama, hanya saja penulis memasukkan satu kata saja yaitu 'selamat' dan meyimpannya dalam file 'selamat'. Sampai pilihan

Kedua Teks ke Suara dapat bekerja dengan baik. Penulis menambah file yang ada dengan kata 'pagi', 'siang', dan 'malam' dan menyimpannya dengan nama yang sesuai. Hal ini dilakukan untuk mempersiapkan uji coba terhadap pilihan 3 dan 4.



Gambar 5.4

Menu Teks ke Suara

Kini akan diuji coba Pilihan Kata , setelah dituliskan angka 3 diikuti Enter, tampak pada layar daftar file yang ada, yaitu :

- q123.qv
- selamat.qv
- pagi.qv
- siang.qv
- malam.qv

Di bagian bawah layar , terlihat tulisan 'Tekan Enter untuk Melanjutkan'. Penulis menekan Enter, maka Kembali muncul Menu Utama.

Pengujian dilanjutkan dengan pilihan Konversi, pertama tampak tulisan 'Jumlah Kata (Maksimum 4) : '. Penulis menuliskan '2', maka tampilkanlah 'Tulis Kata Pertama : '. Penulis memilih 'Selamat'. Lalu muncul pada layar 'Tulis Kata Kedua : '. 'Pagi' dipilih sebagai kata kedua. Kemudian tampak di layar tulisan 'Konversi Teks ke Suara' beberapa saat sebelum berganti dengan 'Selamat' diikuti terdengarnya 'Selamat' dan berlanjut dengan 'Pagi' disebelah 'Selamat' dan diikuti terdengarnya 'Pagi'. Dengan demikian uji coba bisa dikatakan berhasil. Uji coba dilanjutkan dengan mencoba menuliskan kata yang tidak ada di disket, ternyata program dapat menanggapi dengan baik, dengan munculnya tulisan 'Kata pertama tidak ada di directory' dan lain-lain. Kemudian juga dilakukan uji coba untuk konversi tiga dan empat kata. Ternyata hasilnya juga tidak mengecewakan dan sesuai dengan apa yang sudah direncanakan.

V. 2. UJI UKUR

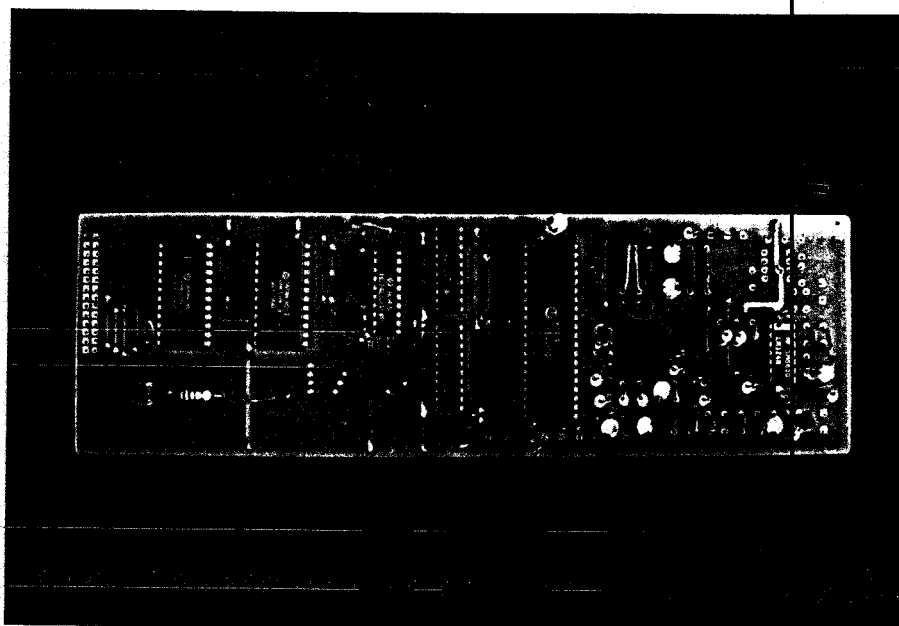
Uji ukur dilakukan di Laboratorium Elektronika - Jurusan Teknik Elektro - FTI - ITS, dengan menggunakan alat-alat ukur :

1. 1 buah Oscilloscope
2. 1 buah Function Generator

3. 1 buah Power Supply

4. 1 buah Digital Voltmeter

Pengujian hanya dilakukan terhadap rangkaian Sintesa Suara Gambar 5.5 meperlihatkan PCB dari rangkaian tersebut, sedang PCB dari rangkaian interface terlihat pada Gambar 5.6. Uji ukur meliputi rangkain Penguat Microphone, Comparator, Low Pass Filter, Clock, dan bentuk gelombang keluaran.

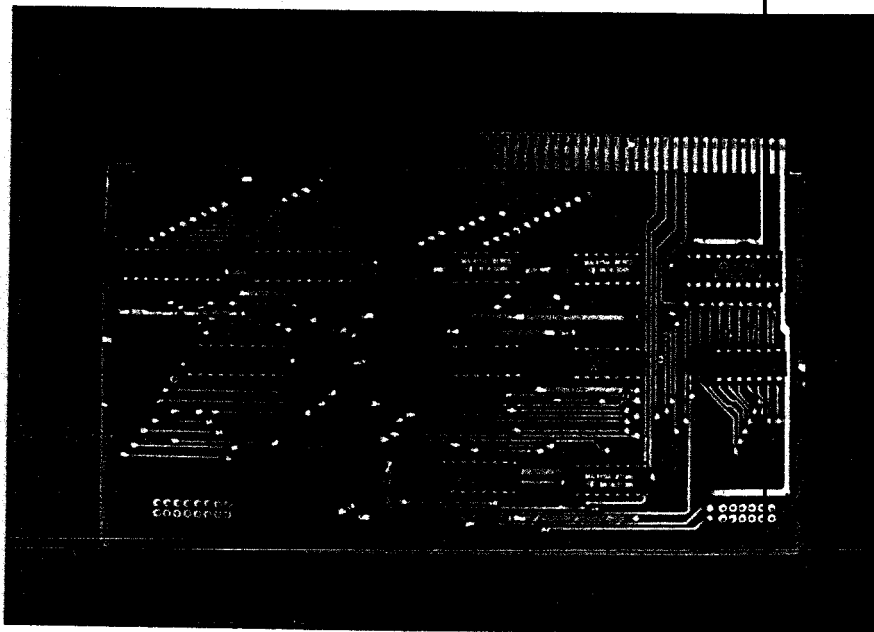


Gambar 5.5

PCB Rangkaian Sintesa Suara

V. 5. 1. Uji Ukur Penguat Microphone

Uji ukur ini dilakukan untuk memperoleh bentuk gelombang output, besarnya penguatan dan tegangan input maksimum. Untuk memperoleh bentuk gelombang output dan besarnya penguatan dilakukan dengan jalan menghubungkan input dari penguat microphone dengan function generator



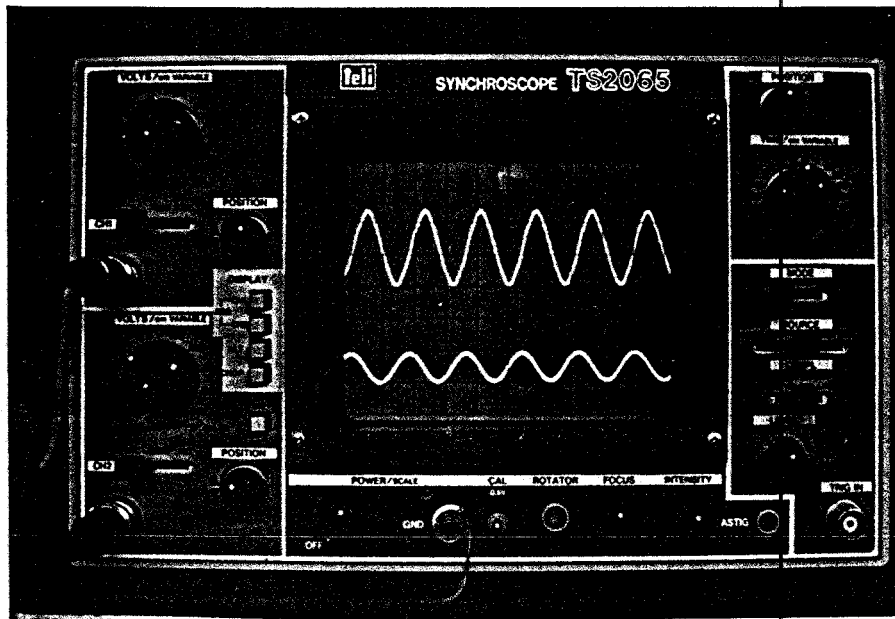
Gambar 5.6

PCB Rangkaian Interface

dan outputnya dengan oscilloscope. Kemudian diamati bentuk gelombang dan besarnya tegangan output untuk beberapa harga tegangan input yang berbeda.

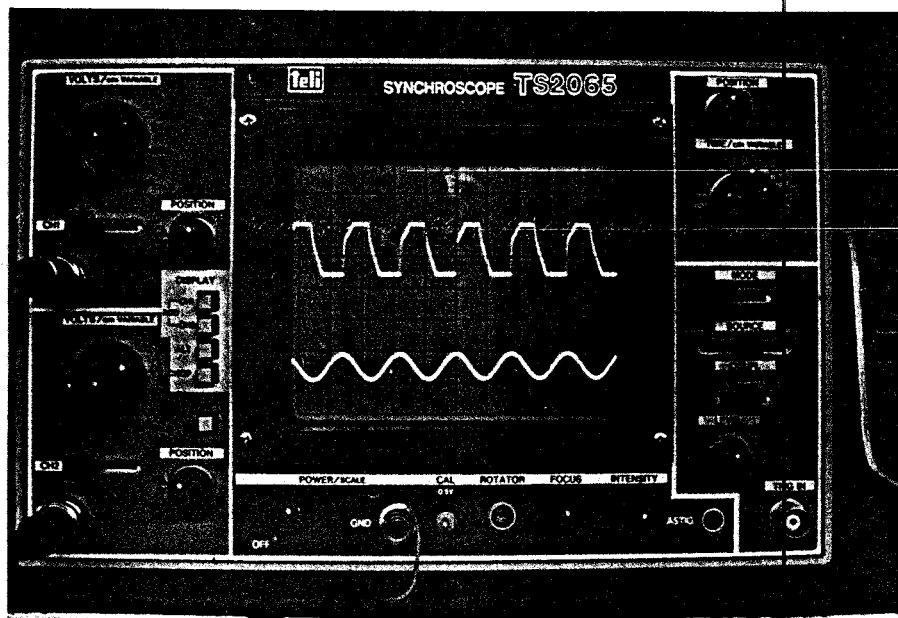
Sedangkan besarnya tegangan input diperoleh dengan jalan menaikkan terus tegangan input sambil mengamati gelombang output. Yang dimaksud dengan tegangan input maksimum adalah tegangan input maksimum sebelum terjadinya pemotongan (clipping) pada gelombang output.

Gambar 5.7 menunjukkan bentuk gelombang input dan output untuk tegangan input $0,2 V_{p-p}$, sedang gambar 5.8 memperlihatkan pemotongan yang terjadi pada output karena tegangan input terlalu besar ($1 V_{p-p}$). Hasil pengukuran tegangan input dan output dari penguat microphone adalah



Gambar 5.7

Bentuk Gelombang Input dan Output Penguat Micropone



Gambar 5.8

Bentuk Gelombang Input dan Output Penguat Microphone

Pada Saat Terjadi Pemotongan

sebagai berikut :

Vin (Vp-p)	Vout (Vp-p)	Penguatan (dB)
0,2	2,4	21,58
0,3	4	22,5
0,5	7	22,92
0,56	7,5	22,54

Dari hasil perhitungan diperoleh gain rata-rata dari penguat microphone, yaitu sebesar 22,3 dB. Sedang tegangan input maksimum adalah 0,56 Vp-p. Biasanya tegangan output dari microphone berkisar antara 3 - 4 mV (8,5 - 11,3 Vp-p), maka penguat ini dapat dikatakan memenuhi syarat sebagai penguat microphone. Apalagi kalau kita amati bentuk gelombang output yang dihasilkan, sinus murni tanpa cacat.

V. 5. 2. Uji Ukur Comparator

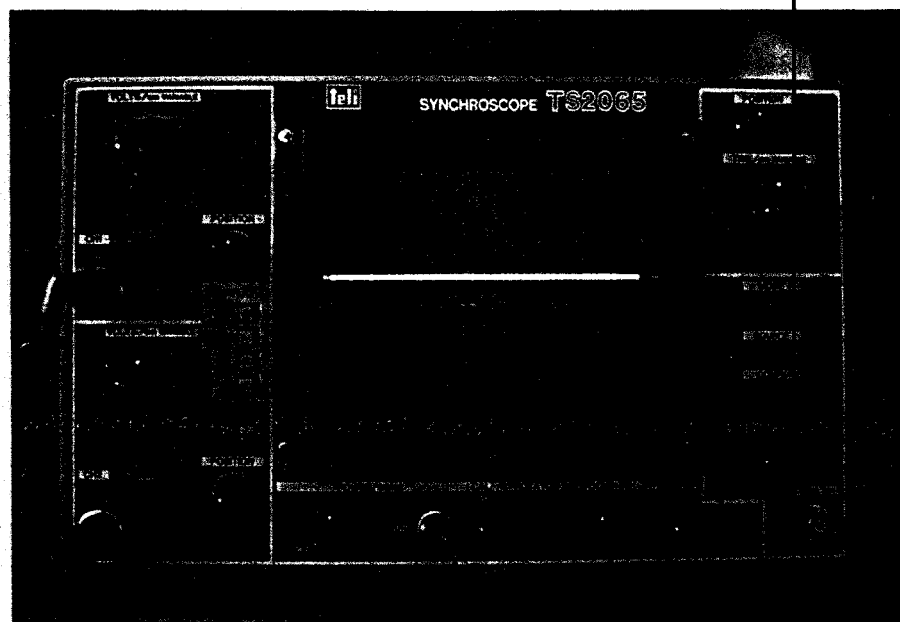
Pengujian dilakukan untuk membuktikan sifat dari comparator, yaitu akan menghasilkan output berlogika 1 bila input pada terminal (+) lebih besar dari pada input pada terminal (-). Untuk itu input comparator diberi tegangan yang berbeda, lalu diamati outputnya.

Ternyata comparator dapat bekerja dengan baik, pada saat tegangan pada terminal (+) lebih besar dari tegangan pada terminal (-) dihasilkan output berlogika 1 (Gambar 5.9), sebaliknya akan dihasilkan output berlogika 0 bila tegangan pada terminal (-) lebih besar dari tegangan pada

terminal (+) (Gambar 5.10).

V. 2. 3. Uji Ukur Low-Pass Filter

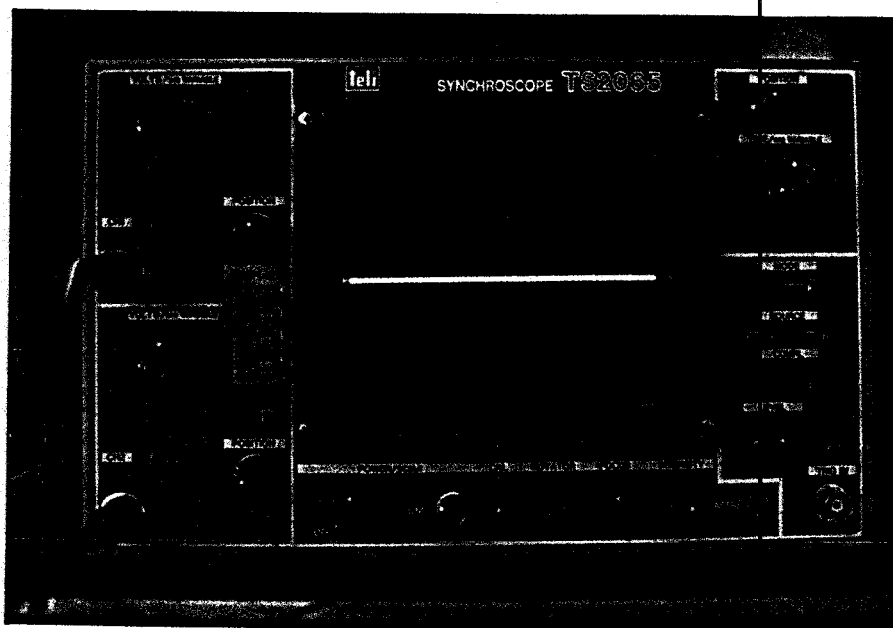
Rangkaian Low-Pass yang dipakai pada sintesa suara dirancang dengan frekwensi cut off sebesar 3 KHz. Dengan uji ukur akan dapat diketahui apakah pada kenyataannya juga demikian. Pengujian dilakukan dengan menghubungkan input dari low pass filter dengan function generator dan outputnya dihubungkan dengan oscilloscope. Frekwensi



Gambar 5. 9

Sinyal Output Comparator

(Terminal (+) > Terminal (-))



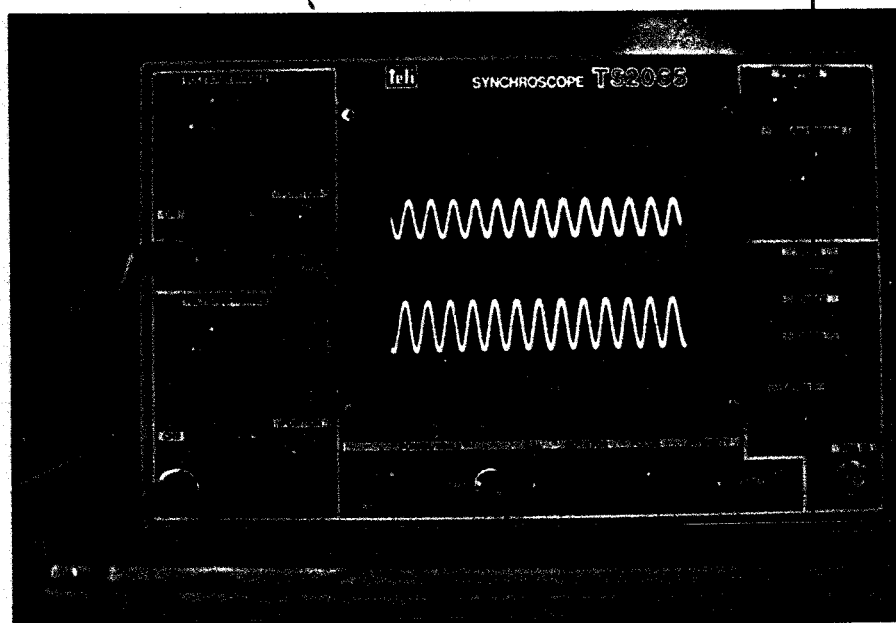
Gambar 5. 10

Sinyal Output Comparator
(Terminal (-) > Terminal (+))

function generator diatur sedemikian rupa sampai didapatkan tegangan output sebesar 0,707 dari tegangan input. Dari pengamatan yang dilakukan ternyata keadaan ini terjadi pada frekwensi 2800 Hz, sehingga ada perbedaan antara perhitungan dengan pengukuran. Gambar 5.11 menunjukkan sinyal input dan sinyal output dari rangkaian low-pass filter.

V. 2. 4. Uji ukur Clock

Uji ukur ini dilakukan terhadap pin 19 dari IC UM 5100 yang merupakan pin clock driver. Pengujian ini dimaksudkan untuk mendapatkan bentuk gelombang dari



Gambar 5.11

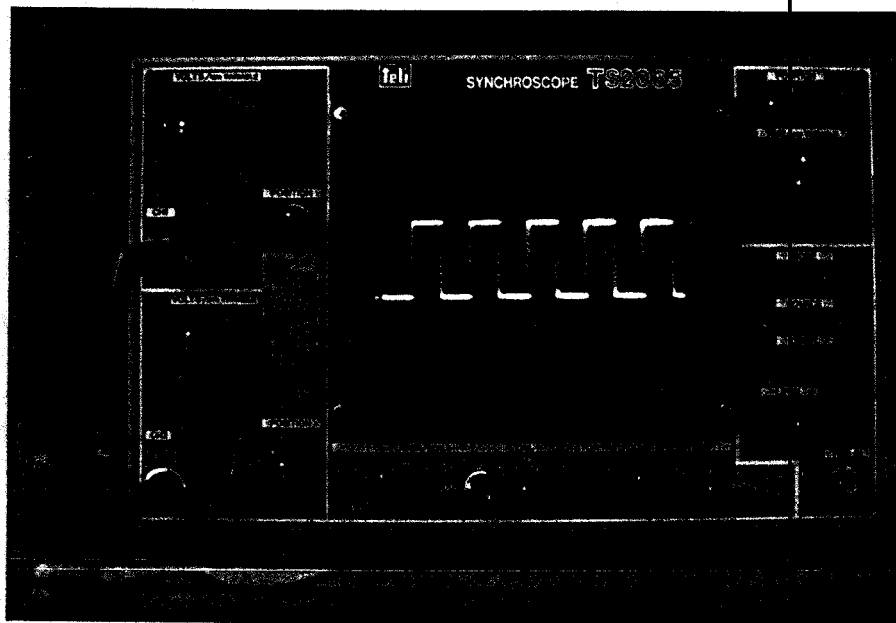
Sinyal Input dan Sinyal Output Low-Pass Filter

sinyal clock dan mengukur berapa frekwensinya. Karena besarnya frekwensi dapat diatur melalui potensiometer yang ada, maka pengamatan direncanakan dilakukan pada dua harga yaitu harga minimum dan maksimum.

Ternyata pengamatan hanya dapat dilakukan pada harga minimum saja karena pada harga maksimum tidak dapat diperoleh bentuk gelombang yang diam. Gambar 5.12 menunjukkan sinyal clock. Dari pengukuran diperoleh frekwensi dari clock yaitu sebesar 1 KHz.

V.2.5. Uji Ukur Gelombang Output

Uji ukur ini bertujuan untuk mengamati bentuk gelombang hasil sintesa dari rangkaian sintesa suara.

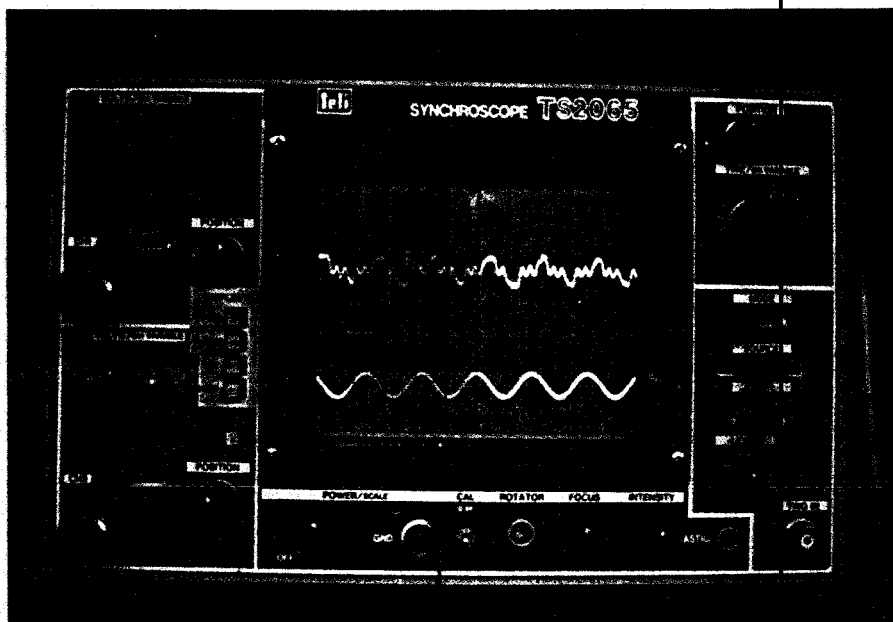


Gambar 5.12

S i n y a l C l o c k

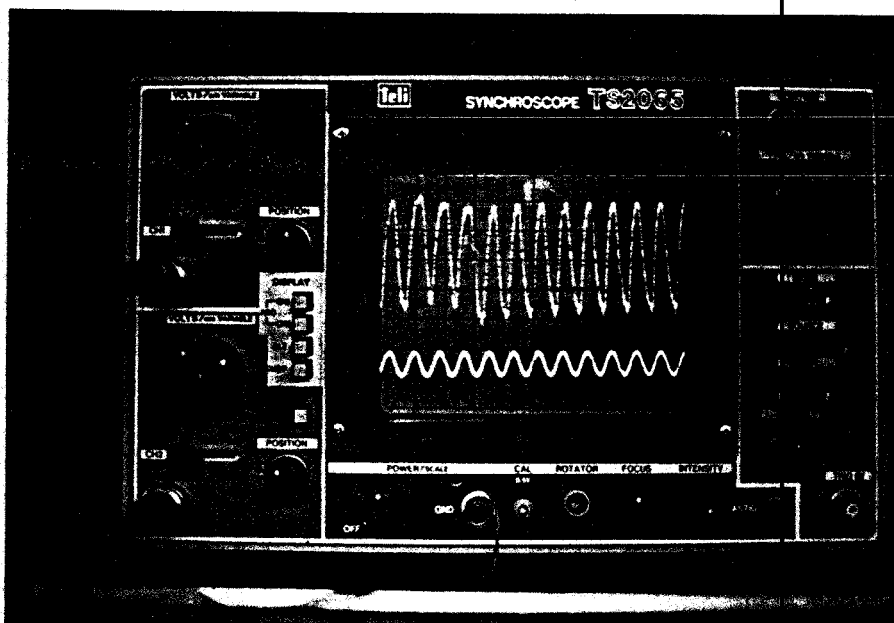
Pengujian dilakukan dengan menghubungkan input penguat microphone dengan function generator yang telah diatur pada frekwensi tertentu, sedang output dari penguat audio dihubungkan dengan oscilloscope. Kemudian sintesa suara diaktifkan pada mode rekam. Setelah proses perekaman selesai, suara di play kembali sambil diamati bentuk gelombangnya pada oscilloscope.

Dari pengamatan yang dilakukan tampak bahwa sinyal output yang dihasilkan oleh rangkaian sintesa suara tidak sinus murni, terlihat masih adanya komponen frekwensi tinggi. Gambar 5.13 menunjukkan sinyal input dan output pada frekwensi 1 KHz. Sedang Gambar 5.14 menunjukkan bentuk sinyal pada frekwensi 1500 Hz.



Gambar 5. 13

Sinyal Input dan Output pada Frekwensi 1 KHz



Gambar 5. 14

Sinyal Input dan Output pada Frekwensi 1500 Hz

B A B V I

P E N U T U P

Dari hasil perencanaan dan pembuatan sintesa suara untuk bahasa Indonesia dengan IC UM 5100 dapat ditarik beberapa kesimpulan bahwa :

1. Dengan menggunakan IC sintesa suara jenis Natural Speech Analysis/Synthesis, dalam hal ini UM 5100, dapat dibuat sebuah sintesa suara untuk bahasa Indonesia.
2. Dengan adanya rangkaian interface, keterbatasan memori dalam menyimpan kata-kata dapat diatasi, karena data suara dapat disimpan dalam bentuk file di disket.
3. Sintesa suara dengan IC UM 5100 yang dinterfacekan ke IBM PC XT dapat menghasilkan suara yang lebih alami walaupun masih terdengar adanya noise.
4. Dalam sebuah disket berkapasitas 360 KB dapat disimpan lebih dari 200 kata.
5. Perbendaharaan kata, pengucapan, dan intonasi kata dapat diubah sesuai kebutuhan.

DAFTAR PUSTAKA

- Bohn, Dennis, Audio Handbook, National Semiconductor Corporation, Santa Clara, 1976.
- Driscoll, Frederick F and Coughlin, Robert F, Operational Amplifiers And Linier Intergrated Circuits, Prentice Hall, Inc, Englewood Cliffs, 1982.
- Eggebrecht, Lewis C, Interfacing To The IBM Personal Computer, H.W. Sams & Co. Inc, Indianapolis, 1983.
- Hall, Douglas V, Microprocessor And Interfacing, Programming And Hardware, McGraw-Hill Book Co, New York, 1986.
- Jung, Walter G, IC Op Amp Cook Book, H.W. Sams & Co. Inc, Indianapolis, 1980.
- Stauugaard, Andrew C. Jr , Robotics And AI : An Introduction To Applied Machine Intelligence, Prentice/Hall International, Inc, New Jersey, 1987.
- Teja, Edward R and Gonnella, Gary W, Voice Technology, Reston Publishing Company, Inc, Virginia.
- Tedeshi, Frank P, The Active Filter Handbook, Tab Books, 1979.
- Yu-Cheng Liu and Gibson, Glenn A, Microcomputer Systems : 8086 / 8088 Family, Prentice/Hall International, Inc,

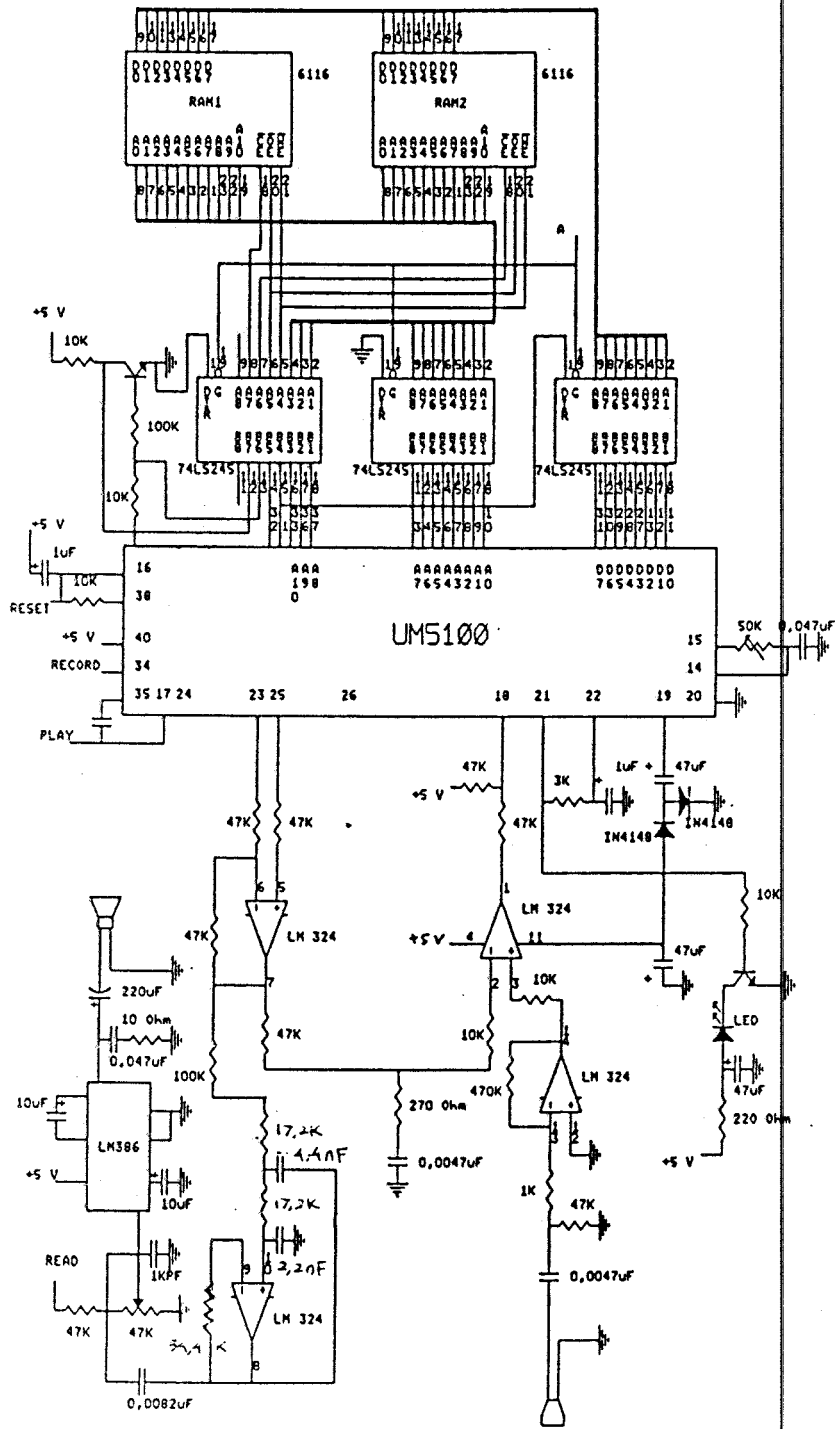
New Jersey, 1986.

Linear Databook, National Semiconductor Corp, 1982.

TTL Databook, Fairchild Inc, 1979.

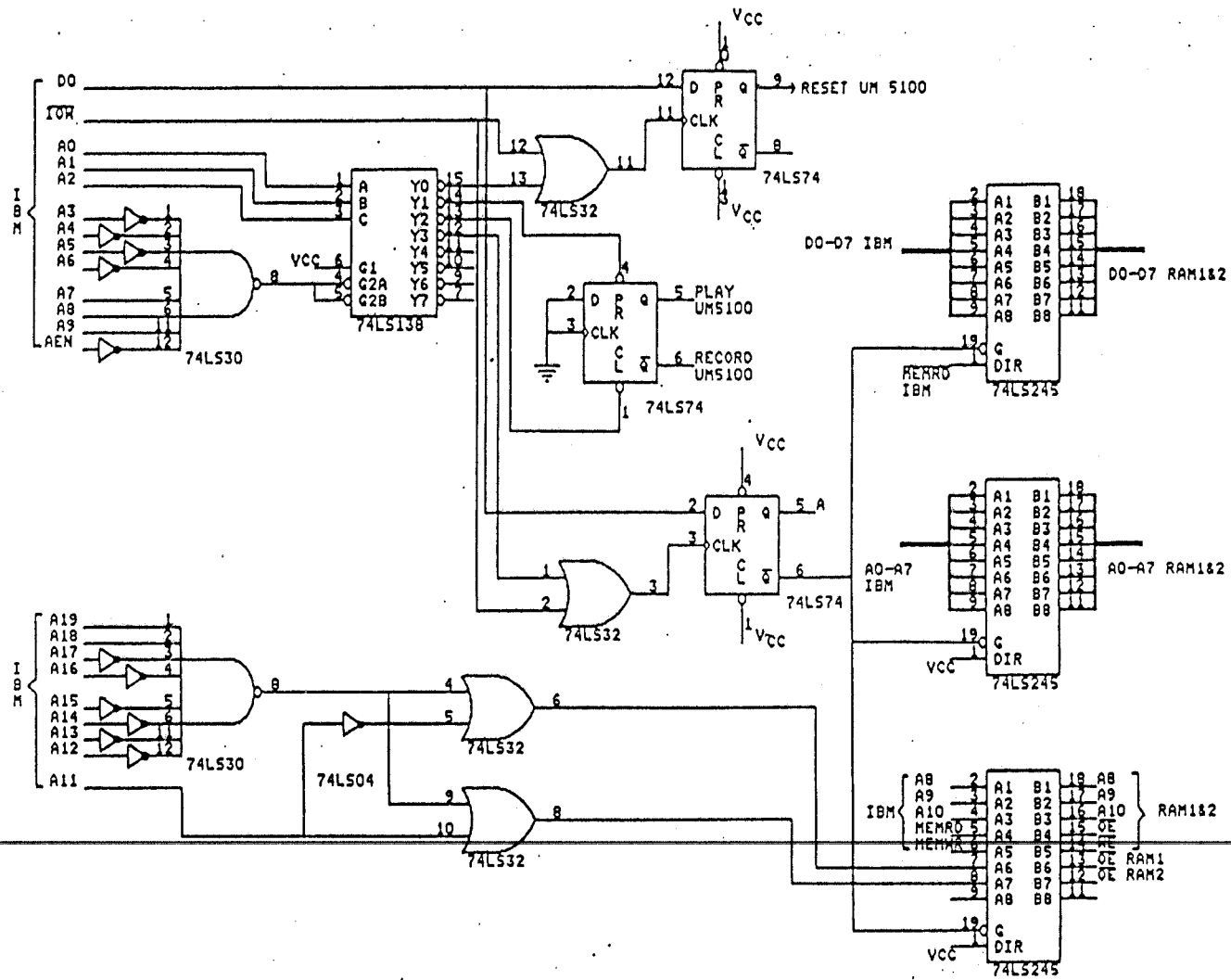
The IBM Personal Computer XT Technical Reference

LAMPIRAN A
RANGKAIAN SINTESA SUARA DAN INTERFACE



RANGKAIAN SINTESA SUARA

RANGKAIAN INTERFACE



LAMPIRAN B
PROGRAM PENGENDALI

```

#####
#
#           PROGRAM PENGENDALI
#   SINTESA SUARA UNTUK BAHASA INDONESIA
#           DENGAN IC UM 5100
#
#           Oleh :
#           V HENGKI SETIAWAN
#           2872200766
#           JURUSAN T. ELEKTRO - FTI - ITS
#           1990
#
#####

```

```
PROGRAM TUGAS_AKHIR;
```

```
USES CRT, DOS;
```

```
TYPE STRING12 = STRING[12];
```

```

VAR I, I1, Y, J1 : CHAR;
    D, C, B, E, H, K, V, J, HH : INTEGER;
    A : BYTE;
    KATA1, KATA3, KATA5, KATA7: STRING12;
    KATA, KATA2, KATA4, KATA6 : STRING12;
    F, Q, P, Z : FILE OF BYTE;

```

```
PROCEDURE MENU;
```

```
BEGIN
```

```

    CLRSCR; TEXTBACKGROUND(BLACK);
    TEXTCOLOR(LIGHTBLUE);
    PORT[$383] := $0; port[$380] := $1;
    WRITE(CHR(201));
    FOR K:=1 TO 78 DO WRITE(CHAR(205));
    WRITELN(CHAR(187));
    FOR K:=2 TO 22 DO BEGIN
        GOTOXY(1, K);
        WRITE(CHAR(186));
        GOTOXY(80, K); WRITE(CHAR(186));
    END;

    WRITE(CHR(200));
    FOR K:=1 TO 78 DO WRITE(CHAR(205));
    WRITELN(CHR(188));
    GOTOXY(8, 23); WRITE('2872200766');
    GOTOXY(26, 3); WRITELN('M E N U - U T A M A');
    WRITELN;
    GOTOXY(30, 5);
    TEXTCOLOR(GREEN); WRITELN('1: P L A Y');
    GOTOXY(30, 7); TEXTCOLOR(RED);
    WRITELN('2: R E K A M');
    GOTOXY(30, 9); TEXTCOLOR(MAGENTA);

```

```

WRITELN('3: S I M P A N ');
GOTOXY(30, 11); TEXTCOLOR(CYAN);
WRITELN('4: P A N G G I L');
GOTOXY(30, 13); TEXTCOLOR(7);
WRITELN('5: TEKS KE SUARA');
GOTOXY(13, 20); TEXTCOLOR(11);
WRITE('SINTESA SUARA UNTUK BAHASA INDONESIA
      DENGAN IC UM 5100');
GOTOXY(30, 15); TEXTCOLOR(BLUE);
WRITE ('6: D O S');
GOTOXY(26, 17); TEXTCOLOR(9);
WRITE('PILIH (1..6) : ');
GOTOXY(56, 23); WRITE('V HENGKI SETIAWAN');
GOTOXY(41, 17);
{$I-}
READ(I);
{$I+}
END;

```

```
PROCEDURE MENU1;
```

```
  BEGIN
```

```

    CLRSCR; TEXTBACKGROUND(BLACK);
    TEXTCOLOR(LIGHTBLUE);
    PORT[$383] := $0; PORT[$380] := $1;
    GOTOXY(19, 1);
    WRITE(CHR(201));
    FOR K:=1 TO 37 DO WRITE(CHAR(205));
    WRITELN(CHAR(187));
    GOTOXY(19, 2);
    FOR K:=2 TO 18 DO BEGIN
        GOTOXY(19, K);
        WRITE(CHAR(186));
        GOTOXY(57, K);
        WRITE(CHAR(186));
    END;

```

```

    GOTOXY(19, 18);
    WRITE(CHR(200));
    FOR K:=1 TO 37 DO WRITE(CHAR(205));
    WRITELN(CHR(188));
    GOTOXY(30, 4); TEXTCOLOR(GREEN);
    WRITELN('1: R E K A M');
    GOTOXY(30, 6); TEXTCOLOR(7);
    WRITELN('2: S I M P A N ');
    GOTOXY(30, 8); TEXTCOLOR(RED);
    WRITELN('3: PILIHAN KATA ');
    GOTOXY(30, 10); TEXTCOLOR(MAGENTA);
    WRITELN('4: KONVERSI ');
    GOTOXY(30, 12); TEXTCOLOR(CYAN);
    WRITELN('5: MENU - UTAMA');
    GOTOXY(27, 20); TEXTCOLOR(11);
    WRITE('T E K S   K E   S U A R A');
    GOTOXY(30, 16); TEXTCOLOR(9);

```

```

WRITE('PILIH (1..5) : ');
{$I-}
READ(I1);
{$I+}
END;

```

```
PROCEDURE DIR;
```

```

VAR
DIRINFO : SEARCHREC;
C : INTEGER;
BEGIN
  FINDFIRST('B:*.QV', READONLY, DIRINFO);
  CLRSCR; C := 0;
  WHILE DOSERROR = 0 DO
  BEGIN
    C := C + 1;
    IF C <= 15 THEN
      WRITELN(DIRINFO.NAME);
    IF (C > 15) AND (C < 31) THEN
      BEGIN
        GOTOXY(15, (C-15));
        WRITELN(DIRINFO.NAME);
      END;
    IF C >= 31 THEN
      BEGIN
        GOTOXY(30, (C-30));
        WRITELN(DIRINFO.NAME);
      END;
    FINDNEXT(DIRINFO);
  END;
  GOTOXY(15, 23); TEXTCOLOR(CYAN);
END;

```

```
PROCEDURE DIR1;
```

```

VAR
DIRINFO : SEARCHREC;
C : INTEGER;

BEGIN
  GOTOXY(46, 16);
  FINDFIRST('B:*.Q', READONLY, DIRINFO);
  CLRSCR; C := 0;
  WHILE DOSERROR = 0 DO
  BEGIN
    C := C + 1;
    IF C <= 15 THEN
      WRITELN(DIRINFO.NAME);
    IF (C > 15) AND (C < 31) THEN

```

```

                BEGIN
                    GOTOXY(15, (C-15));
                    WRITELN(DIRINFO. NAME);
                END;
            IF C >= 31 THEN
                BEGIN
                    GOTOXY(30, (C-30));
                    WRITELN(DIRINFO. NAME);
                END;
            FINDNEXT(DIRINFO);
        END;
        GOTOXY(15, 23); TEXTCOLOR(CYAN);
        WRITE('TEKAN ENTER UNTUK MELANJUTKAN.... ');
        READLN; READLN;
    END;

FUNCTION FILEEXISTS1(FILENAME: STRING) : BOOLEAN;

    VAR
        F : FILE;

    BEGIN
        {$I-}
        ASSIGN(F, KATA1);
        RESET(F);
        CLOSE(F);
        {$I+};
        FILEEXISTS1 := (IORESULT = 0) AND (KATA1 <> '');
    END;

FUNCTION FILEEXISTS2(FILENAME: STRING) : BOOLEAN;

    VAR
        F : FILE;

    BEGIN
        {$I-}
        ASSIGN(F, KATA2);
        RESET(F);
        CLOSE(F);
        {$I+};
        FILEEXISTS2 := (IORESULT = 0) AND (KATA2 <> '');
    END;

FUNCTION FILEEXISTS3(FILENAME: STRING) : BOOLEAN;

    VAR
        F : FILE;

    BEGIN
        {$I-}

```

```
    ASSIGN(F, KATA3);  
    RESET(F);  
    CLOSE(F);  
    {$I+};  
    FILEEXISTS3 := (IORESULT = 0) AND (KATA3 <> '');  
END;
```

```
FUNCTION FILEEXISTS4(FILENAME:STRING) : BOOLEAN;
```

```
    VAR  
        F : FILE;  
  
    BEGIN  
        {$I-}  
        ASSIGN(F, KATA4);  
        RESET(F);  
        CLOSE(F);  
        {$I+};  
        FILEEXISTS4 := (IORESULT = 0) AND (KATA4 <> '');  
    END;
```

```
FUNCTION FILEEXISTS5(FILENAME:STRING) : BOOLEAN;
```

```
    VAR  
        F : FILE;  
  
    BEGIN  
        {$I-}  
        ASSIGN(F, KATA5);  
        RESET(F);  
        CLOSE(F);  
        {$I+};  
        FILEEXISTS5 := (IORESULT = 0) AND (KATA5 <> '');  
    END;
```

```
FUNCTION FILEEXISTS6(FILENAME:STRING) : BOOLEAN;
```

```
    VAR  
        F : FILE;  
  
    BEGIN  
        {$I-}  
        ASSIGN(F, KATA6);  
        RESET(F);  
        CLOSE(F);  
        {$I+};  
        FILEEXISTS6 := (IORESULT = 0) AND (KATA6 <> '');  
    END;
```



```
FUNCTION FILEEXISTS7(FILENAME:STRING) : BOOLEAN;
```

```
VAR
```

```
  F : FILE;
```

```
BEGIN
```

```
  {$I-}
```

```
  ASSIGN(F, KATA7);
```

```
  RESET(F);
```

```
  CLOSE(F);
```

```
  {$I+};
```

```
  FILEEXISTS7 := (IORESULT = 0) AND (KATA7 <> '');
```

```
END;
```

```
PROCEDURE SELESAI;
```

```
  BEGIN
```

```
    CLRSCR;
```

```
    GOTOXY(25, 12);WRITE('S A M P A I J U M P A !!!');
```

```
    DELAY(550);CLRSCR;
```

```
  END;
```

```
PROCEDURE PLAY;
```

```
  BEGIN
```

```
    CLRSCR; TEXTCOLOR(GREEN);
```

```
    GOTOXY(25, 12);WRITE('P L A Y...');
```

```
    PORT[$380] := $0; PORT[$381] := $2;
```

```
    DELAY(3100); TEXTBACKGROUND(BLACK);
```

```
    READLN;
```

```
  END;
```

```
PROCEDURE PLAY1;
```

```
  BEGIN
```

```
    PORT[$380] := $0; PORT[$381] := $2;
```

```
    DELAY(1300); TEXTBACKGROUND(BLACK);
```

```
  END;
```

```
PROCEDURE REKAM;
```

```
  BEGIN
```

```
    CLRSCR; TEXTCOLOR(RED);
```

```
    GOTOXY(22, 12);WRITE('TEKAN ENTER, MASUKKAN  
SUARA...');
```

```
    REPEAT UNTIL KEYPRESSED;
```

```
    CLRSCR;
```

```
    GOTOXY(25, 12);WRITE('R E K A M...');
```

```
    PORT[$380] := $0; PORT[$382] := $2;
```

```
    DELAY(3100);
```

```
    PORT[$380] := $1;
    PLAY;

END;

PROCEDURE REKAM1;

BEGIN
    CLRSCR; TEXTCOLOR(RED);
    GOTOXY(22, 12);
    WRITE('TEKAN ENTER, MASUKKAN SUARA...');
    REPEAT UNTIL KEYPRESSED;
    CLRSCR;
    GOTOXY(25, 12); WRITE('R E K A M...');
    PORT[$380] := $0; PORT[$382] := $2;
    DELAY(1300);
    PORT[$380] := $1;
    CLRSCR;
    GOTOXY(25, 12); WRITE('P L A Y...');
    PLAY1;
    READLN; READLN;
END;

PROCEDURE READNAME(VAR NAME : STRING12);

    VAR CHARCOUNT : BYTE;
        KEYREAD : CHAR;

BEGIN
    NAME := '';
    CHARCOUNT := 0;
    REPEAT
        CHARCOUNT := CHARCOUNT + 1;
        KEYREAD := READKEY;
        WRITE(KEYREAD);
        INSERT(KEYREAD, NAME, CHARCOUNT);
    UNTIL KEYREAD = #13;
        DELETE(KATA, POS(#13, KATA), 1);
END;

PROCEDURE PANGGIL;

    VAR MERY : ARRAY[$0..$fff] OF BYTE;

BEGIN
    GOTOXY(42, 17);
    DIR;
    WRITE('TULIS NAMA FILE : ');
```

```

READNAME(KATA);
CLRSCR;GOTOXY(25,12);WRITE('P A N G G I L ...');
KATA1:='B:\'+KATA+'.QV';
IF FILEEXISTS1(PARAMSTR(1)) THEN
BEGIN
ASSIGN(F,KATA1);RESET(F);
FOR H := $000 TO $fff DO
BEGIN
READ(F,MERY[H]);
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END;
CLOSE(F);
CLRSCR;TEXTCOLOR(GREEN);
GOTOXY(25,12);WRITE(KATA);
PORT[$380] := $0;PORT[$381] := $2;
DELAY(3100);TEXTBACKGROUND(BLACK);
END

ELSE
BEGIN
CLRSCR;GOTOXY(25,12);TEXTCOLOR(100);
WRITE('FILE TIDAK ADA ....');DELAY(1000)
END;
END;

```

```
PROCEDURE SIMPAN2;
```

```

VAR MERY : ARRAY[$0..$fff] OF BYTE;

BEGIN
CLRSCR;GOTOXY(25,12);WRITE('S I M P A N .....');
ASSIGN(F,KATA1);REWRITE(f);
FOR H := $000 TO $fff DO
BEGIN
PORT[$383] := $1;
MERY[H]:=MEM[$C000 :H];
PORT[$383] := $0;
WRITE(F,MERY[H]);
END;
CLOSE(f);TEXTBACKGROUND(BLACK);
END;

```

```
PROCEDURE SIMPAN;
```

```

LABEL ULANG;

BEGIN
ULANG: CLRSCR;GOTOXY(24,12);TEXTCOLOR(magenta);

```

```

WRITE('TULIS NAMA FILE, MAX 8 HURUF', '');
GOTOXY(32, 14);
READNAME(KATA);
KATA1 := 'B:\'+KATA+'.QV';
IF FILEEXISTS1(PARAMSTR(1)) THEN

  BEGIN
    CLRSCR; GOTOXY(25, 12);
    WRITE('FILE SUDAH ADA , DIGANTI ? ( Y=1 ) ');
    READLN(J);
    IF J = 1 THEN
      SIMPAN2
    ELSE GOTO ULANG;

  END

  ELSE
    SIMPAN2;
  END;

PROCEDURE SIMPAN1;

  VAR MERY : ARRAY[0..2000] OF BYTE;

  BEGIN
    CLRSCR; GOTOXY(25, 12); WRITE('S I M P A N . . . . .');
    ASSIGN(F, KATA1); REWRITE(F); H := 0;
    FOR H := 0 TO 1500 DO
      BEGIN
        PORT[$383] := $1;
        MERY[H] := MEM[$C000 : H];
        PORT[$383] := $0;
        WRITE(F, MERY[H]);
      END;
    CLOSE(F); TEXTBACKGROUND(BLACK);
  END;

PROCEDURE SAVE1;

  LABEL ULANG;

  BEGIN
    ULANG: CLRSCR; GOTOXY(24, 12); TEXTCOLOR(magenta);
    WRITE('TULIS NAMA FILE, MAX 8 HURUF', '');
    GOTOXY(33, 14);
    READNAME(KATA);
    KATA1 := 'B:\'+KATA+'.Q';
    if FILEEXISTS1(PARAMSTR(1)) THEN

      BEGIN
        CLRSCR; GOTOXY(25, 12);
        WRITE('FILE SUDAH ADA , DIGANTI ? ( Y=1 ) ');

```

```

    READLN(J);
    IF J = 1 THEN
        SIMPAN1
    ELSE GOTO ULANG;

    END
    ELSE
        SIMPAN1;
    END;

```

```
PROCEDURE SALAM;
```

```

    VAR MERY : ARRAY[$0..$FFF] OF BYTE;

    BEGIN
        ASSIGN(F, 'B:DATANG.Q'); RESET(F);
        FOR H := $0 TO $7FFF DO
            BEGIN
                READ(F, MERY[H]);
                PORT[$383] := $1;
                MEM[$C000:H] := MERY[H];
                PORT[$383] := $0;
            END;
            CLOSE(F);
            PORT[$380] := $0; PORT[$381] := $2;
            DELAY(3100);
        END;

```

```
PROCEDURE KON4;
```

```

    VAR MERY : ARRAY[0..4000] OF BYTE;
        mery1 : ARRAY[0..4000] of BYTE;

    BEGIN
        CLRSCR; GOTOXY(24, 12); TEXTCOLOR(magenta);
        WRITE('TULIS KATA PERTAMA : ');
        READLN(KATA);

        CLRSCR; GOTOXY(24, 12); TEXTCOLOR(magenta);
        WRITE('TULIS KATA KEDUA : ');
        READLN(KATA1);

        CLRSCR; GOTOXY(24, 12); TEXTCOLOR(magenta);
        WRITE('TULIS KATA KETIGA : ');
        READLN(KATA2);

        CLRSCR; GOTOXY(24, 12); TEXTCOLOR(magenta);
        WRITE('TULIS KATA KEEMPAT : ');
        READLN(KATA3);

        CLRSCR; GOTOXY(25, 12);
        WRITE('KONVERSI TEKS KE SUARA.....');

```

```

KATA5:='B:\'+KATA+'.Q';
IF FILEEXISTS5(PARAMSTR(1)) THEN
BEGIN
ASSIGN(F, KATA5); RESET(F);
KATA6:='B:\'+KATA2+'.Q';
IF FILEEXISTS6(PARAMSTR(1)) THEN
BEGIN
ASSIGN(Q, KATA6); RESET(Q);
FOR H := 0 TO 1500 DO
BEGIN
READ(F, MERY[H]);
READ(Q, MERY1[H]);
END; CLOSE(F); CLOSE(Q);

KATA4:='B:\'+KATA1+'.Q';
IF FILEEXISTS4(PARAMSTR(1)) THEN
BEGIN
ASSIGN(Z, KATA4); RESET(Z);
KATA7:='B:\'+KATA3+'.Q';
IF FILEEXISTS7(PARAMSTR(1)) THEN
BEGIN
H := 1501;
ASSIGN(P, KATA7); RESET(P);
FOR H := 1501 TO 3000 DO
BEGIN
READ(Z, MERY[H]);
READ(P, MERY1[H]);
END; CLOSE(Z); CLOSE(P); H := 0;

FOR H := 0 TO 1500 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END; CLRSCR; GOTOXY(25, 12); WRITE(KATA);
PLAY1; H := 1501;

FOR H := 1501 TO 3000 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END; GOTOXY(33, 12); WRITE(KATA1);
PLAY1; H := 0;

FOR H := 0 TO 1500 DO
BEGIN
PORT[$380] := $1;
PORT[$383] := $1;
MEM[$C000:H] := MERY1[H];

```

```

PORT[$383] := $0;
END; GOTOXY(41, 12); WRITE(KATA2);
PLAY1;

FOR H := 1501 TO 3000 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY1[H];
PORT[$383] := $0;
END; GOTOXY(48, 12); WRITE(KATA3);
PLAY1;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA KEEMPAT TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA KEDUA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA KETIGA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA PERTAMA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END;

PROCEDURE KON3;
VAR D, C, B, E, H, hh : INTEGER;
    MERY : ARRAY[0..4000] OF BYTE;
    MERY1 : ARRAY[0..4000] OF BYTE;
BEGIN
CLRSCR; GOTOXY(24, 12); TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA PERTAMA : ');
READLN(KATA);

CLRSCR; GOTOXY(24, 12); TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA KEDUA : ');

```

```

READLN(KATA1);

CLRSCR;GOTOXY(24,12);TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA KETIGA : ');
READLN(KATA2);

CLRSCR;GOTOXY(25,12);
WRITE('KONVERSI TEKS KE SUARA.....');

KATA5:='B:\'+KATA+'.Q';
IF FILEEXISTS5(PARAMSTR(1)) THEN
BEGIN
ASSIGN(F,KATA5);RESET(F);
KATA3:='B:\'+KATA2+'.Q';
IF FILEEXISTS3(PARAMSTR(1)) THEN
BEGIN
ASSIGN(Q,KATA3);RESET(Q);
FOR H := 0 TO 1500 DO
BEGIN
READ(F,MERY[H]);
READ(Q,MERY1[H]);
END;CLOSE(F);CLOSE(Q);

KATA4:='B:\'+KATA1+'.Q';
IF FILEEXISTS4(PARAMSTR(1)) THEN
BEGIN
H := 1501;
ASSIGN(Z,KATA4);RESET(Z);
FOR H := 1501 TO 3000 DO
BEGIN
READ(Z,MERY[H]);
END;CLOSE(Z);H := 0;

FOR H := 0 TO 1500 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END;CLRSCR;GOTOXY(25,12);WRITE(KATA);
PLAY1;H := 1501;

FOR H := 1501 TO 3000 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END;GOTOXY(33,12);WRITE(KATA1);
PLAY1;H := 0;

```



```

FOR H := 0 TO 1500 DO
BEGIN
PORT[$380] := $1;
PORT[$383] := $1;
MEM[$C000:H] := MERY1[H];
PORT[$383] := $0;
END;GOTOXY(41, 12);WRITE(KATA2);
PLAY1;
END
ELSE
BEGIN
CLRSCR;GOTOXY(25, 12);TEXTCOLOR(100);
WRITE('KATA KEDUA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR;GOTOXY(25, 12);TEXTCOLOR(100);
WRITE('KATA KETIGA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR;GOTOXY(25, 12);TEXTCOLOR(100);
WRITE('KATA PERTAMA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END;
END;

PROCEDURE KON2;
VAR D, C, B, E, H, hh : INTEGER;
MERY : ARRAY[0..4000] OF BYTE;
BEGIN
CLRSCR;GOTOXY(24, 12);TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA PERTAMA : ');
READLN(KATA);

CLRSCR;GOTOXY(24, 12);TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA KEDUA : ');
READLN(KATA1);

CLRSCR;GOTOXY(25, 12);
WRITE('KONVERSI TEKS KE SUARA.....');

KATA2:='B:\'+KATA+'.Q';
IF FILEEXISTS2(PARAMSTR(1)) THEN
BEGIN
ASSIGN(F, KATA2);RESET(F);
FOR H := 0 TO 1500 DO

```

```

BEGIN
READ(F, MERY[H]);
END; CLOSE(F);

KATA3 := 'B:\'+KATA1+'.Q';
IF FILEEXISTS3(PARAMSTR(1)) THEN
BEGIN
H := 1501;
ASSIGN(Q, KATA3); RESET(Q);
FOR H := 1501 TO 3000 DO
BEGIN
READ(Q, MERY[H]);
END; CLOSE(Q); H := 0;

FOR H := 0 TO 1500 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END; CLRSCR; GOTOXY(25, 12); WRITE(KATA);
PLAY1; H := 1501;

FOR H := 1501 TO 3000 DO
BEGIN
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END; GOTOXY(33, 12); WRITE(KATA1);
PLAY1;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA KEDUA TIDAK ADA DI DIRECTORY ...');
DELAY(2500);
END;
END
ELSE
BEGIN
CLRSCR; GOTOXY(25, 12); TEXTCOLOR(100);
WRITE('KATA PERTAMA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END;

PROCEDURE KON1;
VAR D, C, B, E, H : INTEGER;
MERY : ARRAY[$0..2000] OF BYTE;
BEGIN
CLRSCR; GOTOXY(24, 12); TEXTCOLOR(MAGENTA);
WRITE('TULIS KATA YG DIPILIH : ');
READLN(KATA);

```

```

CLRSCR;GOTOXY(25,12);
WRITE('KONVERSI TEKS KE SUARA.....');

KATA1:='B:\'+KATA+'.Q';
IF FILEEXISTS1(PARAMSTR(1)) THEN
BEGIN
ASSIGN(F,KATA1);RESET(F);
FOR H := $000 TO 1500 DO
BEGIN
READ(F,MERY[H]);
PORT[$383] := $1;
MEM[$C000:H] := MERY[H];
PORT[$383] := $0;
END;
CLOSE(F);CLRSCR;GOTOXY(25,12);WRITE(KATA,');
PLAY1;
END
ELSE
BEGIN
CLRSCR;GOTOXY(25,12);TEXTCOLOR(100);
WRITE('KATA TIDAK ADA DI DIRECTORY....');
DELAY(2500);
END;
END;

```

```
PROCEDURE KONVERSI;
```

```

BEGIN
CLRSCR;GOTOXY(25,12);TEXTCOLOR(LIGHTBLUE);
WRITE('TULIS JUMLAH KATA (1..4) = ');
READLN(V);
IF V = 1 THEN KON1;
IF V = 2 THEN KON2;
IF V = 3 THEN KON3;
IF V = 4 THEN KON4;
END;

```

```
PROCEDURE TEKSKESUARA;
```

```

LABEL MENU2;

BEGIN
GOTOXY(10,5);READLN;
WHILE(UPCASE(I1)<>'5') DO BEGIN;
MENU1;
CASE I1 OF '1':REKAM1;
           '2':SAVE1;
           '3':DIR1;
           '4':KONVERSI;
           '5':;
END;
END;
I1 := ' ';READLN;
END;

```

```
BEGIN
  CLRSCR; TEXTCOLOR (MAGENTA);
  GOTOXY (25, 12); WRITE ('SELAMAT DATANG.....');
  {SALAM;}
  GOTOXY (10, 5);
  WHILE (UPCASE (i) <> '6') DO BEGIN;
    MENU;
    CASE I OF '1': PLAY;
              '2': REKAM;
              '3': SIMPAN;
              '4': PANGGIL;
              '5': TEKSKESUARA;
              '6': SELESAI;
    END;
  END;
  I := ' '; {READLN;}
END.
```

LAMPIRAN C
LEMBAR DATA



PRELIMINARY

UM5100

VOICE PROCESSOR

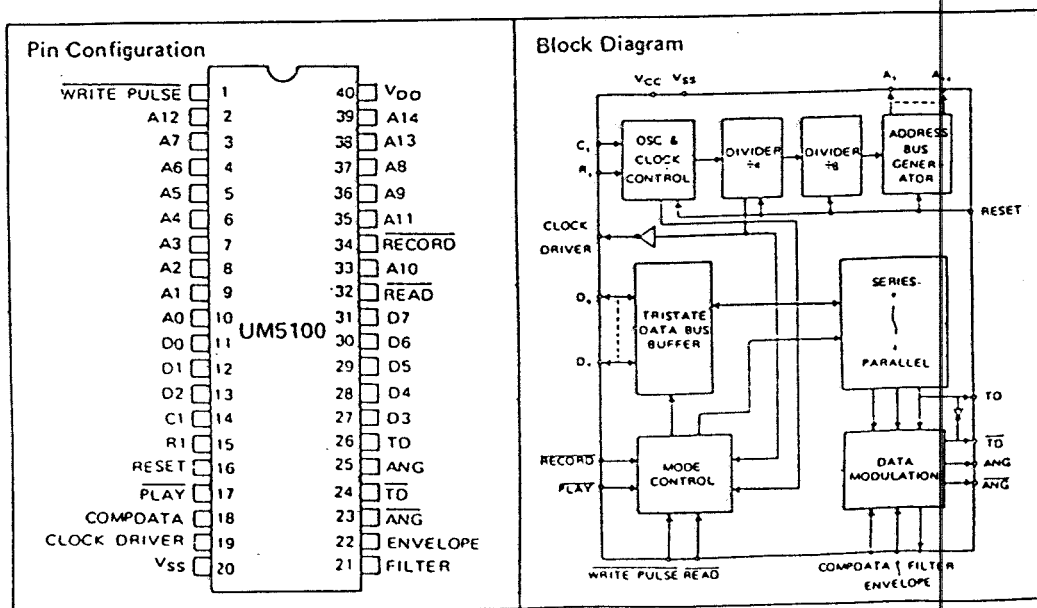
Features

- Voice Recording and Reproducing with External SRAM.
- Voice Reproducing with External EPROM or ROM.
- Continuous Variable Slope Delta Modulation.
- Max. Memory Addressable 256K.
- Single 3V~6V Power Operation.
- Low Power Consumption by CMOS Structure.
- In Expensive RC Oscillation.
- Adjustable Bit Rates from 10K to 28K bps.
- High Quality Voice Regeneration.
- Application for Voice Storage, Security System Telecommunication and Other Voice Fields

General Description

The UM 5100 is a single chip CMOS LSI for voice recording and reproducing with SRAM or only voice producing with EPROM or ROM. It is composed of RC oscillator, Address/data bus, series to parallel (and vice versa) converter, mode controller and delta modulation. Delta modulation circuit is based on continuous variable slope waveform by digital recording and reproducing of voice signals. For low frequency, there are auxiliary outputs to avoid distortion. For high frequency, there

is an internal amplifier, to intensify voice loudness. The RC oscillator is the best option in bit rate from 10K to 28K bps. Memory IC and Audio circuit which includes a microphone speaker, amplifier etc., are externally connected to complete voice recording and reproducing module. All Input pins are CMOS compatible. Applications field Includes: voice storage, voice response, Telecommunication and other voice field.





Pin Description

V_{DD}
Positive power supply +3 ~ +6V operation voltage

V_{SS}
Circuit GND potential.

A₀ ~ A₁₄
Address bus.

D₀ ~ D₇
Data bus.

Record
Input pin, Active Low. Triggering this input pin the chip will go into "speech analysis" mode.

Write Pulse
Output pin, Active Low. This pin generates one pulse each time the clock counting advances eight cycles. This pin is active only in "speech analysis" mode.

Read
Output pin, Active Low. This pin provides output control signal for external memory chip. This pin is active only in "speech synthesis" mode.

PLAY
Input pin, Active Low. Triggering this pin, the chip will go into "speech synthesis" mode.

Reset
Input pin. Active High. When Activated, All the internal counters are cleared and the chip is disabled.

ANG and $\overline{\text{ANG}}$
Analogue voice signal outputs with opposite phase.

Filter
Output pin. Produces envelope waveform with external integrator circuit.

Envelope
Input pin. Envelope signal will go into this pin to modulate voice amplitude.

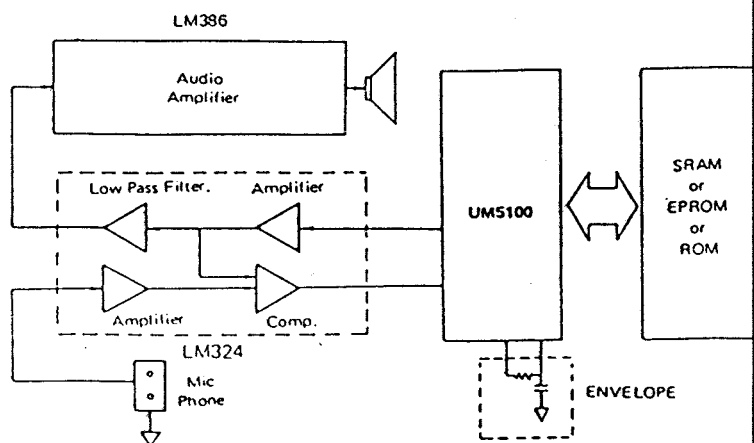
TD and $\overline{\text{TD}}$
Output pin. Auxiliary outputs for low frequency signal.

Compdata.
Input pin. Detects the delta slope which is generated from the comparison of input signal and feedback signal.

C, R
RC Oscillator pins. $f_{osc} = 40 \text{ KHz}$ for $R = 7.2 \text{ K}$ $C = 0.0047 \mu\text{f}$

Clock Driver
Output pin for negative voltage generation.

Talk Back Application Block Diagram





UM5100

Absolute Maximum Ratings

Parameter	Symbol	Rating
Supply Voltage	$V_{DD} - V_{SS}$	0 to 7V
Input Voltage	V_{IN}	V_{SS} to V_{PD}
Operating Temperature	T_{OP}	-10°C to 60°C
Storage Temperature	T_{ST}	-20°C to 80°C

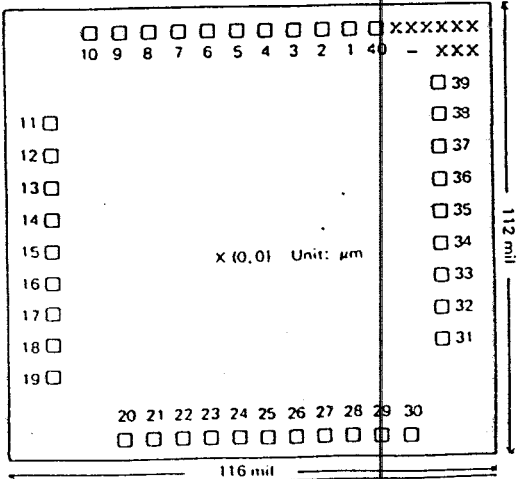
Electrical Characteristics ($V_{DD} = 5V$ $F_{osc} = 40$ KHz unless otherwise specified)

Parameter	Symbol	Limit			Test Conditions
		Min	TYP	MAX	
Supply Voltage	V_{DD}	3V	5V	6V	
Stand-by Current	I_{DD}		1.0 μA		No Oscillator.
Clock Drive Current	I_{drive}	16 mA			$V_{out} = 2.4V$
Clock Sink Current	I_{sink}	16 mA			$V_{out} = 0.4V$
Input Voltage	High V_{IH}	3.5 V		5V	$V_{DD} = 5V$
	Low V_{IL}	0		1.5V	
*Output Current	Drive I_{drive}	3 mA	4 mA		$V_{ou} = 2.4V$
	Sink I_{sink}	3 mA	4 mA		$V_{ou} = 0.5V$
Reset Pulse	T_{Reset}	1 μS			
Write Pulse Width	T_{Write}	200 NS		10 μS	

Note: *Except A₀, D₀ pin.

PAD	X	Y	PAD	X	Y
1	749	1208	21	-730	1208
2	539	1208	22	-520	1208
3	329	1208	23	-310	-1208
4	119	1208	24	-100	-1208
5	-91	1208	25	110	-1208
6	-301	1208	26	319	-1208
7	-511	1208	27	529	-1208
8	-721	1208	28	740	-1208
9	-931	1208	29	950	-1208
10	-1140	1208	30	1161	-1208
11	-1264	697	31	1264	-691
12	-1264	486	32	1264	-481
13	-1264	277	33	1264	-271
14	-1264	67	34	1264	-61
15	-1264	-143	35	1264	149
16	-1264	-354	36	1264	359
17	-1264	-564	37	1264	569
18	-1264	-774	38	1264	779
19	-1264	-984	39	1264	989
20	-956	1216	40	949	1208

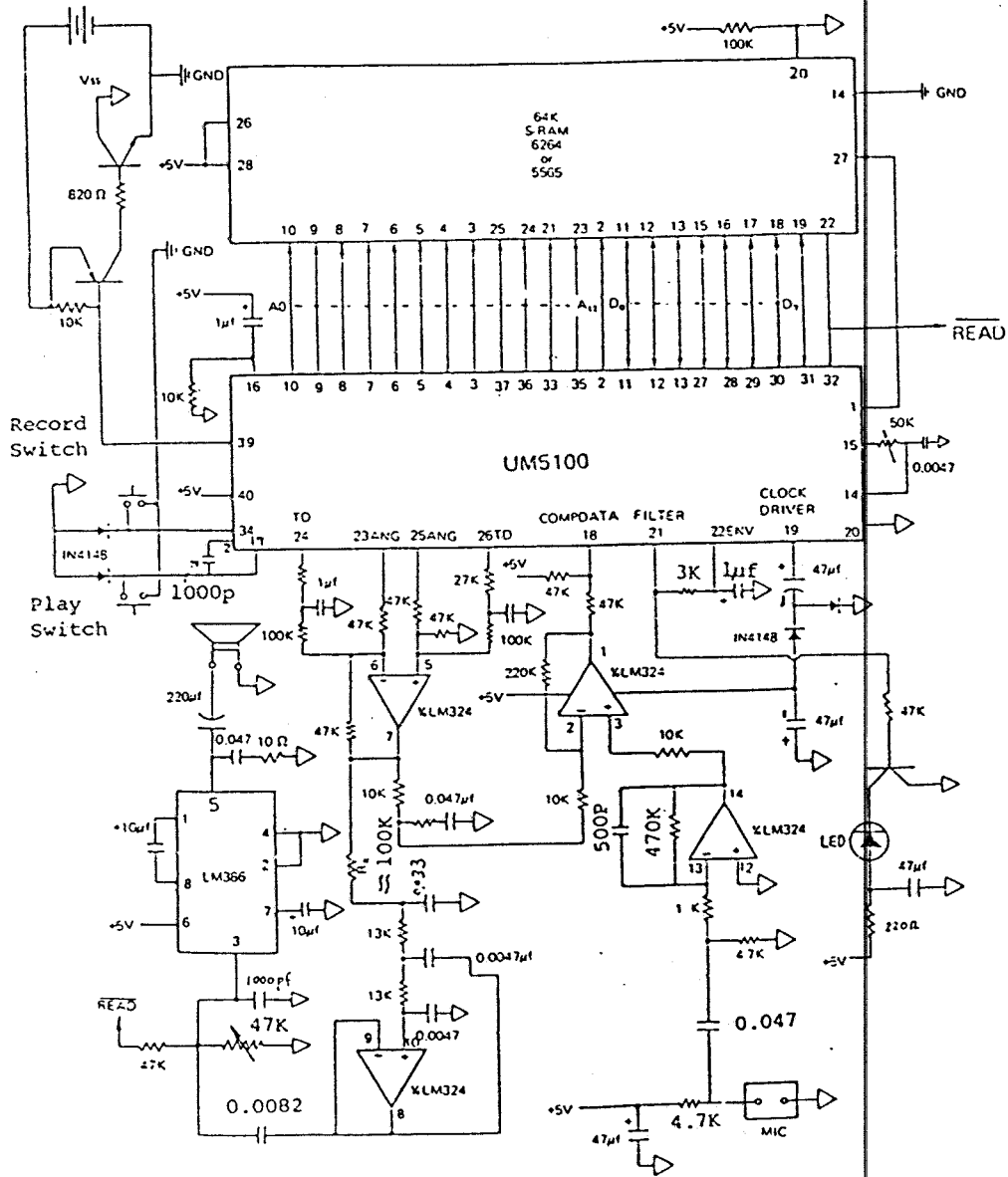
UM5100 Bonding Diagram





UM5100

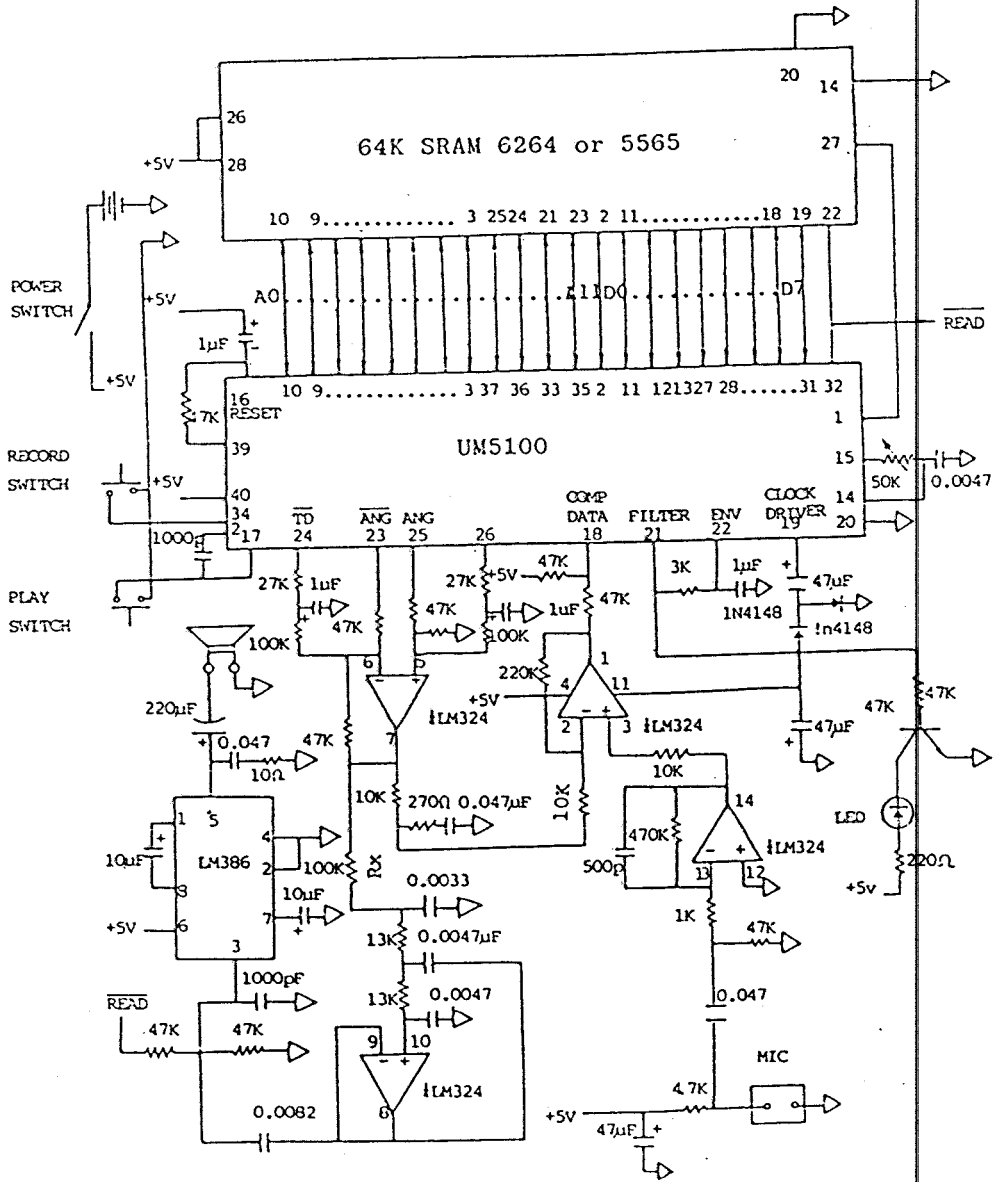
Speech Recording and Playback (With low stand-by Current)





UM5100

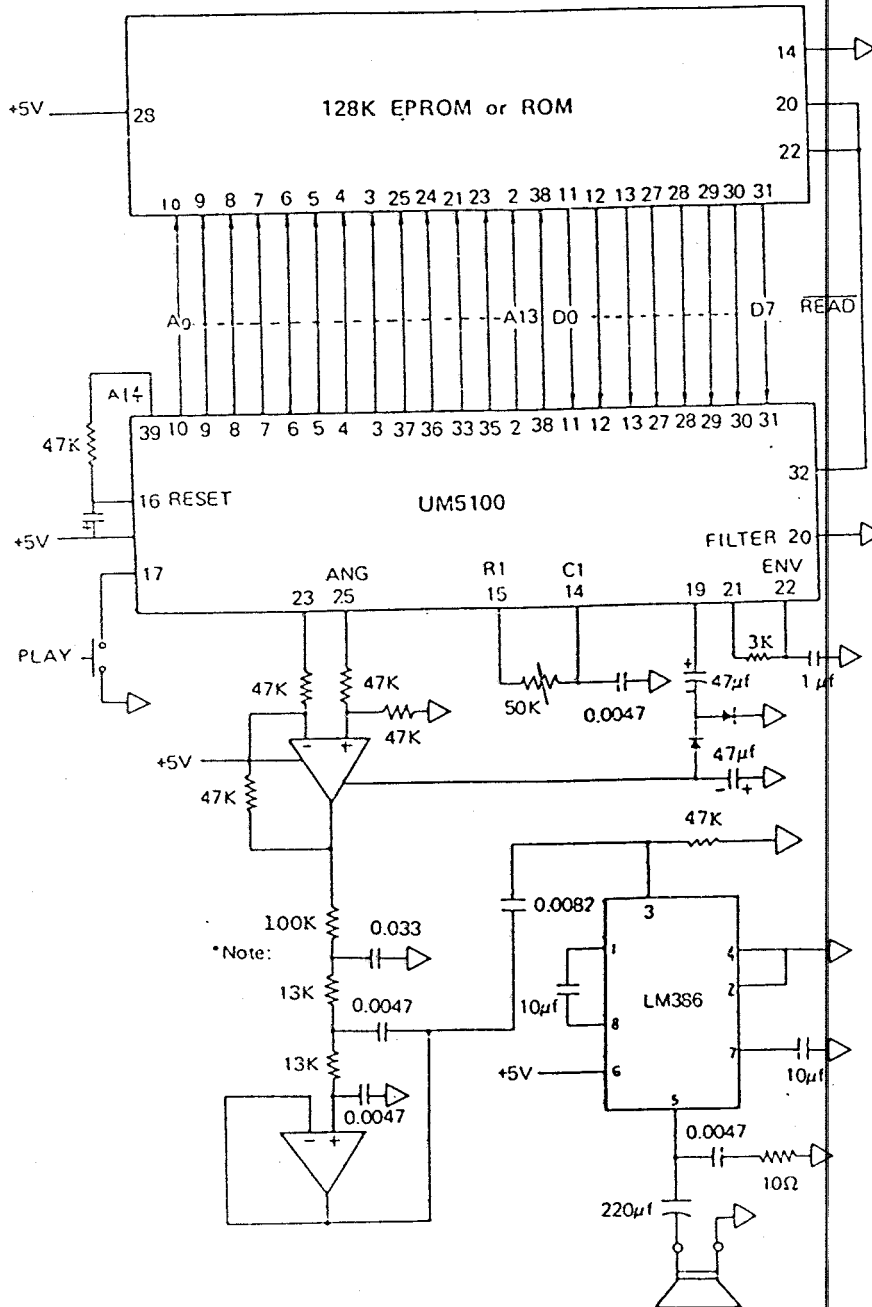
Speech Recording and Playback: (with Power Switch)





UM5100

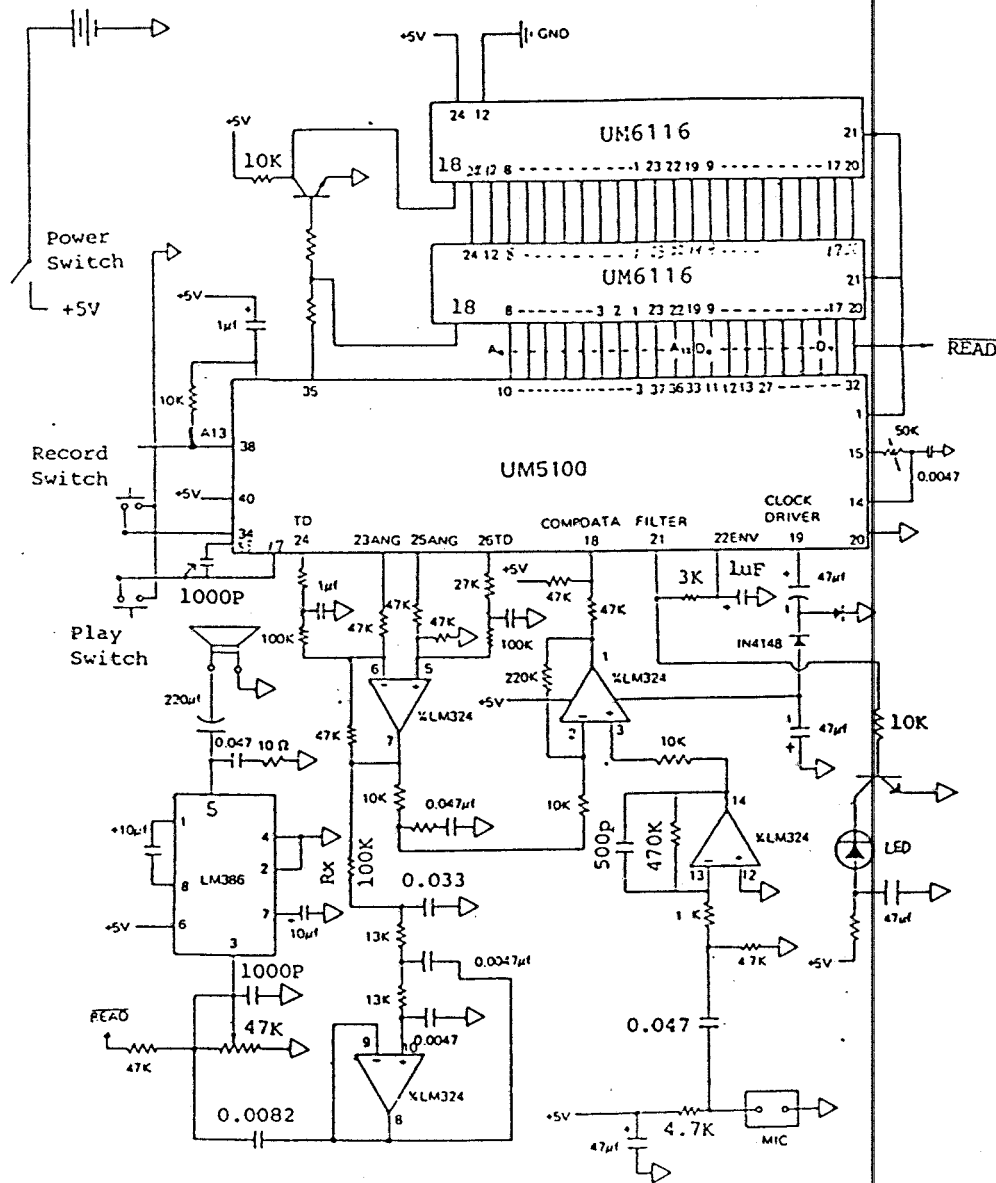
Speech Synthesis:



*Note:



Speech Recording and Playback
(With 16K SRAM x 2)



GENERAL PURPOSE SINGLE SUPPLY OPERATIONAL AMPLIFIER

**LM124/A/224/A/324/A/SAS34
LM158/A/258/A/358A**

DESCRIPTION

The LM124/SAS34 series consists of four independent, high gain, internally frequency compensated operational amplifiers designed specifically to operate from a single power supply over a wide range of voltages.

The LM158 series consists of two operational amplifiers designed as above. Operation from dual supplies is also possible for both series and the low power supply current drain is independent of the magnitude of the power supply voltage.

FEATURES

- Internally frequency compensated for unity gain
- Large dc voltage gain—(100dB)
- Wide bandwidth (unity gain)—1MHz (temperature compensated)
- Wide power supply range
Single supply—(3Vdc to 30Vdc) or dual supplies—(± 1.5 Vdc to ± 15 Vdc)

- Very low supply current drain—essentially independent of supply voltage (1mW/op amp at +5Vdc)
- Low input biasing current—(45nAdc temperature compensated)
- Low input offset voltage—(2mVdc) and offset current—(5nAdc)
- Differential input voltage range equal to the power supply voltage
- Large output voltage—(0Vdc to $V+ - 1.5$ Vdc swing)
- LM124 MII std 883A,B,C available

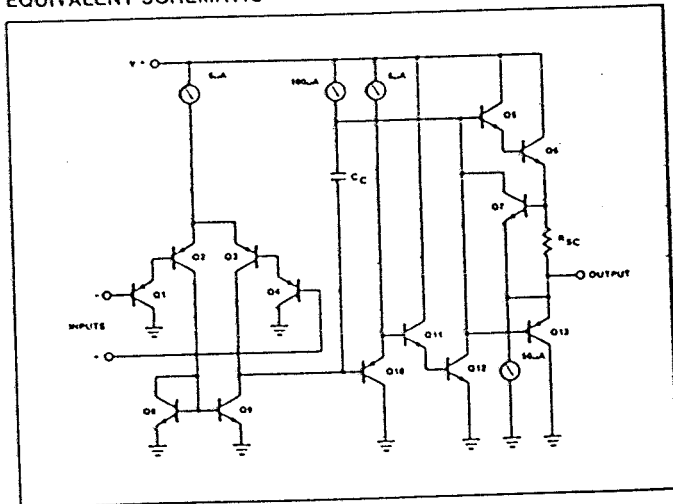
UNIQUE FEATURES

In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.

The unity gain cross frequency is temperature compensated.

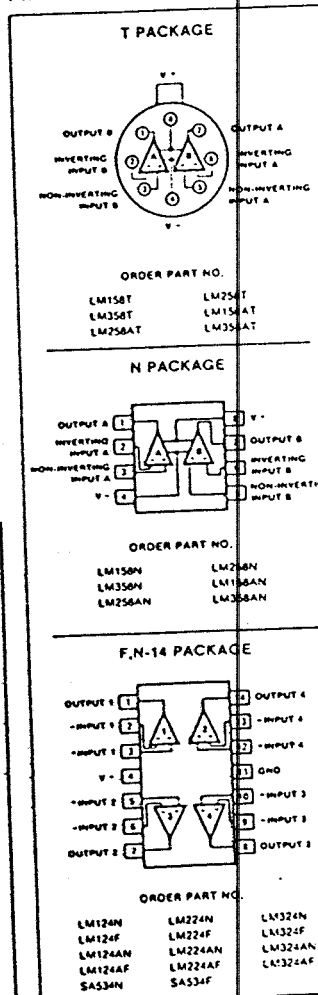
The input bias current is also temperature compensated.

EQUIVALENT SCHEMATIC



LM124/A/224/A/324/A/SAS34-F.N-14.T
LM158/A/258/A/358A-F.N.T

PIN CONFIGURATIONS

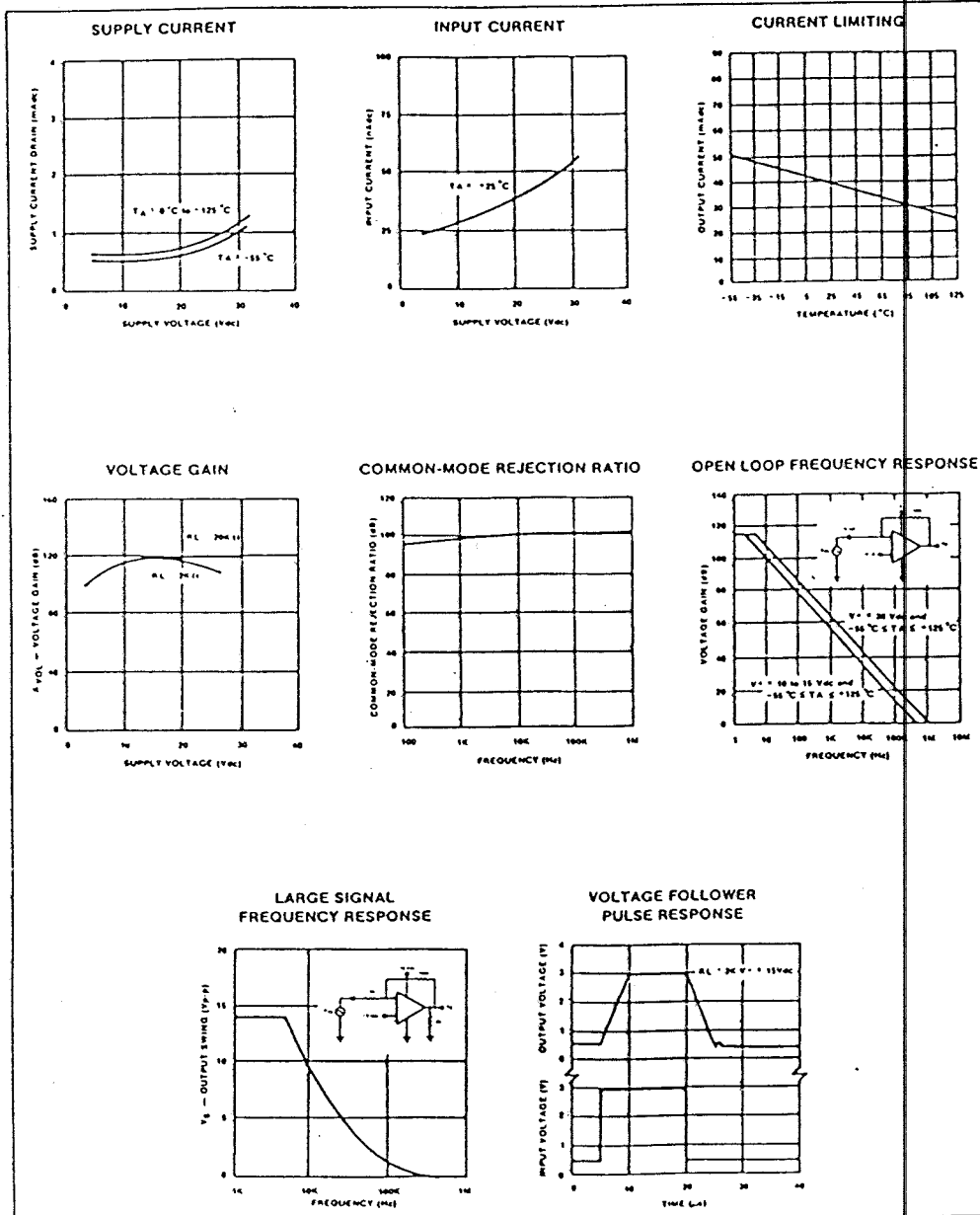


GENERAL PURPOSE SINGLE SUPPLY
OPERATIONAL AMPLIFIER

LM124/A/224/A/324/A/SAS34
LM158/A/258/A/358A

LM124/A/224/A/324/A/SAS34-F,N-14,T
LM158/A/258/A/358A-F,N,T

TYPICAL PERFORMANCE CHARACTERISTICS





LM386 Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value up to 200.

The inputs are ground referenced while the output is automatically biased to one half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

Features

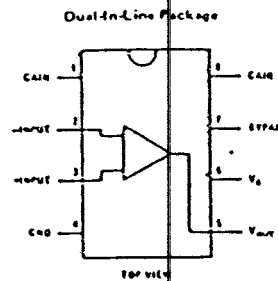
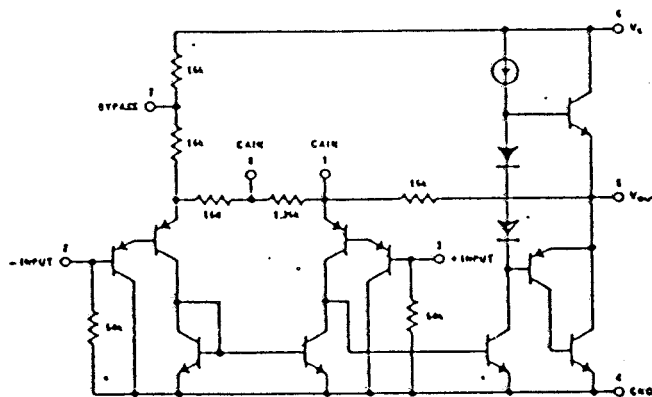
- Battery operation
- Minimum external parts
- Wide supply voltage range 4V-12V or 5V-18V
- Low quiescent current drain 4 mA

- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion
- Eight pin dual-in-line package

Applications

- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

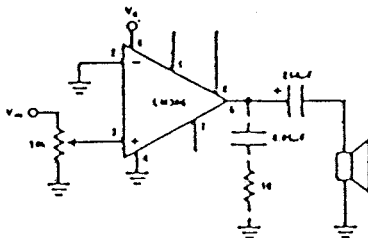
Equivalent Schematic and Connection Diagrams



Order Number LM386N-1,
LM386N-3 or LM386N-4*
See NS Package N08B

Typical Applications

Amplifier with Gain = 20
Minimum Parts



Amplifier with Gain = 200

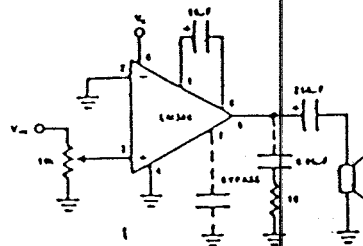


TABLE 4.3.3 Mono Power Amplifier Characteristics

Device	Supply Voltage (V)			Output Power (W) at 10% THD									Quiescent Current (mA)			Fixed Gain (dB)			Gain Control (Typ dB)	Output Protection	
				$R_L = 4\Omega$			$R_L = 8\Omega$			$R_L = 16\Omega$											
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max			
LM386 (8 Pin DIP) [LM389 ¹]	4	6 9 12	12 ⁴		0.32		0.25	0.32				0.18		4, (6 ²)	8, (12 ²)		26		YES (46)	No	
LM388 (14 Pin DIP)	4	6 9 12	12 ⁴	0.45	0.6			0.4				0.2					23	26	30	YES (46)	No
LM390 (14 Pin DIP)	4	6 9	10	0.8	1.0			0.6				0.34					23	26	30	YES (46)	No
LM380 (14 Pin DIP)	8	12 14 18	20		2.5			1.5				0.5					32	34	36	No	YES
LM384 (14 Pin DIP)	12	18 22	26		4.2			4.0				2.2						34		No	YES

¹ Specifications same for $T_A = 25^\circ\text{C}$. For operation at ambient temperatures greater than 25°C the IC must be derated based on a maximum 150°C junction temperature using a thermal resistance obtained from device data sheet.
² LM385 similar to LM386 but includes three additional NPN transistors and a feedback network for a distortion less than 1% Pin DIP.
³ THD = 3%
⁴ Parts selected for higher absolute maximum supply voltage available on special request.

HEX INVERTER

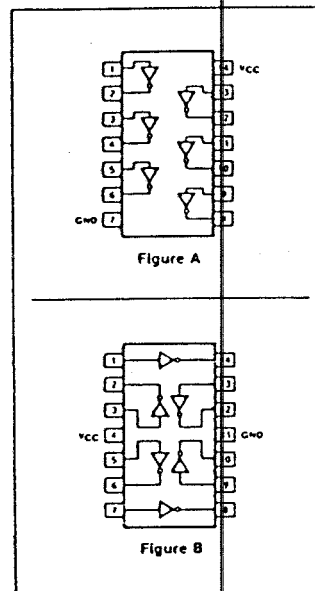
54/74 SERIES "04"

54/7404
54H/74H04
54S/74S04
54LS/74LS04

ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$; $T_A = 0^\circ C$ to $+70^\circ C$		MILITARY RANGES $V_{CC} = 5V \pm 10\%$; $T_A = -55^\circ C$ to $+125^\circ C$	
		Plastic DIP	Fig. A Fig. A	N7404N N74S04N	• N74H04N • N74LS04N
Ceramic DIP	Fig. A Fig. A	N7404F N74S04F	• N74H04F • N74LS04F	S5404F S54S04F	• S54H04F • S54LS04F
Flatpak	Fig. B Fig. A			S5404W S54S04W	• S54H04W • S54LS04W

PIN CONFIGURATIONS



INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	I_{IH} (μA)	40	50	50	20
	I_{IL} (mA)	-1.6	-2.0	-2.0	-0.36
Outputs	I_{OH} (μA)	-400	-500	-1000	-400
	I_{OL} (mA)	16	20	20	4/8(1)

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
I_{CCH}	Supply current	$V_{CC} = \text{Max}, V_{IN} = 0V$			12		26		24	2.5 mA
I_{CCL}	Supply current	$V_{CC} = \text{Max}, V_{IN} \geq 4.5V$			33		58		54	6.6 mA

AC CHARACTERISTICS $T_A = 25^\circ C$ (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		$C_L = 15 pF$ $R_L = 400 \Omega$		$C_L = 25 pF$ $R_L = 280 \Omega$		$C_L = 15 pF$ $R_L = 280 \Omega$		$C_L = 15 pF$ $R_L = 2k \Omega$		
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{PLH}	Propagation delay	Waveform 1			22		10		4.5	15 ns
t_{PHL}	Propagation delay	Waveform 1			15		10		5.0	15 ns

NOTE

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

8-INPUT NAND GATE

54/74 SERIES "30"

54/7430
54H/74H30
54LS/74LS30

PIN CONFIGURATIONS

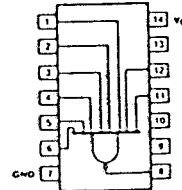


Figure A

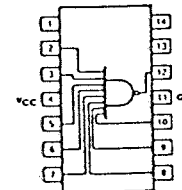


Figure B

ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES V _{CC} = 5V ± 5%; T _A = 0°C to +70°C	MILITARY RANGES V _{CC} = 5V ± 10%; T _A = -55°C to +125°C
Plastic DIP	Fig. A Fig. A	N7430N • N74H30N N74LS30N	
Ceramic DIP	Fig. A Fig. A	N7430F • N74H30F N74LS30F	S5430F • S54H30F S54LS30F
Flatpak	Fig. B Fig. A		S5430W • S54H30W S54LS30W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	I _{IN} (μA)	40	50		20
	I _L (mA)	-1.6	-2.0		-0.4
Outputs	I _{OH} (μA)	-400	-500		-400
	I _{OL} (mA)	16	20		4/8 ^(a)

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
I _{CCH}	Supply current V _{CC} = Max, V _{IN} = 0V		2.0		4.2				0.5	mA
I _{CCL}	Supply current V _{CC} = Max, V _{IN} ≥ 4.5V		6.0		10				1.	mA

AC CHARACTERISTICS T_A = 25°C (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		C _L = 15pF R _L = 400Ω		C _L = 15pF R _L = 280Ω				C _L = 15pF R _L = 2kΩ		
		Min	Max	Min	Max	Min	Max	Min	Max	
t _{PLH}	Propagation delay Waveform 1		22		10				15	ns
t _{PHL}	Propagation delay Waveform 1		15		12				20	ns

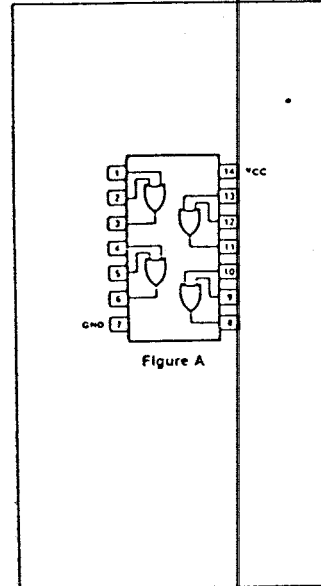
NOTE

- a The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b For family DC characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

QUAD 2-INPUT OR-GATE **54/74 SERIES "32"**

54/7432
54S/74S32
54LS/74LS32

PIN CONFIGURATION



ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$; $T_A = 0^\circ C$ to $+70^\circ C$	MILITARY RANGES $V_{CC} = 5V \pm 10\%$; $T_A = -55^\circ C$ to $+125^\circ C$
Plastic DIP	Fig. A Fig. A	N7432N • N74S32N N74LS32N	
Ceramic DIP	Fig. A Fig. A	N7432F • N74S32F N74LS32F	S5432F • S54S32F S54LS32F
Flatpak	Fig. A Fig. A		S5432W • S54S32W S54LS32W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (See Note a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
Inputs	I_{IH} (μA)	40		50	20
	I_{IL} (mA)	-1.6		-2.0	-0.36
Outputs	I_{OH} (μA)	-800		-1000	-400
	I_{OL} (mA)	16		20	4/8(a)

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (See Note b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
I_{CCH}	Supply current	$V_{CC} = \text{Max}, V_{IN} \geq 4.5V$			22		32		6.2	mA
I_{CCL}	Supply current	$V_{CC} = \text{Max}, V_{IN} = 0V$			38		68		9.8	mA

AC CHARACTERISTICS $T_A = 25^\circ C$ (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		$C_L = 15 \text{ pF}$ $R_L = 400 \Omega$				$C_L = 15 \text{ pF}$ $R_L = 280 \Omega$		$C_L = 15 \text{ pF}$ $R_L = 2 \text{ k}\Omega$		
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{PLH}	Propagation delay	Waveform 2			15		7.0		22	ns
t_{PHL}	Propagation delay	Waveform 2			22		7.0		22	ns

NOTE

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.

DUAL D-TYPE FLIP-FLOP

54/74 SERIES "74"

54/7474
54H/74H74
54S/74S74
54LS/74LS74

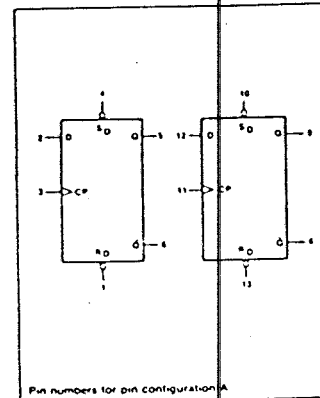
DESCRIPTION

The "74" is a Dual Positive Edge-Triggered D-Type Flip-Flop featuring individual data, clock, set and reset inputs; also complementary Q and Q outputs.

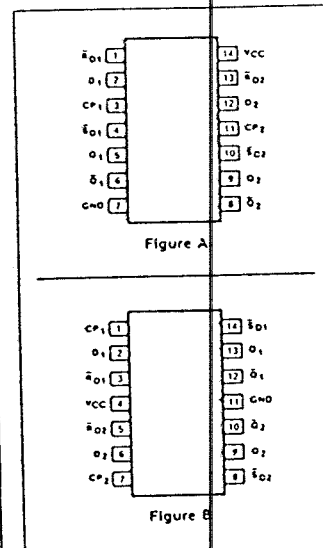
Set (SD) and Reset (RD) are asynchronous active LOW inputs and operate independently of the clock input. Information on the data (D) input is transferred to the Q output

on the LOW-to-HIGH transition of the clock pulse. The D inputs must be stable one setup time prior to the LOW-to-HIGH clock transition for predictable operation. Although the clock input is level sensitive, the positive transition of the clock pulse between the 0.8V and 2.0V levels should be equal to or less than the clock to output delay time for reliable operation.

LOGIC SYMBOL



PIN CONFIGURATION



ORDERING CODE (See Section 9 for further Package and Ordering Information.)

PACKAGES	PIN CONF.	COMMERCIAL RANGES		MILITARY RANGES	
		V _{CC} = 5V ± 5%; T _A = 0°C to +70°C		V _{CC} = 5V ± 10%; T _A = -55°C to +125°C	
Plastic DIP	Fig. A	N7474N	• N74H74N		
	Fig. A	N74S74N	• N74LS74N		
Ceramic DIP	Fig. A	N7474F	• N74H74F	S5474F	• S54H74F
	Fig. A	N74S74F	• N74LS74F	S54S74F	• S54LS74F
Flatpak	Fig. B			S5474W	• S54H74W
	Fig. A			S54S74W	• S54LS74W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (a)

PINS		54/74	54H/74H	54S/74S	54LS/74LS
D Data Input	I _{ih} (μA)	40	50	50	20
	I _{ik} (mA)	-1.6	-2.0	-2.0	-0.36
CP Clock Input	I _{ih} (μA)	80	100	100	40
	I _{ik} (mA)	-3.2	-4.0	-4.0	-0.8
SD Set Input	I _{ih} (μA)	80	100	100	40
	I _{ik} (mA)	-1.6	-2.0	-4.0	-0.8
RD Reset Input	I _{ih} (μA)	120	150	150	60
	I _{ik} (mA)	-3.2	-4.0	-6.0	-1.15
Q & Q-bar Output	I _{oh} (μA)	-400	-500	-1000	-400
	I _{ol} (mA)	16	20	20	4/8 (a)

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (b)

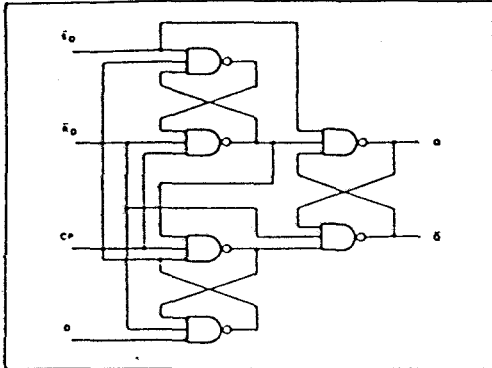
PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
I _{cc} Supply current	V _{CC} = Max V _{CP} = 0V	Mil	30		42		50		8.0	mA
		Com		30		50		50		8.0

NOTES

- a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
- b. For family dc characteristics, see inside front cover for 54/74 and 54H/74H and see inside back cover for 54S/74S and 54LS/74LS specification.

DUAL D-TYPE FLIP-FLOP 54/74 SERIES "74"

LOGIC DIAGRAM



MODE SELECT—TRUTH TABLE

OPERATING MODE	INPUTS				OUTPUTS	
	\bar{S}_D	\bar{R}_D	CP	D	Q	\bar{Q}
Asynchronous Set	L	H	X	X	H	L
Asynchronous Reset (Clear)	H	L	X	X	L	H
Undetermined (c)	L	L	X	X	H	H
Load "1" (Set)	H	H	↑	h	H	L
Load "0" (Reset)	H	H	↑	l	L	H

H = HIGH voltage level steady state.
 h = HIGH voltage level one setup time prior to the LOW to HIGH clock transition.
 L = LOW voltage level steady state.
 l = LOW voltage level one setup time prior to the LOW to HIGH clock transition.
 X = Don't care.

AC CHARACTERISTICS $T_A = 25^\circ\text{C}$ (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		$C_L = 15\text{ pF}$ $R_L = 400\ \Omega$		$C_L = 25\text{ pF}$ $R_L = 280\ \Omega$		$C_L = 15\text{ pF}$ $R_L = 280\ \Omega$		$C_L = 15\text{ pF}$ $R_L = 2\text{ k}\ \Omega$		
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{max} Maximum clock frequency	Waveform 3	15		35		75		25		MHz
t_{PLH} Propagation delay	Waveform 3		25		15		9.0		25	ns
t_{PHL} Clock to output	Waveform 3		40		20		9.0		40	ns
t_{PLH} Propagation delay	Waveform 5		25		20		6.0		25	ns
t_{PHL} Set or reset to output	Waveform 5 CP = HIGH		40		30		13.5		40	ns
t_{PHL} Set or reset to output	Waveform 5 CP = LOW		40		30		8.0		40	ns

AC SET UP REQUIREMENTS: $T_A = 25^\circ\text{C}$ (See Section 4 for Waveforms and Conditions.)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{w(H)}$ Clock pulse width (HIGH)	Waveform 3	30		15		6.0		25		ns
$t_{w(L)}$ Clock pulse width (LOW)	Waveform 3	37		13.5		7.3		15		ns
$t_{w(L)}$ Set or reset pulse width (LOW)	Waveform 5	30		25		7.0		25		ns
$t_s(H)$ Setup time (HIGH) data to clock	Waveform 3	20		10		3.0		25		ns
$t_s(L)$ Setup time (LOW) data to clock	Waveform 3	20		15		3.0		20		ns
t_h Hold time data to clock	Waveform 3	5.0		5.0		2.0		5.0		ns

NOTE

C Both outputs will be High while both \bar{S}_D and \bar{R}_D are LOW, but the output states are unpredictable if \bar{S}_D and \bar{R}_D go HIGH simultaneously.

1-OF-8 DECODER/DEMULTIPLEXER

54/74 SERIES "138"

54S/74S138
54LS/74LS138

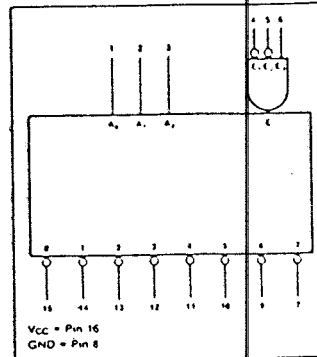
DESCRIPTION

The "138" is a HIGH speed 1-of-8 Decoder/Demultiplexer. The "138" is ideal for HIGH speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using only three "138" devices; or to a 1-of-32 decoder using four "138" devices and one inverter.

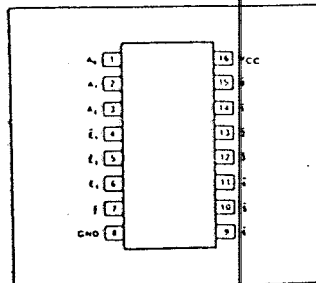
FEATURES

- Demultiplexing capability
- Multiple input enable for easy expansion
- Ideal for memory chip select decoding
- Direct replacement for Intel 3205

LOGIC SYMBOL



PIN CONFIGURATION



ORDERING CODE (See Section 9 for further Package and Ordering Information)

PACKAGES	COMMERCIAL RANGES V _{CC} = 5V ± 5%; T _A = 0°C to 70°C	MILITARY RANGES V _{CC} = 5V ± 10%; T _A = -55°C to +125°C
Plastic DIP	N74S138N • N74LS138N	
Ceramic DIP	N74S138F • N74LS138F	SS4S138F • SS4LS138F
Flatpak		SS4S138W • SS4LS138W

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE (a)

PINS	DESCRIPTION	54/74	54S/74S	54LS/74LS
A ₀ -A ₂	Address inputs I _H (μA) I _L (mA)		50 -2.0	20 -0.36
\bar{E}_1, \bar{E}_2	Enable (Active LOW) inputs I _H (μA) I _L (mA)		50 -2.0	20 -0.36
E ₃	Enable (Active HIGH) input I _H (μA) I _L (mA)		50 -2.0	20 -0.36
$\bar{O}-\bar{7}$	Decoder outputs I _{OH} (μA) I _{OL} (mA)		-1000 20	-400 4/8 ⁽¹⁾

NOTE

(a) The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.

1-OF-8 DECODER/DEMULTIPLEXER

54/74 SERIES "138"

FUNCTIONAL DESCRIPTION

The "138" decoder accepts three binary weighted inputs (A₀, A₁, A₂) and when enabled provides eight mutually exclusive active LOW outputs (0-7). The device features three Enable inputs: two active LOW (\bar{E}_1, \bar{E}_2) and one active HIGH (E₃). Every output will be HIGH unless \bar{E}_1 and \bar{E}_2 are LOW and E₃ is HIGH. This multiple Enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four "138's" and one inverter.

The device can be used as an eight output demultiplexer by using one of the active LOW Enable inputs as the data input and the remaining Enable inputs as strobes. Enable inputs not used must be permanently tied to their appropriate active HIGH or active LOW state.

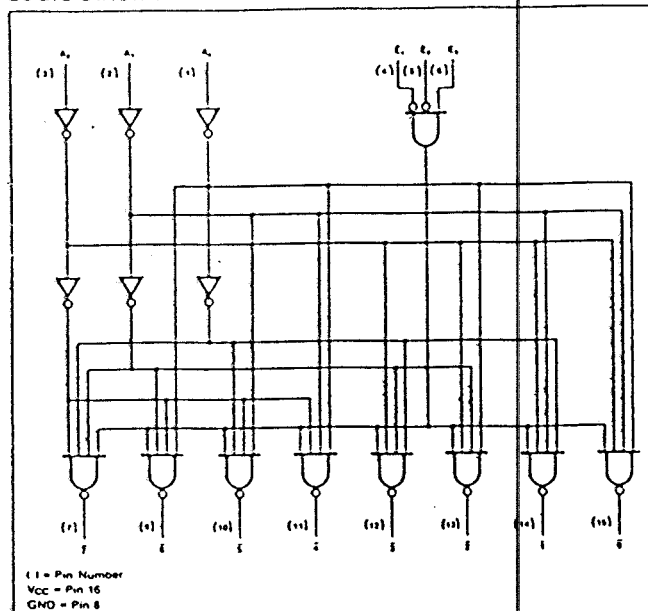
TRUTH TABLE

INPUTS						OUTPUTS							
\bar{E}_1	\bar{E}_2	E ₃	A ₀	A ₁	A ₂	0	1	2	3	4	5	6	7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

NOTES

- H = HIGH voltage level
- L = LOW voltage level
- X = Don't care

LOGIC DIAGRAM



DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (b)

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	
I _{cc}	Supply current				74		10	mA

NOTE

b. For family DC characteristics, see inside front cover for 54/74 and 54LS/74LS, and see inside back cover for 54S/74S and 54LS/74LS specifications.

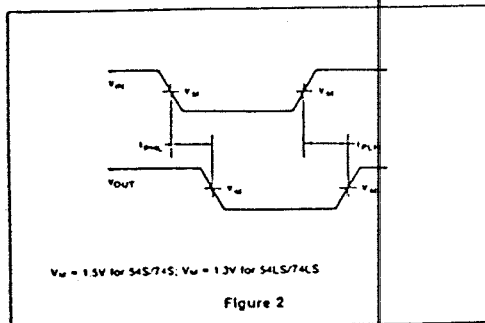
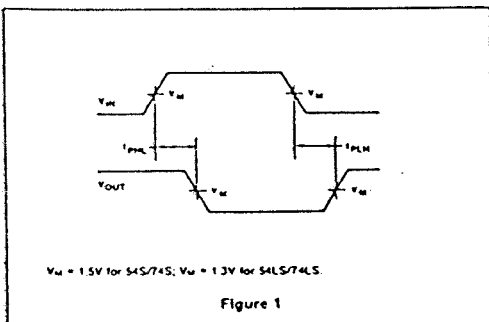
1-OF-8 DECODER/DEMULTIPLEXER

54/74 SERIES "138"

AC CHARACTERISTICS $T_A = 25^\circ\text{C}$ (See Section 4 for Waveforms and Conditions)

PARAMETER	TEST CONDITIONS	54/74		54S/74S		54LS/74LS		UNIT
				$C_L = 15\text{ pF}$ $R_L = 280\Omega$		$C_L = 5\text{ pF}$ $R_L = 2k\Omega$		
		Min	Max	Min	Max	Min	Max	
t_{PLH} t_{PHL}	Propagation delay Address to output				12		27 39	ns ns
t_{PLH} t_{PHL}	Propagation delay E_1 and E_2 to output				8.0 11		18 32	ns ns
t_{PLH} t_{PHL}	Propagation delay E_3 to output				11 11		26 38	ns ns

AC WAVEFORMS



OCTAL TRANSCEIVERS (3-STATE)

54/74 SERIES "245"

54LS/74LS245 (Preliminary Data)

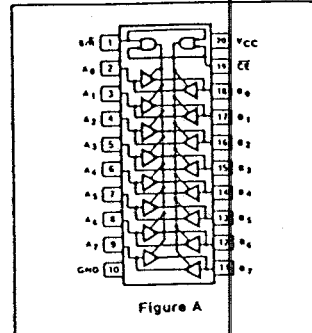
ORDERING CODE (See Section 9 for further Package and Ordering Information)

PACKAGES	PIN CONF.	COMMERCIAL RANGES V _{CC} =5V ± 5%; T _A =0°C to +70°C	MILITARY RANGES V _{CC} =5V ± 10%; T _A =-55°C to +125°C
Plastic DIP	Fig. A	N74LS245N	
Ceramic DIP	Fig. A	N74LS245F	S54LS245F
Flatpak			

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE^(a)

PINS	54/74	54H/74H	54S/74S	54LS/74LS
Inputs	I _H (μA)			20
	I _L (mA)			-0.2
Outputs	I _{OH} (μA)			-12/-15 ^(a)
	I _{OL} (mA)			12/24 ^(a)

PIN CONFIGURATION



FUNCTION TABLE

INPUTS		INPUTS/OUTPUTS	
CE	S/R	A _n	B _n
L	L	A = B	INPUTS
L	H	INPUT	B = A
H	X	(Z)	(Z)

H = HIGH voltage level
L = LOW voltage level
X = Don't care
(Z) = High impedance "off" state

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE^(b)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
V _{OH} Output HIGH voltage	V _{CC} = Min, I _{OH} = -3.0mA							2.4		V
	V _{IN} = 2V, I _{OH} = -12mA							2.0		V
	V _{CE} = V _{IL} , I _{OH} = -15mA ^(c)							2.0		V
V _{OL} Output LOW voltage	V _{CC} = Min, I _{OL} = 12mA								0.4	V
	V _{IN} = V _{IL} , I _{OL} = 24mA ^(c)								0.5	V
I _{OS} Output short circuit current	V _{CC} = Max, V _{OUT} = 0V							-40	120	mA
I _{CCH} Supply current HIGH	V _{CC} = Max, V _A = 4.5V, V _{CE} = 0V, V _{S/R} = 0V								70	mA
I _{CCL} Supply current LOW	V _{CC} = Max, V _A = 0V, V _{CE} = 0V, V _{S/R} = 0V								90	mA
I _{CCZ} Supply current "off"	V _{CC} = Max, V _A = V _B = open, V _{CE} = 4.5V, V _{S/R} = 0V								95	mA

- NOTES
- The slash numbers indicate different parametric values for Military/Commercial temperature ranges respectively.
 - For family DC characteristics see inside front cover for 54/74 and 54H/74H, and see inside back cover for 54S/74S and 54LS/74LS specification.
 - This parameter for Commercial Range only.

OCTAL TRANSCEIVERS (3-STATE)

54/74-SERIES "245"

AC CHARACTERISTICS: $T_A = 25^\circ\text{C}$ (See Section 4 for Waveforms and Conditions)

PARAMETER	TEST CONDITIONS	54/74		54H/74H		54S/74S		54LS/74LS		UNIT
								CL = 45 pF RL = 66 Ω		
		Min	Max	Min	Max	Min	Max	Min	Max	
tPLH Propagation delay	Waveform 2									12 ns
tPHL Propagation delay	Waveform 2									12 ns
tPZH Enable to HIGH	Waveform 8									40 ns
tPZL Enable to LOW	Waveform 7									40 ns
tPHZ Disable from HIGH	Waveform 6	CL = .45pF								35 ns
		CL = 5pF(d)								18 ns
tPLZ Disable from LOW	Waveform 7	CL = 45pF								30 ns
		CL = 5pF(d)								25 ns

NOTE
 1. These tests are for reference only. They represent the delay time to guarantee that a device is disabled and can no longer drive the bus.

LAMPIRAN D
USULAN TUGAS AKHIR

USULAN TUGAS AKHIR

1. Judul Tugas Akhir : PERENCANAAN DAN PEMBUATAN
SINTESA SUARA UNTUK BAHASA
INDONESIA DENGAN IC UM 5100.
2. Bidang Studi : ELEKTRONIKA
3. Ruang Lingkup : - Elektronika Mikro
- Mikroprosesor
- Organisasi Komputer
- Software
4. Latar Belakang : Dewasa ini telah banyak beredar
berbagai macam IC untuk
pemrosesan suara. Yang cukup
banyak dikenal dan dipakai
adalah Votrax SC 01 Speech
Synthesizer. IC ini merupakan
sebuah Phoneme Speech
Synthesizer yang didisain untuk
mensintesa suara dalam bahasa
Inggris. Meskipun dapat juga
digunakan untuk mensintesa suara
dalam Bahasa Indonesia, namun
suara yang dihasilkan kurang
sempurna.
IC jenis lain adalah UM 5100
yang termasuk jenis Natural
Speech Analysis/Synthesis. IC

ini mampu merekam dan mereproduksi suara dengan mengubah sinyal suara menjadi bentuk digital yang kemudian disimpan dalam SRAM/EPROM. Bila diperlukan sinyal digital ini dapat diubah kembali menjadi suara.

Dengan metode ini diharapkan sintesa suara -dalam bahasa apapun- yang diperoleh menjadi lebih sempurna.

Dengan menggunakan komputer, maka suara yang telah diubah dalam bentuk digital dapat diproses untuk berbagai aplikasi sesuai keinginan pemakai.

5. Penelaahan Studi : - Mempelajari macam-macam sintesa dan analisa suara.
- Mempelajari organisasi komputer jenis IBM PC XT.
- Mempelajari teknik interfacing.
6. Tujuan : Merencanakan dan membuat rangkaian sintesa suara untuk bahasa Indonesia yang

dihubungkan dengan komputer IBM PC XT dan mempunyai kemampuan antara lain :

- merekam dan mereproduksi suara
- konversi dari teks ke suara.

7. Langkah-langkah : - Studi literatur

- Perencanaan Hardware dan Software
- Pembuatan Hardware dan Software
- Penulisan naskah Tugas Akhir

8. Relevansi : Rangkaian sintesa suara untuk bahasa Indonesia ini dapat dikembangkan untuk berbagai aplikasi, misalkan :

- Alat bantu bagi penderita cacat
- Pemakaian dalam bidang industri dan robotika

