

TUGAS AKHIR

**PERENCANAAN DAN PEMBUATAN
ALAT PENERIMA TAMU ELEKTRONIK
BERBASIS MIKROKONTROLLER AT89S8252**



**Disusun Oleh :
WAHYU SETIYAWAN
03.57.033**



**KONSENTRASI TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO DIPLOMA III
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
MARET 2007**

SECRET
FEDERAL BUREAU OF INVESTIGATION
UNITED STATES DEPARTMENT OF JUSTICE
COMMUNICATIONS SECTION
WASHINGTON, D. C.

SECRET
COMMUNICATIONS SECTION
WASHINGTON, D. C.



COMMUNICATIONS SECTION
UNITED STATES DEPARTMENT OF JUSTICE
WASHINGTON, D. C.

SECRET

LEMBAR PERSETUJUAN

PERENCANAAN DAN PEMBUATAN ALAT PENERIMA TAMU ELEKTRONIK BERBASIS MIKROKONTROLLER AT 89S8252

TUGAS AKHIR

*Disusun dan diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma
Teknik Elektronika Diploma - III (D – III)*

Disusun Oleh :

NAMA : WAHYU SETIYAWAN
NIM : 03.57.033

Mengetahui,
Ketua jurusan Teknik Elektro D-III


Menyetujui,
Dosen Pembimbing



(**H. Choirul Saleh, MT**)

(**Bambang Prio H. ST, MT**)

**KONSENTRASI TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO DIPLOMA III
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
MARET 2007**



Berbaktillah kepada mereka (Ibu dan Bapak) dengan penuh kerendahan diri dan ucapkanlah " Wahai Tuhanku " kasihilah mereka berdua, sebagaimana mereka memeliharaaku dengan penuh kasih sayang sewaktu masih kecil.

Tuhan Tidak melakukan kejahatan, dia memberi akal dan pengetahuan kepada kita sehingga kita bisa di bimbing untuk melawan perangkap-perangkap kesalahan dan kehancuran.

**Ingatlah 5 perkara sebelum 5 perkara
Sehat sebelum Sakit
Muda sebelum Tua
Kaya sebelum Miskin
Lapang sebelum Sempit
Hidup sebelum Mati**



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Wahyu Thanks For
Allah SWT Sang Pencipta....Limpahan Rahmatmu...
Hidayahmu...Izinmu...

Kubisa Menyelesaikan ini semua dengan baik

Puji Syukur kehadiran Allah SWT, yang telah memberi
Rahman dan Rohim-Nya kepada junjungan
Nabi besar Muhamadd SAW beserta keluarga dan para
sahabat beliau. Dan tak lupa saya ucapkan
Terima Kasih kepada:

>.Kedua orang Tua koe tercinta :

**ABADI BUDIANTO
TARMIATI**

Yang telah merawatku mulai aku lahir sampai sekarang,
Terima kasih atas segala kasih sayang dan cintamu, serta
biaya meteri dan do'a yang tulus ikhlas yang telah diberikan
selama ini. Kupersembahkan karya kecil ini untuk kalian.

>.Tuk kakak koe Ari M.R, Makasih atas Do'a serta biaya yang
di berikan selama ini juga dukungannya. Sukses buat Mas Ari.

>.Tuk kakak koe Dedy dan Mbak Erika. Makasih mas atas
dukungan yang di berikan, maaf selama ini dah ngerepoti.
Moga si kecil Falero cepet besar.....!

>.Belahan Jiwa koe Puspita Dewi Yanti (Tita) Thanks atas
dukungan, semangat yang diberikan serta doa Tulus ikhlas
juga kasih sayang & cinta yang diberikan ke aQ,
akhirnya Q bisa lulus juga...!!

>.Terima Kasih untuk keluarga di Pandaan atas perhatian dan
dukungannya selama ini.

>.Tuk Mas Yudi Makasih tlah membantu menyelesaikan
Tugas AkhirQ dari awal
hingga akhir.....!

>.Spesial Thank's untuk sobat karibq Andi yang telah banyak
membantu penyelesaian tugas akhirq, makasih juga buat
komputernya, gak akan q lupa kebaikan u....., semoga
persahabatan kita nggak akan berakhir sampai disini, key.....!!!

ily wanna be

- >. Tuk Sobat koe kontrakan Griyashanta G 204 :
Bayu-See_204 (terus berkreatifitas n berkarya, cepetan nyusul ya lulus....)
 - Arik-SheeWo (piye mbe jadi kan beli kameranya...???)
 - Yudi-Kopleh (akhirnya selesai juga ya kulnya)
 - Lukmawan-jamin (sukses buat kamu, jangan suka merusak pintu yo bozz!!!!)
 - Endin-Gendon (tetep semangat kul..., jo mulah mulih wae, ojo sering ngapusi cah cilik2....)
 - Feri-peyek (jangan menyerah tetep optimis...)
 - Andra-Mendring (kapan selesai-kulnya)
 - Rangga-Ndondo (jo Tidur kelamaan, suwun printemya)
- Makasih atas bantuan kalian semuanya, jangan lupain aQ ya sobat, persahabatn kita semua jangan pernah berakhir sampai di sini ya, sukses buat kalian semuanya.....!!!

Gamegraps 204

- >. Boeat kang Anjas, makasih dah beri dorongan semangat, kapan punya momongan...?!!!
- >. Buat ganyonk su pengky (ojo kemplo-kemplo, sing wis rabi yo wis ben!!!)
- >. To Kidu (du insap...but keep'in rock n roll!!!)
- >. Boeat Deena, Thanks Denn dah di semangatn n akhirnya koe bisa nyusul kamu LULUS bareng.....!!! Moga Persahabatan kita gak akan putus.....!!!
- >. Yuli, Diaf ma Desi, tetep semangat kul n cepetan nyusul lulus...!!!
- >. Albert-Gempil & Mbak Desi makasih atas bantuannya selama ini, moga kalian bisa bersatu selamanya, jangan lupa nti di undang, key.....!!!

> Pak Choirul juga Pak Eko, Makasih dah menguji saya, walaupun dengan perdebatan yang lama, akhirnya saya bisa lulus.....!!!

> Pak Bambang, makasih atas bimbingan TA-nya selama ini, meskipun lama akhirnya bisa selesai juga TA-nya.....!!!

> Temen-temen seperjuangan koe elektronika D-III 2003, Dayat, Andi, Yuda, Syahrul, (akhirnya kita bisa lulus dan wisuda bareng), Reza(sukses buat kamu,jangan lupa kalau dah jadi orang sukses), Nurman(cepatan nyusul ya, jangan kelamaan), Suharyono-Ambon (tetep semangat kerjakan TA), Arip-Penceng(kamu bisa Rip nyusul Aq...), Ditto(makasih atas segala bantuannya selama ini), Venny(jangan lupain Aq ya....). Kalian adalah sobat baik koe, jangan sampai berhenti persahabatan kita....???

Koe seneng bisa kenal kalian semua.....!!!

> Tuk Dyanning soulmateku makasih untuk dorongan semangatnya, gimana kita bisa lulus bareng kan.....???

> Akhirnyasemua beban yang slama nie kutanggung dapat koe selesaikan. Tapi jalan masih panjang, bersemangatlah!!! Oiya lupa buat si "LG 14 inch" dan juga si "Astrea Impresa AE"(mereka adalah saksi bisu yang tau apa yang aQ lakukan & kemana aQ pergi.....!!!

Dan semua pihak yang telah banyak membantu Aq, yang mana tidak dapat di tulis satu persatu, Aq ucapkan terima kasih.



KATA PENGANTAR

Dengan mengucapkan Alhamdulillah, puja dan puji syukur kehadirat-Mu Ya Allah SWT atas berkat rahmat, hidayah dan segala karunia-Nya, sehingga dapat menyelesaikan tugas akhir yang berjudul “ **PERENCANAAN DAN PEMBUATAN ALAT PENERIMA TAMU ELEKTRONIK BERBASIS MIKROKONTROLER AT89S8252** “ ini dengan lancar. Tugas Akhir ini dikerjakan sebagai salah satu syarat untuk menyelesaikan studi di Jurusan Teknik Elektro Diploma III, Konsentrasi Teknik Elektronika Diploma III Institut Teknologi Nasional Malang.

Keberhasilan penyelesaian laporan Tugas Akhir ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penulis menyampaikan ucapan terima kasih kepada :

1. Bapak Ir. H. Choirul Saleh, MT selaku Ketua Jurusan Teknik Elektro DIII.
2. Bapak Bambang Prio H.ST,MT selaku Sekretaris Jurusan Teknik Elektro D-III.
3. Bapak Bambang Prio H.ST,MT selaku Dosen Pembimbing Tugas Akhir.
4. Buat semua Dosen – Dosen di jurusan teknik elektronika D-III Terima kasih atas ilmu yang diberikan kepada saya sehingga bisa menjadi manusia yang berguna.

5. Keluarga, kedua orang tua, saudara-saudara dan kakak yang telah banyak memberikan do'a restu, dorongan semangat, dan biaya baik secara moril maupun materil.
6. Teman – Teman semua, Elektronika D-III angkatan 2003, Makasih atas semua bantuannya.
7. Semua teman – teman Kontrakan GS G 204, Bayu (siong), Arik (shewo), Lukmawan (jamin), Yudi (kopleh), Endin (cemeng), Fery (peyek), Andra (mendring), Rangga (ndondos) dan Deena Terima Kasih atas bantuan kalian semua.

Meskipun telah dikaji ulang dan dikerjakan dengan sungguh-sungguh, namun penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, karena keterbatasan pengetahuan dan keterampilan yang penulis miliki, sehingga segala kritik dan saran penulis terima untuk dijadikan pedoman didalam menyusun laporan berikutnya yang lebih sempurna.

Akhir kata semoga Tugas Akhir ini dapat memberikan manfaat bagi seluruh masyarakat ilmiah pada khususnya dan pembaca pada umumnya.

Malang, Maret 2007

Penulis

ABSTRAKSI

“ PERENCANAAN DAN PEMBUATAN ALAT PENERIMA TAMU ELEKTRONIK ”

Wahyu Setiyawan, 03.57.033. Laporan Akhir, Jurusan Teknik Elektronika D-III, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang 2007. Dosen Pembimbing : Bambang Prio H.ST,MT. E-mail : wahyu_sipe@yahoo.com

Seiring perkembangan jaman yang semakin maju membuat masyarakat sering meninggalkan rumah dikarenakan aktifitasnya. Dalam menjalani kegiatan di luar rumah, sering kali penghuni enggan untuk meninggalkan rumah terlalu lama. Selain faktor keamanan, terdapat faktor lainnya yaitu terhambatnya komunikasi dan informasi antara tamu dengan penghuni rumah. Untuk mendapatkan efisiensi dan efektivitas dalam menjalani kegiatan di luar rumah, maka dibuat suatu alat penerima tamu elektronik. Adapun tujuan dari pembuatan alat ini adalah untuk memastikan bahwa jika tamu mencari salah satu penghuni rumah yang ia maksud, jika orangnya ada di rumah, pasti tamu akan ketemu langsung dengan penghuni rumah yang ia cari. Pilihan Alternatif jika seluruh penghuni rumah tidak berada di rumah maka tamu bisa meninggalkan pesan.

Alat penerima tamu elektronik ini terdiri dari Mikrokontroler AT89S8252 sebagai pengontrol utama, LCD sebagai media penampil pesan dan Suara yang dihasilkan ketika dilakukan pemanggilan kepada penghuni rumah lewat speaker pemanggil adalah berasal dari IC ISD 2560 yang sebelumnya telah diisi suara dengan kapasitas penyimpanan maksimum 60 detik, serta Keypad sebagai media penulis. Prinsip kerja alat ini adalah ketika penghuni berada di rumah maka LCD akan menampilkan daftar nama-nama anggota keluarga yang berada di rumah. Jika penghuni tidak berada di rumah maka tamu dapat menuliskan pesannya dengan cara tamu harus menekan tombol bel dahulu, tekan tombol ENT pada keypad untuk memulai proses penulisan serta tekan tombol-tombol yang ada pada keypad untuk melakukan penulisan dan terakhir tekan tombol ENT agar pesan yang tertulis dapat tersimpan.

Dengan melalui pengujian pada alat penerima tamu elektronik ini yang mana pengujian dilakukan pada setiap blok rangkaian yang ada dengan hasil pengujian yang sudah benar, seperti salah satu hasil pengujian LCD yang data keluarannya sesuai yang diinginkan. Pembuatan alat penerima tamu elektronik ini dari perangkat keras dan lunak sudah sesuai dengan apa yang direncanakan, sehingga alat yang telah dibuat keseluruhannya akan berfungsi sesuai apa yang diinginkan

Kata kunci : Keypad, Mikrokontroler AT 89S8252, LCD M1632, ISD 2560.

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PENGESAHAN	ii
KATA PENGANTAR.....	iii
ABSTRAKSI.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
DAFTAR FLOWCHART	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Metodologi Penulisan.....	4
1.6 Sistematika Pembahasan	5
BAB II LANDASAN TEORI	6
2.1 Pendahuluan.....	6
2.1.1 Mikrokontroler AT89S8252.....	6
2.1.2 Konfigurasi Kaki – Kaki MCU AT89S8252	10
2.1.3 Karakteristik Osilator	14

2.1.4 Organisasi Memori MCU AT89S8252	14
2.1.4.1 Memori Program	15
2.1.4.2 Memori Data	16
2.1.4.3 EEPROM Dan RAM.....	18
2.1.4.3.1 Register Kontrol Untuk EEPROM.....	19
2.1.4.3.2 Register Dual Data Pointer.....	19
2.1.4.3.3 Memprogram Flash Dan EEPROM	19
2.1.5 SFR (Special Function Register).....	20
2.1.5.1 SFR Untuk Timer 2.....	24
2.1.5.2 SFR Untuk Wacthdog Dan Memori.....	25
2.1.5.3 SFR Pengontrol SPI (Serial Peripheral Interface).....	26
2.1.5.4 Power Control (PCON).....	27
2.1.6 UART (Universal Asynchronous Transmitter/Receiver).....	28
2.1.6.1 Pengiriman Data Secara Parallel	28
2.1.6.2 Pengiriman Data Secara Serial.....	28
2.1.7 Sistem Interrupt.....	29
2.1.8 Timer Dan Counter	30
2.1.9 WDT (Programmable Watchdog Timer)	33
2.1.10 Timer 2.....	34
2.1.11 Metode Pengalamatan	35
2.1.12 Bahasa Assembler MCS – 51.....	36

2.2 ISD (Information Storage Devices) 2560	37
2.2.1 Konfigurasi Pin – Pin ISD 2560.....	41
2.3 <i>Liquid Crystal Display</i> (LCD) M1632.....	44
2.3.1 Register	47
2.3.2 Busy Flag	48
2.3.3 Address Counter	48
2.3.4 DDRAM.....	48
2.3.5 CGRAM	49
2.4 Keypad	51
2.5 IC Audio Amplifier LM 386.....	52
2.5.1 Penguatan Yang Terkontrol	53
2.5.2 Bias Masukkan	54
2.6 Loudspeaker	55
2.6.1 Bagian – Bagian Dari Loudspeaker	55
2.6.2 Ukuran Loudspeaker	56
2.7 Push Button.....	57
2.8 LED (Light Emitting Dioda).....	57

BAB III METEDOLOGI PENELITIAN.....	59
3.1 Pendahuluan.....	59
3.2 Perencanaan Perangkat Keras	59
3.2.1 Cara Kerja Rangkaian Keseluruhan	60
3.2.1.1 Prinsip Kerja Alat Bila Penghuni Tidak Ada Di Rumah	60
3.2.1.2 Prinsip Kerja Alat Untuk Penghuni Yang Ada Di Rumah	61
3.2.2 Fungsi Komponen Dari Rangkaian Sistem	61
3.3 Perencanaan Rangkaian Keypad Matrik 4 x 4	63
3.4 Perencanaan Rangkaian Tombol.....	64
3.5 Perencanaan Minimum Sistem AT89S8252	65
3.5.1 Rangkaian Clock Internal.....	67
3.5.2 Rangkaian Reset.....	68
3.6 Perencanaan Rangkaian LCD M1632.....	70
3.7 Perencanaan Rangkaian ISD 2560	72
3.7.1 Mode Recording.....	73
3.7.2 Mode Playback.....	74
3.8 Perencanaan Rangkaian LM 386	75
3.9 Perencanaan Rangkaian LED Indikator	76
3.10 Perencanaan Perangkat Lunak	77

3.10.1	Proses Inisialisasi	78
3.10.2	Proses Pemutaran Dan Perekaman Suara Serta Perubahan Status	80
3.10.3	Proses Pembacaan Pesan.....	82
3.10.4	Proses Pemanggilan Dan Penulisan Pesan.....	84
BAB IV PENGUJIAN ALAT.....		86
4.1	Pendahuluan	86
4.2	Pengujian Rangkaian Keypad Matrik 4 x 4	86
4.2.1	Prosedur Pengujian.....	87
4.2.2	Hasil Pengujian	87
4.3	Pengujian Rangkaian Minimum Sistem AT 89S8252	90
4.3.1	Hasil Pengujian	90
4.4	Pengujian Rangkaian LCD (Liquid Crystal Display)	91
4.4.1	Tujuan	91
4.4.2	Peralatan Yang Dibutuhkan	91
4.4.3	Prosedur Pengujian.....	91
4.4.4	Hasil Pengujian	92
4.5	Pengujian Rangkaian ISD 2560	93

4.5.1 Tujuan	93
4.5.2 Peralatan Yang Digunakan.....	93
4.5.3 Langkah Pengujian IC ISD	95
4.5.4 Hasil Pengujian	96
4.6 Pengujian Rangkaian Keseluruhan	96
BAB V PENUTUP.....	99
5.1 Kesimpulan	99
5.2 Saran.....	99
DAFTAR PUSTAKA.....	xviii
GAMBAR KESELURUHAN	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

No	Keterangan Gambar	Halaman
1.	Blok Diagram AT89S8252	9
2.	Bentuk Fisik AT89S8252.....	10
3.	Karakteristik Oscillator	14
4.	Memori Program MCS-51	15
5.	Memori Data MCS-51.....	16
6.	Memori Data Internal (RAM) Dan SFR MCS-51	17
7.	Ruang Special Function Register	22
8.	Koneksi Master Atau Slave SPI.....	26
9.	Blok Diagram ISD 2560.....	39
10.	Konfigurasi Pin Dari ISD 2560.....	40
11.	LCD 1632.....	46
12.	Proses Scanning Keypad Matrik 4 x 4	51
13.	Blok Diagram Audio Amplifier LM 386	53
14.	Simbol Dari Loudspeaker	56
15.	Push Button	57
16.	Simbol Bentuk Dan Fisik Light Emitting Dioda.....	58
17.	Diagram Blok Secara Keseluruhan	60
18.	Rangkaian Keypad Matrik 4 x 4	63
19.	Perencanaan Rangkaian Tombol.....	64
20.	Rangkaian Minimum Sistem.....	65
21.	Rangkaian Clock	67
22.	Rangkaian Reset.....	69

23. Rangkaian Display LCD 16 X 2	70
24. Rangkaian ISD 2560	72
25. Perencanaan Rangkaian LM 386	76
26. Perencanaan Rangkaian LED Indikator	76
27. Pengujian Rangkaian Keypad Dana LED Pada MCU AT89S8252	87
28. Pengujian Rangkaian MCU AT 89S8252	90
29. Pengujian rangkaian LCD Dengan MCU AT 89S8252.....	92
30. Foto Pengujian LCD	93
31. Foto Pengujian Rangkaian Modul Pengisian IC ISD 2560.....	95
32. Foto Alat Keseluruhan Tampak Depan.....	98
33. Foto Alat Keseluruhan Tampak Samping	98

DAFTAR TABEL

No	Keterangan Tabel	Halaman
1.	Perbandingan Beberapa Jenis Mikrokontroller.....	7
2.	Fungsi Alternatif Pin Port.....	11
3.	Fungsi Alternatif Pin Port.....	12
4.	Register Untuk Watchdog Dan EEPROM.....	19
5.	Pembagian Alamat Pada SFR.....	21
6.	Register TCON 2.....	24
7.	SFR Untuk Watchdog Dan Memori.....	25
8.	Register Power Control.....	27
9.	Register Serial Control.....	29
10.	Register TCON.....	31
11.	Keterangan Register TCON.....	31
12.	Tabel Register TMOD.....	32
13.	Kombinasi M0 Dan M1 Pada Register TMOD.....	32
14.	Pemilihan Periode Waktu Watchdog Timer.....	33
15.	Mode Operasi Timer 2.....	35
16.	Seri ISD 2500.....	38
17.	Fungsi Dari Pin Address Pada Mode Operasional ISD 2560.....	41
18.	Register Selection.....	48
19.	Fungsi Masing-Masing Pin Atau Terminal LCD M1632.....	49
20.	Instruksi LCD.....	50
21.	Fungsi Penyemat (Pin) LCD.....	71
22.	Sinyal Kontrol ISD.....	73

23. Hasil Pengujian Fungsi Tombol-tombol pada keypad	88
24. Hasil Pengujian Papan Tombol Matrik Keypad 4 x 4.....	89
25. Hasil Pengujian Minimum Sistem	91
26. Hasil Pengujian LCD (Liquid Crystal Display).....	92
27. Hasil Pengujian Pengalamatan rekaman suara Pada ISD 2560	96

DAFTAR FLOWCHART

No	Keterangan Flowchart	Halaman
1.	Proses Inisialisasi	79
2.	Proses Play Dan Record ISD Serta Perubahan Status.....	81
3.	Proses Baca Pesan Tersimpan.....	83
4.	Proses Pemanggilan Pemanggilan Penghuni Dan Ketik Pesan Text	85

BAB I

PENDAHULUAN

1.1 Latar belakang

Perkembangan teknologi dibidang elektronika sekarang ini sangat pesat, sehingga dalam pengaplikasiannya dapat menciptakan berbagai peralatan yang mampu mendukung kinerja manusia dalam kehidupan sehari-hari. Yang menjadi tolak ukur keberhasilan dalam pengaplikasian perangkat elektronika tersebut adalah seberapa besar manfaat yang dapat dirasakan atau dicapai dari penciptaan peralatan tersebut.

Penggunaan mikrokontroller sebagai unit-unit kendali sudah sangat luas. Hal ini dikarenakan peralatan-peralatan yang terkontrol dengan mikrokontroller memberikan kemudahan-kemudahan dalam penggunaannya. Salah satu contohnya adalah dengan adanya penciptaan alat dengan Sistem Penerima Tamu Elektronik

Selama ini alat penerima tamu yang sering dipakai masih banyak menggunakan sistem manual yang sudah ada dipasaran. Jadi apabila ada tamu yang berkunjung ke suatu rumah, orang tersebut harus menekan tombol bel rumah tersebut terlebih dahulu setelah itu maka akan ada orang yang keluar dari rumah tersebut untuk menemui tamu. Benar tidaknya penghuni yang dicari oleh tamu tersebut belum tentu akan benar karena tamu akan menanyakan terlebih dahulu yang dicari, apakah ada di rumah atau sedang keluar rumah, kalau ada maka tamu akan langsung ketemu dengan orang yang dicari dan apabila orang yang dicari sedang keluar maka salah satu penghuni akan memberi tahu bahwa orang yang dicari tersebut tidak ada di rumah. Jadi tamu bisa tahu yang dicari apabila ada

orang yang menemuinya dan memberi tahu tamu tersebut. maka ini adalah salah satu kekurangan dari alat penerima tamu manual ini. Sebelumnya alat penerima tamu elektronik yang ada hanya menekankan pada sistem penerimaan tamu elektroniknya saja tanpa ada pilihan jika seluruh penghuni rumah tidak ada, bias meninggalkan pesan singkat.

Melihat permasalahan diatas maka penulis terfikir untuk membuat suatu alat dengan sistem pengontrolan yang otomatis, maka dipakai sebuah IC AT 89S8252 yang dapat digunakan sebagai alat penerima tamu yang dapat difungsikan sama dengan alat penerima tamu sistem manual. Alat penerima tamu elektronik ini memiliki kelebihan yang lebih efisien bila digunakan pada rumah, karena penghuni akan langsung mengetahui bahwa ada tamu yang mencarinya melalui speaker panggil yang sebelumnya IC ISD 2560 telah diisi suara. Tamu juga akan mengetahui langsung apakah penghuni yang dicari itu ada atau tidak ada melalui LCD sebagai displaynya. Dengan alat ini maka akan dapat mempermudah bagi pemilik rumah mengetahui informasi apakah ada tamu yang datang untuk mencari salah satu dari penghuni rumah. Apabila seluruh penghuni rumah sedang tidak ada semua dan ada tamu yang datang untuk berkunjung maka dipastikan tamu tidak akan bertemu dengan penghuni rumah, kalau terjadi seperti ini maka ada pilihan bagi tamu yang tidak bisa bertemu dengan penghuni rumah secara langsung yaitu dengan cara meninggalkan pesan singkat untuk penghuni rumah. Jadi tamu tidak akan kecewa apabila sewaktu berkunjung ke rumah semua penghuni rumah tidak ada dikarenakan masih ada pilihan bagi tamu untuk meninggalkan pesan singkat

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah dapat dijabarkan sebagai berikut:

1. Bagaimana merubah sistem sebuah alat penerima tamu manual menjadi suatu sistem alat penerima tamu elektronik
2. Bagaimana merancang dan merealisasikan alat penerima tamu elektronik menggunakan Mikrokontroler AT89S8252.
3. Bagaimana membuat perangkat lunak yang digunakan untuk mengoperasikan alat ini.

1.3 Tujuan

Adapun Tujuan dari pembuatan alat ini adalah untuk mempermudah seluruh penghuni rumah mendapat informasi tentang tamu yang datang untuk mencari salah satu dari penghuni rumah.

1.4 Batasan Masalah

Agar permasalahan dalam Tugas Akhir ini tidak meluas dan lebih focus serta menghindari salah persepsi dan pengertian tentang perancangan alat maka dalam hal ini penulis membatasi perancangan alat ini dari segi :

1. Sistem rangkaian dari alat penerima tamu elektronik ini hanya simulasi
2. Penekanan perencanaan dan pembuatan alat ini adalah untuk sistem penerima tamu secara elektronik bukan pada cara penerimaan pesannya.

3. Pesan singkat adalah pilihan bagi tamu apabila ingin meninggalkan pesan sewaktu penghuni rumah keluar atau tidak ada semua.
4. Pengaplikasian alat ini digunakan untuk rumah.
5. Tidak membahas masalah catu daya yang digunakan.
6. Tidak membahas kinerja penulisan pada menu keypad.
7. Tidak ada rahasia pesan antar penghuni rumah.
8. Bagi penghuni rumah pesan bukan untuk perorangan.

1.5 Metodologi Penulisan

Perencanaan serta pembahasan yang dilaksanakan dalam tugas akhir ini adalah bersifat praktis yaitu perencanaan serta pembahasan yang mengarah pada pembuatan alat secara nyata. Awal pembahasan ini dimulai dengan membahas masing-masing rangkaian tersebut, kemudian diikuti dengan perencanaan dan pembuatan perangkat lunak maupun alat.

Metode yang digunakan dalam penulisan tugas akhir ini adalah :

1. Studi Pustaka

Memperoleh data dengan cara membaca dan mempelajari buku literatur yang ada hubungannya dengan penyusunan tugas akhir ini.

2. Studi Lapangan

Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.

3. Pengolahan Data

Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

1.6 Sistematika Pembahasan

Agar pembaca lebih cepat memahami alur dari laporan ini maka sistematika pembahasan dalam Laporan Tugas Akhir ini perlu penulis kemukakan. Adapun sistematikanya adalah sebagai berikut :

BAB I : PENDAHULUAN

Berisi tentang latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi penulisan dan sistematika pembahasan.

BAB II : LANDASAN TEORI

Pada bab ini diuraikan tentang teori-teori dasar yang digunakan sebagai landasan permasalahan yang akan dibahas dalam penulisan dan peralatan yang akan digunakan.

BAB III : METEDOLOGI PENELITIAN

Berisi tentang penjelasan dan tata cara perencanaan dan pembuatan alat yang akan dibuat.

BAB IV : PENGUJIAN ALAT

Bab ini berisi hasil dari pengukuran dari perencanaan yang kemudian diuji. Hasil yang didapat kemudian dibahas dan dibandingkan antara data dari hasil pengujian dengan teori dasarnya.

BAB V : PENUTUP

Berisi tentang kesimpulan dan saran.

BAB II

LANDASAN TEORI

2.1 Pendahuluan

Landasan teori sangat membantu untuk dapat memahami suatu sistem. Selain dari pada itu dapat juga dijadikan sebagai bahan acuan di dalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi mikrokontroler AT89S8252, ISD 2560, LCD, IC LM 386, Keypad dan perangkat pendukung lainnya.

2.1.1 Mikrokontroler AT89S8252

Perbedaan mendasar antara mikrokontroler dan mikroprosesor terletak pada kelengkapan isinya yaitu mikrokontroler sudah dilengkapi dengan berbagai macam alat kontrol selain memiliki CPU juga dilengkapi memori (ROM & RAM) input maupun output yang merupakan kelengkapan minimum system, sedangkan mikroprosesor kesemuanya itu tidak dimiliki secara internal melainkan terpisah sebuah mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal (*Single Chip Microcomputer*) yang dapat berdiri sendiri.

Mikrokontroler AT89S8252 merupakan versi CHMOS dari AT89S8252 yaitu versi NMOS. NMOS merupakan kependekan dari *N - Channel Metal Oxide Silicon* dan kompatibel dengan MCS-51 mikrokomputer yang merupakan produksi dari ATMEL. Seri AT89S8252 terdiri dari beberapa jenis tingkat

kecepatan mulai dari 12 MHz sampai dengan 24 MHz. Digunakan untuk beberapa keperluan mulai dari komersial, industri, otomotif, dan militer. Perbandingan antara mikrokontroler MCS-51 dengan beberapa produk mikrokontroler buatan Atmel dapat dilihat pada table berikut :

Tabel 2.1 Perbandingan Beberapa Jenis Mikrokontroler

Sumber : Ibnu Malik 2003 : 7

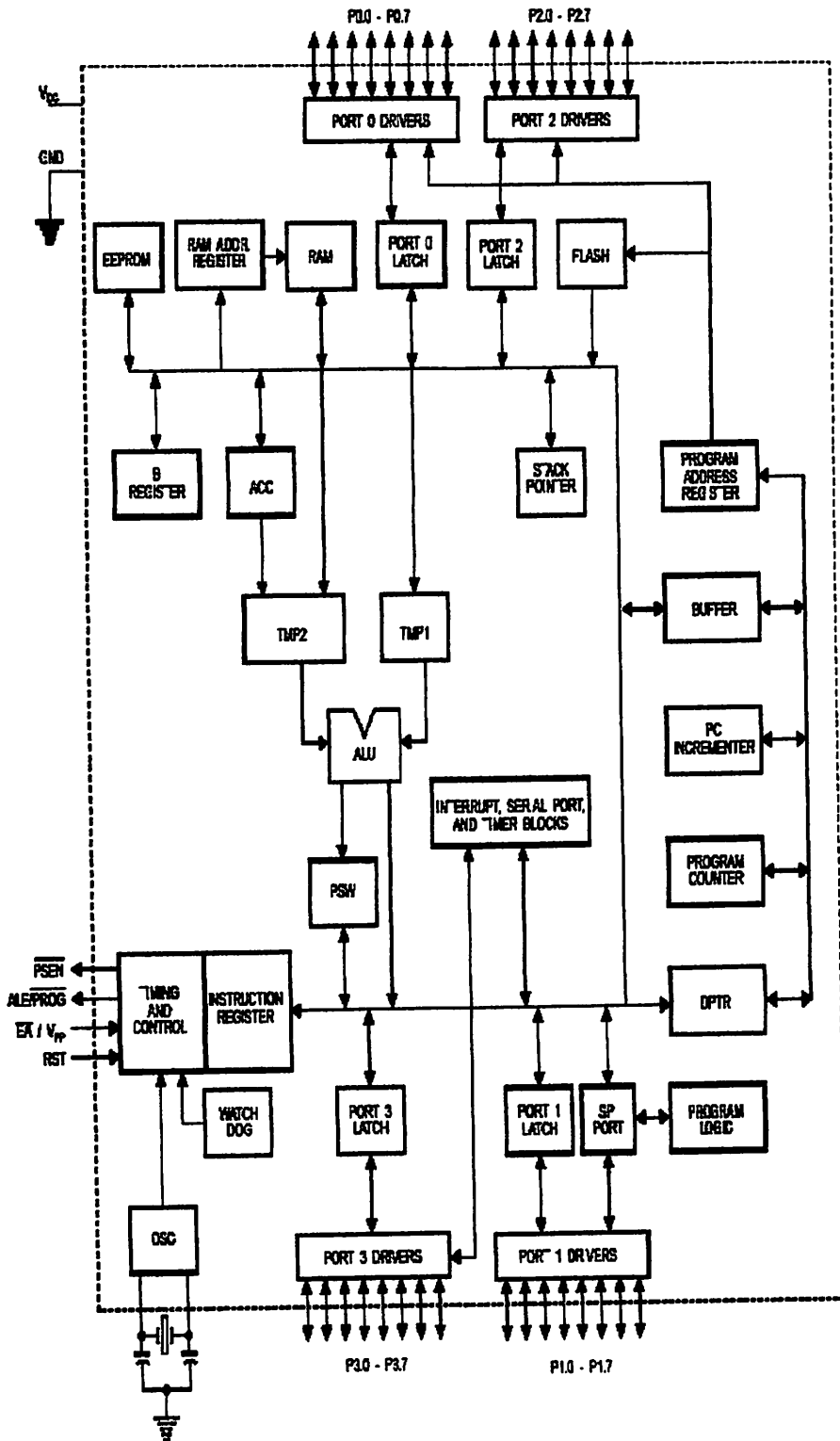
Produk	ROM	RAM	MEM Lain	EFF. I/O	Frek Max
8031	N/A	128 Byte	N/A	2 X 8 Bit	12 MHZ
8751	4K	128 Byte	N/A	4 x 8 Bit	12 MHZ
8051	4K	128 Byte	N/A	4 x 8 Bit	12 MHZ
AT89S51	4K	128 Byte	N/A	4 x 8 Bit	24 MHZ
AT89C52	8K	265 Byte	N/A	4 x 8 Bit	24 MHZ
AT89S8252	8K	265 Byte	2K EEPROM	4 x 8 Bit	24 MHZ

Mikrokontroler AT89S8252 diproduksi oleh Atmel Corp, yang dilengkapi dengan 8 KB *flash memory* yang bisa diprogram dan dihapus. Selain *flash memory*, AT89S8252 juga menyediakan 2KB EEPROM, 256 byte RAM internal, 32 jalur I/O, tiga buah *timer/counter* 16-bit, sebuah *watchdog timer* yang dapat diprogram, dua buah data pointer, enam buah *vector interrupt*, *on-chip oscillator*, rangkaian clock dan sebuah port serial (*full-duplex*), yang sangat bermanfaat dalam melakukan komunikasi serial AT89S8252.

Arsitektur dari MCU AT89S8252 adalah sebagai berikut :

1. Compatible with MCS-51™ Products.
2. 8 K Bytes of In-System Reprogrammable Downloadable Flash Memory.
 - SPI Serial Interface for Program Downloading.
 - Endurance: 1,000 Write/Erase Cycles.
3. 2 K Bytes EEPROM.
 - Endurance: 100,000 Write/Erase Cycles.

4. 2,7 V to 6 V Operating Range.
5. Fully Static Operation: 0 to 24 MHz.
6. 3 - Level Program Memory Lock.
7. 256 x 8-bit Internal RAM.
8. 32 Programmable I/O Lines.
9. 3 buah *timer/counter* 16-bit.
10. 9 Interrupt Sources.
11. Programmable UART Serial Channel.
12. SPI Serial Interface.
13. Low Power Idle and Power Down Modes.
14. Interrupt Recovery From Power Down.
15. Programmable Watchdog Timer.
16. Dual Data Pointer.
17. Power Off Flag.

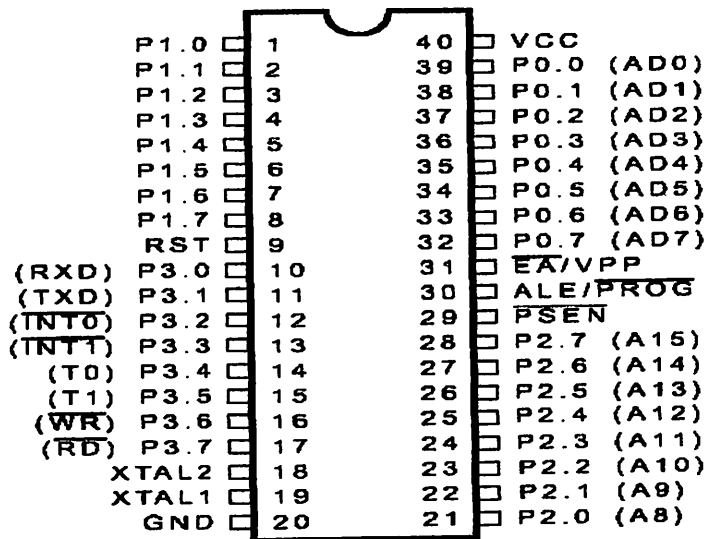


Gambar 2-1 Blok Diagram AT89S8252

Sumber : Data Sheet AT 89S8252 : 3

2.1.2 Konfigurasi Kaki-kaki MCU AT89S8252

Berikut ini adalah bentuk fisik dari AT89S8252.



Gambar 2-2 Bentuk Fisik AT89S8252

Sumber : Data Sheet AT 89S8252 : 2

Fungsi-fungsi dari tiap-tiap pin sebagai berikut :

a. Gnd (*Ground*) Pin 20

Pin ini digunakan sebagai titik referensi untuk setiap pin pada mikrokontroller.

b. Vcc (*Supply Tegangan*), Pin 40

Dihubungkan dengan sumber tegangan + 5 V.

c. Port 0 (P0.0 – P0.7), Pin 32 - 39

Port 0 merupakan sebuah port I/O *bi-directional* 8 bit. Sebagai port output setiap pin dapat menerima arus dari 8 buah input TTL. Pada saat port 0 diberi logika satu, tiap pin dapat digunakan sebagai input dengan *high impedance*. Port 0 juga digunakan sebagai *bus byte* terendah dari alamat atau data pada saat mikrokontroler mengakses memori eksternal. Pada

mode ini, P0 mempunyai *pull-up* sudah tersedia dalam mikrokontroler. P0 juga digunakan untuk menerima kode selama proses pemrograman dan mengeluarkan kode

d. *Port 1 (P1.0 – P1.7), Pin 1 - 8*

Port 1 merupakan port I/O *bi-directional* dengan *pull-up internal*. *Buffer output* Port 1 mampu menerima atau memberikan arus untuk 4 input TTL. Pada saat P1 diberi logika satu, P1 akan di *pull high* dengan *pull-up internal* dan dapat digunakan sebagai input. Sebagai input, tiap pin pada P1 (yang di *pull-low*) akan mengeluarkan arus I_L karena adanya *pull-up internal*. Beberapa pin P1 mempunyai fungsi tambahan. Pin P1.0 dan P1.1 dapat digunakan sebagai pencacah dan *trigger* untuk *timer/counter 2*. Sedangkan P1.4, P1.5, P1.6 dan P1.7 dapat digunakan sebagai port SPI, I/O data dan I/O *shift clock*. P1 juga digunakan untuk menerima alamat terendah selama proses pemrograman dan verifikasi.

Tabel 2-2 Fungsi Alternatif Pin Port

Sumber : Data Sheet AT 89S8252 : 4

Pin Port	Fungsi
P1.0	T2 (Input Pencacah Eksternal Untuk Timer/Counter2), <i>clock-out</i>
P1.1	T2EX (Timer/Counter2 <i>capture/reload</i> trigger dan kontrol arah)
P1.4	SS (Input pemilih port slave)
P1.5	MOSI (Master Data Output, Slave Data Input) - pin untuk jalur SPI
P1.6	MISO (Master Data Input, Slave Data Output) - pin untuk jalur SPI
P1.7	SCK (Master Clock Output, Slave Clock Input) - pin untuk jalur SPI

e. *Port 2 (P2.0 – P2.7), Pin 21 – 28*

Port 2 merupakan sebuah port I/O 8 bit yang sifatnya *bidirectional* dengan *pull-up internal*. *Buffer* dari keluaran P2 mampu menerima atau memberikan arus pada empat buah input TTL. Jika P2 diberi logika satu,

maka setiap pin akan di *pull-high* dengan *pull-up internal* dan bisa digunakan sebagai input. Sebagai input, jika P2 di *pull low* maka setiap pin mampu memberikan arus I_L karena adanya *pull-up internal*.

f. *Port 3* (P3.0 – P3.7), Pin 10 - 17

Port 3 merupakan sebuah I/O *bi-directional* 8 bit yang mempunyai *pull-up internal* atau *buffer* keluaran port 3 mampu menerima atau memberi arus kepada empat buah TTL. Pada saat P3 diberi logika satu, setiap pin tersebut akan di *pull-up* secara internal dan dapat digunakan sebagai input. Sebagai input, jika P3 di *pull-low*, setiap pin pada port tersebut akan memberikan arus I_L karena adanya *pull-up*. *Port 3* juga menyediakan fungsi tambahan seperti terlihat pada tabel berikut :

Tabel 2-3 Fungsi Alternatif Pin Port
Sumber : Data Sheet AT 89S8252 : 4

Pin Port	Fungsi
P3.0	RXD (masukan data serial atau sebagai keluaran data).
P3.1	TXD (keluaran pengiriman data untuk serial <i>port (asynchronous)</i> atau sebagai keluaran <i>clock (synchronous)</i>).
P3.2	INT 0 (masukan interrupt external 0).
P3.3	INT 1 (masukan interrupt external 1).
P3.4	T0 (masukan dari pewaktu/pencacah 0).
P3.5	T1 (masukan dari pewaktu/pencacah 1).
P3.6	WD (sinyal penulisan memori data eksternal).
P3.7	RD (sinyal pembacaan memori data eksternal).

g. RST (*Reset*), Pin 9

Merupakan pin untuk memberikan sinyal reset pada mikrokontroller. Reset yang digunakan adalah sebuah pulsa dengan logika satu atau HV. Pulsa yang diberikan minimal sebesar 2 *cycle* mesin pada saat osilator bekerja atau aktif.

h. ALE (*Address Latch Enable*)/Prog, Pin 30

Pin ALE (aktif tinggi) mengeluarkan pulsa output untuk menyangga (*latch*) satu byte alamat rendah selama mengakses ke memori eksternal. ALE dapat mengendalikan 8 beban TTL. Pin ini juga merupakan input pulsa program yang aktif rendah selama pemrograman EPROM. Pada operasi normal, ALE dikeluarkan pada suatu kecepatan yang konstan yaitu $1/6$ dari frekuensi osilator dan dapat digunakan untuk *timing* eksternal atau untuk tujuan membuat *clock*.

i. PSEN (*Program Strobe Enable*), Pin 29

Pin ini aktif rendah yang merupakan *strobe* pembacaan ke program memori eksternal.

j. XTAL 1 Dan XTAL 2, Pin 18 - 19

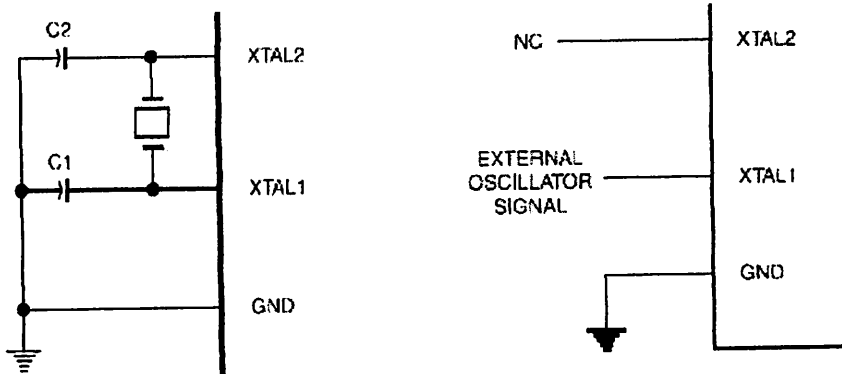
Merupakan input untuk *amplifier oscillator inverting* dan input untuk rangkaian clock internal. XTAL2 merupakan keluaran dari *amplifier oscillator inverting*.

k. EA (*External Access*)/VPP, Pin 31

Merupakan sinyal ENABLE yang digunakan untuk mengakses memori eksternal. EA harus di strap ke vcc supaya dapat mengeksekusi program internal. Pin ini juga menerima VPP sebesar 12V selama pemrograman paralel.

2.1.3 Karakteristik Osilator

XTALI dan XTAL2 masing-masing adalah sebuah input dan output dari sebuah *amplifier inverting* yang dapat digunakan sebagai osilator. Resonator yang digunakan dapat berupa kristal maupun keramik. Untuk memberikan clock pada AT89S8252 dari luar, XTALI harus terhubung ke rangkaian clock sedangkan XTAL2 dibiarkan dalam keadaan tidak tersambung. *Duty cycle* dari sumber clock luar diberikan oleh sebuah flip-flop pembagi - 2.



a) *Oscillator Connector*

b) *External Clock Configuration*

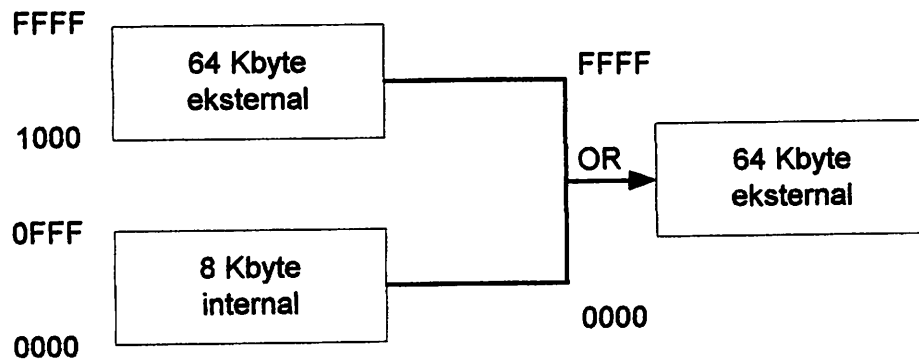
Gambar 2-3 Karakteristik *Oscillator*

Sumber : Data Sheet AT 89S8252 : 17

2.1.4 Organisasi Memori MCU AT89S8252

Mikrokontroler keluarga MCS-51 memiliki memori program dan memori data yang terpisah. Pemisahan ini dilakukan secara logika sehingga CPU dapat mengakses sampai 64 *Kbyte* memori program dan 64 *Kbyte* memori data. Lebar memori data internal adalah 8 bit dan 16 bit (register PC dan register DPTR).

2.1.4.1 Memori Program



Gambar 2-4 Memori Program MCS-51

Sumber : Ibnu Malik 2003 : 8

Memori program menggunakan alamat sepanjang 64 Kbyte dengan 8 Kbyte (alamat \$0000 sampai dengan \$0FFF) yang merupakan memori internal sehingga 60 Kbyte merupakan memori eksternal. Dapat menggunakan 64 Kbyte memori eksternal sebagaimana yang ditunjukkan pada gambar memori data.

Memori program merupakan tempat penyimpanan data permanen. Memori program lebih dikenal dengan nama *Read Only Memory* (ROM). Data dalam ROM tidak akan terhapus meskipun catu daya dimatikan atau dikenal sebagai sifat *non-volatile*. Karena sifatnya yang demikian ROM dapat digunakan untuk menyimpan program.

Ada beberapa tipe ROM, antara lain :

a. ROM (*Read Only Memory*)

Merupakan memori yang sudah diprogram oleh pabrik (ROM murni).

b. PROM (*Programmable Read Only Memory*)

Merupakan memori yang dapat diprogram oleh pemakai tetapi tidak dapat diprogram ulang.

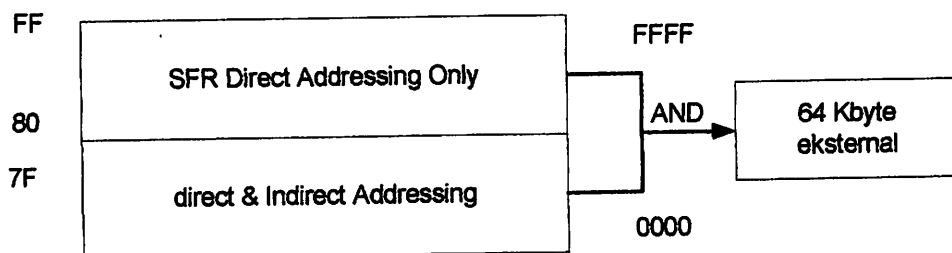
c. EPROM (*Erasable Programmable Read Only Memory*)

Merupakan PROM yang dapat diulang. ROM ini juga terdapat pada mikrokontroler 8751, hal ini ditandai dengan adanya jendela kaca pada konstruksi IC 8751 yang digunakan untuk menghapus atau memperbaiki program yang sudah ada.

d. EEPROM (*Electrical Erasable Programmable Read Only Memory*)

Pada prinsipnya hampir sama dengan EPROM, tetapi perbedaannya terletak pada pengosongan atau penghapusan program. Untuk EPROM dapat dihapus dengan menggunakan sinar *ultra violet*, sedangkan pada EEPROM pengisian program dapat dilakukan langsung atau menumpuk program lama dengan program yang baru. EEPROM lebih fleksibel dibandingkan EPROM.

2.1.4.2 Memori Data



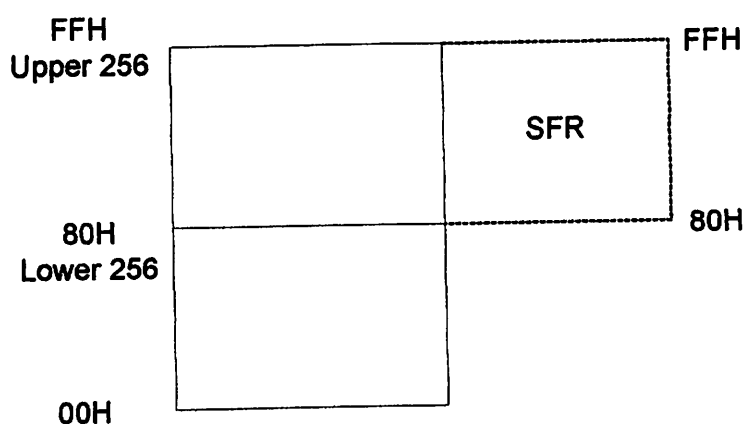
Gambar 2-5 Memori Data MCS-51

Sumber : Ibnu Malik 2003 : 9

Memori data merupakan tempat penyimpanan data yang bersifat sementara atau *volatile*. Dengan kata lain data akan hilang bila tidak dicatu. Memori data lebih dikenal dengan nama RAM (*Random Access Memory*), yaitu dapat dilakukan pembacaan dan penulisan data alamat yang tersedia.

Memori MCS-51 mempunyai 256 bytes RAM internal ditambah sejumlah register fungsi khusus atau *Special Function Register (SFR)*. (*Advanced Mikrodevides*, 1988.1.6). Selain mempunyai memori internal, MCU AT89S8252 mempunyai memori eksternal yang memiliki pengalamatan sampai 64 Kbytes.

Pada keluarga mikrokontroler MCS-51, ruang memori data eksternal terbagi menjadi 3 blok yang disebut *lower 256*, *upper 256* dan ruang SFR, sebagaimana ditunjukkan pada gambar di bawah ini.



Gambar 2-6 Memori Internal (RAM) dan SFR MCS-51

Sumber : Ibnu Malik 2003 : 10

Pada *lower 256* lokasi memori dibagi menjadi 3 bagian :

1. Register bank 0 – 3

Lokasi bank register dimulai dari alamat 00H – 01H yang terdiri dari 32 bytes. Register bank ini terdiri dari 4 buah register 8 bit yang dapat dipilih melalui pengaturan *program status word* register.

2. *Bit Addressing*

Terdiri dari 16 *bytes* yang dimulai dari 20H – 2FH. Masing-masing dari 128 bit lokasi ini dapat dialamati secara langsung yaitu dari \$00H sampai \$7FH.

3. *Scratch Pad Ar*

Lokasi dari alamat \$30H sampai \$7FH atau sebanyak 80 *bytes* yang dapat digunakan sebagai alamat bagi RAM.

2.1.4.3 EEPROM Dan RAM

AT89S8252 mempunyai 2 KB EEPROM internal untuk menyimpan data dan 256 byte RAM internal. 128 byte teratas RAM menduduki ruang memori yang sama secara paralel dengan SFR. Artinya 128 byte tersebut mempunyai alamat yang sama dengan SFR. Namun secara fisik, 128 byte teratas RAM terpisah dari SFR.

Pada saat sebuah intruksi mengakses sebuah lokasi diatas 7FH, mode pengalamatan yang digunakan dalam menentukan apakah CPU harus mengakses 128 byte teratas RAM. Instruksi yang menggunakan pengalamatan langsung akan mengakses SFR, sedangkan pengalamatan tak langsung akan mengakses 128 byte teratas RAM. Daerah RAM ini juga digunakan sebagai ruang untuk *stack pointer*.

Penggunaan EEPROM harus diawali dengan mengatur bit EEMEN dalam register WBCON. Alamat EEPROM dimulai dari 000H sampai 7FFH . Jika EEMEN diset '0' dan EEMWE diset '1' maka mikrokontroller akan mengakses memori data eksternal. EEMWE harus diset ke '0' jika tidak ada memori

EEPROM yang digunakan. Umumnya penulisan ke EEPROM membutuhkan waktu 2,5 ms untuk setiap karakter. Status penulisan ini dapat dipantau melalui bit RDY/BSY (*read only*) yang terdapat dalam SFR WMCON. RDY/BSY = 0 berarti penulisan belum selesai dan RDY/BSY = 1 berarti penulisan telah selesai dan siklus penulisan selanjutnya dapat diinisialisasi.

2.1.4.3.1 Register Kontrol Untuk EEPROM

EEMEN dan EEMWE pada register WMCON digunakan untuk memilih 2 KB EEPROM dan untuk *meng-enable* penulisan ke EEPROM.

Tabel 2-4 Register untuk Watchdog dan EEPROM

Sumber : Ibnu Malik 2003 : 38

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

2.1.4.3.2 Register Dual Data Pointer

Untuk memfasilitasi pengaksesan dua memori, EEPROM dan memori data eksternal, AT89S8252 dilengkapi dengan dua buah register data pointer 16 bit. DP0 pada SFR dengan lokasi 82H-83H dan DP1 pada 84H-85H. Bit DPS = 0 dalam SFR WMCON akan memilih DP0 dan DPS = 1 akan memilih DP1. Sebelum mengakses register data pointer, bit DPS harus diinisialisasi ke nilai yang sesuai.

2.1.4.4 Memprogram Flash Dan EEPROM

AT89S8252 menyediakan memori flash yang dapat ditulis maupun dibaca sebesar 8 KB dan 2 KB EEPROM. AT89S8252 mendukung pemrograman secara

paralel dan serial. Pemrograman secara paralel dilakukan pada tegangan tinggi sebesar 12v dan secara serial dilakukan pada tegangan rendah sebesar 5v. Pemrograman secara serial memberikan kemudahan kepada user karena tidak memerlukan rangkaian khusus, sedangkan pemrograman secara paralel kompatibel dengan paralel-programmer konvensional.

Pada mode serial, memori untuk kode dan data dipetakan oleh alamat yang berbeda. Sedangkan pada mode paralel, keduanya menempati memori yang bersebelahan yaitu 0000H-1FFFFH untuk kode dan 2000H-27FFFH untuk data. Pada mode serial, untuk memprogram ulang memori tidak perlu menghapus chip terlebih dahulu kecuali jika bit-bit lock digunakan pada pemrograman sebelumnya. Sedangkan pada mode paralel, untuk memprogram ulang memori yang sudah berisi kode, chip harus dihapus terlebih dahulu (memori flash dan EEPROM).

2.1.5 SFR (Special Function Register)

Selain memiliki SFR (Special Function Register) seperti halnya pada mikrokontroler MCS-51, mikrokontroler Atmel AT89S8252 memiliki tambahan SFR. Hal ini tak lain adalah karena terdapatnya tambahan fitur pada mikrokontroler ini. Jadi SFR tambahan ini adalah SFR untuk mengontrol alat tambahan pada mikrokontroler Atmel AT89S8252. Pembagian alamat pada SFR adalah sebagai berikut :

Tabel 2- 5 Pembagian Alamat Pada SFR

Sumber : Data Sheet AT 89S8252 : 6

Simbol	Nama Register	Nilai pada saat reset	Alamat
ACC	Accumulator	0000 _H	0E0 _H
B	Register B	00 _H	0F0 _H
PSW	Program Status Word	00 _H	0D0 _H
SP	Stack Pointer	07 _H	81 _H
DPTR	Data Pointer 2 byte	-	-
DPL	Low bytes	0000 _H	82 _H
DPH	High bytes	0000 _H	83 _H
P0	Port 0	FF _H	80 _H
P1	Port 1	FF _H	90 _H
P2	Port 2	FF _H	0A0 _H
P3	Port 3	FF _H	0B0 _H
IP	Interupt priority control	XXX00000 _B	0B8 _H
IE	Interupt enable control	0XX00000 _B	0A8 _H
TMOD	Timer/counter mode control	00 _H	89 _H
TCON	Timer/ counter control	00 _H	88 _H
TH0	Timer/counter 0 high byte	00 _H	8C _H
TL0	Timer counter 0 low byte	00 _H	8A _H
TH1	Timer / counter 1 high byte	00 _H	8D _H
TL1	Timer/ counter 1 low byte	00 _H	8B _H
SCON	Serial control	00 _H	9B _H
SBUF	Serial data buffer	Independen	99 _H
PCON	Power control	HMOS 0XXXXXXXX _B	87 _H
		CHMOS 0XXX0000 _B	

Pada 256 bytes atas (*upper 256*) ditempati oleh register yang mempunyai fungsi khusus yang disebut dengan *Special Fuction Register (SFR)*. Ruang dari register fungsi khusus ini adalah dari 80H sampai FFH. Berikut ini adalah contoh isi vector alamat pada *Special Function Register*.

FFH	
E0H	Acc
B0H	Port 3
A0H	Port 2
90H	Port 1
80H	Port 0

Gambar 2-7 Ruang Special Function Register

Sumber : Data Sheet AT 89S8252 : 7

Beberapa macam register fungsi khusus yang sering digunakan adalah sebagai berikut ini :

a. *Accumulator* (A atau ACC)

Dari namanya dapat diketahui bahwa fungsi *Accumulator* adalah sebagai pengumpul. SFR ini merupakan register yang paling banyak dipakai. Untuk semua operasi aritmatika biasanya menggunakan atau melibatkan *accumulator*. Alamat dari *accumulator* adalah E0H.

b. Register B

Register B beralamat di F0H. Register ini digunakan untuk operasi perkalian dan pembagian.

c. *Program Status Word* (PSW)

Register ini meliputi bit-bit : CY (*Carry*), AC (*Auxillary Carry*), FO sebagai flag, RS0 dan RS1 untuk pemilih register bank, OV (*Over Flow*), dan *parity flag*.

d. *Data Pointer High* (DPH) Dan *Data Pointer Low* (DPL)

DPTR adalah register yang digunakan untuk pengalamatan tidak langsung. Register ini digunakan untuk mengakses memori program baik internal

maupun eksternal. DPTR dikontrol oleh 2 buah register 8 bit yaitu DPH dan DPL.

e. *Port 0, Port 1, Port 2, Port 3*

Pada keluarga 8051 masing-masing *port* dapat dialamati langsung baik secara *byte* atau bit. Masing-masing *port* merupakan *port bi-directional (input/output)* yaitu :

1. *Port 0* digunakan sebagai pengalamatan memori dari luar.
2. *Port 1* digunakan sebagai I/O dari mikrokontroler.
3. *Port 2* digunakan sebagai pengalamatan memori dari luar.
4. *Port 3* berisi sinyal kontrol seperti *interrupt serial*, WR, dan RD.

f. *Register Prioritas Interrupt (Interrupt Priority Register /IP).*

Merupakan register yang berisi bit-bit untuk mengaktifkan prioritas dari suatu *interrupt* yang ada pada mikrokontroler pada taraf yang diinginkan.

g. *Interrupt Enable Register*

Merupakan register yang berisi bit-bit untuk menghidupkan atau mematikan sumber-sumber *interrupt*.

h. *Timer/Counter Control Register*

Pengendalian kerja dari timer atau counter dilakukan dengan pengaturan register yang berhubungan kerja dari timer atau counter mode kontrol. Untuk mengaktifkan timer atau counter yang meliputi penentuan fungsi sebagai timer atau sebagai counter serta pemilihan mode operasi yang dapat diatur melalui TMOD.

i. *Serial Control Buffer*

Register ini digunakan untuk menampung data masukan (SBUF *in*) atau keluaran (SBUF *out*) dari *serial port*.

Pada Mikrokontroler AT89S8252 terdapat SFR tambahan. SFR tambahan itu meliputi : T2CON (Timer 2 register dengan alamat 0C8H), T2MOD (Timer 2 mode dengan alamat 0C9H), WMCON (Watchdog and memori control register dengan alamat 96H), SPCR (SPI Control register dengan alamat D5H), spsr (Status register dengan alamat AAH), SPDR (SPI data register dengan alamat 86H). SFR tambahan ini antara lain :

2.1.5.1 SFR Untuk Timer 2

Pada Mikrokontroler Atmel AT89S8252 terdapat tambahan sebuah timer/counter yang diberi nama timer 2 (sehingga Mikrokontroler Atmel AT89S8252 memiliki 3 timer/counter yaitu timer/counter 0, timer/counter 1 dan timer/counter 2).

Pada timer atau counter 2 ini dikendalikan oleh special function register yang bernama T2CON (timer 2 control), T2MOD (timer 2 mode) dan sepasang register RCAP2H, RCAP2L merupakan register capture/reload untuk timer 2 dalam 16 bit capture mode atau auto reload mode. Register T2CON yang beralamat di 0C8H memiliki bit-bit sebagai berikut :

Tabel 2-6 Register T2CON

Sumber : Ibnu Malik 2003 : 37

MSB				LSB			
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Keterangan :

TF2 : Bendera overflow timer 2, diset oleh timer 2 dan harus diclear lewat software.

EXF2 : Bendera luar timer 2 diset pada saat suatu capture atau reload disebabkan oleh transisi negative pada T2EX dan $EXEN2 = 1$.

RCLK : Receive clock enable. Jika diset menyebabkan serial port menggunakan pulsa overflow timer 2 sebagai detak penerimaan pada serial port untuk mode 1 dan 3.

TCLK : Transmit Clock Enable. Jika diset menyebabkan serial port menggunakan pulsa overflow timer 2 sebagai detak pengiriman.

EXEN2: Timer 2 Exsternal enable.

TR2 : Bit untuk mengatur start/stop untuk timer 2. jika $TR2 = 1$ timer akan aktif.

C/T2 : Bit pemilihan timer atau counter untuk timer 2.

2.1.5.2 SFR Untuk Watchdog Dan Memori

Untuk menggunakan watchdog timer atau memori, maka dapat dilakukan dengan mengatur SFR yang bernama WMCON dengan alamat 96H. Bit-bit pada SFR ini dapat dijelaskan sebagai berikut. Nilai SFR pada saat reset adalah 0000 0000B.

Tabel 2-7 SFR Untuk Watchdog dan Memori

Sumber : Ibnu Malik 2003 : 38

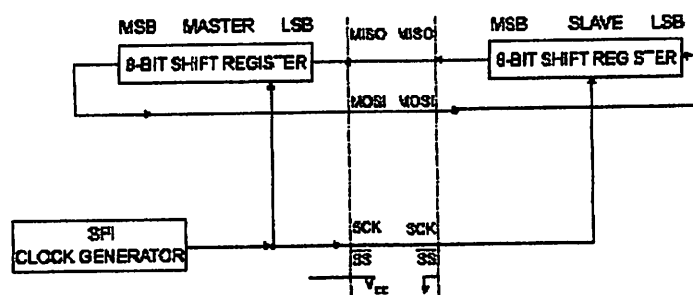
MSB						LSB	
PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTTRST	WDTEN

Keterangan :

- PS 0,1,2 : Ketiga bit ini merupakan prescaler untuk watchdog timer.
- EEMWE : Bit pengaktif penulisan EEPROM data memori.
- EEMEN : Bit pengaktif pengaksesan internal EEPROM.
- DPS : Jika bit ini dideklarasi akan memilih bank pertama dari data pointer register (DPO). Jika bit ini diset akan terpilih bank kedua (DP1).
- WDTRST : Tiap saat bit ini diset ke 1 oleh software pengguna suatu pulsa akan dihasilkan untuk mereset watchdog timer.
- WDTEN : Bit pengaktif watchdog timer.

2.1.5.3 SFR Pengontrol SPI (Serial Peripheral Interface)

SPI merupakan interface yang mampu melakukan komunikasi *sinkron full duplex* dengan kecepatan tinggi antar AT89S8252 atau antara AT89S8252 dan perangkat lain yang membutuhkan kecepatan tinggi. Port ini dapat digunakan untuk sambungan *master/slave* antara CPU dengan SPI ditunjukkan oleh Gambar 2.8 berikut :



Gambar 2-8 Koneksi Master/ Slave SPI

Sumber : Ibnu Malik 2003 : 40

Fitur SPI pada Atmel AT89S8252 meliputi :

- a. Full Duplex, 3 kawat dengan transfer data secara sinkron.
- b. Operasi master atau slave.
- c. Frekuensi maximum 6 MHZ.
- d. Sistem data transfer MSB dahulu atau LSB dahulu.
- e. 4 bit rate terprogram.
- f. Bendera sela pada akhir transmisi.
- g. Write Collision flag protection.
- h. Bangun dari mode idle (hanya untuk mode slave).

2.1.5.4 Power Control (PCON)

Register power control beralamat di 87H. Berguna untuk mengatur kebutuhan daya mikrokontroller. Dengan adanya register pengatur daya ini memungkinkan mikrokontroller ke mode idle atau sleep yang mana akan lebih menghemat pemakaian daya. Selain itu ada bit-bit pada register PCON ini untuk mengatur baud rate pada serial port. Berikut bit-bit pada PCON adalah sebagai berikut :

Tabel 2-8 Register Power Control

Sumber : Ibnu Malik 2003 : 16

MSB				LSB			
SMOD	-	-	-	GF1	GF0	PD	IDL

Keterangan :

SMOD : Mode generator baud rate. Jika SMOD = 1, baud rate UART akan digandakan

GF1 : Flag general purpose

GF 0 : Flag general purpose

PD : Flag Power Down. Mode power down aktif jika diberi logika ' 1'

IDL : Flag Idle. Mode Idle aktif jika diberi logika ' 1'

2.1.6 Serial Port Atau UART (*Universal Asynchronous Transmitter/Receiver*)

Adanya serial port pada mikrokontroler MCS-51 menambah daya guna dari mikrokontroler ini. Serial port adalah suatu port (terminal) untuk mengirim data secara serial. Sebagaimana diketahui pengiriman data ada dua jenis yaitu :

2.1.6.1 Pengiriman Data Secara Paralel

Pengiriman data secara paralel adalah suatu sistem pengiriman data dimana keseluruhan bit dikirimkan secara bersamaan. Pada pengiriman secara paralel, data dikirimkan dalam format 8 bit sekaligus. Keuntungan pengiriman secara paralel adalah bahwa data dikirimkan sangat cepat, namun kelemahannya adalah bahwa diperlukan jumlah penghantar (kawat) yang banyak.

2.1.6.2 Pengiriman Data Secara Serial

Selain secara paralel, data dapat pula dikirimkan ke tempat lain secara serial. Dengan demikian data dikirimkan bit per bit (satu per satu). Keuntungan pengiriman data secara seri adalah bahwa diperlukan lebih sedikit penghantar untuk pengiriman data, namun kerugiannya adalah bahwa waktu yang diperlukan untuk mengirimkan data menjadi semakin lama.

Pada mikrokontroler MCS-51, serial port dikontrol oleh beberapa register yaitu SCON (serial port control) beralamat di 98H dan SBUF (serial data buffer) yang beralamat di 99H. Data dari dan menuju ke serial port akan melalui register

SBUF. Setelah serial port dikonfigurasi (mode operasi dan baute rate), maka penulisan ke SBUF akan memulai pengiriman secara serial. Isi dari register SCON adalah sebagai berikut :

Tabel 2-9 Register Serial Control (SCON)

Sumber : Ibnu Malik 2003 : 18

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	T1	R1

Keterangan :

SM0 : Bit 0 mode port serial MSB.

SM1 : Bit 1 mode port serial LSB.

SM2 : Mode untuk multiprosesor.

REN : Flag untuk menerima data.

TB8 : Bit ke-9 yang akan dikirim pada mode 2 dan 3.

RB8 : Pada mode 1, jika SM2=0, RB8 merupakan bit stop yang diterima. Pada mode 2 dan 3, RB8 merupakan bit ke-9 dari data yang diterima.

T1 : Flag transmit.

R1 : Flag receive.

2.1.7 Sistem Interrupt

Mikrokontroller AT89S8252 memiliki 5 buah sumber interupt, 2 eksternal interupsi (INT0 dan INT1), 2 timer interupsi (timer 0 dan 1) dan serial port interupsi

INT0 : Interrupt pada P3.2 (kaki 12).

INT1 : Interrupt pada P3.3 (kaki 13).

Timer 0 : Timer pada P3.4 (kaki 14).

Timer 1 : Timer pada P3.5 (kaki 15).

Port serial : Jika pengiriman atau penerimaan suatu frame telah lengkap.

Saat terjadinya interrupt mikrokontroler secara otomatis akan menuju subrutin pada alamat tersebut. Setelah interrupt servis selesai dikerjakan Mikrokontroler akan mengerjakan program semula. Dua sumber eksternal adalah INTO dan INTI, kedua interupsi eksternal akan aktif transisi tergantung isi ITO dan ITI pada register TCON. Interrupt TO dan TI aktif pada saat timer yang sesuai mengalami roll over. Interupsi serial akan dibangkitkan dengan melakukan operasi OR pada R1 dan T1. Tiap – tiap sumber interupsi dapat enable atau disable secara software. Tingkat prioritas semua sumber interupsi dapat diprogram sendiri – sendiri dengan set atau clear bit pada SFRS (interrupt priority). Register akan berperan dalam mengatur aktif tidaknya interupsi adalah, interrupt enable register.

2.1.8 Timer Dan Counter

Mikrokontroler AT89C51 mempunyai dua buah timer atau counter 16 bit yang dapat diatur melalui perangkat lunak, yaitu, timer atau counter 0 dan timer atau counter 1. Periode waktu timer atau counter secara umum ditentukan dengan persamaan berikut:

- o Sebagai timer/ counter 8 bit

$$T = (255 - TLx) * 1 / (F_{osc} / 12)$$

Dimana TLX adalah register TLO atau TL1

- o Sebagai timer / counter 16 bit

$$T = (65535 - THx TLx) * 1 / (F_{osc} / 12)$$

Dimana :

THx = isi register TH0 atau TH1

TLx = isi register TLO atau TL1

Selain timer mode, ada lagi register yang digunakan untuk mengontrol timer yaitu register TCON. Pada register TCON ini bit-bitnya ada yang digunakan untuk mengatur timer dan ada pula bit-bit untuk mengatur interupsi (sela). Bit-bit pada register ini adalah sebagai berikut:

Tabel 2-10 Register TCON

Sumber : Ibnu Malik 2003 : 31

MSB				LSB			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Tabel 2-11 Keterangan Register TCON

Sumber : Ibnu Malik 2003 :31

Simbol	Posisi	Fungsi
TF1	TCON. 7	Timer 1 over flow flag, diset oleh perangkat keras saat timer atau counter menghasilkan over flow.
TR1	TCON. 6	Bit untuk menjalankan timer 1. diset oleh software untuk membuat timer ON atau OFF.
TF0	TCON. 5	Timer 0 over flag. Diset oleh hardware.
TR0	TCON. 4	Bit untuk menjalankan timer 0. Diset atau clear oleh software untuk membuat timer ON atau OFF.
IE1	TCON. 3	Eksternal interrupt 1 Edge.
IT1	TCON. 2	Interrupt 1 type control bit. Diset atau clear oleh software untuk menspesifikasi sisi turun atau level rendah dari intrupsi eksternal.
IE0	TCON. 1	Eksternal interrupt 0 edge flag.
IT0	TCON. 0	Interrupt 0 type control bit.

Pengontrolan pemilihan mode oprasi Timer atau counter adalah register timer mode (TMOD) yang mana definisi bit-bitnya adalah sebagai berikut :

Tabel 2-12 Tabel Register TMOD

Sumber : Ibnu Malik 2003 : 29

MSB				LSB			
Timer/Counter 1				Timer/Counter 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

Keterangan :

GATE : Saat Trx dalam TCON diset 1 dan GATE = 1, Timer atau counter x akan berjalan ketika Trx = 1 (timer dikontrol oleh software).

C/T : Pemilihan fungsi timer atau counter. Clear (0) untuk operasi timer dengan masukan dari sistem *clock internal*. Set (1) untuk operasi counter dengan masukan dari pin TO dan T1.

M1 : Bit pemilih mode 1.

M0 : Bit pemilih mode 0.

Tabel 2-13 Kombinasi M0 dan M1 Pada Register TMOD

Sumber : Ibnu Malik 2003 : 29

M1	M0	Mode	Operasi
0	0	0	Timer 13 bit.
1	1	1	Timer atau Counter 16 bit.
1	0	2	Timer atau reload 8 bit (pengisian otomatis).
1	1	3	TLO adalah timer atau counter 8 bit yang dikontrol oleh kontrol bit standart timer 0. THO adalah timer 8 bit dan di kontrol oleh bit timer 1. Atau split mode.

Jadi pada mikrokontroller MCS-51 ini, Timer/Counter dapat difungsikan menjadi 4 mode yaitu 13 bit timer, 16 bit timer, 8 bit auto reload dan split mode. 13 bit mode timer digunakan untuk menjaga kompatibilitas dengan pendahulunya yaitu mikrokontroller 8048. Namun saat ini timer 13 bit ini jarang digunakan.

2.1.9 Programmable Watchdog Timer (WDT)

Pada mikrokontroler Atmel AT89S8252 juga dilengkapi dengan watchdog timer. Watchdog timer ini menggunakan detak tersendiri. Untuk mengatur rentang waktu (periode) pada WDT ini maka terdapat bit prescaler yang dapat mengatur rentang waktu yang dibutuhkan.

Bit prescaler ini adalah bit PS0, PS1 dan PS 2 pada register WMCON. Periode waktu pada WDT ini berkisar dari 16 mili detik sampai 2048 mili detik. Karena bit prescalernya ada tiga maka akan ada 8 buah kemungkinan seperti yang tertera pada table dibawah ini :

Tabel 2-14 Pemilihan Periode Waktu Watchdog Timer

Sumber : Ibnu Malik 2003 : 45

PS 2	PS 1	PS 0	Periode
0	0	0	16 Mili Detik
0	0	1	32 Mili Detik
0	1	0	64 Mili Detik
0	1	1	128 Mili Detik
1	0	0	256 Mili Detik
1	0	1	512 Mili Detik
1	1	0	1024 Mili Detik
1	1	1	2048 Mili Detik

Watchdog timer (WDT) dilumpuhkan oleh power on reset (POR) dan selama power down . WDT diaktifkan dengan menseting bit WDEN pada SFR WMCON (alamat 96H). Jika perhitungan waktu WDT telah selesai (time out) tanpa ada reset atau dilumpuhkan maka suatu pulsa reset internal akan dihasilkan untuk mereset CPU.

2.1.10 Timer 2

Pada mikrokontroller Atmel AT89S8252 terdapat tambahan Timer 2, timer yang lainnya adalah timer 0 dan timer 1 yang juga terdapat pada standard MCS-51. Pada standard MCS-51 biasanya digunakan timer 1 sebagai penghasil baud rate. Nah pada mikrokontroller Atmel AT89S8252 ini selain dapat menggunakan timer 1 sebagai penghasil baud rate (untuk menjaga kompatibilitas dengan MCS-51) juga dapat menggunakan timer 2 sebagai penghasil baud rate untuk serial port.

Timer 2 ini merupakan timer atau counter yang berukuran 16 bit yang dapat beroperasi sebagai timer (dengan detak dari sistem detak mikrokontroller) atau dapat beroperasi sebagai penghitung kejadian (even counter) dengan detak dari luar. Untuk mengatur fungsi ini dilakukan dengan mengatur bit C/T2 pada SFR T2CON. Terlihat bahwa jika bit ini tinggi maka akan terpilih fungsi counter (C), tetapi jika bit ini rendah maka akan terpilih fungsi timer 2 (T2).

Timer 2 ini memiliki 3 mode operasi yaitu capture, auto reload (up dan down counting) dan baud rate generator. Untuk memilih mode ini dilakukan dengan mengatur bit pada SFR T2CON (timer 2 control register). Timer 2 ini terdiri dari 2 buah timer 8 bit register yaitu TH2 dan TL2. Pada fungsi timer, register TL2 dinaikkan (increment) tiap siklus mesin terdiri dari 12 periode osilasi, maka count rate menjadi $1/12$ dari frekuensi osilator. Pada fungsi counter, register dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pena yang bersesuaian (dalam hal ini pin T2 atau P1.0). Pada fungsi ini, masukan luar akan disampling selama S5P2 dari tiap siklus mesin. Tabel berikut menunjukkan mode operasi yang dapat dijalankan pada timer 2, yaitu :

Tabel 2-15 Mode Operasi Timer 2

Sumber : Ibnu Malik 2003 : 46

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16 Bit Auto Reload
0	1	1	16 Bit Capture
1	X	1	Baud Rate Generator
X	X	0	Off

2.1.11 Metode Pengalamatan

Metode pengalamatan yang digunakan pada MCS-51 terbagi menjadi dua jenis, yaitu pengalamatan langsung dan pengalamatan tidak langsung. (Moh. Ibnu Malik, 1997:36)

1. Pengalamatan Tak Langsung

Operasi pengalamatan tak langsung menunjukkan ke sebuah register yang berisi lokasi alamat memori yang akan digunakan dalam suatu operasi. Lokasi yang nyata tergantung dari isi register saat instruksi dijalankan. Untuk melakukan pengalamatan tak langsung digunakan :

- a. ADD A, @R0 : Tambahkan isi R0 dengan Acc dan hasilnya di Acc.
- b. DEC @R1 : Kurangi isi dari alamat R1.

2. Pengalamatan Langsung

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung. Untuk melakukan hal tersebut digunakan tanda

#. Misalnya :

- a. MOV A,#01H : Isi Acc dengan data 01H.
- b. MOV DPTR,#19H : Isi DPTR dengan data 19H.

Pengalamatan data langsung dari 0 sampai 127 akan mengakses RAM internal, sedangkan pengalamatan dari 128 sampai 255 akan mengakses register perangkat keras. Misalnya :

- a. `MOV P3,A` : Pindahkan isi Acc ke alamat *Port 3* (B0H).
- b. `INC 50` : Naikkan lokasi 50 (desimal) dalam memori.

2.1.12 Bahasa Assembler MCS-51

Bahasa assembler mikrokontroller MCS-51 dengan mikrokontroller Atmel AT89S8252 adalah sama, karena mikrokontroller AT89S8252 merupakan turunan dari mikrokontroller MCS-51. Bahasa assembler digunakan dalam setiap operasi CPU dalam bentuk bahasa yang disusun berurutan dalam pernyataannya. Masing-masing pernyataan akan diterjemahkan ke dalam instruksi bahasa mesin atau sering disebut *operation code/opcode*. Dalam penulisan bahasa mesin ini, terdapat berbagai macam kelompok instruksi, diantaranya :

a. Perpindahan Data

Instruksi ini digunakan untuk memindahkan data antar register memori.

Register memori antar muka register dan antar muka memori.

Contonya : `MOV A,R0` : Memindahkan isi register R0 ke Acc.

`MOV A,@R0` : Memindahkan isi alamat R0 ke Acc.

b. Operasi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, perkalian, maupun pembagian.

Contohnya : `ADD A,#data` : Menambah Acc dengan data.

`ADC A,#data` : Menambah Acc dengan data dan carry.

- INC R6 : Menambah isi R6 dengan 1.
 DEC R7 : Mengurangi isi R5 dengan 1.
 MUL AB : Mengalikan isi Acc dengan isi register B.
 DIV AB : Membagi isi Acc dengan isi register B.

c. Operasi Percabangan

Instruksi ini mengubah urutan normal pelaksanaan suatu program untuk melaksanakan pada lain tempat yang kita perlukan pada saat itu.

Contohnya :

1. CJNE (*Compare Jump Not Equal*)

Instruksi ini membandingkan isi lokasi memori tertentu dengan isi Acc, jika sama instruksi ini selanjutnya akan dieksekusi. Jika tidak sama eksekusi akan kembali ke alamat kode yang telah ditunjuk.

2. JB (*Jump if Bit Set*)

Instruksi ini akan menguji suatu alamat bit isi satu, eksekusi akan menuju ke alamat kode dan jika tidak instruksi akan dilanjutkan.

3. JNB (*Jump if Bit Not Set*)

Instruksi ini menguji suatu alamat bit. Jika berisi 0 maka eksekusi akan menuju ke alamat kode. Jika berisi 1 maka instruksi selanjutnya yang akan dieksekusi.

2.2 ISD (Information Storage Devices) 2560

ISD merupakan rangkaian yang digunakan untuk merekam pesan suara yang dapat diputar ulang. Salah satu jenis IC *voice recorder* adalah ISD (*Information Storage Device*). Satu *chip* ISD dapat digunakan untuk merekam lebih dari satu

pesan suara yang selanjutnya dapat diputar ulang atau *playback*. *Chipcorder* ISD menggunakan teknologi CMOS EEPROM sehingga dapat digunakan untuk merekam pesan suara yang berbeda dan dapat diubah-ubah. *Chipcorder* seri ISD2500 adalah jenis IC *voice recorder* yang mempunyai waktu penyimpanan suara selama 60 detik dengan spesifikasi yang mudah digunakan karena terkemas dalam satu *chip* dan mengurangi pemakaian komponen eksternal, kualitas suara yang identik dengan aslinya, tidak memerlukan pemrograman untuk merekam, dapat dialamati sehingga memungkinkan untuk merekam pesan suara lebih dari satu, memiliki mode *power down* otomatis ketika proses *playback* dan perekaman berakhir, memiliki sinyal *End-Of-Message* (EOM) sebagai pembatas antara rekaman pertama dengan berikutnya, pesan suara dapat disimpan selama 100 tahun (*tipycal*), dan catu daya tunggal. *Chipcorder* seri ISD2500 dikontrol oleh satu sinyal CE yang diset rendah yang berfungsi untuk menjalankan dan memutar ulang hasil rekaman. Dengan melakukan mode pengalamatan pada pin *input address* maka beberapa pesan atau suara dapat direkam dalam satu *chipcorder*. Jumlah alamat yang dimiliki adalah 960 alamat untuk 60 detik waktu rekam sehingga 1 alamat memiliki waktu rekam selama 1/16 detik.

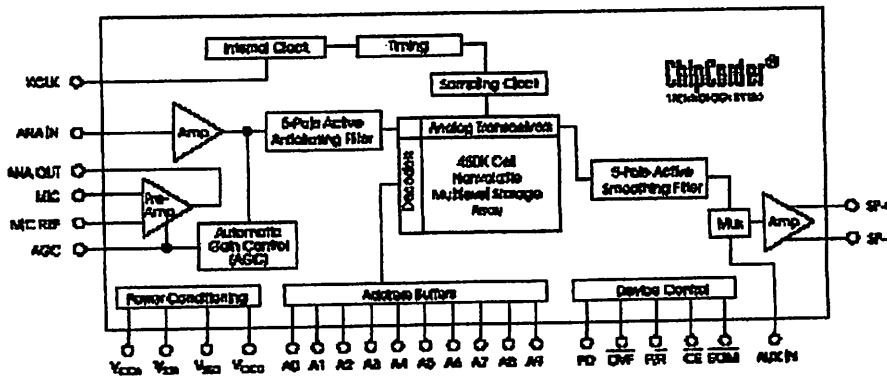
Tabel 2-16 Seri ISD 2500

Sumber : Data Sheet ISD 2560 : 2

Tipe	Waktu (Detik)	Sample Rate (KHZ)	Filter Band (HZ)
ISD 2560	60	8.0	3400
ISD 2575	75	6.4	2700
ISD 2590	90	5.3	2300
ISD 25120	120	4.0	1700

Seperti pada sampel rate tabel diatas banyak pilihan, yang memungkinkan pengguna untuk memilih kualitas suara yang diinginkan. Semakin bertambah

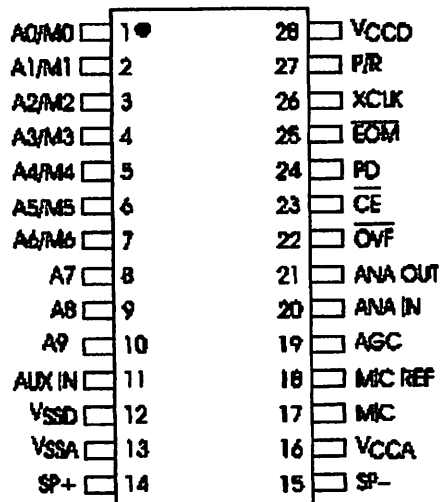
durasi penyimpanan semakin berkurang frekuensi sampling dan bandwidth yang akan mengakibatkan perubahan kualitas suara. Sampel suara disimpan secara langsung ke dalam sebuah chip memori nonvolatil tanpa digitalisasi dan kompresi seperti solusi lainnya. Blok diagram seri ISD 2560 dapat dilihat pada gambar dibawah ini :



Gambar 2-9 Blok Diagram ISD 2560

Sumber : Data Sheet ISD 2560 : 1

Konfigurasi kaki IC ISD2500 diperlihatkan dalam Gambar 2.10 Fungsi dari pin-pin tersebut adalah *Power Down Input* (PD) bila dalam keadaan tidak merekam atau memutar ulang, pin PD diberi pulsa tinggi dan power yang sangat rendah. *Chip Enable Input* (CE) di set rendah untuk menjalankan semua baik pemutaran ulang dan merekam suara, *Playback/record Input* (P/R) masukan P/R berfungsi mengunci pin CE. Masukan tinggi untuk memutar ulang sedangkan masukan rendah berfungsi untuk keadaan merekam.



Gambar 2.10 Konfigurasi *Pin* Dari ISD 2560

Sumber : Data Sheet ISD 2560 : 4

Pada RUN Output (EOM) penulisan bersifat *non-volatile* atau tidak bisa dihapus sehingga pengisian secara otomatis pada tiap-tiap pesan perekaman. *Overflow Output* (OVF) sinyal pulsa yang dihasilkan rendah (LOW) dikarenakan tempat pada memori kurang. *Mikrofon input* (MIC) pin ini biasanya dikopling dengan sebuah kapasitor untuk melewatkan semua sinyal input. *Mikrofon Reference* (MICREF) dihubungkan ke bagian ground *mikrofon* untuk mengurangi *noise* rekaman. *Output analog* (ANA OUT) merupakan sinyal dari *mikrofon* dikuatkan oleh rangkaian *pre-amplifier mikrofon* dalam chip dan pin ANA OUT merupakan *output* dari hasil penguatan tersebut. ANA IN digunakan sebagai input sinyal analog (termasuk *mikrofon*) melalui kopling kapasitor.

Operational external clock (XCLK) dapat dipicu (*clocked*) secara eksternal melalui pin XCLK, apabila tidak digunakan maka pin ini harus dihubungkan ke ground. *Address input* (A0-A9) pin *address input* memiliki dua fungsi yaitu sebagai bit alamat dan bit mode operasional, tergantung pada level

Most Significant Bit (MSB) dari alamatnya (A8 dan A9), apabila kedua MSB diberi logika 0 maka semua *input* pada pin ini dianggap sebagai bit-bit alamat dan digunakan sebagai alamat awal dari proses perekaman atau *playback*. Apabila kedua MSB diberi logika 1, *input* pada pin ini tidak dibaca sebagai bit alamat tetapi sebagai bit-bit mode operasional dimana fungsi masing-masing pin tersebut berbeda sebagaimana Tabel 2.18 berikut :

Tabel 2-17 Fungsi Dari Pin Address Input Pada Mode Operasional ISD 2560

Sumber : Data Sheet ISD 2560 : 7

Pin	Fungsi
M0	Melewati rekaman suara pertama dan memainkan rekaman berikutnya dengan sangat cepat.
M1	Menggabungkan beberapa suara menjadi satu rekaman dengan satu sinyal EOM.
M2	Tidak digunakan.
M3	Hanya memutar rekaman di alamat 0.
M4	Merekam atau membaca banyak data secara berurutan.
M5	Membolehkan penulisan atau perekaman berhenti sejenak.
M6	Menyederhanakan pin alat.

2.2.1 Konfigurasi Pin – Pin ISD 2560

- VCCA dan VCCD pin 16 dan 28 :

Analog dan *digital* sirkuit yang terdapat didalam chip ISD2560 menggunakan bus power yang terpisah untuk meminimalisasi *noise*. Pin power ini harus dihubungkan sedekat mungkin dengan sumber tegangan.

- VSSA dan VSSD (ground) pin 13 dan 12 :

Sama seperti VCCA dan VCCD, analog dan digital sirkuit didalam ISD 2560 menggunakan *bus ground* yang terpisah untuk meminimalisasi *noise*, *pin power* ini harus sedekat mungkin dengan *ground*.

- *Power Down Input (PD)* Pin : 24

Jika tidak digunakan untuk melakukan operasi *record (play back)*, harus diberi logika tinggi. Ketika pulsa *overflow (OVF)* rendah, PD harus berlogika tinggi untuk mereset address pointer kembali ke awal *record atau play back*.

- *Chip Enable Input (\overline{CE})* Pin 23 :

Berfungsi untuk mengaktifkan semua operasi *record dan play back*.

- *Play back/Record Input (P/\overline{R})* Pin 27 :

(P/\overline{R}) input ditahan dengan adanya transisi turun dari pin \overline{CE} . Logika tinggi akan memilih *play back cycle* dan logika rendah untuk memilih *record cycle*.

- *End-Of-Message Run Output (\overline{EOM})* Pin 25 :

Sebuah alamat akan dimasukkan secara otomatis diakhir masing-masing perekaman. Pulsa output (\overline{EOM}) akan rendah untuk setiap periode dari T_{EOM} .

- *Microphone Input (MIC)* Pin 17 :

Digunakan untuk mentransfer sinyal suara ke on-chip *preamplifier*.

- *Microphone Reference Input (MIC REF)* Pin 18 :

Merupakan input pembalik ke *microphone preamplifier* yang memberikan *noise-cancelling* atau *common-mode rejection* input ke IC ketika dihubungkan ke sebuah *microphone* diferensial.

- *Overflow Output (\overline{OVF})* Pin 22 :

Sinyal pulsa rendah pada akhir tempat memori, mengindikasikan bahwa IC ini telah terpenuhi dan pesan telah melebihi kapasitas. Keluaran (\overline{OVF})

kemudian diikuti masukan \overline{CE} sampai pulsa PD telah mereset. Pin ini juga berfungsi untuk menambah beberapa IC ISD 2560 untuk menambah durasi *record/play back*.

- *Automatik Gain Control (AGC)* pin 19 :

Kegunaan dari *AGC* untuk menambah atau mengurangi secara otomatis penguatan (*gain*) dari *pre-amplifier*. Dan juga meluaskan batas dari sinyal input yang dapat digunakan oleh *microphone* tanpa terjadi distorsi. *AGC* ini dapat secara dinamis meluaskan batas dari suara yang terekam dari suara bisikan sampai suara yang keras. Untuk menggunakan fasilitas *AGC* ini, resistor dan kapasitor luar (*eksternal*) harus dihubungkan secara paralel antara pin *AGC* dengan ground.

- *Analog Output (ANA OUT)* pin 21 :

Sinyal dari *microphone* dikuatkan dan dikeluarkan melalui *ANA OUT* pin. Penguatan tegangan dari *pre-amp* tergantung dari *AGC* pin. *Pre-amplifier* ini mempunyai penguat maksimum sekitar 24 dB untuk tingkat masukan sinyal kecil.

- *Analog Input (ANA IN)* pin 20 :

Kapasitor *eksternal* (Luar) menghubungkan antara *ANA IN* ke *ANA OUT* pin. Harga dari kapasitor luar bersama-sama dengan 3 K Ω input impedansi di *ANA IN*, dapat dipilih sendiri untuk memberikan keadaan *Cut Off* (terputus).

- *Eksternal Clock Input (XCLK)* pin 26 :

Untuk ISD 2560 mempunyai sebuah internal pulldown. Frekuensi clock sampling internal kurang lebih 1% dari spesifikasi. Frekuensi ini bervariasi dari $\pm 2,25\%$ berada pada suhu kamar dan dalam range tegangan operasi.

Internal clock mempunyai toleransi $\pm 5\%$ pada temperatur dan tegangan kerja. Digunakan untuk menambah kristal clock bila dibutuhkan pewaktuan yang lebih besar dan presisi. Bila pin XLCK ini tidak digunakan, pin ini harus dihubungkan dengan ground.

- *Speaker Output* (SP+ dan SP-) pin 14 dan 15 :

ISD 2560 telah mempunyai sebuah driver *on-chip differential speaker* yang sanggup memikul beban 50Mw dengan *Impedansi* sebesar 16Ω dari AUX IN. Speaker output berada pada level V_{SA} selama proses *record* dan *power down*.

- *Auxillary Input* (AUX IN) Pin 11 :

Dihubungkan langsung ke kaki keluaran *amplifier* dan keluaran *speaker* ketika \overline{CE} , P/\overline{R} berada pada logika tinggi dan *play back* tidak aktif.

- *Adress Mode Input* (AX/MX) Pin 1-10 dan 1-7 :

Mempunyai dua fungsi tergantung pada level dari dua most significant bit (MSB) dari alamat tersebut (A8 – A9). Jika salah satu atau keduanya dari MSB berlogika rendah, semua input dianggap sebagai bit alamat dan digunakan sebagai awal alamat untuk proses record atau play back terbaru.

Alamat input ini ditahan transisi turun dari \overline{CE} . Jika kedua MSB berlogika tinggi, address/mode input dianggap sebagai bit mode.

2.3 Liquid Crystal Display (LCD) M1632

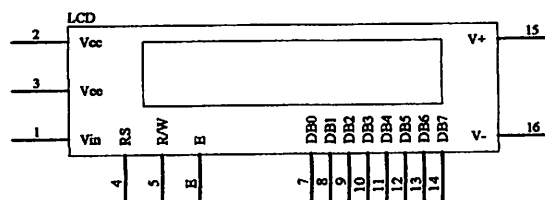
Liquid crystal display (LCD) merupakan komponen elektronika yang digunakan untuk menampilkan suatu karakter baik itu angka maupun karakter tertentu, sehingga tampilan tersebut dapat dilihat secara visual. Pemakaian LCD

ini banyak digunakan karena daya yang dibutuhkan relatif kecil, selain itu dapat menampilkan angka, huruf atau simbol dan karakter tertentu meskipun pada komponen ini dibatasi sumber cahaya eksternal/internal, suhu dan *life time*.

LCD terdiri atas tumpukan tipis atau sel dari dari dua lembar kaca dengan pinggirannya tertutup rapat. Antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang tembus cahaya. Permukaan luar masing-masing keeping kaca mempunyai lapisan tembus cahaya seperti *oxida (tien oxida)* atau *oxida indium (indium oxida)* sel mempunyai ketebalan 1×10^{-5} meter dan diisi dengan kristal cair.

Kristal cair adalah suatu bahan yang akan mengalir seperti sebuah cairan tetapi struktur molekulnya seperti benda padat. Pada LCD terdapat suatu unit penghamburan cahaya, yang mana terdapat suatu proses *nematic liquid crystal*.

Pada proses tersebut permukaan penghantar *idium oksida* yang tembus pandang, sehingga saat cahaya datang dan melewatinya struktur kristal cair akan kelihatan bersih. Jika diberikan tegangan pada permukaan penghantar, susunan molekul terganggu yang menyebabkan perbedaan penyebaran pada daerah yang terbentuk. Cahaya datang dipantulkan dalam arah yang berbeda pada titik temu antara penyebaran indeks yang berbeda pada daerah dengan hasil hamburan sinar yang menampakkan suatu lapisan kaca, dimana hubungan antara permukaan berlawanan, antara yang satu dengan yang lain.



Gambar 2-11 LCD 1632

Sumber : Data Sheet LCD M1632

LCD yang digunakan pada perancangan alat ini adalah tipe M1632. LCD tipe ini merupakan suatu jenis tampilan yang menggunakan *liquid crystal* dan menampilkan suatu karakter secara dot matrik. LCD ini memiliki ciri-ciri sebagai berikut :

1. LCD ini terdiri dari 32 karakter dengan dua baris masing-masing 16 karakter dengan *display* dot matrik 5x7 dan kursor.
2. Karakter generator ROM dengan 8 tipe karakter (Untuk program write).
3. Karakter generator RAM dengan 8 bit karakter.
4. 80x8 *display* data RAM.
5. Dapat *di-interfacekan* dengan MPU 8 atau 4 bit.
6. Dilengkapi dengan fungsi hambatan , *display clear*, *cursor home*, *display ON/OFF*, *cursor ON/OFF*, *display karakterblink*, *cursor shift*, *display shift*.
7. Internal data.
8. Power On-Reset.
9. +5 V Single power supllly.
10. RAM data dan RAM generator karakter dapat dibaca dari MPU.
11. Range temperature operasi 0-60 °C.

LCD ini mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontrol CMOS didalamnya. Kontroler tersebut sebagai pembangkit dari karakter ROM atau RAM dan display data RAM. Semua fungsi tampilan di kontrol oleh suatu instruksi dan modul LCD dapat dengan mudah untuk *di-interfacekan* dengan mikrokontroler. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih termultiplex dengan bus alamat, serta 3 bit sinyal kontrol. Sementara pengendalian dot matrik LCD dilakukan secara eksternal oleh kontroler yang sudah ada pada modul LCD. Dasar-dasar pengoperasian LCD ini terdiri dari pengoperasian dasar pada *register, busy flag, adres counter, display data RAM*.

2.3.1 Register

Kontroler dari LCD mempunyai dua buah register 8 bit yaitu instruksi register (IR) dan data register (DR) yang dapat dilihat pada Tabel 2.7. IR menyimpan kode instruksi seperti *display clear, cursor shift* dan informasi alamat dari *display data RAM (DDRAM)* serta karakter generator (CGRAM). Data register juga bisa menyimpan data yang telah ditulis atau dibaca oleh DDRAM maupun CGRAM. Bila data ditulis dalam DDRAM atau CGRAM dari mikroprosesor unit (MPU) maka data dalam DR secara otomatis akan ditulis dalam DDRAM atau CGRAM. Dan dalam membaca data dari DDRAM atau CGRAM maka alamat data yang penting akan ditulis dari IR. Kemudian data tersebut dibaca oleh DR dan MPU membaca data tersebut dari DR. Setelah operasi pembacaan selesai maka data pada DDRAM atau CGRAM yang alamatnya telah dibaca masuk ke DR dan operasi pembacaan dilanjutkan.

Tabel 2-18 Register Selection

Sumber : Data Sheet LCD M1632 : 8

RS	R/W	Operasi
0	0	IR selection, IR write Operasi internal : <i>display clear.</i>
0	1	<i>Bussy Flag(DB7) address Counter(DB0-DB6) read.</i>
1	0	DR selection DR write Operasi internal : DR ke DDRAM atau CGRAM.
1	1	DR selection DR read Operasi internal: DDRAM atau CGRAM ke DR.

2.3.2 Busy Flag

Busy flag menunjukkan bahwa modul siap untuk menerima instruksi, selanjutnya seperti terlihat dalam tabel, register seleksi sinyal akan melalui DB7 jika RS = 0 dan RW = 1 jika nilai 1 maka modul LCD sedang melakukan kerja internal dan instruksi tidak akan diterima. Oleh karena itu status flag harus dicek dahulu sebelum instruksi dijalankan. Jika instruksi dijalankan tanpa pengecekan status flag terlebih dahulu, maka instruksi akan gagal.

2.3.3 Address Counter

Address counter menunjukkan lokasi memori dalam LCD. Pemilihan lokasi alamat itu diberikan lewat register intruksi (IR) yang mana alamat ini ditransfer dari IR ke AC. Bila *display* data ditulis atau dibaca dari DDRAM atau CGRAM maka AC akan mengurangi atau menambah *mode set entry* data masuk.

2.3.4 DDRAM

DDRAM mempunyai kapasitas 80 x 8 bit dan menyimpan data 80 bit kode karakter. Karakter Pada LCD masing-masing pin punya *range* alamat sendiri-

sendiri, alamat itu diekspresikan dengan bilangan heksadesimal. Untuk line 1 *range-nya* antara 00H - 0FH, sedang untuk line 2 antara 40H - 4FH. Beberapa lokasi penyimpanan yang tidak digunakan untuk *display* digunakan untuk *general data* RAM.

2.3.5 CGRAM

CGRAM digunakan untuk membuat karakter tersendiri melalui program. Untuk lebih mengetahui fungsi dari masing-masing pin atau terminal dari tipe LCD ini dapat dilihat dalam Tabel 2.20 berikut:

Tabel 2-19 Fungsi Masing-Masing Pin Atau Terminal LCD tipe M1632

Sumber : Data Sheet LCD M1632 : 7

Nama Sinyal	No terminal	I/O	Tujuan	Fungsi
DB0-DB7	8	I/O	MPU	Sebagai lalu lintas data dan intruksi ke/dari MPU, lower byte
E	1	Input	MPU	Sinyal penanda operasi start (read/write)
R/W	1	Input	MPU	Sinyal seleksi register 0 = write 1 = read
RS	1	Input	MPU	Sinyal seleksi register 0 = Register instruksi (write) <i>busy flag</i> dan address conter (read) 1 = Data register (write dan read)
VLCD	1	-	PSU	Untuk driver LCD guna pengaturan contrast
VDD	1	-	PSU	+5 volt
VSS	1	-	PSU	Ground terminal = 0 volt
V+BL	1	-	PSU	4-4,2 V 50-200 Ma
V-BL	1	-	PSU	0 V (GND)

Intruksi-intruksi yang digunakan LCD tipe ini dapat dilihat dalam tabel

2.21. berikut :

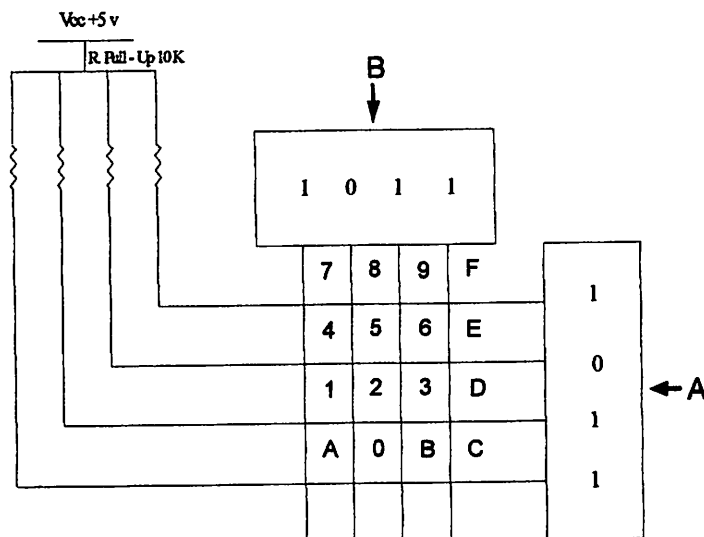
Tabel 2-20 Instruksi LCD

Sumber : Data Sheet LCD M1632 : 16

No	Instruksi	RS	RW	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
1	Display clear	0	0	0	0	0	0	0	0	0	1
2	Cursor Home	0	0	0	0	0	0	0	0	1	*
3	Entri mode set	0	0	0	0	0	0	0	1	I/ O	S
4	Display on/of control	0	0	0	0	0	0	1	D	O C	B
5	Cursor / display shift	0	0	0	0	0	1	S/ L	R /L	*	*
6	Function set	0	0	0	0	1	D L	1	*	*	*
7	CG RAM adress set	0	0	0	1	Alamat Karakter					
8	DD RAM Adrees set	0	0	1	Tampilan alamat data						
9	BF / Adress set	0	1	B F	Tampilan alamat bus						
10	Penulisan ke CG RAM Atau ke DD RAM	1	0	Tulis Data							
11	Pembaca an CG RAM	1	1	Baca Data							

2.4 Keypad

Keypad disini digunakan untuk memasukkan data acuan , proses scanning matrik keypad pada dasarnya mendecoder penekanan suatu tombol dengan konfigurasi matrik. Diumpamakan port B dioperasikan sebagai output scanning bagian kolom. Scanning dilakukan secara berurutan dari kolom paling kiri sampai kolom paling kanan, kolom yang aktif akan berada pada kondisi low. Untuk mengetahui ada tidaknya tombol ditekan, Maka harus dilakukan pembacaan terhadap port (Diumpamakan port A) yang dioperasikan sebagai input dari setiap baris pada kolom yang sedang aktif. Jika tidak ada tombol yang ditekan maka semua kondisi baris akan high karena dipull-up oleh resistor pull-up Vcc. Jika salah satu baris, tombol ditekan maka tombol tersebut terletak pada kolom yang sedang aktif, kondisi baris yang terbaca pada port A adalah low. Hal ini dapat dijelaskan pada gambar 2. berikut :



Gambar 2-12 Proses Scanning Keypad Matrik 4 x 4

Diumpamakan pin Y1-Y4 sebagai baris dan pin X1-X4 sebagai kolom. Yaitu apabila salah satu pin misalnya Y1 terhubung dengan X1 maka data outputnya akan = 0 sedangkan untuk Y1 dan X2 data outputnya akan = 1. Diumpamakan tombol 5 ditekan maka pada proses kerjanya port B dikirim data 0111. Dalam keadaan ini dilakukan pembacaan pada port A dan hasil yang yang diperoleh adalah 1111. Ini berarti tidak ada tombol yang ditekan pada kolom paling kiri. Selanjutnya pada port B dikirim data 1011 dan dilakukan pembacaan dari port A. Hasil diperoleh dari pembacaan tersebut yaitu 1011. Ini berarti ada tombol yang ditekan pada kolom kedua baris kedua. Melalui software yang di buat dapat diketahui kode dari tombol yang ditekan. Untuk menghindari pembacaan yang salah karena adanya dua tombol atau lebih yang ditekan bersamaan, maka proses scanning dilakukan terhadap seluruh tombol keypad dan perhitungan jumlah tombol yang ditekan.

2.5 IC Audio Amplifier LM 386

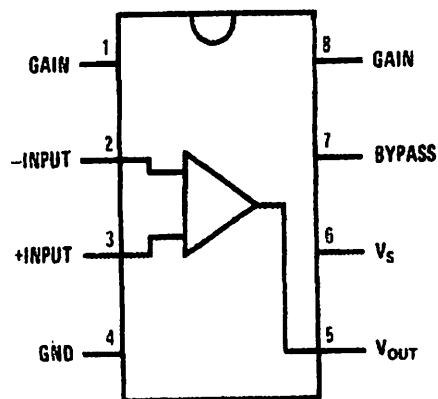
LM 386 ini adalah power amplifier yang dirancang untuk digunakan dalam aplikasi yang menggunakan tegangan rendah (low voltage).Komponen ini digunakan untuk memperkuat sinyal keluaran yang berupa audio *amplifie dari ISD*. Penguatan secara internal mulai dari 20 kali untuk mempertahankan bagian luar menghitung rendah, tetapi dengan penambahan resistor dan kapasitor external antara pin 1 dan 8 akan meningkatkan penguatan, nilainya dari 20 sampai 200 kali.

Masukan-masukan (input) disesuaikan dengan pertanahan sementara output ini membias secara otomatis ke satu setengah tegangan supply. Daya

tenang ini hanya 24 miliwatt dimana operasi dari tegangan adalah 6 volt membuat LM 386 yang ideal untuk operasi baterai.

Fitur – fitur yang ada pada IC LM 386 antara lain :

1. Ukuran tegangan yaitu = 4V-12V atau 5V-15V.
2. Konsumsi arus sebesar 4 mA.
3. Penguatan gain dari 20 – 200.
4. Mempunyai Vref pentanahan (harus ditanahkan).
5. Distorsinya yang rendah : 0,2%($A_v = 20, V_s = 6V, R_l = 8\Omega, P_o = 125mW, f = 1Khz$).



Gambar 2-13 Blok Diagram Audio Amplifier LM 386

Sumber : Data Sheet LM 386 : 1

2.5.1 Penguatan Yang Terkontrol

Untuk membuat LM 386 menjadi suatu amplifier yang serbaguna, 2 pin (1 dan 8) disediakan untuk keuntungan pengendalian, dengan pin 1 dan 8 terbuka maka resistor 1,35 K Ω diset untuk penguatan 20 kali. Jika kapasitor diletakkan

antara pin 1 dan 8 maka resistor $1,35 \text{ K}\Omega$ sebagai bypass sehingga keuntungannya akan mendekati mencapai 200 kali.

Jika suatu resistor ditempatkan secara urut penguatannya dapat mencapai dari 20 – 200 kali. Untuk memperoleh kendali dapat juga dipakai kapasitif dengan menggabungkan suatu resistor atau FET dari pin 1 ke ground.

Komponen eksternal tambahan dapat diletakkan secara paralel dengan umpan balik resistor internal. Untuk memperoleh tanggapan respon frekuensi, untuk aplikasi yang berdiri sendiri sebagai contoh kita dapat mengganti kerugian bus pembicara yang lemah dengan frekuensi yang membentur alur umpan balik, dapat dilakukan dengan memasang rangkaian RC dari pin 1 ke pin 5 selama 6 db bus efektif menaikkan tegangan 15 K nilai yang paling rendah untuk baiknya operasi stabil $R = 10 \text{ K}$, jika pin 8 terbuka. Pin 1 dan 8 pada posisi bypass maka R lebih rendah $2 \text{ K}\Omega$ yang dapat dipakai.

2.5.2 Bias Masukan

Gambar skematik ditampilkan untuk kedua masukan dengan bias ke ground dengan $50 \text{ K}\Omega$ resistor. Arus transistor masukan adalah sekitar 250 nA sehingga tegangan masukan sekitar 12,5 mV dimana masih terbuka, jika sumber tegangan DC yang mengendalikan LM 386 lebih tinggi dibanding $250 \text{ K}\Omega$ akan memberikan sangat sedikit tambahan offset (sekitar 2,5 mV dimasukkan, 50 mV dikeluarkan)

Untuk resistansi sumber DC antara nilai – nilai ini kita dapat menghapus offset kelebihan dengan meletakkan suatu resistor dari masukan ke ground. Nilai

yang sama untuk sumber resistansi DC tentu saja semua masalah offset dihilangkan jika masukkan secara kapasitif digabungkan.

Ketika menggunakan LM 386 dengan penguatan tinggi (membypass resistor $1,35\text{ K}\Omega$ antara pin 1 dan 8) diperlukan untuk membypass masukkan yang tidak dipakai, mencegah penurunan derajat dari penguatan dan ketidakstabilan. Ini dilakukan dengan menghubungkan singkat kapasitor sebesar $0,1\mu\text{F}$ ke ground tergantung pada resistansi sumber masukkan yang dikendalikan.

2.6 Loudspeaker

Loudspeaker berfungsi mengubah sinyal listrik menjadi gelombang audio. Loudspeaker adalah suatu transduser yang membangkitkan gelombang suara dari suatu sinyal AC. Prinsip kerja loudspeaker ini adalah berdasarkan pergerakan kumparan (*voice coil*). Apabila suatu sinyal diberikan pada kumparan yang ditempelkan pada sebuah diafragma yang mempunyai sifat relatif bebas bergerak pada medan magnet maka diafragma ini selanjutnya akan menghasilkan getaran suara yang dapat didengar.

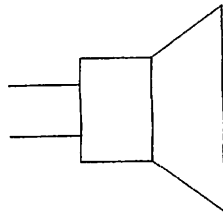
2.6.1 Bagian – Bagian Dari Loudspeaker

Loudspeaker terdiri dari beberapa bagian antara lain :

1. Magnet radial (berbentuk bulat), atau magnet tetap yang mempunyai gaya magnet yang sangat kuat.
2. Kumparan (*moving coil*), berupa gulungan kawat email yang menempati ruang bebas pada celah magnet.
3. Kertas membran berbentuk kerucut (*conus*), diujung kerucut melekat kumparan dimana kedua ujung kawat masing – masing dilekatkan pada

kerucut dan untuk saluran keluar disambungkan pada rangkanya yang dilapisi isolator.

4. Kuker merupakan tabung kertas yang terletak pada seputar badan magnet dicelah udara antara kutub utara dan kutub selatan badan magnet.
5. Rangka dari logam.



Gambar 2-14 Simbol Dari *Loudspeaker*

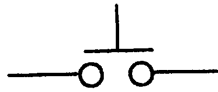
2.6.2 Ukuran Loudspeaker

Setiap loudspeaker mempunyai tiga macam ukuran yaitu :

1. Garis tengah kerucut diukur dengan satuan inci (1 inci = 2,54 cm). Dalam prakteknya terdapat loudspeaker dengan ukuran 2 inci, 3 inci, 4 inci, 6inci, 8 inci dan seterusnya.
 2. Daya listrik yang dibutuhkan loudspeaker diukur dengan satuan watt.
 3. Impedansi loudspeaker diukur dengan satuan ohm. Impedansi adalah hambatan listrik yang terdapat dalam kumparan bila dilalui arus listrik AC.
- Pada Tugas akhir ini, loudspeaker yang digunakan mempunyai ukuran $3\frac{7}{8}\Omega/0,5w$. Berarti garis tengah kerucutnya 3 inci, impedansinya 8 ohm dengan daya maksimumnya sebesar 10 watt.

2.7 Push Button

Push Button merupakan mikro switch yang digunakan sebagai inputan atau pemberi sinyal pada mikrokontroler untuk diproses sesuai dengan program yang telah dibuat.



Gambar 2-15 Push Button

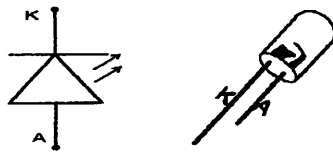
2.8 LED (*Light Emitting Diode*)

Light emitting diode adalah dioda semi konduktor khusus yang dirancang untuk memancarkan cahaya bila ada arus yang melaluinya. prinsip kerjanya adalah pada waktu led infra merah dibias *forward*, elektron dari pita konduksi melewati *junction* dan jatuh kedalam *hole* pita valensi, sehingga elektron-elektron tersebut memancarkan energi, pada dioda penyearah biasa energi ini dikeluarkan sebagai energi panas, tetapi pada led energi ini dipancarkan sebagai cahaya. Led infra merah memancarkan cahaya yang tidak kelihatan oleh mata.

Led merupakan PN *junction* yang memancarkan cahaya. Apabila anoda diberi tegangan positif dan katoda ke *ground*, maka led menjadi "ON" dan arus akan mengalir dari anoda ke katoda. Pada reaksi semikonduktor suatu dioda akan terjadi perpindahan elektron dari tipe N menuju tipe P dan perpindahan hole dari tipe P ke tipe N, proses rekombinasi antara elektron dan hole mengakibatkan pelepasan energi berupa pancaran cahaya.

LED harus dioperasikan di dalam ukuran kerja tegangan dan arus tertentu untuk mencegah kerusakan yang tidak dapat diubah lagi. Sebagian besar LED

membutuhkan 1,5 V sampai 2,2 V untuk memberi bias maju yang dapat mengatasi aman arus sebesar 10 sampai dengan 20 mA untuk kecerahan maximum. LED biasanya dihubungkan seri dengan tahanan yang membatasi tegangan dan arus pada nilai yang dikehendaki. Keuntungan utama penggunaan LED sebagai sumber cahaya dibandingkan dengan bola lampu cahaya biasa adalah penggunaan daya yang jauh lebih rendah, jauh lebih lama umurnya (lebih dari 20 tahun) dan beroperasi dengan kecepatan tinggi. Simbol dan bentuk fisik LED seperti gambar 2.16 berikut ini.



(a.) Simbol

(b.) Bentuk Fisik

Gambar 2-16 Simbol Bentuk dan Fisik Light Emitting Dioda

BAB III

METODELOGI PENELITIAN

3.1. Pendahuluan

Dalam Bab metodologi penelitian ini akan dibahas pembuatan seluruh sistem perangkat yang ada pada alat Penerima Tamu Elektronik Berbasis Mikrokontroler AT89S8252, secara garis besar terdapat dua bagian perangkat yang ada yaitu:

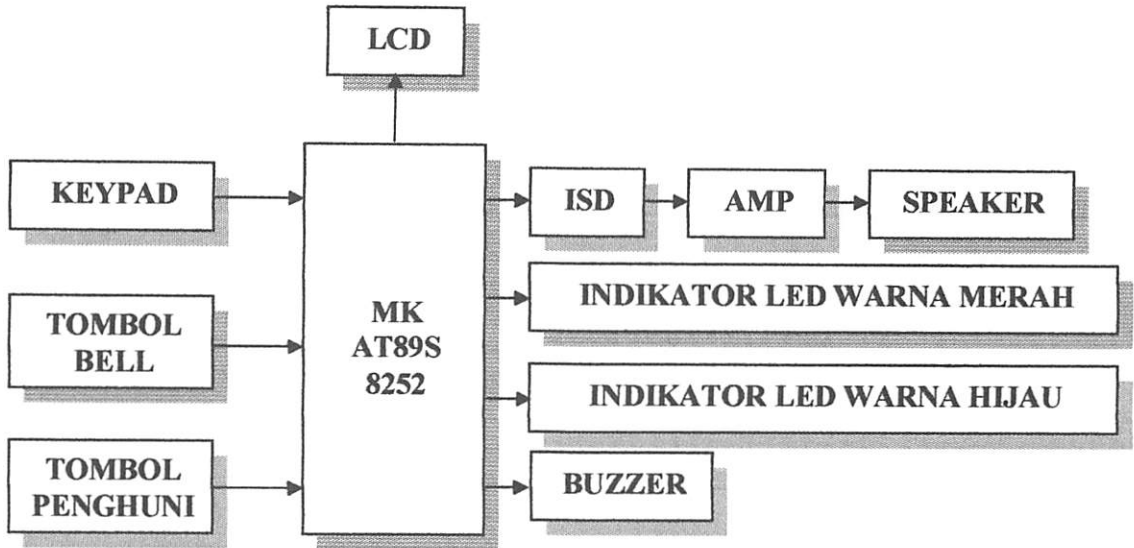
1. Perencanaan perangkat keras
2. Perencanaan perangkat lunak

Pada perencanaan perangkat keras akan meliputi penjelasan dari perencanaan diagram blok sistem dan juga perencanaan minimum sistem dari mikrokontroler AT89S8252, beserta *peripheral* yang digunakan. Pada perencanaan perangkat lunak akan meliputi penjelasan dari perangkat lunak yang digunakan pada minimum sistem *mikrokontroler* AT89S8252. Akan tetapi perangkat tersebut dalam kerjanya akan saling menunjang antara perangkat satu dengan perangkat yang lainnya sehingga alat yang direncanakan dapat berjalan sesuai dengan perencanaannya.

3.2 Perencanaan Perangkat Keras

Dalam Tugas Akhir ini perencanaan dan pembuatan Alat Penerima Tamu Elektronik berbasis *mikrokontroler* AT89S8252 sebagai kontrol utama dan menggunakan komponen lain sebagai komponen pendukung. Sebelum membuat perangkat keras terlebih dahulu direncanakan blok diagram yang akan dibuat dan

kemudian membahasnya sesuai dengan blok diagram tersebut. Adapun blok diagram alat tersebut adalah sebagai berikut:



Gambar 3–1 Diagram Blok Secara Keseluruhan

Sumber : Perencanaan.

Gambar diatas adalah diagram blok dari rangkaian sistem. Rangkaian alat pengontrol tersebut terdiri dari rangkaian *keypad*, rangkaian pengontrol sistem *mikrokontroller* AT 89S8252, rangkaian *LCD* MI632, rangkaian tampilan suara ISD 25560 dan lain - lainnya.

3.2.1 Cara Kerja Rangkaian Keseluruhan

3.2.1.1 Prinsip Kerja Alat Bila Penghuni Tidak Ada Di Rumah

Keseluruhan jalannya rangkaian diatur oleh program yang telah dibuat dalam mikrokontroler sebagai pengendali sistem secara keseluruhan. Penghuni yang akan meninggalkan rumah diharuskan merubah statusnya dengan menekan tombol COR dan tombol panah. Dikarenakan status penghuni telah dirubah

maka LCD tidak menampilkan nama penghuni yang meninggalkan rumah. Apabila ada tamu maka tamu bisa meninggalkan pesan, disaat penghuni kembali ke rumahnya maka ia dapat melihat pesan yang masuk sesuai yang ditampilkan LCD.

3.2.1.2 Prinsip Kerja Alat Untuk Penghuni Yang Berada Di Rumah

Sedangkan prinsip kerja bila penghuni berada di rumah adalah setelah tamu menekan bel maka LCD akan menampilkan nama-nama dan kode anggota keluarga yang ada di rumah. Tamu dapat menekan kode tertentu untuk memanggil anggota keluarga yang dicari.

3.2.2 Fungsi Komponen Dari Rangkaian Sistem

Pada gambar blok diagram rangkaian keseluruhan diatas, dapat dilihat beberapa blok diagram yang masing-masing memiliki fungsi :

1. Keypad

Digunakan sebagai media penulisan pesan dan pemilih nama panggilan.

2. Tombol Bel

Berfungsi untuk mengaktifkan agar buzzer dapat bekerja.

3. Tombol Penghuni

Berfungsi untuk merubah status dari penghuni rumah bila mau keluar rumah dan sebaliknya.

4. Mikrokontroler AT89S8252

Berfungsi untuk mengontrol komponen – komponen yang terhubung secara keseluruhan.

5. LCD M1632

Digunakan sebagai alat untuk menampilkan menu serta pesan yang masuk.

6. ISD 2560

Berfungsi sebagai media perekam dan pemutar suara.

7. Loudspeaker

Berfungsi untuk memanggil penghuni rumah dengan suara yang sudah diatur.

8. Indikator berupa lampu LED, terdiri dari LED warna hijau dan LED warna merah.

Berfungsi LED warna merah digunakan untuk indikator bahwa penghuni tidak berada di rumah. Sedangkan LED warna hijau digunakan sebagai indikator adanya pesan masuk.

9. Operasional Amplifier

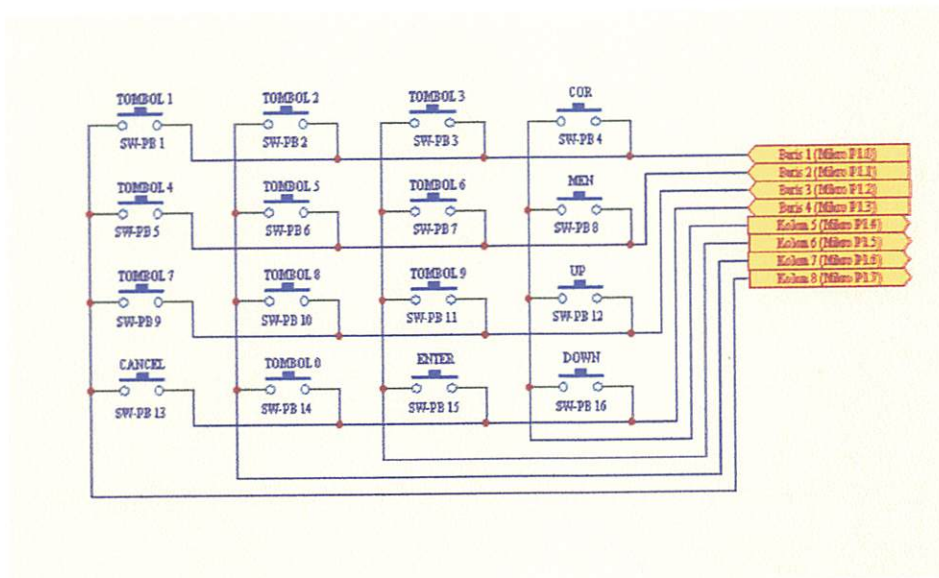
Digunakan sebagai penguat sinyal suara dari ISD 2560.

10. Buzzer

Berfungsi sebagai pembangkit suara.

3.3 Perencanaan Rangkaian Keypad Matrik 4 x 4.

Keypad digunakan sebagai masukan data referensi dan mengubah data yang diinginkan. Data tersebut dirubah dalam bentuk kode biner oleh rangkaian keypad. Membaca tombol ke mikrokontroller merupakan pemrograman input karena data diberikan untuk dibaca oleh mikrokontroller. Bila kita menggunakan tombol yang jumlahnya cukup banyak, maka untuk menghindari pemakaian jumlah port yang besar, kita bisa menyusun tombol tersebut secara matrix. Keypad yang dipakai yaitu berukuran 4 x 4 (4 bagian kolom dan 4 bagian baris) yang terdiri dari saklar angka 0 sampai 9 sebagai masukan data dan karakter Enter, Cancel, Up, Down, Cor dan Men yang digunakan untuk merubah status penghuni rumah dan untuk baca pesan singkat. Dengan menggunakan matrix 4 x 4 sehingga dengan membutuhkan 8 kaki port dapat menterjemahkan sebanyak 16 tombol. Keypad matrix 4 x 4 ditunjukkan pada gambar berikut :



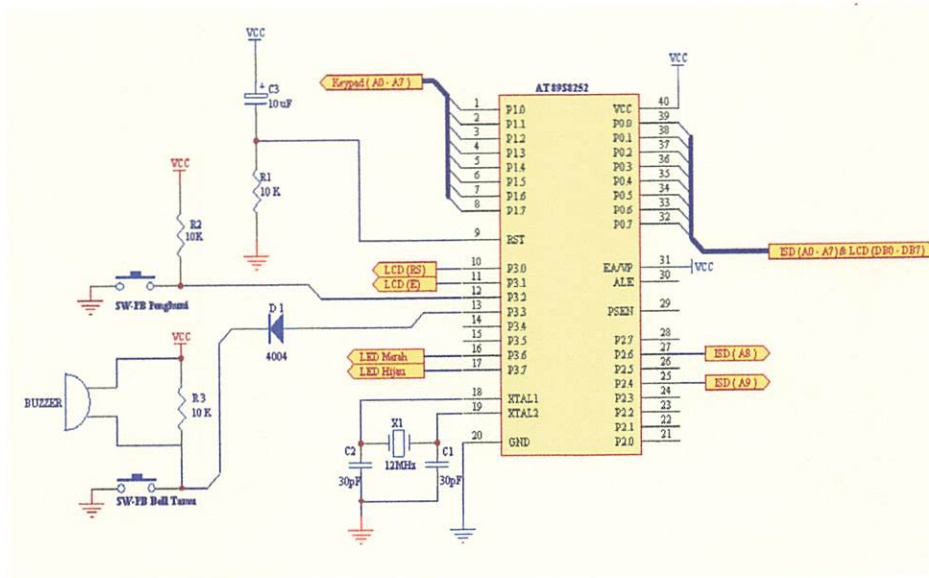
Gambar 3–2 Rangkaian Keypad Matrix 4 x 4

Sumber : Ricky 2004

Adapun cara kerja keypad yang direncanakan dapat dijelaskan sebagai berikut : Setiap kali penekanan tombol akan terjadi suatu persilangan antara baris X dengan kolom Y. kondisi logic hasil penekanan tombol keypad tersebut dihubungkan pada Port Input (P1.0 – P1.7) melalui kaki X1-X4 dan Y1-Y4. Keadaan penekanan tombol persilangan antara baris X dan kolom Y akan dibaca dan untuk sementara disimpan dimemory internal mikrokontroller sehingga persilangan antara baris dan kolom dapat dikirimkan ke MCU.

3.4 Perencanaan Rangkaian Tombol

Pada sistem ini digunakan juga beberapa komponen tambahan berupa tombol *push button* 2 buah, yang rangkaiannya dapat dilihat pada gambar berikut :



Gambar 3–3 Perencanaan Rangkaian Tombol

Untuk membuat alat yang dikehendaki, maka perlu adanya pembagian port-port yang digunakan. Port-port yang digunakan dalam Mikrokontroller itu adalah sebagai berikut :

- Port P0.0 - P0.7

Digunakan sebagai *output* mikrokontroler dan merupakan alamat bagi IC ISD2560 (A0-A7) serta dihubungkan ke LCD (DB0-DB7).

- Port P1.0 - P1.7

Dihubungkan dengan port-port yang ada pada keypad berfungsi sebagai inputan data masuk sehingga diolah oleh MK.

- Port P2.5 Dan P2.7.

Dihubungkan pada pin PD dan P/\bar{R} pada ISD2560 yang berfungsi sebagai pengontrol ISD sebagai perekam atau pemutar suara

- Port P2.6 dan P2.4

Digunakan sebagai *output* mikrokontroler dan merupakan alamat bagi IC ISD2560 (A8 dan A9)

- RXD dan TXD atau P3.0 dan P3.1

Dihubungkan dengan LCD (RS dan E) agar dapat menampilkan data.

- XTAL2 dan XTAL1 atau pin 18 dan 19

Dihubungkan dengan 2 kapasitor dan kristal yang merupakan komponen pendukung dari sebuah mikrokontroller.

- INT0 dan INT1 atau P3.2 dan P3.3

Dihubungkan ke tombol penghuni dan tombol bel tamu.

- WR dan RD atau P3.6 dan P3.7

Dihubungkan dengan LED indikator warna merah dan hijau

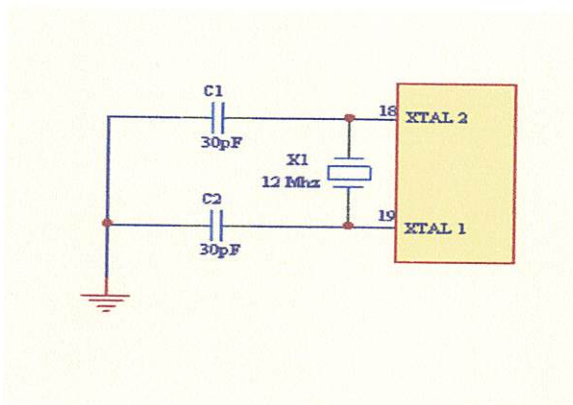
3.5.1 Rangkaian Clock Internal

Kecepatan proses pengolahan data pada mikrokontroler ditentukan oleh Clock (pewaktu) yang dikendalikan oleh mikrokontroler tersebut. Pada Mikrokontroler AT 89S8252 terdapat internal Clock. Internal Clock generator berfungsi sebagai sumber Clock, tapi masih memerlukan rangkaian tambahan untuk membangkitkan clock yang diperlukan. Rangkaian Clock ini terdiri dari dua buah kapasitor dan sebuah kristal yang dirangkai sedemikian rupa dan kemudian dihubungkan dengan Pin 18 dan 19 pada AT 89S8252. Cara mengkases clock internal yang terdapat pada chip mikrokontroler yaitu sebuah kristal pada pin XTAL 1 dan pin XTAL 2 dengan dua buah kapasitor yang masing – masing dihubungkan ke kaki kristal dan dihubungkan ke ground.

Dalam perancangan rangkaian ini menggunakan:

1. $C = 30 \text{ pF}$. Penentuan besarnya kapasitansi disesuaikan dengan spesifikasi pada data sheet AT 89S8252.
2. Kristal 12 MHz (berdasarkan data sheet AT 89S8252)

Untuk gambar rangkaian clock dari mikrokontroler AT89S8252 ditunjukkan pada gambar berikut :



Gambar 3–5 Rangkaian Clock

Sumber : Data Sheet AT 898252

Dengan menggunakan kristal diatas maka dapat dihitung waktu yang diperlukan untuk satu siklus mesin, yaitu :

$$F = 12 \text{ MHz}$$

Sehingga $T = \frac{1}{f}$

$$T = \frac{1}{12\text{MHz}} = \frac{1}{12} \mu\text{s}$$

Maka untuk satu siklus mesin dari mikrokontroller besarnya adalah :

$$\text{Time} = 12 \times T$$

$$\text{Time} = 12 \times \frac{1}{12} \mu\text{s}$$

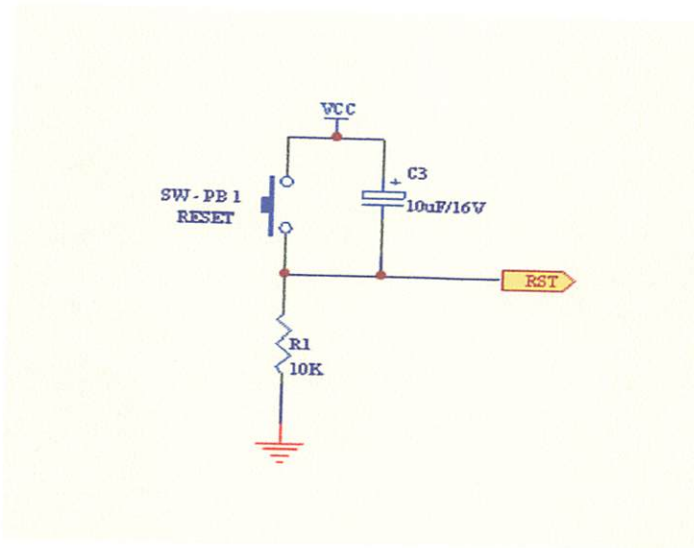
$$\text{Time} = 1 \mu\text{s}$$

Mikrokontroller memiliki oscillator internal (*On Chip Oscillator*) yang dapat digunakan sebagai clock bagi CPU. Untuk menggunakan oscillator internal diperlukan sebuah kristal atau resonator keramic antara pin Xtal 1 dan Xtal 2 dan sebuah kapasitor ke *ground*. Untuk kristalnya dapat digunakan frekuensi dari 6 sampai 12 MHz. Sedangkan kapsitor bernilai antara 27 pF sampai 33 pF.

3.5.2 Rangkaian Reset

Rangkaian reset bertujuan agar mikrokontroller dapat melakukan proses dari awal jika terjadi kerancuan yang diakibatkan oleh banyak hal, seperti kelebihan arus, perubahan tegangan, dan lain lain. Rangkaian reset untuk mikrokontroller AT89S8252 direncanakan agar mempunyai kemampuan power on reset yaitu reset yang terjadi ketika sistem dinyalakan untuk pertama kali. Reset juga dapat dilakukan secara manual dengan menekan tombol reset yang

berupa switch push button. Perencanaan gambar rangkaian reset adalah sebagai berikut :



Gambar 3–6 Rangkaian Reset

Sumber : Data Sheet AT 898252

Rangkaian Reset terbentuk oleh komponen R dan C yang sudah baku (ditetapkan oleh perusahaan pembuat IC AT 89S8252). Nilai R yang dipakai adalah 10 K Ω dan C 10 μ F. Besarnya nilai tahanan dan kapasitor pada rangkaian reset akan menentukan waktu/lama pulsa reset.

Dengan Rumus :

$$\tau = R \times C$$

Agar dapat terjadi secara normal maka nilai τ harus lebih besar dari 30 siklus mesin.

$$\tau \gg 30 \times \text{Time}$$

Karena nilai Time diatas adalah 1 μ s maka :

$$\tau \gg 30 \times 1 \mu\text{s}$$

$$\tau \gg 30 \mu\text{s}$$

Dengan mengambil nilai R dan C sebesar $10\text{ K}\Omega$ dan $10\ \mu\text{F}$ maka besarnya τ dapat dicari sebagai berikut :

$$\tau = R \times C$$

$$\tau = 10\text{ K}\Omega \times 10\ \mu\text{F}$$

$$\tau = 10 \cdot 10^3 \times 10 \cdot 10^{-6}$$

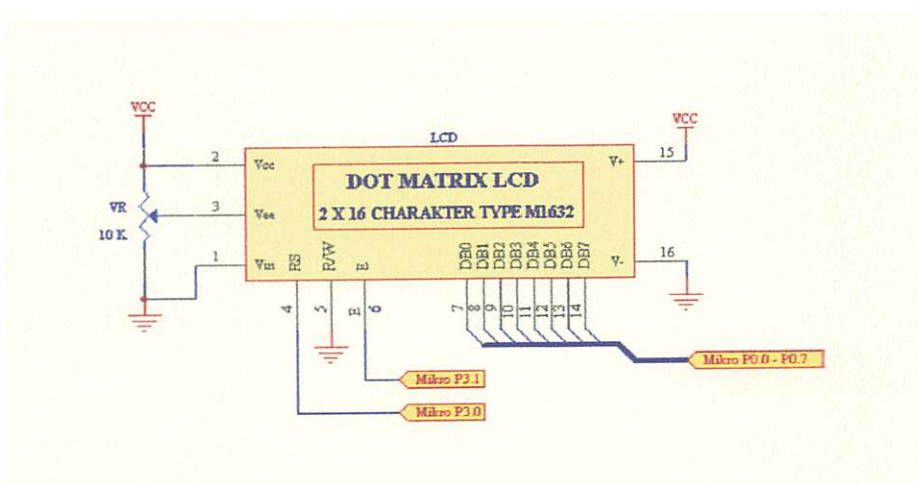
$$\tau = 100 \times 10^{-3}$$

$$\tau = 0,1\ \mu\text{s}$$

Dengan demikian nilai τ jauh lebih besar dibanding nilai minimumnya.

3.6 Perencanaan Rangkaian LCD M1632

Dalam sistem ini direncanakan menggunakan sebuah layar penampil yang berupa *Liquid Cristal Display* (LCD) dot matrik 16×2 , yaitu mempunyai 2 baris tampilan dan masing – masing terdiri atas 16 karakter tiap barisnya. Penampil kristal cair tipe TM 1632 ini dilengkapi pula dengan *back light* berwarna hijau. Rangkaian penampil kristal cair ditunjukkan pada gambar di bawah ini :



Gambar 3–7 Rangkaian Display LCD 16×2

Sumber : Ricky 2004

Peraga LCD mempunyai 14 penyemat (Pin) yang fungsinya ditunjukkan pada tabel seperti berikut ini :

Tabel 3-1 Fungsi Penyemat (Pin) LCD

Sumber : Data Sheet LCD M1632

Nama Penyemat	Fungsi
<input checked="" type="checkbox"/> DB0-DB7	<input checked="" type="checkbox"/> Merupakan saluran data (data bus), yang berisi perintah dan data yang akan ditampilkan pada LCD.
<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Sinyal operasi awal sinyal ini mengaktifkan baca tulis data.
<input checked="" type="checkbox"/> R/W	<input checked="" type="checkbox"/> Sinyal seleksi tulis dan baca. Tulis = 0. Baca = 1.
<input checked="" type="checkbox"/> RS	<input checked="" type="checkbox"/> Sinyal pemilih register keadaan. Instruksi register tulis = 0. Instruksi register baca = 1.
<input checked="" type="checkbox"/> Vlc	<input checked="" type="checkbox"/> Untuk mengendalikan kecerahan LCD.
<input checked="" type="checkbox"/> Vcc	<input checked="" type="checkbox"/> + 5 Volt, supli tegangan.
<input checked="" type="checkbox"/> Vss	<input checked="" type="checkbox"/> Terminal ground.

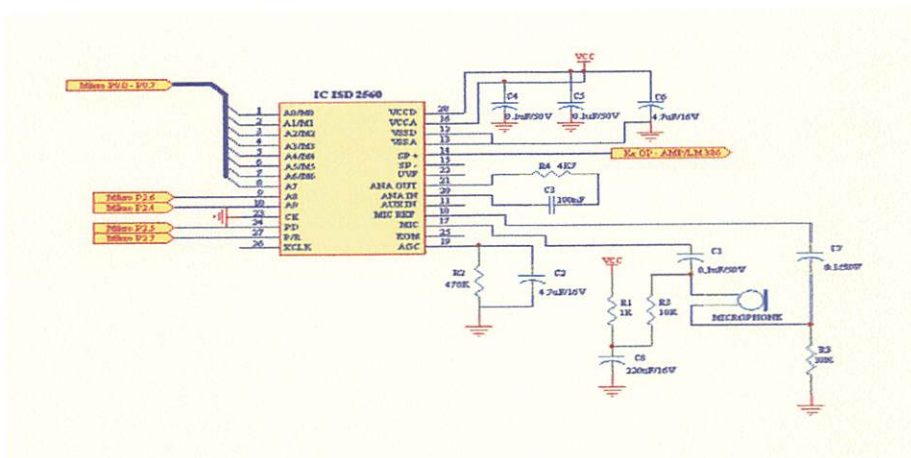
Sinyal-sinyal yang dipergunakan oleh LCD adalah data bus, RS, R/W, dan E. Sinyal E dihubungkan ke Port 3.1 untuk mengaktifkan LCD. LCD akan aktif jika Mikrokontroller memberikan instruksi tulis pada alamat LCD. Sedangkan Port 3.0 dipergunakan untuk memberikan sinyal RS yang membedakan data yang diberikan pada LCD. Sinyal RS diberikan ke LCD untuk membedakan sinyal antara instruksi program atau instruksi penulisan data. Untuk pin R/W akan berlogika low (0) apabila dihubungkan dengan ground maka LCD hanya difungsikan untuk menulis program atau data ke display. Untuk mengambil data

dari Mikrokontroler maka Pin-pin data data dihubungkan dengan Port 0.0 sampai Port 0.7 yang merupakan Pin-pin data dari Mikrokontroler.

Vr1 pada pin 3 (V_{EE}) digunakan untuk mengatur kontras dari karakter yang ditampilkan, sedangkan pada pin 15 ($V+$) dihubungkan ke Vcc dan pin 16 dihubungkan ke ground.

3.7 Perencanaan Rangkaian ISD 2560

Untuk penyampaian informasi dalam tampilan suara, yang digunakan dalam perencanaan alat ini adalah IC ISD 2560. Pin alamat A0 sampai A7 ISD 2560 dihubungkan dengan mikrokontroler P0.0 – P0.7, sedangkan pin A8 dan A9 dihubungkan ke P2.6 dan P2.4. Selanjutnya ISD pin 24 (\overline{PD}) dan pin 27 ($\overline{P/R}$) dihubungkan ke mikrokontroler port P2.5 dan port P2.7 sebagai pin kontrol untuk memutar suara, CE dihubungkan ke Ground. Rangkaian ISD 2560 dengan mikrokontroler AT89S8252 untuk memutar suara ditunjukkan pada gambar berikut :



Gambar 3–8 Rangkaian ISD 2560

Sumber : Ricky 2004

Dalam perencanaan IC penyimpanan sura yang digunakan memakai IC ISD 2560 yang mempunyai dua fungsi yaitu mode messege address dan mode operasional. Dalam perencanaan ISD 2560 ini dioperasikan pada mode message address. Berdasarkan data sheet ISD 2560 mampu merekam sura dengan waktu perekaman selama 60 detik. Untuk pemberitahuan (suara) yang disimpan pada ISD 2560 lebih efisien maka data suara yang sama tidak perlu dituliskan (direkam) lagi, cukup dengan jalan mencuplik tiap segment.

Tabel 3-2 Sinyal Kontrol ISD

Sumber : Data Sheet ISD 2560

Langkah	Fungsi	Cara
1	Power up chip dan memilih mode rekam atau playback	1. PD = Low 2. P/R = Terserah
2	Menset alamat untuk rekam	Menset alamat A0 – A9
3A	Playback mulai	P/R = High CE = Pulsa Low
3B	Record mulai	P/R = Low CE = Low
4A	Playback berhenti	Otomatis
4B	Record berhenti	PD Dan CE = High

Untuk lebih jelas dari aplikasi yang tertulis pada tabel diatas, tentang mode recording dan mode playback dapat diuraikan sebagai berikut :

3.7.1 Mode Recording

1. Pin PD diberi Low, biasanya menggunakan resistor pulldown.
2. Pin P/R diberi Low juga.
3. Pin CE diberi pulsa Low. Recording dimulai (start), pada saat EOM diberi high yang menunjukkan bahwa operasi sedang berlangsung.

4. Pin CE diberi pulsa Low. Recording istirahat (pause), pada saat EOM diubah low, internal address sebagai petunjuk tidak dibersihkan, tapi EOM disimpan dimemori pada point pesan yang selesai. Pin P/R diberi high pada waktu itu, berikutnya CE akan mulai (start). Playback pada address (alamat) 0.
5. Pin CE diberi pulsa low. Recording dimulai (start) pada alamat berikut, kemudian sebelum meng-set EOM kembali high.
6. Ketika rangkaian recording (perekaman) selesai, akhirnya CE berpulsa low maka akan selesai putaran record (rekam).

3.7.2 Mode Playback

1. Pin PD diberi low.
2. Pin P/R diberi High.
3. Pin CE diberi pulsa low, playback mulai (start) EOM menjadi high menunjukkan bahwa operasi sedang berlangsung.
4. Jika CE diberi pulsa low atau EOM tetap selama operasi, maka bagian akan pause. Address internal sebagai petunjuk tidak dibersihkan dan EOM kembali keadaan low. Pin P/R dapat ditukar pada saat itu, berikut operasi record (rekam) tidak akan reset, address petunjuk dan recording memulai dimana playback diselesaikan.
5. Pin CE diberi pulsa low lagi. Playback mulai (start), EOM keadaan high menunjukkan operasi sedang berlangsung.
6. Sambungan playback dilangkah 4 dan 5 hingga PD berpulsa high atau overflow.

7. Jika overflow, CE = low akan men-reset address pointer dan playback start dari permulaan, kemudian PD berpulsa maka bagian itu men-reset address
0. Pada rangkaian ISD 2560 ini menggunakan tegangan sebesar 5 Vdc (Vcc). Vcc dihubungkan ke V_{CCA} , V_{CCD} , V_{SSA} dan V_{SSD} serta komponen yang lain seperti pada gambar. V_{CCA} dan V_{CCD} merupakan tegangan input ground menjadikan power supply ground berimpedansi rendah.

Untuk XCLK pada ISD 2560 mempunyai frekuensi sample rate 8.0 KHz (berdasarkan data sheet), dengan demikian dapat dihitung waktu yang diperoleh dengan rumus :

$$F = 1/T$$

$$\text{Maka : } T = 1/8.0$$

$$= 0,125 \text{ mS}$$

3.8 Perencanaan Rangkaian LM 386

Penguat audio dapat menggunakan transistor, operasional amplifier ataupun IC audio. Pada sistem ini di gunakan suatu komponen IC LM 386 yang berfungsi untuk memperkuat suara lemah yang berasal dari rangkaian ISD untuk diumpankan ke loudspeaker yang berguna untuk memanggil penghuni rumah. Pemilihan penguat audio didasarkan pada minimalisasi level tegangan Vcc yang dibutuhkan, serta penguatan suara agar saat memutar kembali rekaman dapat terdengar. Perencanaan dari rangkaian penguat audio yang menggunakan IC LM 386 dapat dilihat seperti pada gambar berikut :

3.10 Perencanaan Perangkat Lunak

Setelah semua perangkat keras telah selesai dikerjakan, pada tahap selanjutnya adalah pembuatan perangkat lunak (*software*) yang akan mendukung kerja dari sistem, maka dipergunakan bahasa assembler MCS-51. Perangkat lunak disini adalah perintah (program) didalam memori yang harus dilaksanakan oleh mikrokontroller dan juga menentukan bagaimana cara sistem rangkaian ini akan bekerja. Pada bagian inilah semua tata kerja rangkaian ditentukan.

Memori merupakan fasilitas utama dalam suatu mikrokontroller karena didalam memori ini disimpan perintah-perintah yang harus dikerjakan. Memori disini dapat dibedakan menurut fungsinya yaitu memori program dan memori data. Memori program adalah tempat menyimpan program, dan program ini hanya bisa dibaca namun tidak dapat diganti atau ditambah, sedangkan memori data adalah tempat menyimpan data dan data ini dapat dibaca, ditambah atau diganti.

Didalam merancang *software* haruslah dipertimbangkan dari *hardware* yang ada. Pada pemanfaatan *hardware* juga harus dipertimbangkan penggunaan komponen dengan kecepatan proses eksekusi yang dilakukan program.

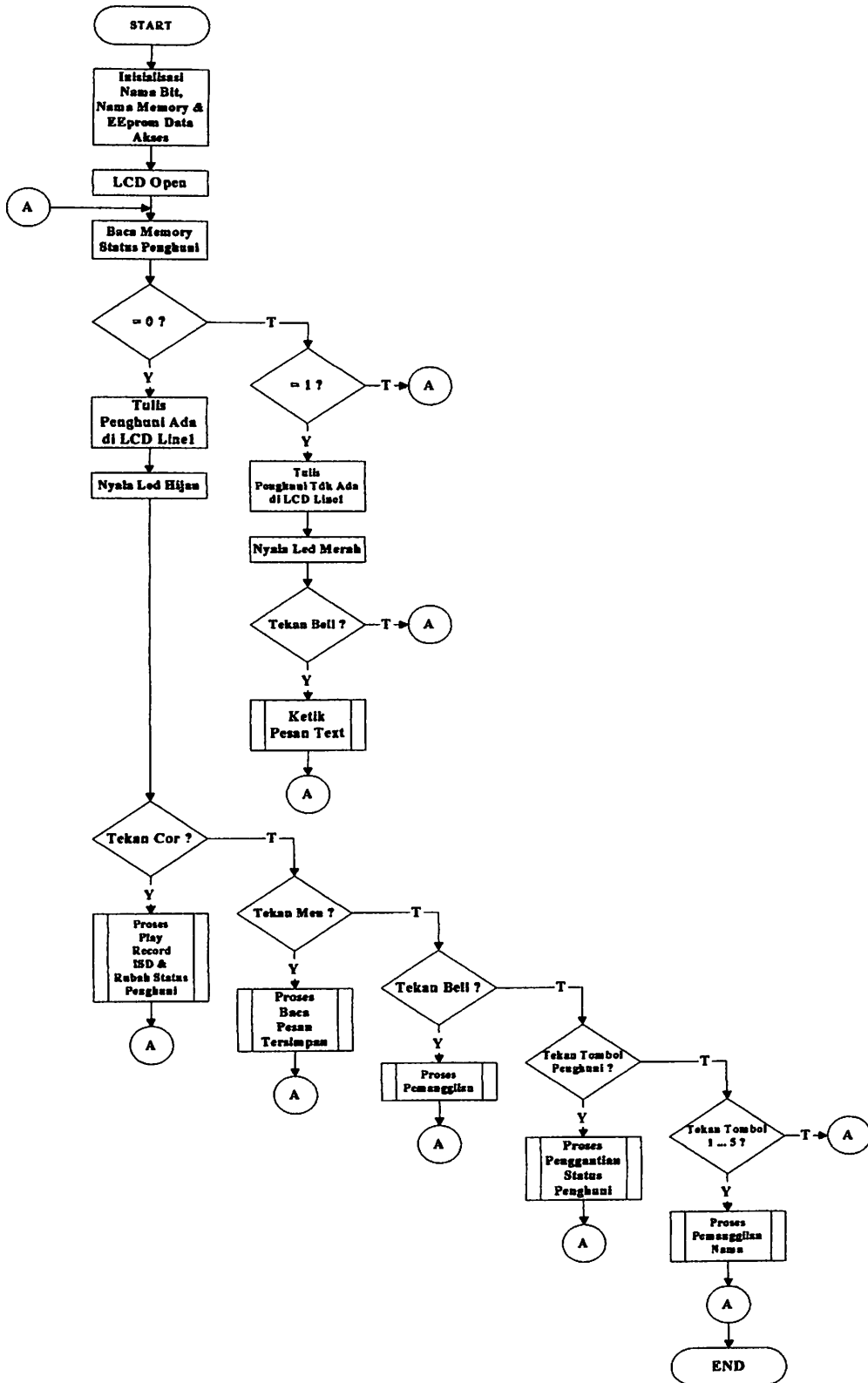
Sebelum kita membuat program, untuk mempermudah pekerjaan, kita membuat diagram alur dari program yang akan dibuat. Sehingga dengan begitu kita akan mudah menentukan urutan kerja dari program. Diagram alur menunjukkan proses-proses yang dijalankan alat Penerima Tamu Elektronik. Adapun proses-prosesnya adalah proses inisialisasi, proses pemutaran dan perekaman suara, proses pembacaan pesan, proses pemanggilan dan proses penulisan pesan. Adapun cara membuat *software* dalam pembuatan alat ini menggunakan *flowchart*-nya adalah sebagai berikut :

3.10.1 Proses Inisialisasi

Proses inisialisasi adalah proses awal dalam membuat suatu program. Proses inisialisasi terdiri dari proses pengecekan port, LCD dan RAM pada mikrokontroler, kemudian LCD akan menampilkan menu atau kata pengantar. Start merupakan proses awal, dimana program telah dijalankan.

Proses kedua adalah proses pengecekan penghuni rumah, jika 0 berarti penghuni ada, maka LCD menampilkan menu atau kata pengantar penghuni ada dan LED hijau menyala. Jika 1 berarti penghuni tidak ada sehingga LCD menampilkan menu atau kata pengantar penghuni tidak ada dan LED merah menyala.

Tombol COR pada keypad digunakan sebagai tombol untuk melakukan proses perekaman, pemutaran suara dan perubahan status penghuni. Proses pembacaan pesan dilakukan dengan menekan Tombol MENU. Sedangkan Bel dan Keypad digunakan sebagai proses pemanggilan dan pengetikan. Gambar dari proses inisialisasi ditunjukkan pada Flowcart 3 -1 berikut :



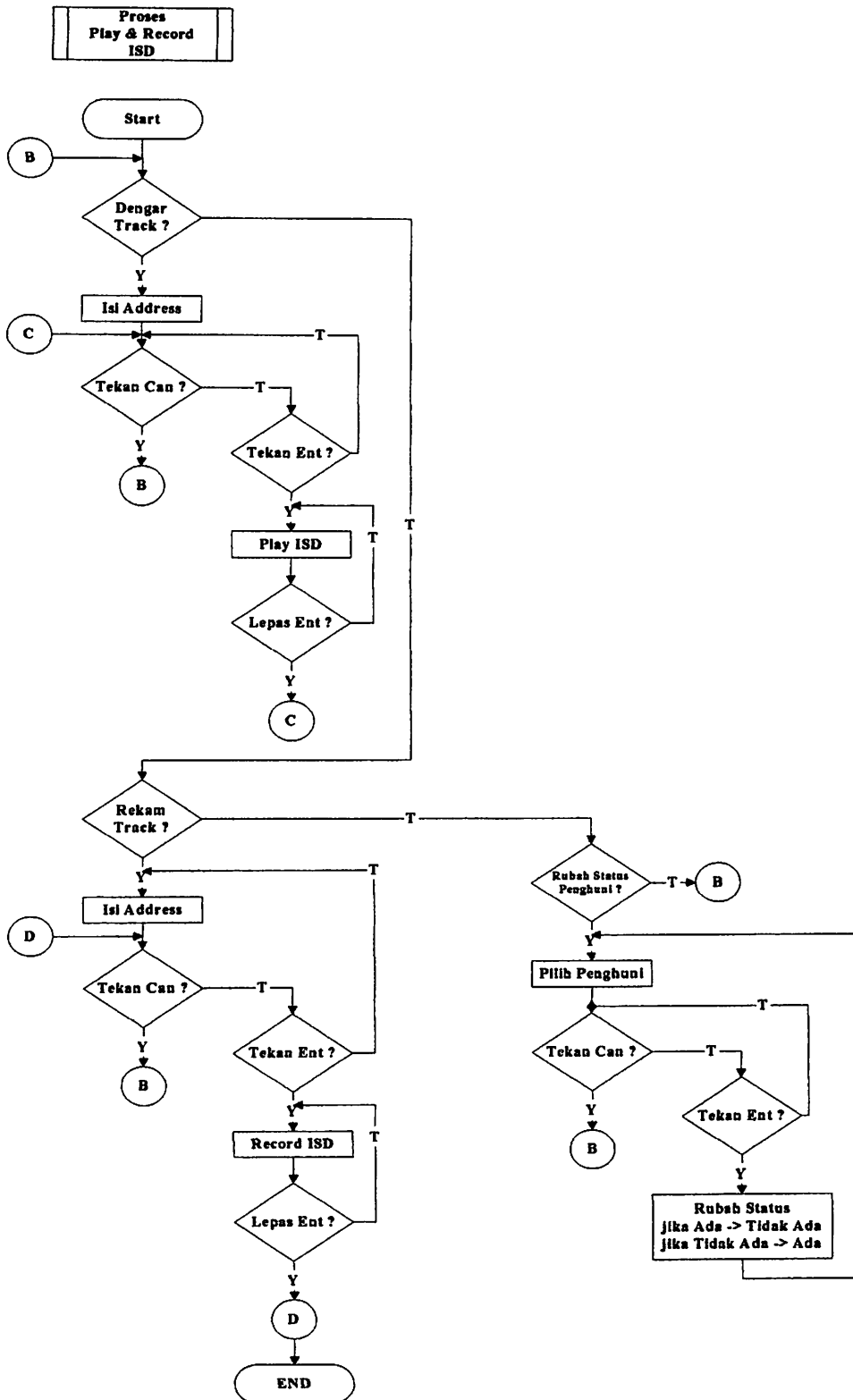
Flowchart 3-1 Proses Inisialisasi

Sumber : Perencanaan.

3.10.2 Proses Pemutaran Dan Perekaman Suara Serta Perubahan Status

Proses pengisian suara dilakukan pada saat proses perekaman dan pemutaran suara. Proses ini memerlukan alamat yang sesuai. Awal proses perekaman suara dan proses pemutaran adalah dengan menekan tombol COR kemudian isi pada alamat yang dikehendaki dan tekan tombol ENT untuk menjalankan instruksi.

Proses perubahan status dapat dilakukan dengan menekan tombol COR kemudian pilih statusnya. Gambar dari proses pemutaran dan perekaman suara diperlihatkan pada Flowchart 3 - 2 berikut :

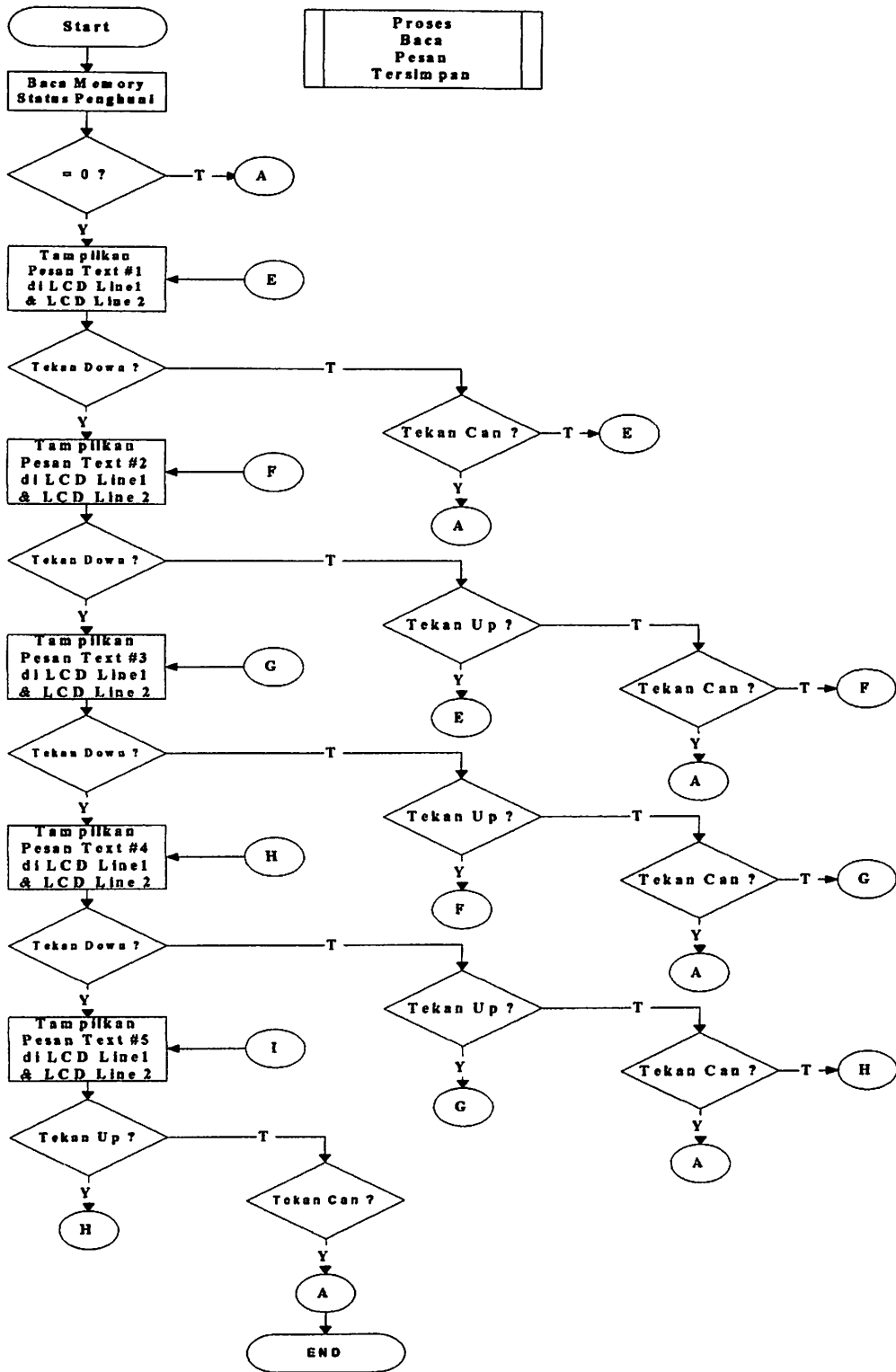


Flowchart 3-2 Proses Play Dan Record ISD Serta Perubahan Status

Sumber : Perencanaan.

3.10.3 Proses Pembacaan Pesan

Proses pembacaan pesan merupakan proses yang sangat penting dikarenakan melalui proses ini penghuni memperoleh informasi. Proses pembacaan pesan diawali dengan pembacaan memori status penghuni, jika memori status = 0 maka pesan 1 akan tampil pada LCD. Tekan tombol panah bawah untuk menampilkan pesan selanjutnya untuk kembali ke pesan awal dengan menekan tombol panah atas. Untuk kembali ke MENU awal dengan menekan tombol CAN. Gambar dari proses pembacaan pesan ditunjukkan pada Flowchart 3 - 3.berikut :



Flowchart 3-3 Proses Baca Pesan Tersimpan

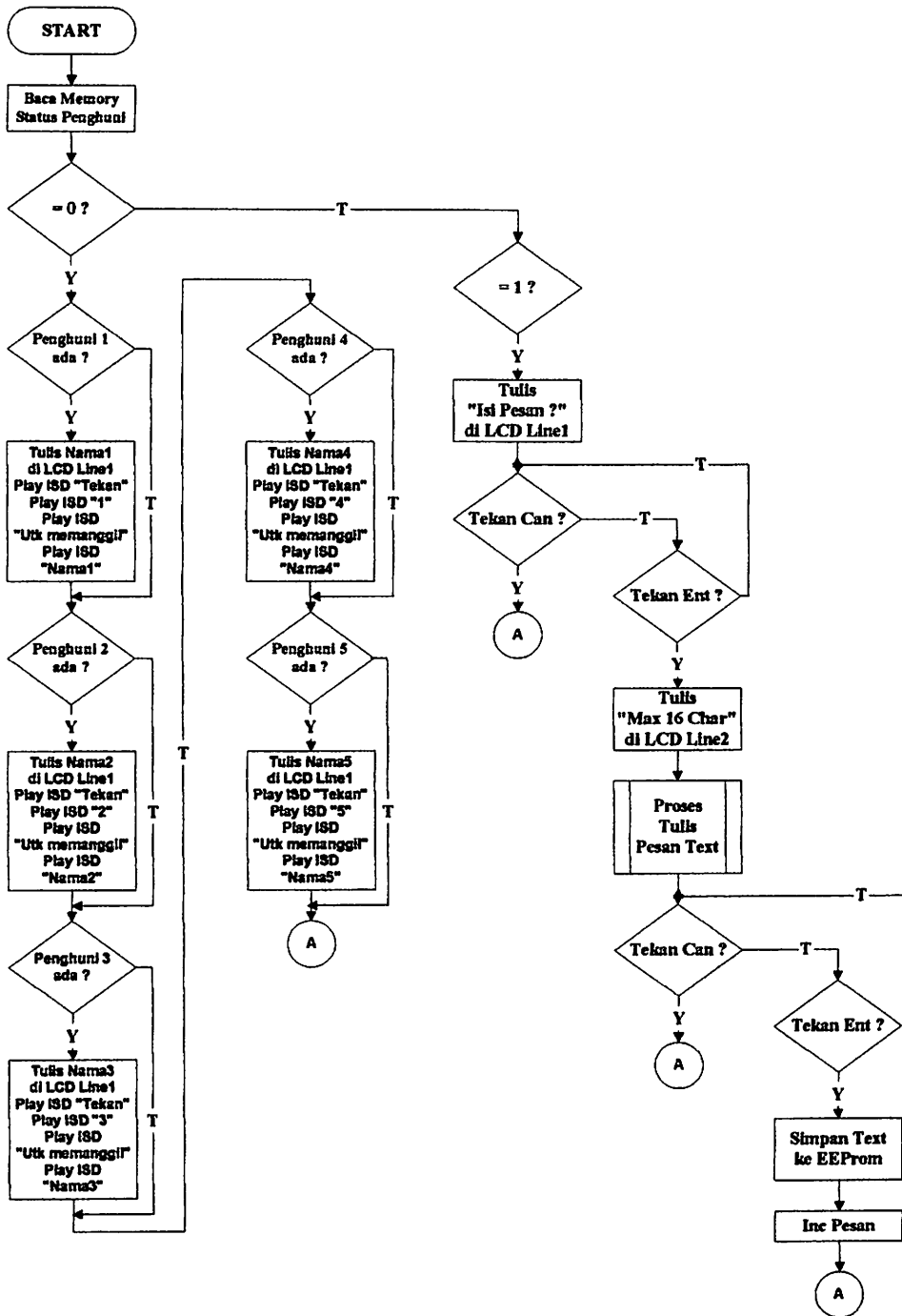
Sumber : Perencanaan.

3.10.4 Proses Pemanggilan Dan Penulisan Pesan

Proses awal pemanggilan adalah melakukan pembacaan memori status penghuni, jika memori status penghuni = 0 maka pada LCD tertulis nama penghuni yang ada di rumah sehingga ISD memutar nama anggota keluarga yang ada. Apabila ingin memanggil nama 1 maka tekan tombol 1 pada keypad dan seterusnya.

Penulisan pesan dilakukan bila nilai memori status = 1 kemudian, tekan ENT untuk menulis pesan maksimal. Penulisan pesan ini maksimal 5 pesan dengan panjang pesan yang dapat ditulis maksimal 16 karakter. Untuk menyimpan pesan dilakukan dengan menekan tombol ENT. Gambar dari proses pemanggilan dan penulisan pesan ditunjukkan pada Flowchart 3 - 4.berikut :

Proses Pemanggilan atau Ketik Pesan Text



Flowchart 3-4 Proses Pemanggilan Penghuni Dan Ketik Pesan Text

Sumber : Perencanaan.

BAB IV

PENGUJIAN ALAT

4.1 Pendahuluan

Setelah perangkat keras dan perangkat lunak selesai dibuat, selanjutnya adalah dilakukan pengujian terhadap alat yang dibuat. Tujuan pengujian adalah untuk mengetahui keadaan masukan atau keadaan keluaran dari tiap-tiap blok rangkaian yang direncanakan. Sehingga untuk mengetahui apakah alat yang telah dirancang dapat bekerja sesuai dengan yang diharapkan, maka perlu pengujian dan analisis sistem kerja alat, sehingga dengan pengujian ini dapat diketahui apakah alat yang direncanakan dapat berfungsi dengan baik dan sesuai dengan yang diharapkan atau tidak. Untuk tujuan ini, pengujian dilakukan dengan urutan rangkaian sebagai berikut :

- Pengujian rangkaian Keypad Matrik 4 x 4.
- Pengujian rangkaian Minimum AT 89S8252.
- Pengujian rangkaian LCD.
- Pengujian rangkaian ISD 2560.

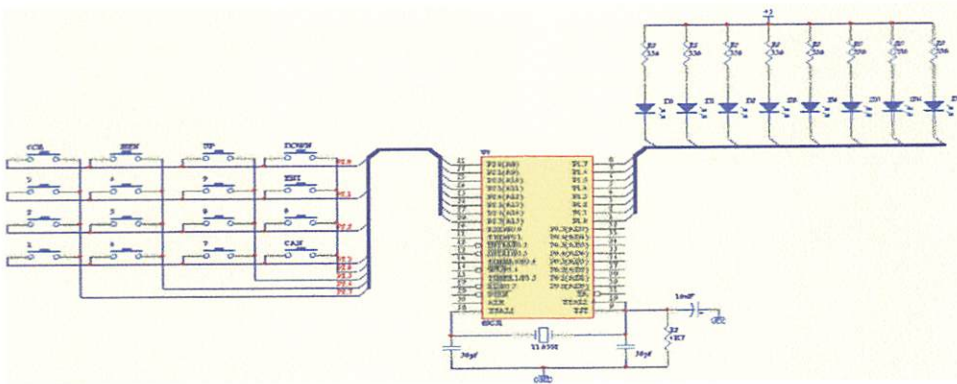
4.2 Pengujian Rangkaian Matrik Keypad 4 x 4

Pengujian ini bertujuan untuk mengetahui konfigurasi logika keluaran dari unit papan tombol saat tombol ditekan. Keypad yang digunakan adalah jenis matrik 4 x 4. Dalam pengujian ini keluaran yang diamati adalah proses scanning yang terjadi pada lajur baris dan kolom. Untuk mengetahui kebenaran rangkaian

keypad yang telah dibuat maka keluaran dari rangkaian keypad ini akan ditampilkan ke port 1 MCU AT89S8252.

4.2.1 Prosedur Pengujian

1. Jalankan program pengujian keypad. Menyiapkan catu daya dengan keluaran 5V.
2. Menyusun rangkaian pengujian papan tombol seperti pada gambar 4-1, serta memastikan bahwa hubungan antar pin pada masing-masing saklar telah benar. Langkah ini dilakukan dalam keadaan tidak ada penekanan tombol.
3. Tekan sembarang tombol kemudian amati hasil penekanan tombol yang telah dilakukan.
4. Hasil dari pengujian rangkaian papan tombol terdapat pada tabel 4-1.



Gambar 4-1 Pengujian rangkaian keypad dan LED pada MCU AT 89S8252

Sumber : Hasil pengujian

4.2.2 Hasil Pengujian

Dari hasil pengujian keypad terlihat pada tabel (4-2), pada saat tombol “1” ditekan, maka yang terhubung adalah baris 1 (B1) dan kolom 1 (K1). Dari tabel

(4-2) sudah jelas. Nilai “0” pada tabel baris dan kolom menunjukkan bahwa antara baris dan kolom tersebut terhubung sesuai dengan penekanan tombol keypad.

Dalam pengujian keypad selain tombol angka yang diuji, juga tombol yang lain seperti MEN, COR, CAN, ENT, PANAH ATAS atau BAWAH yang ada pada keypad kemudian membandingkan hasilnya dengan program yang digunakan untuk mengontrol keypad tersebut. Hasil dari pengujian tombol fungsi keypad ditunjukkan pada tabel di bawah.

Dari data hasil pengujian keypad dapat diketahui bahwa saat tombol ditekan, maka keluaran port 1 MCU AT 89S8252 akan berlogika sesuai dengan tombol yang ditekan. Dari tabel (4-1) hasil pengujian terlihat bahwa rangkaian tombol yang telah direalisasikan sesuai dengan unjuk kerja perencanaan. Data hasil pengujian papan tombol keypad ditunjukkan pada tabel di bawah yaitu sebagai berikut :

Tabel 4-1 Hasil pengujian fungsi tombol-tombol pada keypad

Sumber : Hasil pengujian

Nama Tombol	Fungsi
MEN	Untuk membaca pesan masuk.
COR	Untuk merubah status penghuni, untuk merekam atau mendengar suara.
CAN	Untuk kembali ke proses awal.
ENT	Perintah untuk melakukan aktifitas atau enter.
Panah Atas dan Panah Bawah	Sebagai tombol pemilih yang sesuai dengan apa yang diinginkan.

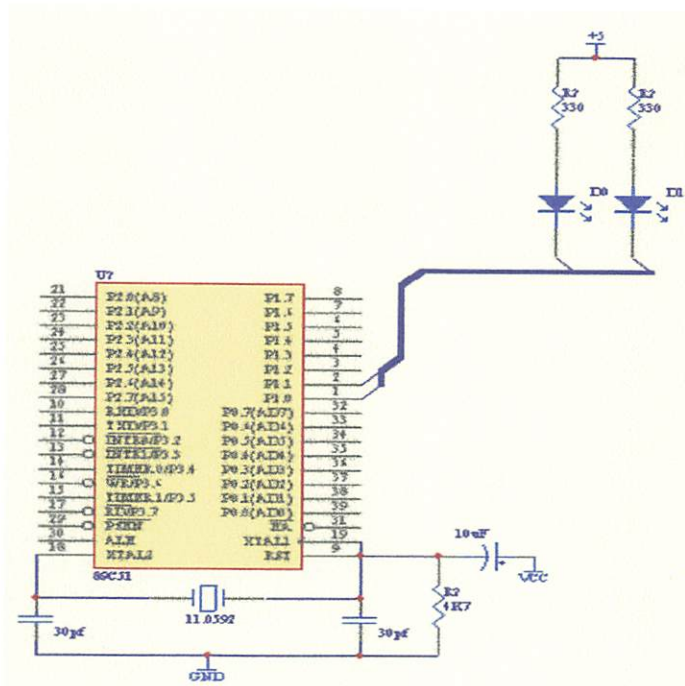
Tabel 4-2 Hasil pengujian papan tombol matrik keypad 4 x 4.

Sumber : Hasil pengujian

Tombol	Baris				Kolom				HEKSA
	B1	B2	B3	B4	K1	K2	K3	K4	
1	0	1	1	1	0	1	1	1	77H
2	0	1	1	1	1	0	1	1	7BH
3	0	1	1	1	1	1	0	1	7DH
COR	0	1	1	1	1	1	1	0	7EH
4	1	0	1	1	0	1	1	1	B7H
5	1	0	1	1	1	0	1	1	BBH
6	1	0	1	1	1	1	0	1	BDH
MEN	1	0	1	1	1	1	1	0	BEH
7	1	1	0	1	0	1	1	1	D7H
8	1	1	0	1	1	0	1	1	DBH
9	1	1	0	1	1	1	0	1	DDH
UP	1	1	0	1	1	1	1	0	DEH
CAN	1	1	1	0	0	1	1	1	E7H
0	1	1	1	0	1	0	1	1	EBH
ENT	1	1	1	0	1	1	0	1	EDH
DOWN	1	1	1	0	0	1	1	0	EEH

4.3 Pengujian Rangkaian Minimum Sistem AT 89S8252

Tujuan pengujian pada gambar 4-2 adalah untuk mengetahui apakah rangkaian minimum dapat bekerja sesuai dengan program yang dimasukkan. Rangkaian minimum AT 89S8252 dihubungkan ke LED melalui port 1.0 dan port 1.1 Adapun rangkaiannya ditunjukkan pada gambar 4-2 berikut :



Gambar 4-2 Pengujian rangkaian MCU AT 89S8252

Sumber : Hasil pengujian

4.3.1 Hasil Pengujian

Dari pengujian rangkaian minimum sistem diatas menunjukkan led pada port 1.0 dan port 1.1 dapat menyala secara bergantian.

Tabel 4-3 Hasil pengujian minimum sistem

Sumber : Hasil pengujian

Data Pada Port 1	D1	D0
1111110B	Mati	Hidup
1111101B	Hidup	Mati

4.4 Pengujian Rangkaian LCD (Liquid Crystal Display)

4.4.1 Tujuan

Dalam pengujian rangkaian LCD pada gambar 4-3 memiliki tujuan yaitu untuk mengetahui apakah LCD dapat bekerja dengan baik dalam menampilkan huruf dan angka pada layar LCD.

Sebelum pengujian dilakukan dan mendapatkan hasil yang diinginkan maka tahapan pengujian untuk LCD adalah sebagai berikut :

4.4.2 Peralatan Yang Dibutuhkan

1. Power Supply 5 Volt.
2. Sistem Mikrokontroller.
3. LCD M1632.

4.4.3 Prosedur Pengujian

1. Menyusun rangkaian seperti dalam gambar 4-3.
2. Menjalankan program ke LCD .
2. Mengamati tampilan pada LCD.



Gambar 4-4 Foto pengujian LCD

Sumber : Hasil pengujian

4.5 Pengujian Rangkaian ISD 2560

4.5.1 Tujuan

Untuk mengetahui kesesuaian data biner yang nantinya dari Mikrokontroller, dalam pengujian IC ISD data biner diwakili dengan saklar deep switch yang dirangkai dengan resistor pack sebagai pull up. Pengalamatan oleh saklar disesuaikan dengan alamat suara yang telah direkam dalam IC ISD 2560.

4.5.2 Peralatan Yang Digunakan

Peralatan yang di butuhkan untuk pengujian ISD 2560 antara lain :

1. Modul pengisian IC ISD 2560
2. IC ISD 2560
3. Speaker
4. Condencor mic
5. Saklar deep switch
6. Saklar push button

Fungsi dari masing-masing peralatan yang digunakan dalam pengujian IC ISD 2560 adalah sebagai berikut :

1. Modul Pengisian IC ISD 2560

Dalam pengisian dan pengalamatan suara yang direkam dalam IC ISD 2560 modul yang digunakan adalah kit modul pengisian yang dirancang sesuai dengan data sheet konfigurasi pin IC ISD 2560.

2. IC ISD 2560

IC ISD (Information Storage Devices) 2560 berfungsi sebagai penampung suara yang akan direkam, lama durasi waktu rekam yang mampu di simpan dalam IC ISD tergantung dari tipe, untuk ISD 2560 durasinya 60 detik.

3. Speaker

Berfungsi untuk mengubah gelombang listrik menjadi gelombang audio atau suara.

4. Condensor Mic

Berfungsi sebagai penerima inputan suara yang akan di rekam ke dalam IC ISD 2560.

5. Saklar Deep Switch

Saklar-saklar ini difungsikan sebagai pengganti inputan pengalamatan dari mikrokontroller.

6. Saklar Push Button

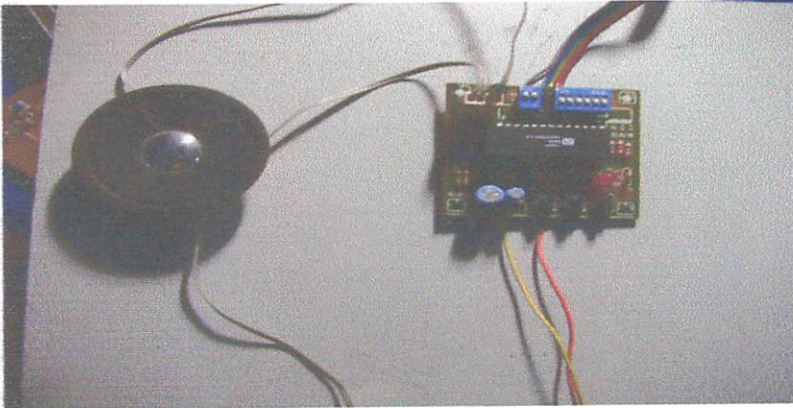
Saklar push button dipasangkan pada pin-pin Play dan pin Rec dari IC ISD.

4.5.3 Langkah Pengujian IC ISD

Langkah-langkah pengujian IC ISD 2560 adalah sebagai berikut :

- a. Menempatkan posisi pengalamatan deep switch pada posisi 0000 0000 H, yaitu saklar pada posisi low semua, terhubung pada ground.
- b. Pengisian suara pada alamat pertama 0000 0000 H dengan menekan push button Rec selama perekaman suara.
- c. Mencari alamat akhir dari pengisian suara dengan mengubah poisisi deep switch sampai pada suara akhir perekaman.
- d. Mengulang langkah b,c,d, sesuai dengan data suara yang hendak di rekam pada IC ISD.

Dibawah ini adalah gambar foto rangkaian modul pengisian IC ISD 2560 :



Gambar 4-5 Foto pengujian rangkaian modul pengisian IC ISD 2560

Sumber : Hasil pengujian

4.5.4 Hasil Pengujian

Setelah melakukan pengujian pengisian IC ISD 2560 diperoleh data hasil perekaman sebagai berikut :

Tabel 4-5 Hasil pengujian pengalamatan rekaman suara pada ISD 2560

Sumber : Hasil pengujian

Alamat Awal	Heksa	Rec	Play Back
1000 0000	80 H	Wahyu	Wahyu
1001 0110	96 H	Riris	Riris
1011 0010	B2 H	Icha	Icha
1100 1000	C8 H	Fa'us	Fa'us
1110 0100	E4 H	Andi	Andi
1010 0000	A0 H	Ada Tamu	Ada Tamu

Hasil pengujian ISD dalam pengisian suara, suara yang dihasilkan atau dikeluarkan akan sama persis dengan suara yang kita rekam sebelumnya dengan isi satu kalimat pada alamat pertama atau satu kali rekaman.

4.6 Pengujian Rangkaian Keseluruhan

Tujuan dari pengujian sistem secara keseluruhan adalah untuk mengetahui cara kerja piranti setelah perangkat keras dan perangkat lunak diintegrasikan bersama-sama. Pengujian sistem secara keseluruhan diuji atas beberapa tahap berdasarkan tampilan menu dan mengikuti diagram alir atau flowchart dalam perencanaan pada BAB III.

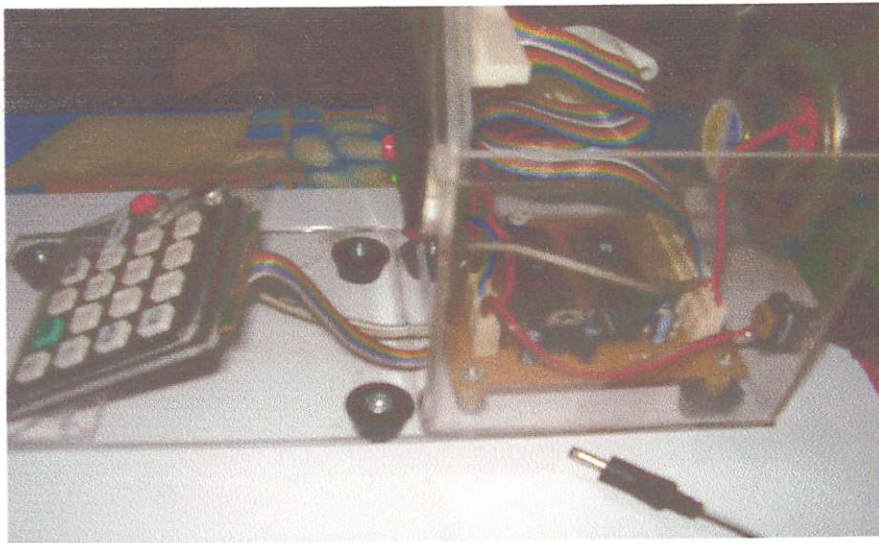
- Memasang catu daya yang dibutuhkan yaitu sebesar 12 V.

- Maka akan langsung keluar menu di LCD Dengan tulisan Penghuni ada. Bila salah satu atau beberapa penghuni rumah ada.
- Bila tombol bel tamu ditekan maka buzzer berbunyi dan LCD akan menampilkan menu status penghuni rumah dan no pemanggil penghuni yang telah di set terlebih dahulu.
- Jika nomer pemanggil untuk tamu yang ada pada di keypad ditekan dan status penghuni rumah ada maka ISD akan bekerja untuk memanggil nama yang sesuai dengan yang tampil di LCD.
- Penghuni yang dicari tamu akan keluar untuk menemui tamu yang datang.
- Jika menu pada LCD tertulis penghuni tidak ada dan lampu indikator led merah menyala maka, jika bel tamu ditekan, di LCD akan muncul tulisan silahkan tinggalkan pesan.
- Jika tamu ingin tinggalkan pesan tekan ENT pada keypad maka tamu bisa mengetik pesan melalui keypad yang ada.

Dengan Pengujian rangkaian keseluruhan telah di lakukan maka dapat disimpulkan bahwa alat sudah dapat bekerja sesuai dengan perencanaan. Gambar dari alat keseluruhan adalah sebagai berikut :



Gambar 4-6 Foto alat keseluruhan tampak depan



Gambar 4-7 Foto alat keseluruhan tampak samping

3. Dalam media penyampain pesan yang sebelumnya memakai keypad dapat dirubah dengan memakai ISD agar memudahkan tamu bila meninggalkan pesan kepada salah satu penghuni rumah.

BAB V

PENUTUP

5.1 Kesimpulan

1. Cara merubah alat tamu manual ke elektronik yaitu dengan merubah sistem yang sebelumnya manual menjadi elektronik dengan menambahkan berbagai komponen elektronika yang sebelumnya tidak ada menjadi ada seperti, mikrokontroller, keypad, LCD, ISD.
2. Dalam pembuatan alat penerima tamu elektronik ini yang dipakai komponen utamanya adalah Mikrokontroller jenis AT 89S8252. Karena tipe IC ini sudah terdapat memori EEPROM 2 KB sehingga tidak menambah lagi memori *eksternal* yang bilamana diperlukan.
3. Dalam pembuatan alat penerima tamu elektronik ini pembuatan perangkat lunak sudah sesuai dengan apa yang direncanakan sehingga alat yang dibuat juga akan berfungsi sesuai yang diinginkan.

5.2 Saran

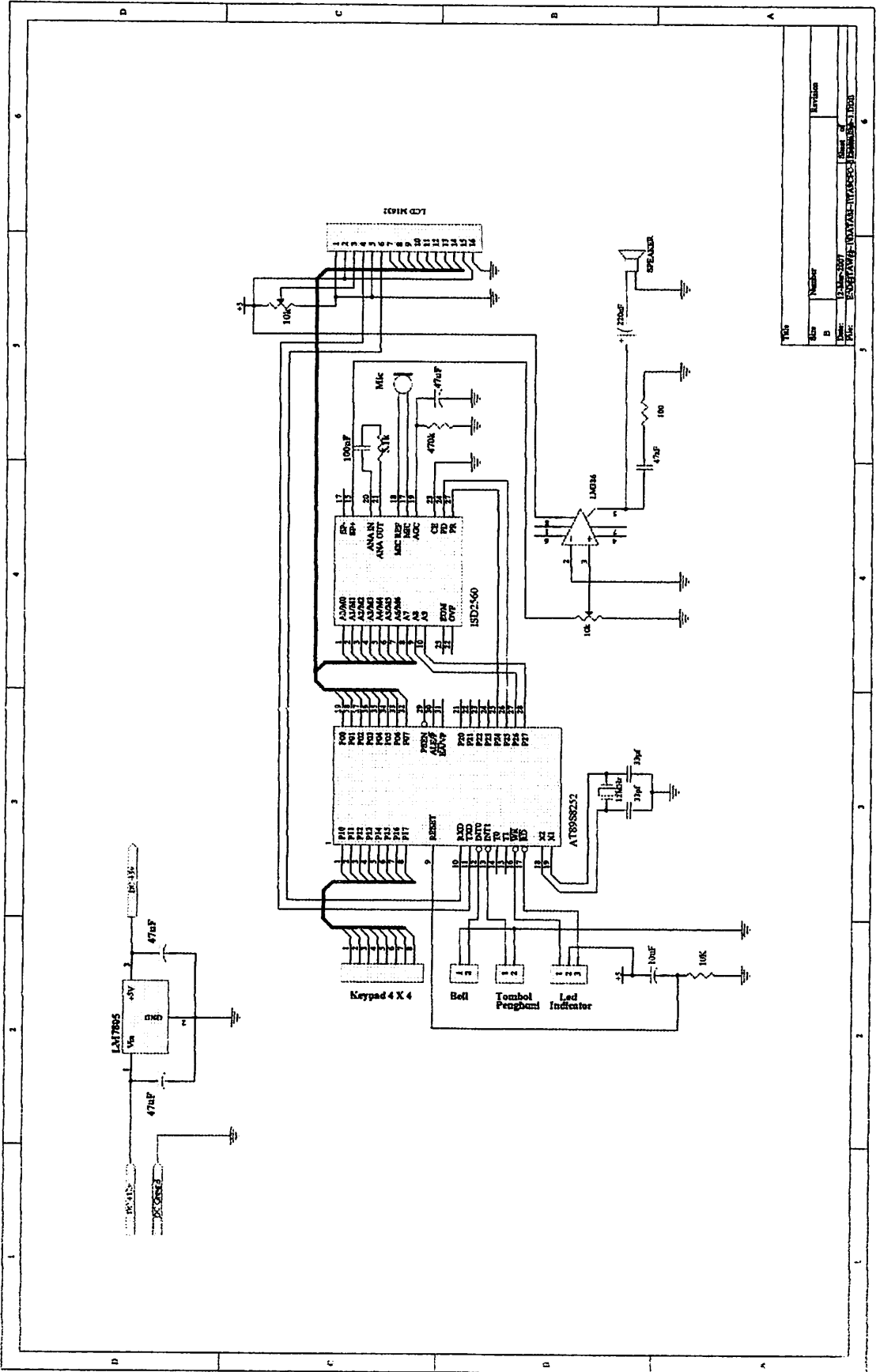
1. Melakukan penambahan durasi waktu, karena pada alat penerima tamu elektronik ini hanya mampu merekam nama panggilan 5 orang saja, bisa dikembangkan lagi lebih dari 5 orang.
2. Alat masih bersifat umum dalam penyampaian pesan, sehingga perlu adanya penambahan program agar pesan dapat diterima hanya untuk orang yang bersangkutan.

DAFTAR PUSTAKA

- [1]. Data Sheet Mikrokontroller AT89S8252. *Atmel Corp.*
- [2]. Ibnu Malik, Moh, (2003), Belajar Mikrokontroller ATMEL AT 89S8252, penerbit Gava Media, Yogyakarta.
- [3]. Data Sheet ISD 2560.
- [4]. Data Sheet Liquid Crystal Display M1632.
- [5]. Data Sheet LM 386.
- [6]. *Design Protel 99 se.*
- [7]. ATMEL Data Book, 1999.
- [8]. *National Telekomunikasi Databook, USA, National semikonduktor Coporation.*
- [9]. MCS 51 Microcontroller *Family User's Manual* : I-18.
- [10]. Malvino Albert Paul, (1990), Prinsip-prinsip Elektronika, Erlangga, Jakarta.
- [11]. Ricky 2004, Peneliti Sebelumnya.

LAMPIRAN - LAMPIRAN

- *SOFTWARE ALAT*
- *DATA SHEET KOMPONEN*



No.		Revisi	
No.	Number		
B	13.04.2007	Date of	
Doc	BUKAWI-001A05-ITANSC-EMISS-1001	Revision	
Doc			



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-III

LEMBAR ASISTENSI BIMBINGAN TUGAS AKHIR

Nama : Wahyu Setiyawan
Nim : 03.57.033
Waktu Bimbingan : 29/11/06 s/d 29/03/07
Judul : Perencanaan Dan Pembuatan Alat Penerima Tamu
Elektronik Berbasis Mikrokontroler AT89S8252.

No	Tanggal	Materi	Paraf
①	12/12 '06	Acc Bab I BAB II tambahkan Teori OP-AMP	A
②	18/12 '06	ACC BAB II	A
③	4/2 '07	Revisi BAB III	A
④	23/2 '07	Acc BAB III tambahkan Data sumber Pustaka	A
⑤	28/2 '07	BAB IV Perbaiki + Pustaka?	A
6	2/3 '07	Pustaka perbaiki	A
7	7/3 '07	Acc BAB I-IV, kesimpulan perbaiki	A

P. 10/3 '07
uji Alat Acc
Ace Meju

Malang, 20
Mengetahui
Dosen Pembimbing


(Bambang Prio Hartono, ST, MT)



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-III
PROGRAM STUDI TEKNIK ELEKTRONIKA
MALANG

LEMBAR PERBAIKAN TUGAS AKHIR

Dari hasil ujian komprehensif Jenjang Diploma III (D-III) Jurusan Teknik Elektro Program Studi Elektronika yang diselenggarakan pada :

Hari : Rabu
Tanggal : 21 Maret 2007

Telah dilakukan perbaikan Tugas Akhir oleh :

Nama : Wahyu Setiyawan
Nim : 03.57.033
Jurusan : Teknik Elektro D-III
Program Studi : Teknik Elektronika
Judul : Perencanaan dan Pembuatan Alat
Penerima Tamu Elektronik Berbasis
Mikrokontroler AT 89S8252.

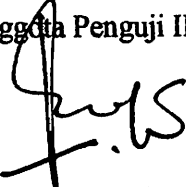
No	Materi Perbaikan	Keterangan
1	Perbaikan Gambar Rangkaian	es

Telah Diperiksa/Disetujui :

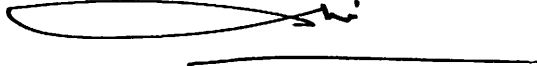
Anggota Penguji I


(Ir. H. Choirul Saleh, MT)

Anggota Penguji II


(Ir. Eko Nurcahyo)

Dosen Pembimbing



(Bambang Prio H, ST, MT)



**INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-III
PROGRAM STUDI TEKNIK ELEKTRONIKA
MALANG**

**BERITA ACARA UJIAN TUGAS AKHIR
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Wahyu Setiyawan
Nim : 03.57.033
Jurusan : Teknik Elektro D - III
Program Studi : Teknik Elektronika
Judul Tugas Akhir : Perencanaan dan Pembuatan Alat
Penerima Tamu Elektronik Berbasis
Mikrokontroler AT 89S8252.

Dipertahankan dihadapan team penguji majelis Tugas Akhir Jenjang Diploma III
(D-III) pada :

Hari : Rabu
Tanggal : 21 Maret 2007
Dengan Nilai : 84,50 (A) ⁶



Panitia Ujian Tugas Akhir

(Ir.Mochtar Asroni, MSME)
Ketua Majelis Penguji

(Ir. H Choirul Saleh, MT)
Sekretaris Majelis Penguji

Anggota Penguji

(Ir. Choirul Shaleh, MT)
Pertama

(H. Eko Nurcahyo)
Kedua

org 0h

Hibt Bit P2.4
IStr Bit P2.5
Lobt Bit P2.6
ISPR Bit P2.7
Rest Bit P3.0
Enbl Bit P3.1
Spng Bit P3.2
Bell Bit P3.3
Led1 Bit P3.6
Led2 Bit P3.7
Wmcn Data 96h
Eemn Equ 00001000b
Eemw Equ 00010000b
Wtdg Equ 00000010b
Dt10 Equ 30h
Dt11 Equ 31h
Dt12 Equ 32h
Dt13 Equ 33h
Dt14 Equ 34h
Dt15 Equ 35h
Dt16 Equ 36h
Dt17 Equ 37h
Dt18 Equ 38h
Dt19 Equ 39h
Dt1A Equ 3Ah
Dt1B Equ 3Bh
Dt1C Equ 3Ch
Dt1D Equ 3Dh
Dt1E Equ 3Eh
Dt1F Equ 3Fh
Dt20 Equ 40h
Dt21 Equ 41h
Dt22 Equ 42h
Dt23 Equ 43h
Dt24 Equ 44h
Dt25 Equ 45h
Dt26 Equ 46h
Dt27 Equ 47h
Dt28 Equ 48h
Dt29 Equ 49h
Dt2A Equ 4Ah
Dt2B Equ 4Bh
Dt2C Equ 4Ch
Dt2D Equ 4Dh
Dt2E Equ 4Eh
Dt2F Equ 4Fh

```

Kpd0 Equ 50h
Kpd1 Equ 51h
Kpd2 Equ 52h
Kpd3 Equ 53h
Kpd4 Equ 54h
Kpd5 Equ 55h
Kpd6 Equ 56h
Kpd7 Equ 57h
Kpd8 Equ 58h
Kpd9 Equ 59h
Stps Equ 60h           ; register status pesan
Stpg Equ 61h           ; register status penghuni
Stp1 Equ 62h           ; register status penghuni 1
Stp2 Equ 63h           ; register status penghuni 2
Stp3 Equ 64h           ; register status penghuni 3
Stp4 Equ 65h           ; register status penghuni 4
Stp5 Equ 66h           ; register status penghuni 5
Stch Equ 67h           ; register status character
Cntr Equ 68h           ; register counter timeout ganti character
Dad0 Equ 69h           ; register data address #0 input keypad
Dad1 Equ 6Ah           ; register data address #1 input keypad
Dad2 Equ 6Bh           ; register data address #2 input keypad
Daf0 Equ 6Ch           ; register data buffer 0
Daf1 Equ 6Dh           ; register data buffer 1
Dalo Equ 6Eh           ; register data address lo
Dahi Equ 6Fh           ; register data address hi
Bicr Equ 70h           ; register lama bicara
Hurf Equ 71h           ; register huruf
Dly0 Equ 72h           ; register delay 0
Dly1 Equ 73h           ; register delay 1
Dly2 Equ 74h           ; register delay 2
;
init: lcall delays
      lcall lcd_op
      setb IStr
      setb ISPR
;
mulai: lcall rdstpg
       mov  A,Stpg
       cjne A,#0,cekpng
       mov  DPTR,#pgnada
       setb Led1
       clr  Led2
cekpng: cjne A,#1,tlspng
       mov  DPTR,#pgtdad
       clr  Led1
       setb Led2
tlspng: lcall line1

```

```

    mov  huruf,#16
    lcall tulis
    mov  DPTR,#kosong
    lcall line2
    mov  huruf,#16
    lcall tulis
;
scn0: lcall scnkpd
      cjne R1,#13,scn1
      lcall tg_lps
      mov  R5,#1
      ljmp isiisd
scn1: cjne R1,#14,scn2
      lcall tg_lps
      mov  R4,#1
      ljmp bcps0
scn2: jb  Bell,scn3
      lcall adabel
      ljmp mulai
scn3: jb  Spng,scn4
      lcall stspng
      ljmp mulai
scn4: lcall tknpgl
      ljmp scn0
;
bcps0: lcall rdstpg
       mov  A,Stpg
       cjne A,#0,bcpsx
       ljmp bcpsy
bcpsx: ljmp mulai
bcpsy: lcall rdstps
       mov  DPTR,#chrctr
       lcall line1
       mov  A,#10
       lcall wr_chr
       mov  A,Stps
       lcall wr_chr
       mov  A,#10
       lcall wr_chr
       mov  DPTR,#psnbaru
       mov  Hurf,#13
       lcall tulis
       mov  DPTR,#bcpesan
       mov  Hurf,#16
       lcall line2
       lcall tulis
       lcall tg_lps
;

```

```
bcps1: lcall scnkpd
       cjne R1,#11,bcps2
       ljmp mulai
bcps2: cjne R1,#12,bcps1
;
bcps3: cjne R4,#1,bcps4
       mov DPTR,#00
bcps4: cjne R4,#2,bcps5
       mov DPTR,#32
bcps5: cjne R4,#3,bcps6
       mov DPTR,#64
bcps6: cjne R4,#4,bcps7
       mov DPTR,#96
bcps7: cjne R4,#5,bcps8
       mov DPTR,#128
bcps8: lcall rdmem
       lcall tmpls
       lcall tg_lps
;
bcps9: lcall scnkpd
       cjne R1,#15,bcps11
       dec R4
       cjne R4,#0,bcps10
       mov R4,#1
bcps10: ljmp bcps3
bcps11: cjne R1,#16,bcps13
       inc R4
       cjne R4,#6,bcps12
       mov R4,#5
bcps12: ljmp bcps3
bcps13: cjne R1,#11,bcps9
       ljmp mulai
```

```

;
stspng: lcall rdstpg
        mov  A,Stpg
        cjne A,#0,stspg1
        jnb  Spng,$
        mov  Stps,#0
        lcall wrstps
        mov  Stp1,#1          ;\
        lcall wrspg1         ;|
        mov  Stp2,#1         ;|
        lcall wrspg2         ;|
        mov  Stp3,#1         ;| rubah status penghuni = tidak ada
        lcall wrspg3         ;|
        mov  Stp4,#1         ;|
        lcall wrspg4         ;|
        mov  Stp5,#1         ;|
        lcall wrspg5         ;/
        mov  Stpg,#1
        lcall wrstpg
        clr  Led1
        setb Led2
        ljmp stspg3
stspg1: cjne A,#1,stspg3
        jnb  Spng,$
        mov  Stpg,#0
        lcall wrstpg
        setb Led1
        clr  Led2
        mov  R6,#1
        lcall rbhsts
        lcall rdstps
        mov  A,Stps
        cjne A,#0,stspg2
        ljmp stspg3
stspg2: mov  R4,#1
        ljmp bcps0
stspg3: ret
;
tknpg1: lcall rdstpg
        mov  A,Stpg
        cjne A,#0,tknpgx
        sjmp tknpg0
tknpgx: ljmp tknpg5
tknpg0: lcall rdspg1
        mov  A,Stp1
        cjne A,#0,tknpg1
        lcall scnkp1
        cjne R1,#1,tknpg1

```

```

mov   DPTR,#sysmg1
lcall line1
mov   huruf,#16
lcall tulis
mov   DPTR,#pglnm1
lcall line2
mov   huruf,#16
lcall tulis
lcall nmpng1
lcall adatamu
ljmp  mulai
tknpg1: lcall rdspg2
mov   A,Stp2
cjne  A,#0,tknpg2
lcall scnkp
cjne  R1,#2,tknpg2
mov   DPTR,#sysmg1
lcall line1
mov   huruf,#16
lcall tulis
mov   DPTR,#pglnm2
lcall line2
mov   huruf,#16
lcall tulis
lcall nmpng2
lcall adatamu
ljmp  mulai
tknpg2: lcall rdspg3
mov   A,Stp3
cjne  A,#0,tknpg3
lcall scnkp
cjne  R1,#3,tknpg3
mov   DPTR,#sysmg1
lcall line1
mov   huruf,#16
lcall tulis
mov   DPTR,#pglnm3
lcall line2
mov   huruf,#16
lcall tulis
lcall nmpng3
lcall adatamu
ljmp  mulai
tknpg3: lcall rdspg4
mov   A,Stp4
cjne  A,#0,tknpg4
lcall scnkp
cjne  R1,#4,tknpg4

```

```

    mov  DPTR,#sysmg1
    lcall line1
    mov  huruf,#16
    lcall tulis
    mov  DPTR,#pglnm4
    lcall line2
    mov  huruf,#16
    lcall tulis
    lcall nmpng4
    lcall adatamu
    ljmp  mulai
tknpg4: lcall rdspg5
    mov  A,Stp5
    cjne A,#0,tknpg5
    lcall scnkp4
    cjne R1,#5,tknpg5
    mov  DPTR,#sysmg1
    lcall line1
    mov  huruf,#16
    lcall tulis
    mov  DPTR,#pglnm5
    lcall line2
    mov  huruf,#16
    lcall tulis
    lcall nmpng5
    lcall adatamu
    ljmp  mulai
tknpg5: ret
;
adabel: mov  DPTR,#belbny
    lcall line2
    mov  huruf,#16
    lcall tulis
    jnb  Bell,$
;
    mov  A,Stpg
pggpng: cjne A,#0,isilmp
    ljmp  pggpn0
isilmp: ljmp  isipsn
pggpn0: lcall rdspg1
    mov  A,Stp1
    cjne A,#0,pggpn1
    mov  DPTR,#tkntbl
    lcall line1
    mov  huruf,#7
    lcall tulis
    mov  DPTR,#chrctr
    mov  A,#1

```

```

lcall wr_chr
mov A,#10
lcall wr_chr
mov DPTR,#pnggl
mov hurf,#7
lcall tulis
mov DPTR,#pglnm1
lcall line2
mov hurf,#16
lcall tulis
lcall delayi
pggp1: lcall rdspg2
mov A,Stp2
cjne A,#0,pggp2
mov DPTR,#tkntbl
lcall line1
mov hurf,#7
lcall tulis
mov DPTR,#chrctr
mov A,#2
lcall wr_chr
mov A,#10
lcall wr_chr
mov DPTR,#pnggl
mov hurf,#7
lcall tulis
mov DPTR,#pglnm2
lcall line2
mov hurf,#16
lcall tulis
lcall delayi
pggp2: lcall rdspg3
mov A,Stp3
cjne A,#0,pggp3
mov DPTR,#tkntbl
lcall line1
mov hurf,#7
lcall tulis
mov DPTR,#chrctr
mov A,#3
lcall wr_chr
mov A,#10
lcall wr_chr
mov DPTR,#pnggl
mov hurf,#7
lcall tulis
mov DPTR,#pglnm3
lcall line2

```



```

    mov    huruf,#16
    lcall  tulis
    lcall  delayi
pggp3: lcall  rdspg4
    mov    A,Stp4
    cjne  A,#0,pggp4
    mov    DPTR,#tkntbl
    lcall  line1
    mov    huruf,#7
    lcall  tulis
    mov    DPTR,#chrctr
    mov    A,#4
    lcall  wr_chr
    mov    A,#10
    lcall  wr_chr
    mov    DPTR,#pnggl
    mov    huruf,#7
    lcall  tulis
    mov    DPTR,#pglm4
    lcall  line2
    mov    huruf,#16
    lcall  tulis
    lcall  delayi
pggp4: lcall  rdspg5
    mov    A,Stp5
    cjne  A,#0,pggp5
    mov    DPTR,#tkntbl
    lcall  line1
    mov    huruf,#7
    lcall  tulis
    mov    DPTR,#chrctr
    mov    A,#5
    lcall  wr_chr
    mov    A,#10
    lcall  wr_chr
    mov    DPTR,#pnggl
    mov    huruf,#7
    lcall  tulis
    mov    DPTR,#pglm5
    lcall  line2
    mov    huruf,#16
    lcall  tulis
    lcall  delayi
pggp5: ret
;
isipsn: cjne  A,#1,isitxt
    mov    DPTR,#psntxt
    lcall  line1

```

```

    mov  huruf,#16
    lcall tulis
    mov  DPTR,#kosong
    lcall line2
    mov  huruf,#16
    lcall tulis
isips0: lcall scnkpdl
    cjne R1,#11,isips1
    ljmp mulai
isips1: cjne R1,#12,isips0
    mov  DPTR,#maxchr
    lcall line2
    mov  huruf,#16
    lcall tulis
    lcall delayl
;
isitxt: mov  DPTR,#kosong
    lcall line1
    mov  huruf,#16
    lcall tulis
    mov  DPTR,#kosong
    lcall line2
    mov  huruf,#16
    lcall tulis
    lcall kpdrst
    lcall dtarst
    mov  Stch,#0
    lcall cursor
    lcall tlschr
;
chr01: lcall scnkpdl
    cjne R1,#1,chr02
    inc  Kpd1
    mov  A,Kpd1
    cjne A,#8,rsch1
    mov  Kpd1,#1
rsch1: mov  Cntr,#75
    lcall tlsch1
;
chr02: lcall scnkpdl
    cjne R1,#2,chr03
    inc  Kpd2
    mov  A,Kpd2
    cjne A,#5,rsch2
    mov  Kpd2,#1
rsch2: mov  Cntr,#75
    lcall tlsch2
;

```

```

chr03: lcall scnkp
       cjne R1,#3,chr04
       inc Kpd3
       mov A,Kpd3
       cjne A,#5,rsch3
       mov Kpd3,#1
rsch3: mov Cntr,#75
       lcall tlsch3
;
chr04: lcall scnkp
       cjne R1,#4,chr05
       inc Kpd4
       mov A,Kpd4
       cjne A,#5,rsch4
       mov Kpd4,#1
rsch4: mov Cntr,#75
       lcall tlsch4
;
chr05: lcall scnkp
       cjne R1,#5,chr06
       inc Kpd5
       mov A,Kpd5
       cjne A,#5,rsch5
       mov Kpd5,#1
rsch5: mov Cntr,#75
       lcall tlsch5
;
chr06: lcall scnkp
       cjne R1,#6,chr07
       inc Kpd6
       mov A,Kpd6
       cjne A,#5,rsch6
       mov Kpd6,#1
rsch6: mov Cntr,#75
       lcall tlsch6
;
chr07: lcall scnkp
       cjne R1,#7,chr08
       inc Kpd7
       mov A,Kpd7
       cjne A,#6,rsch7
       mov Kpd7,#1
rsch7: mov Cntr,#75
       lcall tlsch7
;
chr08: lcall scnkp
       cjne R1,#8,chr09
       inc Kpd8

```

```

    mov  A,Kpd8
    cjne A,#5,rsch8
    mov  Kpd8,#1
rsch8: mov  Cntr,#75
    lcall tisch8
;
chr09: lcall scnkpdp
    cjne R1,#9,chr10
    inc  Kpd9
    mov  A,Kpd9
    cjne A,#6,rsch9
    mov  Kpd9,#1
rsch9: mov  Cntr,#75
    lcall tisch9
;
chr10: lcall scnkpdp
    cjne R1,#0,chr11
    inc  Kpd0
    mov  A,Kpd0
    cjne A,#2,rsch10
    mov  Kpd0,#1
rsch10: mov  Cntr,#75
    lcall tisch0
;
chr11: lcall scnkpdp
    cjne R1,#15,chr12
    mov  R3,#10
    lcall switch
    dec  Stch
    lcall cursor
    lcall tischr
chr12: cjne R1,#16,chr13
    mov  R3,#10
    lcall switch
    inc  Stch
    lcall cursor
    lcall tischr
chr13: cjne R1,#11,chr14
    lcall tg_lps
    ljmp mulai
chr14: cjne R1,#12,chr15
    lcall tg_lps
    lcall rdstps
    inc  Stps
    mov  A,Stps
    cjne A,#6,ckmxps
    mov  Stps,#5
ckmxps: lcall wrstps

```

```

    mov R4,Stps
wrps0: cjne R4,#1,wrps1
    mov DPTR,#00
wrps1: cjne R4,#2,wrps2
    mov DPTR,#32
wrps2: cjne R4,#3,wrps3
    mov DPTR,#64
wrps3: cjne R4,#4,wrps4
    mov DPTR,#96
wrps4: cjne R4,#5,wrps5
    mov DPTR,#128
wrps5: lcall wrmem
    ljmp mulai
;
chr15: lcall jeda
    mov A,Cntr
    cjne A,#0,mkgnch
    ljmp chr01
mkgnch: lcall jeda
    djnz Cntr,timout
    inc Stch
    lcall kpdrst
    lcall cursor
    lcall tlschr
timout: ljmp chr01
;
tlsch1: mov R2,Kpd1
    cjne R2,#1,ctk12
    mov R3,#37
ctk12: cjne R2,#2,ctk13
    mov R3,#38
ctk13: cjne R2,#3,ctk14
    mov R3,#39
ctk14: cjne R2,#4,ctk15
    mov R3,#40
ctk15: cjne R2,#5,ctk16
    mov R3,#41
ctk16: cjne R2,#6,ctk17
    mov R3,#42
ctk17: cjne R2,#7,ctk18
    mov R3,#1
ctk18: lcall tislcd
    ret
;
tlsch2: mov R2,Kpd2
    cjne R2,#1,ctk22
    mov R3,#11
ctk22: cjne R2,#2,ctk23

```

```

    mov R3,#12
cttk23: cjne R2,#3,cttk24
    mov R3,#13
cttk24: cjne R2,#4,cttk25
    mov R3,#2
cttk25: lcall tislcd
    ret
;
tlsch3: mov R2,Kpd3
    cjne R2,#1,cttk32
    mov R3,#14
cttk32: cjne R2,#2,cttk33
    mov R3,#15
cttk33: cjne R2,#3,cttk34
    mov R3,#16
cttk34: cjne R2,#4,cttk35
    mov R3,#3
cttk35: lcall tislcd
    ret
;
tlsch4: mov R2,Kpd4
    cjne R2,#1,cttk42
    mov R3,#17
cttk42: cjne R2,#2,cttk43
    mov R3,#18
cttk43: cjne R2,#3,cttk44
    mov R3,#19
cttk44: cjne R2,#4,cttk45
    mov R3,#4
cttk45: lcall tislcd
    ret
;
tlsch5: mov R2,Kpd5
    cjne R2,#1,cttk52
    mov R3,#20
cttk52: cjne R2,#2,cttk53
    mov R3,#21
cttk53: cjne R2,#3,cttk54
    mov R3,#22
cttk54: cjne R2,#4,cttk55
    mov R3,#5
cttk55: lcall tislcd
    ret
;
tlsch6: mov R2,Kpd6
    cjne R2,#1,cttk62
    mov R3,#23
cttk62: cjne R2,#2,cttk63

```

```

    mov    R3,#24
cttk63: cjne R2,#3,cttk64
    mov    R3,#25
cttk64: cjne R2,#4,cttk65
    mov    R3,#6
cttk65: lcall tislcd
    ret
;
tisch7: mov    R2,Kpd7
    cjne  R2,#1,cttk72
    mov    R3,#26
cttk72: cjne R2,#2,cttk73
    mov    R3,#27
cttk73: cjne R2,#3,cttk74
    mov    R3,#28
cttk74: cjne R2,#4,cttk75
    mov    R3,#29
cttk75: cjne R2,#5,cttk76
    mov    R3,#7
cttk76: lcall tislcd
    ret
;
tisch8: mov    R2,Kpd8
    cjne  R2,#1,cttk82
    mov    R3,#30
cttk82: cjne R2,#2,cttk83
    mov    R3,#31
cttk83: cjne R2,#3,cttk84
    mov    R3,#32
cttk84: cjne R2,#4,cttk85
    mov    R3,#8
cttk85: lcall tislcd
    ret
;
tisch9: mov    R2,Kpd9
    cjne  R2,#1,cttk92
    mov    R3,#33
cttk92: cjne R2,#2,cttk93
    mov    R3,#34
cttk93: cjne R2,#3,cttk94
    mov    R3,#35
cttk94: cjne R2,#4,cttk95
    mov    R3,#36
cttk95: cjne R2,#5,cttk96
    mov    R3,#9
cttk96: lcall tislcd
    ret
;

```

```

tlsch0: mov  R2,Kpd0
        cjne R2,#1,ctk02
        mov  R3,#0
ctk02: lcall tislcd
        ret
;
tislcd: lcall switch
        lcall tlsru
        ret
;
tlsru:  mov  DPTR,#chrctr
        lcall line1
        mov  A,Dt10
        lcall wr_chr
        mov  A,Dt11
        lcall wr_chr
        mov  A,Dt12
        lcall wr_chr
        mov  A,Dt13
        lcall wr_chr
        mov  A,Dt14
        lcall wr_chr
        mov  A,Dt15
        lcall wr_chr
        mov  A,Dt16
        lcall wr_chr
        mov  A,Dt17
        lcall wr_chr
        mov  A,Dt18
        lcall wr_chr
        mov  A,Dt19
        lcall wr_chr
        mov  A,Dt1A
        lcall wr_chr
        mov  A,Dt1B
        lcall wr_chr
        mov  A,Dt1C
        lcall wr_chr
        mov  A,Dt1D
        lcall wr_chr
        mov  A,Dt1E
        lcall wr_chr
        mov  A,Dt1F
        lcall wr_chr
        lcall line2
        mov  A,Dt20
        lcall wr_chr
        mov  A,Dt21

```



```
lcall wr_chr
mov A,Dt22
lcall wr_chr
mov A,Dt23
lcall wr_chr
mov A,Dt24
lcall wr_chr
mov A,Dt25
lcall wr_chr
mov A,Dt26
lcall wr_chr
mov A,Dt27
lcall wr_chr
mov A,Dt28
lcall wr_chr
mov A,Dt29
lcall wr_chr
mov A,Dt2A
lcall wr_chr
mov A,Dt2B
lcall wr_chr
mov A,Dt2C
lcall wr_chr
mov A,Dt2D
lcall wr_chr
mov A,Dt2E
lcall wr_chr
mov A,Dt2F
lcall wr_chr
lcall tg_lps
ret
```

```
;
tmplps: mov DPTR,#chrctr
lcall line1
mov A,Dt10
lcall wr_chr
mov A,Dt11
lcall wr_chr
mov A,Dt12
lcall wr_chr
mov A,Dt13
lcall wr_chr
mov A,Dt14
lcall wr_chr
mov A,Dt15
lcall wr_chr
mov A,Dt16
lcall wr_chr
```

```
mov  A,Dt17
lcall wr_chr
mov  A,Dt18
lcall wr_chr
mov  A,Dt19
lcall wr_chr
mov  A,Dt1A
lcall wr_chr
mov  A,Dt1B
lcall wr_chr
mov  A,Dt1C
lcall wr_chr
mov  A,Dt1D
lcall wr_chr
mov  A,Dt1E
lcall wr_chr
mov  A,Dt1F
lcall wr_chr
lcall line2
mov  A,Dt20
lcall wr_chr
mov  A,Dt21
lcall wr_chr
mov  A,Dt22
lcall wr_chr
mov  A,Dt23
lcall wr_chr
mov  A,Dt24
lcall wr_chr
mov  A,Dt25
lcall wr_chr
mov  A,Dt26
lcall wr_chr
mov  A,Dt27
lcall wr_chr
mov  A,Dt28
lcall wr_chr
mov  A,Dt29
lcall wr_chr
mov  A,Dt2A
lcall wr_chr
mov  A,Dt2B
lcall wr_chr
mov  A,Dt2C
lcall wr_chr
mov  A,Dt2D
lcall wr_chr
mov  A,Dt2E
```

```

    lcall wr_chr
    mov  A,Dt2F
    lcall wr_chr
    ret
;
isiisd: mov  A,Stpg
        cjne A,#0,isisdx
        ljmp  isisd0
isisdx: ljmp  mulai
isisd0: cjne R5,#1,isisd1
        mov  DPTR,#dsuara
        lcall line1
        mov  huruf,#16
        lcall tulis
        mov  DPTR,#kosong
        lcall line2
        mov  huruf,#16
        lcall tulis
isisd1: cjne R5,#2,isisd2
        mov  DPTR,#rsuara
        lcall line1
        mov  huruf,#16
        lcall tulis
        mov  DPTR,#kosong
        lcall line2
        mov  huruf,#16
        lcall tulis
isisd2: cjne R5,#3,isisd3
        mov  DPTR,#lhtsts
        lcall line1
        mov  huruf,#16
        lcall tulis
        mov  DPTR,#kosong
        lcall line2
        mov  huruf,#16
        lcall tulis
isisd3: lcall tg_lps
isisd4: lcall scnkpd
        cjne R1,#15,isisd6
        dec  R5
        cjne R5,#0,isisd5
        mov  R5,#1
isisd5: ljmp  isiisd
isisd6: cjne R1,#16,isisd8
        inc  R5
        cjne R5,#4,isisd7
        mov  R5,#3
isisd7: ljmp  isiisd

```

```

isisd8: cjne R1,#11,isisd9
        ljmp mulai
isisd9: cjne R1,#12,isisd4
;
isisdA: cjne R5,#1,isisdB
        ljmp plyisd
isisdB: cjne R5,#2,isisdC
        ljmp recisd
isisdC: cjne R5,#3,isisdA
        mov R6,#1
        acall rbhsts
        ljmp mulai
;
plyisd: mov DPTR,#adres
        lcall line2
        mov hurf,#11
        lcall tulis
        mov DPTR,#chrctr
        lcall tg_tkn
        mov Dad0,R1
        mov A,R1
        lcall wr_chr
        lcall tg_lps
        lcall tg_tkn
        mov Dad1,R1
        mov A,R1
        lcall wr_chr
        lcall tg_lps
        lcall tg_tkn
        mov Dad2,R1
        mov A,R1
        lcall wr_chr
        lcall tg_lps
;
dloop: mov DPTR,#dsuara
        lcall line1
        mov hurf,#16
        lcall tulis
        mov DPTR,#adres
        lcall line2
        mov hurf,#11
        lcall tulis
        mov DPTR,#chrctr
        mov A,Dad0
        lcall wr_chr
        mov A,Dad1
        lcall wr_chr
        mov A,Dad2

```

```

    lcall wr_chr
    mov  A,#10
    lcall wr_chr
    mov  A,#10
    lcall wr_chr
;
ply0: lcall scnkpd
      cjne R1,#11,ply1
      ljmp mulai
ply1: cjne R1,#12,ply0
      mov  DPTR,#plying
      lcall line1
      mov  huruf,#16
      lcall tulis
      lcall datadd           ; switch data to address (dec to hex)
      mov  A,Dahi
ply2: cjne A,#0,ply3
      clr  Lobt
      clr  Hibt
ply3: cjne A,#1,ply4
      setb Lobt
      clr  Hibt
ply4: cjne A,#2,ply5
      clr  Lobt
      setb Hibt
ply5: mov  P0,Dalo
      lcall jeda
      lcall jeda
      clr  IStr
      lcall tg_lps
      setb IStr
      ljmp dloop
;
recisd: mov  DPTR,#rsuara
        lcall line1
        mov  huruf,#16
        lcall tulis
        mov  DPTR,#adres
        lcall line2
        mov  huruf,#11
        lcall tulis
        mov  DPTR,#chrctr
        lcall tg_tkn
        mov  Dad0,R1
        mov  A,R1
        lcall wr_chr
        lcall tg_lps
        lcall tg_tkn

```

```

    mov  Dad1,R1
    mov  A,R1
    lcall wr_chr
    lcall tg_ips
    lcall tg_tkn
    mov  Dad2,R1
    mov  A,R1
    lcall wr_chr
    lcall tg_ips
;
rloop: mov  DPTR,#rsuara
    lcall line1
    mov  huruf,#16
    lcall tulis
    mov  DPTR,#adres
    lcall line2
    mov  huruf,#11
    lcall tulis
    mov  DPTR,#chrctr
    mov  A,Dad0
    lcall wr_chr
    mov  A,Dad1
    lcall wr_chr
    mov  A,Dad2
    lcall wr_chr
    mov  A,#10
    lcall wr_chr
    mov  A,#10
    lcall wr_chr
;
rec0: lcall scnkpdp
    cjne R1,#11,rec1
    ljmp mulai
rec1: cjne R1,#12,rec0
    mov  DPTR,#recing
    lcall line1
    mov  huruf,#16
    lcall tulis
    lcall datadd           ; switch data to address (dec to hex)
    mov  A,Dahi
rec2: cjne A,#0,rec3
    clr  Lobt
    clr  Hibt
rec3: cjne A,#1,rec4
    setb Lobt
    clr  Hibt
rec4: cjne A,#2,rec5
    clr  Lobt

```

```

    setb  Hibt
rec5: mov  P0,Dalo
    lcall jeda
    lcall jeda
    clr   ISPR
    clr   IStr
    lcall tg_ips
    setb  IStr
    setb  ISPR
    ljmp  rloop
;
datadd: mov  Daf0,Dad0
    mov   A,Dad1
    mov   B,#10
    mul   AB
    mov   B,Dad2
    add   A,B
    mov   Daf1,A
;
    mov   Dalo,#0           ; conter lo addr -> 0
    mov   Dahi,#0          ; conter hi addr -> 0
    mov   A,Daf0           ; cek angka #1 = 0
    cjne  A,#0,datad1      ; tidak -> conter addr
    ljmp  datad3           ; ulang sebanyak 100 kali angka #1
datad1: mov  R7,#100       ; benar -> angka #2 tambah #3
datad2: lcall cntadr
    djnz  R7,datad2
    djnz  Daf0,datad1
;
datad3: mov  A,Daf1        ; cek angka #2 tambah #3 = 0
    cjne  A,#0,datad4      ; tidak -> conter sebanyak jumlah angka tsb
    ljmp  datad5           ; benar -> keluar
datad4: lcall cntadr
    djnz  Daf1,datad4
datad5: ret
;
cntadr: inc  Dalo
    mov   A,Dalo
    cjne  A,#0,cntadd
    inc   Dahi
cntadd: ret
;
rbhsts: cjne R6,#1,rbhst1
    mov   DPTR,#pglnm1
    lcall line1
    mov   huruf,#16
    lcall tulis
    lcall rdspg1

```

```

    mov    A,Stp1
rbst00: cjne  A,#0,rbst01
        mov    DPTR,#sttada
        ljmp   rbst02
rbst01: mov    DPTR,#stttdk
rbst02: lcall  line2
        mov    huruf,#16
        lcall  tulis
;
rbhst1: cjne  R6,#2,rbhst2
        mov    DPTR,#pglnm2
        lcall  line1
        mov    huruf,#16
        lcall  tulis
        lcall  rdspg2
        mov    A,Stp2
rbst10: cjne  A,#0,rbst11
        mov    DPTR,#sttada
        ljmp   rbst12
rbst11: mov    DPTR,#stttdk
rbst12: lcall  line2
        mov    huruf,#16
        lcall  tulis
;
rbhst2: cjne  R6,#3,rbhst3
        mov    DPTR,#pglnm3
        lcall  line1
        mov    huruf,#16
        lcall  tulis
        lcall  rdspg3
        mov    A,Stp3
rbst20: cjne  A,#0,rbst21
        mov    DPTR,#sttada
        ljmp   rbst22
rbst21: mov    DPTR,#stttdk
rbst22: lcall  line2
        mov    huruf,#16
        lcall  tulis
;
rbhst3: cjne  R6,#4,rbhst4
        mov    DPTR,#pglnm4
        lcall  line1
        mov    huruf,#16
        lcall  tulis
        lcall  rdspg4
        mov    A,Stp4
rbst30: cjne  A,#0,rbst31
        mov    DPTR,#sttada

```



```

    ljmp rbst32
rbst31: mov  DPTR,#stttdk
rbst32: lcall line2
        mov  hurf,#16
        lcall tulis
;
rbhst4: cjne R6,#5,rbhst5
        mov  DPTR,#pglnm5
        lcall line1
        mov  hurf,#16
        lcall tulis
        lcall rdspg5
        mov  A,Stp5
rbst40: cjne A,#0,rbst41
        mov  DPTR,#sttada
        ljmp rbst42
rbst41: mov  DPTR,#stttdk
rbst42: lcall line2
        mov  hurf,#16
        lcall tulis
;
rbhst5: lcall tg_lps
rbhst6: lcall scnkpD
        cjne R1,#15,rbhst8
        dec  R6
        cjne R6,#0,rbhst7
        mov  R6,#1
rbhst7: ljmp rbhsts
rbhst8: cjne R1,#16,rbhstA
        inc  R6
        cjne R6,#6,rbhst9
        mov  R6,#5
rbhst9: ljmp rbhsts
rbhstA: cjne R1,#11,rbhstB
        ljmp ckrbpB
rbhstB: cjne R1,#12,rbhst6
;
ckrbp0: cjne R6,#1,ckrbp2
        lcall rdspg1
        mov  A,Stp1
        cjne A,#0,ckrbp1
        mov  Stp1,#1
        lcall wrspg1
        ljmp rbhsts
ckrbp1: mov  Stp1,#0
        lcall wrspg1
        ljmp rbhsts
;

```

```

ckrbp2: cjne R6,#2,ckrbp4
        lcall rdspg2
        mov  A,Stp2
        cjne A,#0,ckrbp3
        mov  Stp2,#1
        lcall wrspg2
        ljmp rbhsts
ckrbp3: mov  Stp2,#0
        lcall wrspg2
        ljmp rbhsts
;
ckrbp4: cjne R6,#3,ckrbp6
        lcall rdspg3
        mov  A,Stp3
        cjne A,#0,ckrbp5
        mov  Stp3,#1
        lcall wrspg3
        ljmp rbhsts
ckrbp5: mov  Stp3,#0
        lcall wrspg3
        ljmp rbhsts
;
ckrbp6: cjne R6,#4,ckrbp8
        lcall rdspg4
        mov  A,Stp4
        cjne A,#0,ckrbp7
        mov  Stp4,#1
        lcall wrspg4
        ljmp rbhsts
ckrbp7: mov  Stp4,#0
        lcall wrspg4
        ljmp rbhsts
;
ckrbp8: cjne R6,#5,ckrbpA
        lcall rdspg5
        mov  A,Stp5
        cjne A,#0,ckrbp9
        mov  Stp5,#1
        lcall wrspg5
        ljmp rbhsts
ckrbp9: mov  Stp5,#0
        lcall wrspg5
        ljmp rbhsts
;
ckrbpA: ljmp ckrbp0
ckrbpB: lcall rdspg1
        lcall rdspg2
        lcall rdspg3

```

```

lcall rdspg4
lcall rdspg5
mov  A,Stp1
cjne A,#1,ckrbpC
mov  A,Stp2
cjne A,#1,ckrbpC
mov  A,Stp3
cjne A,#1,ckrbpC
mov  A,Stp4
cjne A,#1,ckrbpC
mov  A,Stp5
cjne A,#1,ckrbpC
mov  Stps,#0
lcall wrstps
mov  Stpg,#1
lcall wrstpg
ckrbpC: ret
;
nminf1: clr  Lobt
        clr  Hibt
        mov  P0,#0
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nminf2: clr  Lobt
        clr  Hibt
        mov  P0,#16
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nminf3: clr  Lobt
        clr  Hibt
        mov  P0,#32
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr

```

```

    lcall delays
    ret
;
nminf4: clr  Lobt
        clr  Hibt
        mov  P0,#48
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nminf5: clr  Lobt
        clr  Hibt
        mov  P0,#64
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nmpng1: clr  Lobt
        clr  Hibt
        mov  P0,#80
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nmpng2: clr  Lobt
        clr  Hibt
        mov  P0,#96
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nmpng3: clr  Lobt

```

```

    clr  Hibt
    mov  P0,#112
    lcall jeda
    clr  IStr
    mov  Bicr,#12
    lcall talk
    setb IStr
    lcall delays
    ret
;
nmpng4: clr  Lobt
        clr  Hibt
        mov  P0,#128
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
nmpng5: clr  Lobt
        clr  Hibt
        mov  P0,#144
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
satu:  clr  Lobt
        clr  Hibt
        mov  P0,#160
        lcall jeda
        clr  IStr
        mov  Bicr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
dua:   clr  Lobt
        clr  Hibt
        mov  P0,#176
        lcall jeda
        clr  IStr

```

```

    mov  Bcr,#12
    lcall talk
    setb IStr
    lcall delays
    ret
;
tiga: clr  Lobt
      clr  Hibt
      mov  P0,#192
      lcall jeda
      clr  IStr
      mov  Bcr,#12
      lcall talk
      setb IStr
      lcall delays
      ret
;
empat: clr  Lobt
        clr  Hibt
        mov  P0,#208
        lcall jeda
        clr  IStr
        mov  Bcr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
lima:  clr  Lobt
        clr  Hibt
        mov  P0,#224
        lcall jeda
        clr  IStr
        mov  Bcr,#12
        lcall talk
        setb IStr
        lcall delays
        ret
;
tekan: clr  Lobt
        clr  Hibt
        mov  P0,#240
        lcall jeda
        clr  IStr
        mov  Bcr,#12
        lcall talk
        setb IStr
        lcall delays

```

```

    ret
;
adatamu:setb  Lobt
    clr  Hibt
    mov  P0,#0
    lcall jeda
    clr  IStr
    mov  Bicr,#12
    lcall talk
    setb IStr
    lcall delays
    ret
;
mmnggil:setb  Lobt
    clr  Hibt
    mov  P0,#16
    lcall jeda
    clr  IStr
    mov  Bicr,#12
    lcall talk
    setb IStr
    lcall delays
    ret
;
line1: mov  R0,#80h
    lcall w_ins
    ret
;
line2: mov  R0,#0C0h
    lcall w_ins
    ret
;
tulis: clr  A
    movc A,@A+DPTR
    mov  R0,A
    inc  DPTR
    lcall w_chr
    djnz huruf,tulis
    ret
;
wr_chr: movc A,@A+DPTR
    mov  R0,A
    lcall w_chr
    ret
;
W_ins: clr  Enbl
    clr  Rest
    mov  P0,R0

```

```

    setb Enbl
    clr Enbl
    lcall jeda
    ret
;
W_chr: clr Enbl
    setb Rest
    mov P0,R0
    setb Enbl
    clr Enbl
    lcall jeda
    ret
;
Lcd_op: lcall delays
    mov R0,#3Fh ; 3Fh -> 8 bit data, 2Fh -> 4bit data
    lcall w_ins
    lcall w_ins
    mov R0,#0Dh ; 0Eh -> cursor, 0Dh -> Blink
    lcall w_ins
    mov R0,#06h
    lcall w_ins
    mov R0,#01h
    lcall w_ins
    mov R0,#0Ch
    lcall w_ins
    lcall delays
    ret
;
scnkpd: lcall jeda
    mov R1,#10
coll: mov P1,#11111110b
    mov A,P1
c1b1: cjne A,#11101110b,c1b2
    mov R1,#1
c1b2: cjne A,#11011110b,c1b3
    mov R1,#2
c1b3: cjne A,#10111110b,c1b4
    mov R1,#3
c1b4: cjne A,#01111110b,col2
    mov R1,#13
;
col2: mov P1,#11111101b
    mov A,P1
c2b1: cjne A,#11101101b,c2b2
    mov R1,#4
c2b2: cjne A,#11011101b,c2b3
    mov R1,#5
c2b3: cjne A,#10111101b,c2b4

```



```

    mov R1,#6
c2b4: cjne A,#01111101b,col3
    mov R1,#14
;
col3: mov P1,#11111011b
    mov A,P1
c3b1: cjne A,#11101011b,c3b2
    mov R1,#7
c3b2: cjne A,#11011011b,c3b3
    mov R1,#8
c3b3: cjne A,#10111011b,c3b4
    mov R1,#9
c3b4: cjne A,#01111011b,col4
    mov R1,#15
;
col4: mov P1,#11110111b
    mov A,P1
c4b1: cjne A,#11100111b,c4b2
    mov R1,#11
c4b2: cjne A,#11010111b,c4b3
    mov R1,#0
c4b3: cjne A,#10110111b,c4b4
    mov R1,#12
c4b4: cjne A,#01110111b,back
    mov R1,#16
back: ret
;
tg_tkn: lcall jeda
    lcall scnkpd
tg_tk0: cjne R1,#16,tg_tk1
    ljmp tg_tkn
tg_tk1: cjne R1,#15,tg_tk2
    ljmp tg_tkn
tg_tk2: cjne R1,#14,tg_tk3
    ljmp tg_tkn
tg_tk3: cjne R1,#13,tg_tk4
    ljmp tg_tkn
tg_tk4: cjne R1,#12,tg_tk5
    ljmp tg_tkn
tg_tk5: cjne R1,#11,tg_tk6
    lcall tg_lps
    ljmp mulai
tg_tk6: cjne R1,#10,tg_tk7
    ljmp tg_tkn
tg_tk7: ret
;
tg_lps: lcall jeda
    lcall scnkpd

```

```

    cjne R1,#10,tg_lps
    ret
;
wt_wr: mov  A,Wmcn
      anl  A,#Wtdg
      jz   wt_wr
      ret
;
switch: mov  A,Stch
        cjne A,#0,swtc01
        mov  Dt10,R3
swtc01: cjne A,#1,swtc02
        mov  Dt11,R3
swtc02: cjne A,#2,swtc03
        mov  Dt12,R3
swtc03: cjne A,#3,swtc04
        mov  Dt13,R3
swtc04: cjne A,#4,swtc05
        mov  Dt14,R3
swtc05: cjne A,#5,swtc06
        mov  Dt15,R3
swtc06: cjne A,#6,swtc07
        mov  Dt16,R3
swtc07: cjne A,#7,swtc08
        mov  Dt17,R3
swtc08: cjne A,#8,swtc09
        mov  Dt18,R3
swtc09: cjne A,#9,swtc10
        mov  Dt19,R3
swtc10: cjne A,#10,swtc11
        mov  Dt1A,R3
swtc11: cjne A,#11,swtc12
        mov  Dt1B,R3
swtc12: cjne A,#12,swtc13
        mov  Dt1C,R3
swtc13: cjne A,#13,swtc14
        mov  Dt1D,R3
swtc14: cjne A,#14,swtc15
        mov  Dt1E,R3
swtc15: cjne A,#15,swtc16
        mov  Dt1F,R3
swtc16: cjne A,#16,swtc17
        mov  Dt20,R3
swtc17: cjne A,#17,swtc18
        mov  Dt21,R3
swtc18: cjne A,#18,swtc19
        mov  Dt22,R3
swtc19: cjne A,#19,swtc20

```

```
    mov    Dt23,R3
swtc20: cjne A,#20,swtc21
    mov    Dt24,R3
swtc21: cjne A,#21,swtc22
    mov    Dt25,R3
swtc22: cjne A,#22,swtc23
    mov    Dt26,R3
swtc23: cjne A,#23,swtc24
    mov    Dt27,R3
swtc24: cjne A,#24,swtc25
    mov    Dt28,R3
swtc25: cjne A,#25,swtc26
    mov    Dt29,R3
swtc26: cjne A,#26,swtc27
    mov    Dt2A,R3
swtc27: cjne A,#27,swtc28
    mov    Dt2B,R3
swtc28: cjne A,#28,swtc29
    mov    Dt2C,R3
swtc29: cjne A,#29,swtc30
    mov    Dt2D,R3
swtc30: cjne A,#30,swtc31
    mov    Dt2E,R3
swtc31: cjne A,#31,swtc32
    mov    Dt2F,R3
swtc32: ret
;
cursor: mov    A,Stch
    cjne A,#0,crsr01
    mov    Dt10,#43
crsr01: cjne A,#1,crsr02
    mov    Dt11,#43
crsr02: cjne A,#2,crsr03
    mov    Dt12,#43
crsr03: cjne A,#3,crsr04
    mov    Dt13,#43
crsr04: cjne A,#4,crsr05
    mov    Dt14,#43
crsr05: cjne A,#5,crsr06
    mov    Dt15,#43
crsr06: cjne A,#6,crsr07
    mov    Dt16,#43
crsr07: cjne A,#7,crsr08
    mov    Dt17,#43
crsr08: cjne A,#8,crsr09
    mov    Dt18,#43
crsr09: cjne A,#9,crsr10
    mov    Dt19,#43
```

```
csr10: cjne  A,#10,csr11
        mov   Dt1A,#43
csr11: cjne  A,#11,csr12
        mov   Dt1B,#43
csr12: cjne  A,#12,csr13
        mov   Dt1C,#43
csr13: cjne  A,#13,csr14
        mov   Dt1D,#43
csr14: cjne  A,#14,csr15
        mov   Dt1E,#43
csr15: cjne  A,#15,csr16
        mov   Dt1F,#43
csr16: cjne  A,#16,csr17
        mov   Dt20,#43
csr17: cjne  A,#17,csr18
        mov   Dt21,#43
csr18: cjne  A,#18,csr19
        mov   Dt22,#43
csr19: cjne  A,#19,csr20
        mov   Dt23,#43
csr20: cjne  A,#20,csr21
        mov   Dt24,#43
csr21: cjne  A,#21,csr22
        mov   Dt25,#43
csr22: cjne  A,#22,csr23
        mov   Dt26,#43
csr23: cjne  A,#23,csr24
        mov   Dt27,#43
csr24: cjne  A,#24,csr25
        mov   Dt28,#43
csr25: cjne  A,#25,csr26
        mov   Dt29,#43
csr26: cjne  A,#26,csr27
        mov   Dt2A,#43
csr27: cjne  A,#27,csr28
        mov   Dt2B,#43
csr28: cjne  A,#28,csr29
        mov   Dt2C,#43
csr29: cjne  A,#29,csr30
        mov   Dt2D,#43
csr30: cjne  A,#30,csr31
        mov   Dt2E,#43
csr31: cjne  A,#31,csr32
        mov   Dt2F,#43
csr32: ret
;
kpdrst: mov   Kpd0,#0
        mov   Kpd1,#0
```

```
mov Kpd2,#0
mov Kpd3,#0
mov Kpd4,#0
mov Kpd5,#0
mov Kpd6,#0
mov Kpd7,#0
mov Kpd8,#0
mov Kpd9,#0
ret
```

```
;
```

```
dtarst: mov Dt10,#10
mov Dt11,#10
mov Dt12,#10
mov Dt13,#10
mov Dt14,#10
mov Dt15,#10
mov Dt16,#10
mov Dt17,#10
mov Dt18,#10
mov Dt19,#10
mov Dt1A,#10
mov Dt1B,#10
mov Dt1C,#10
mov Dt1D,#10
mov Dt1E,#10
mov Dt1F,#10
mov Dt20,#10
mov Dt21,#10
mov Dt22,#10
mov Dt23,#10
mov Dt24,#10
mov Dt25,#10
mov Dt26,#10
mov Dt27,#10
mov Dt28,#10
mov Dt29,#10
mov Dt2A,#10
mov Dt2B,#10
mov Dt2C,#10
mov Dt2D,#10
mov Dt2E,#10
mov Dt2F,#10
ret
```

```
;
```

```
rdmem: orl Wmcn,#Eemn
movx A,@DPTR
cjne A,#43,rdmm10
mov A,#10
```

```
rdmm10: mov  Dt10,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm11
        mov  A,#10
rdmm11: mov  Dt11,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm12
        mov  A,#10
rdmm12: mov  Dt12,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm13
        mov  A,#10
rdmm13: mov  Dt13,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm14
        mov  A,#10
rdmm14: mov  Dt14,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm15
        mov  A,#10
rdmm15: mov  Dt15,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm16
        mov  A,#10
rdmm16: mov  Dt16,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm17
        mov  A,#10
rdmm17: mov  Dt17,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm18
        mov  A,#10
rdmm18: mov  Dt18,A
        inc  DPTR
        movx A,@DPTR
        cjne A,#43,rdmm19
        mov  A,#10
rdmm19: mov  Dt19,A
        inc  DPTR
        movx A,@DPTR
```

```

    cjne A,#43,rdmm1A
    mov  A,#10
rdmm1A: mov  Dt1A,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm1B
    mov  A,#10
rdmm1B: mov  Dt1B,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm1C
    mov  A,#10
rdmm1C: mov  Dt1C,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm1D
    mov  A,#10
rdmm1D: mov  Dt1D,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm1E
    mov  A,#10
rdmm1E: mov  Dt1E,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm1F
    mov  A,#10
rdmm1F: mov  Dt1F,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm20
    mov  A,#10
rdmm20: mov  Dt20,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm21
    mov  A,#10
rdmm21: mov  Dt21,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm22
    mov  A,#10
rdmm22: mov  Dt22,A
    inc  DPTR
    movx A,@DPTR
    cjne A,#43,rdmm23
    mov  A,#10
rdmm23: mov  Dt23,A

```

```

inc DPTR
movx A,@DPTR
cjne A,#43,rdmm24
mov A,#10
rdmm24: mov Dt24,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm25
mov A,#10
rdmm25: mov Dt25,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm26
mov A,#10
rdmm26: mov Dt26,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm27
mov A,#10
rdmm27: mov Dt27,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm28
mov A,#10
rdmm28: mov Dt28,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm29
mov A,#10
rdmm29: mov Dt29,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm2A
mov A,#10
rdmm2A: mov Dt2A,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm2B
mov A,#10
rdmm2B: mov Dt2B,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm2C
mov A,#10
rdmm2C: mov Dt2C,A
inc DPTR
movx A,@DPTR
cjne A,#43,rdmm2D

```



```

    mov    A,#10
rdmm2D: mov    Dt2D,A
    inc   DPTR
    movx  A,@DPTR
    cjne  A,#43,rdmm2E
    mov   A,#10
rdmm2E: mov    Dt2E,A
    inc   DPTR
    movx  A,@DPTR
    cjne  A,#43,rdmm2F
    mov   A,#10
rdmm2F: mov    Dt2F,A
    xrl   Wmcn,#Eemn
    ret
;
wrmem: orl   Wmcn,#Eemw
    orl   Wmcn,#Eemw
    mov   A,Dt10
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt11
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt12
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt13
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt14
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt15
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt16
    movx  @DPTR,A
    lcall wt_wr
    inc   DPTR
    mov   A,Dt17
    movx  @DPTR,A
    lcall wt_wr

```

```
inc DPTR
mov A,Dt18
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt19
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1A
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1B
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1C
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1D
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1E
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt1F
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt20
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt21
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt22
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt23
movx @DPTR,A
lcall wt_wr
```

```
inc DPTR
mov A,Dt24
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt25
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt26
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt27
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt28
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt29
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2A
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2B
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2C
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2D
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2E
movx @DPTR,A
lcall wt_wr
inc DPTR
mov A,Dt2F
movx @DPTR,A
lcall wt_wr
```

```

    xrl  Wmcn,#Eemw
    xrl  Wmcn,#Eemn
    ret
;
rdspg1: orl  Wmcn,#Eemn
        mov  DPTR,#200
        movx A,@DPTR
        mov  Stp1,A
        xrl  Wmcn,#Eemn
        mov  A,Stp1
        cjne A,#0FFh,rdsp1
        mov  Stp1,#0
rdsp1:  ret
;
wrspg1: orl  Wmcn,#Eemn
        orl  Wmcn,#Eemw
        mov  DPTR,#200
        mov  A,Stp1
        movx @DPTR,A
        lcall wt_wr
        xrl  Wmcn,#Eemw
        xrl  Wmcn,#Eemn
        ret
;
rdspg2: orl  Wmcn,#Eemn
        mov  DPTR,#201
        movx A,@DPTR
        mov  Stp2,A
        xrl  Wmcn,#Eemn
        mov  A,Stp2
        cjne A,#0FFh,rdsp2
        mov  Stp2,#0
rdsp2:  ret
;
wrspg2: orl  Wmcn,#Eemn
        orl  Wmcn,#Eemw
        mov  DPTR,#201
        mov  A,Stp2
        movx @DPTR,A
        lcall wt_wr
        xrl  Wmcn,#Eemw
        xrl  Wmcn,#Eemn
        ret
;
rdspg3: orl  Wmcn,#Eemn
        mov  DPTR,#202
        movx A,@DPTR
        mov  Stp3,A

```

```

    xrl  Wmcn,#Eemn
    mov  A,Stp3
    cjne A,#0FFh,rdsp3
    mov  Stp3,#0
rdsp3: ret
;
wrspg3: orl  Wmcn,#Eemn
        orl  Wmcn,#Eemw
        mov  DPTR,#202
        mov  A,Stp3
        movx @DPTR,A
        lcall wt_wr
        xrl  Wmcn,#Eemw
        xrl  Wmcn,#Eemn
        ret
;
rdspg4: orl  Wmcn,#Eemn
        mov  DPTR,#203
        movx A,@DPTR
        mov  Stp4,A
        xrl  Wmcn,#Eemn
        mov  A,Stp4
        cjne A,#0FFh,rdsp4
        mov  Stp4,#0
rdsp4: ret
;
wrspg4: orl  Wmcn,#Eemn
        orl  Wmcn,#Eemw
        mov  DPTR,#203
        mov  A,Stp4
        movx @DPTR,A
        lcall wt_wr
        xrl  Wmcn,#Eemw
        xrl  Wmcn,#Eemn
        ret
;
rdspg5: orl  Wmcn,#Eemn
        mov  DPTR,#204
        movx A,@DPTR
        mov  Stp5,A
        xrl  Wmcn,#Eemn
        mov  A,Stp5
        cjne A,#0FFh,rdsp5
        mov  Stp5,#0
rdsp5: ret
;
wrspg5: orl  Wmcn,#Eemn
        orl  Wmcn,#Eemw

```

```

    mov DPTR,#204
    mov A,Stp5
    movx @DPTR,A
    lcall wt_wr
    xrl Wmcn,#Eemw
    xrl Wmcn,#Eemn
    ret
;
rdstpg: orl Wmcn,#Eemn
        mov DPTR,#254
        movx A,@DPTR
        mov Stpg,A
        xrl Wmcn,#Eemn
        mov A,Stpg
        cjne A,#0FFh,rdstpx
        mov Stpg,#0
rdstpx: ret
;
wrstpg: orl Wmcn,#Eemn
        orl Wmcn,#Eemw
        mov DPTR,#254
        mov A,Stpg
        movx @DPTR,A
        lcall wt_wr
        xrl Wmcn,#Eemw
        xrl Wmcn,#Eemn
        ret
;
rdstps: orl Wmcn,#Eemn
        mov DPTR,#255
        movx A,@DPTR
        mov Stps,A
        xrl Wmcn,#Eemn
        mov A,Stps
        cjne A,#0FFh,rdstpy
        mov Stps,#0
rdstpy: ret
;
wrstps: orl Wmcn,#Eemn
        orl Wmcn,#Eemw
        mov DPTR,#255
        mov A,Stps
        movx @DPTR,A
        lcall wt_wr
        xrl Wmcn,#Eemw
        xrl Wmcn,#Eemn
        ret
;

```

```

jeda: djnz Dly0,$
      ret
;
delays: mov Dly1,#255
dlys: lcall jeda
      djnz Dly1,dlys
      ret
;
delayl: mov Dly2,#10
dlyl: lcall delays
      djnz Dly2,dlyl
      ret
;
delayi: mov Dly2,#20
dlyi: lcall delays
      djnz Dly2,dlyi
      ret
;
talk: lcall delays
      djnz Bicr,talk
      ret
;
logo: DB ' Wahyu '
      DB ' Elektro '
      DB 'Alat Terima Tamu'
      DB ' Pemanggil Otomat '
maxchr: DB 'Text Max 16 Chr '
pgnada: DB ' Penghuni Ada '
belbny: DB ' Bel Berbunyi '
tkntbl: DB ' Tekan: '
pnggl: DB ' Panggil '
sysmgl: DB 'System Memanggil'
pglnm1: DB ' wahyu '
pglnm2: DB ' Riris '
pglnm3: DB ' Icha '
pglnm4: DB ' Fa`us '
pglnm5: DB ' Andi '
pgtdad: DB ' Penghuni Tdk Ada'
lhtsts: DB ' Lihat Status ? '
sttada: DB 'Status: Ada '
stttkd: DB 'Status: Tdk Ada '
psntxt: DB ' Pesan Text ? '
psnbaru:DB ' Pesan Baru '
bcpesan:DB ' Baca Pesan ? '
dsuara: DB ' Dengar Suara ? '
plying: DB ' Playing.... '
rsuara: DB ' Rekam Suara ? '
recing: DB ' Recording.... '

```

```
address: DB ' Address: '  
kosong: DB ' '  
chrctr: DB '0123456789 abcdefghijklmnopqrstuvwxyz,?!-&_ '  
  
;  
    end
```


Features

Compatible with MCS-51™ Products
8K Bytes of In-System Reprogrammable Downloadable Flash Memory
– SPI Serial Interface for Program Downloading
– Endurance: 1,000 Write/Erase Cycles
2K Bytes EEPROM
– Endurance: 100,000 Write/Erase Cycles
1.0V to 6V Operating Range
Fully Static Operation: 0 Hz to 24 MHz
Three-Level Program Memory Lock
256 x 8-bit Internal RAM
32 Programmable I/O Lines
Three 16-bit Timer/Counters
Five Interrupt Sources
Programmable UART Serial Channel
SPI Serial Interface
Low Power Idle and Power Down Modes
Interrupt Recovery From Power Down
Programmable Watchdog Timer
Dual Data Pointer
Power Off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode suspends the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but ceases the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



8-Bit Microcontroller with 8K Bytes Flash

AT89S8252

0401D-A-12/97





8-Bit
Microcontroller
with 8K Bytes
Flash

AT8928252

1990-A-0000

4-102



Features

- Compatible with MCS 81™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
- SPI Serial Interface for Program Downloading
- Endurance: 1,000 Write/Erase Cycles
- 2 Bytes EEPROM
- Endurance: 100,000 Write/Erase Cycles
- 1.0V to 5V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-bit Internal RAM
- 12 Programmable I/O Lines
- Three 18-bit Timers/Counters
- Multiple Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- On-Chip Data Pointer
- Power Off Flag

Description

The AT8928252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash, programmable and erasable read-only memory and 256 bytes of EEPROM. The device is manufactured using Atmel's high density non-volatile memory technology and is compatible with the industry standard 80C81 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be programmed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT8928252 is a powerful microcomputer which provides a highly flexible and cost-effective solution to many embedded control applications.

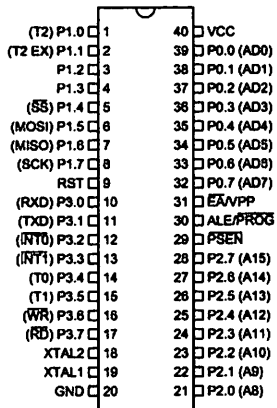
The AT8928252 provides the following standard features: 8K bytes of Downloadable Flash, 256 bytes of EEPROM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 18-bit timers/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT8928252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from. The Lock Bit 2 has been activated.

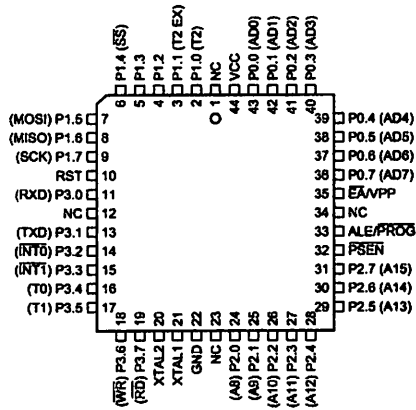


Pin Configurations

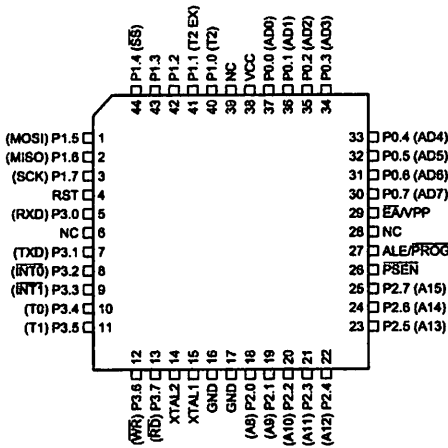
PDIP



PLCC



PQFP/TQFP



Pin Description

Supply voltage.

Ground.

Port 0
Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

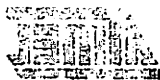
Port 0 can also be configured to be the multiplexed lower address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.



Configurations

Table 1

Configuration	Number of Configurations
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	1
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1
78	1
79	1
80	1
81	1
82	1
83	1
84	1
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

Table 2

Configuration	Number of Configurations
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	1
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1
78	1
79	1
80	1
81	1
82	1
83	1
84	1
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

Configuration

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

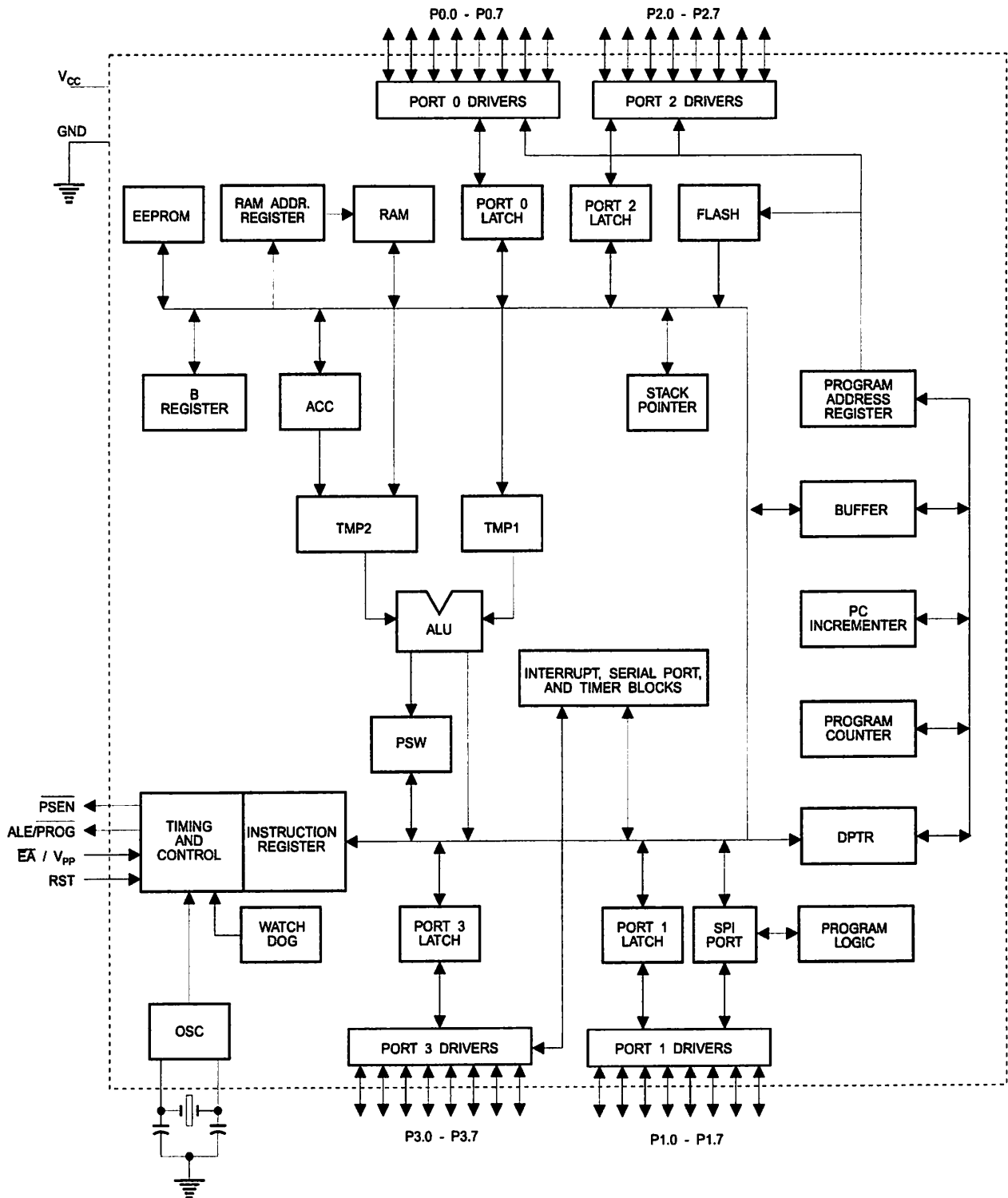
that the output of the program is the same as the output of the program when the input is the same as the input of the program. The program is written in C++ and runs on a Windows operating system.

Table 1

Table 1 shows the number of configurations for each of the 100 configurations. The number of configurations is 1 for each configuration. The number of configurations is 1 for each configuration. The number of configurations is 1 for each configuration.

Some of the configurations provide additional features. For example, configuration 1 provides an external trigger input (P1) and the internal trigger input (P2) and the internal trigger input (P3) respectively.

Block Diagram





Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
1.0	T2 (external count input to Timer/Counter 2), clock-out
1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
1.4	\overline{SS} (Slave port select input)
1.5	MOSI (Master data output, slave data input pin for SPI channel)
1.6	MISO (Master data input, slave data output pin for SPI channel)
1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ PTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	\overline{WR} (external data memory write strobe)
P3.7	\overline{RD} (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/ \overline{PROG}

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (\overline{PROG}) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

\overline{PSEN}

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, \overline{PSEN} is activated twice each machine cycle, except that two \overline{PSEN} activations are skipped during each access to external data memory.

\overline{EA}/V_{PP}

External Access Enable. \overline{EA} must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, \overline{EA} will be internally latched on reset.

\overline{EA} should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

AL1
 out to the inverting oscillator amplifier and input to the
 external clock operating circuit.

AL2
 output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XX00000	87H



er software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the unlisted bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable read/write. The DPS bit selects one of two DPTR registers available.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H						Reset Value = 0000 0000B		
8-bit Addressable								
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{T2}$	CP/ $\overline{RL2}$
	7	6	5	4	3	2	1	0
Symbol	Function							
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.							
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).							
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.							
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.							
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.							
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.							
C/ $\overline{T2}$	Timer or counter select for Timer 2. C/ $\overline{T2}$ = 0 for timer function. C/ $\overline{T2}$ = 1 for external event counter (falling edge triggered).							
CP/ $\overline{RL2}$	Capture/Reload select. CP/ $\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/ $\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.							

Internal Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 8-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H				Reset Value = 0000 0010B				
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
Bit	7	6	5	4	3	2	1	0
Symbol	Function							
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.							
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.							
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.							
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1							
WDTRST RDY/ \overline{BSY}	Watchdog Timer Reset and EEPROM Ready/ \overline{Busy} Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/ \overline{BSY} flag in a Read-Only mode during EEPROM write. RDY/ \overline{BSY} = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/ \overline{BSY} bit equals "0" and is automatically reset to "1" when programming is completed.							
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.							



MDTEN	Matched Time Enable Bit. MDTEN = 1 enables the matched time, and MDTEN = 0 disables the matched time. The BDAVSA bit is set to 0, and is automatically reset to 1, when programming is completed.
MDTBS1	Matched Time Reset and EEPROM Ready/Busy Flag. Each time this bit is set to 1, by user software, a write to second bank, DB1.
DB2	Data Point Register Select. DB2 = 0 selects the first bank of data point registers, DB0, and DB2 = 1 selects the second of external data memory. When EEMEN = 0, MOAX with DB1B accesses external data memory.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOAX instruction with DB1B will access on-chip EEPROM.
EEMWE	MOAX instruction. User software should set this bit to 0, when EEPROM write is completed.
DB0	EEPROM Data Memory Write Enable Bit. Set this bit to 1, before initiating write to on-chip EEPROM with the MOAX instruction.
DB1	DB1. When all three bits are set to 1, the nominal period is 50ms.
DB2	DB2. This bit is for the Matched Time. When all three bits are set to 0, the matched time, and a positive period of 10ms.
Symbol	Function

	7	6	5	4	3	2	1	0
	DB2	DB1	DB0	EEMWE	EEMEN	DB2	MDTBS1	MDTEN

WMC00 address = 80H

Base: Data = 0000 0010B

Table 2. WMC00—Matched Time and Memory Control Register

The user should always initialize the DB2 bit to the 0 in SFR WMC00 selects DB0 and DB2 = 1 selects these locations 82H-83H and DB1 at 84H-85H. Bit DB2 bit 0 is 1 Point Register are provided: DB0 at SFR 80H, EEPROM and external data memory, two banks of Data Point Register. To external accessing not

and is not affected by RESET.
power up. It can be set and reset under software control bit 4 (PCOIN4) in the PCOIN SFR. POE is set to 1, during Power Off Flag. The Power Off Flag (POE) is located at Point Register.
appropriate value before accessing the respective Data



Table 4. SPCR—SPI Control Register

SPCR Address = D5H				Reset Value = 0000 01XXB				
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Bit	7	6	5	4	3	2	1	0
Symbol	Function							
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.							
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.							
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.							
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.							
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.							
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.							
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{OSC} , is as follows:							
	SPR1	SPR0						SCK = F_{OSC} , divided by
	0	0						4
	0	1						16
	1	0						64
	1	1						128

Table 5. SPSR—SPI Status Register

SPSR Address = AAH				Reset Value = 00XX XXXXB				
	SPIF	WCOL	—	—	—	—	—	
Bit	7	6	5	4	3	2	1	0
Symbol	Function							
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.							
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.							

Table 6. SPDR—SPI Data Register

SPDR Address = 86H				Reset Value = unchanged			
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
7	6	5	4	3	2	1	0

Data Memory—EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at $V_{CC} = 5V$) are within $\pm 30\%$ of the nominal.

The WDT is disabled by Power-on Reset and during Power Down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $C/\overline{TR2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/ $\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

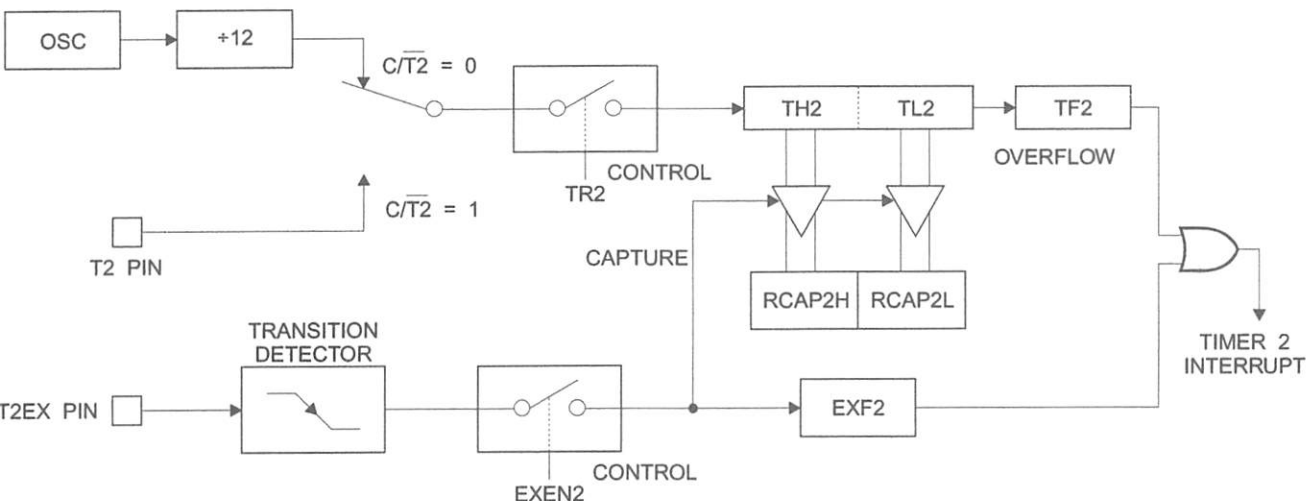
In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to

Figure 1. Timer 2 in Capture Mode



FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in

RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

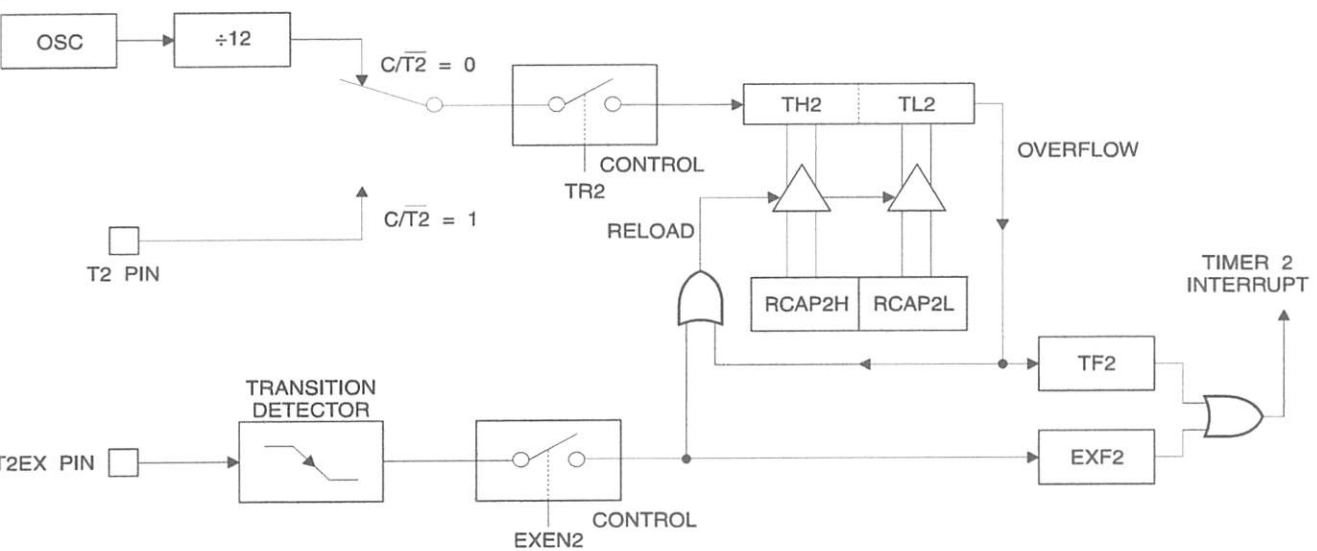


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H							Reset Value = XXXX XX00B	
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN
Symbol	Function							
—	Not implemented, reserved for future use.							
T2OE	Timer 2 Output Enable bit.							
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.							

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

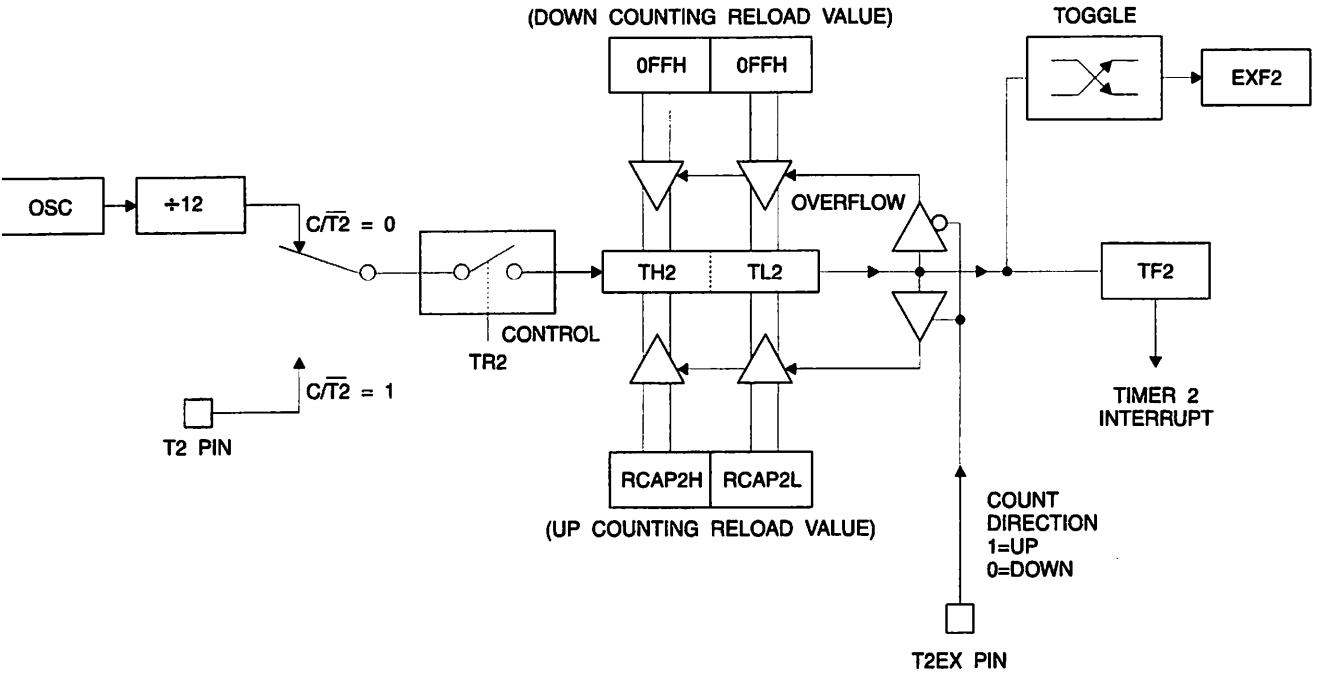
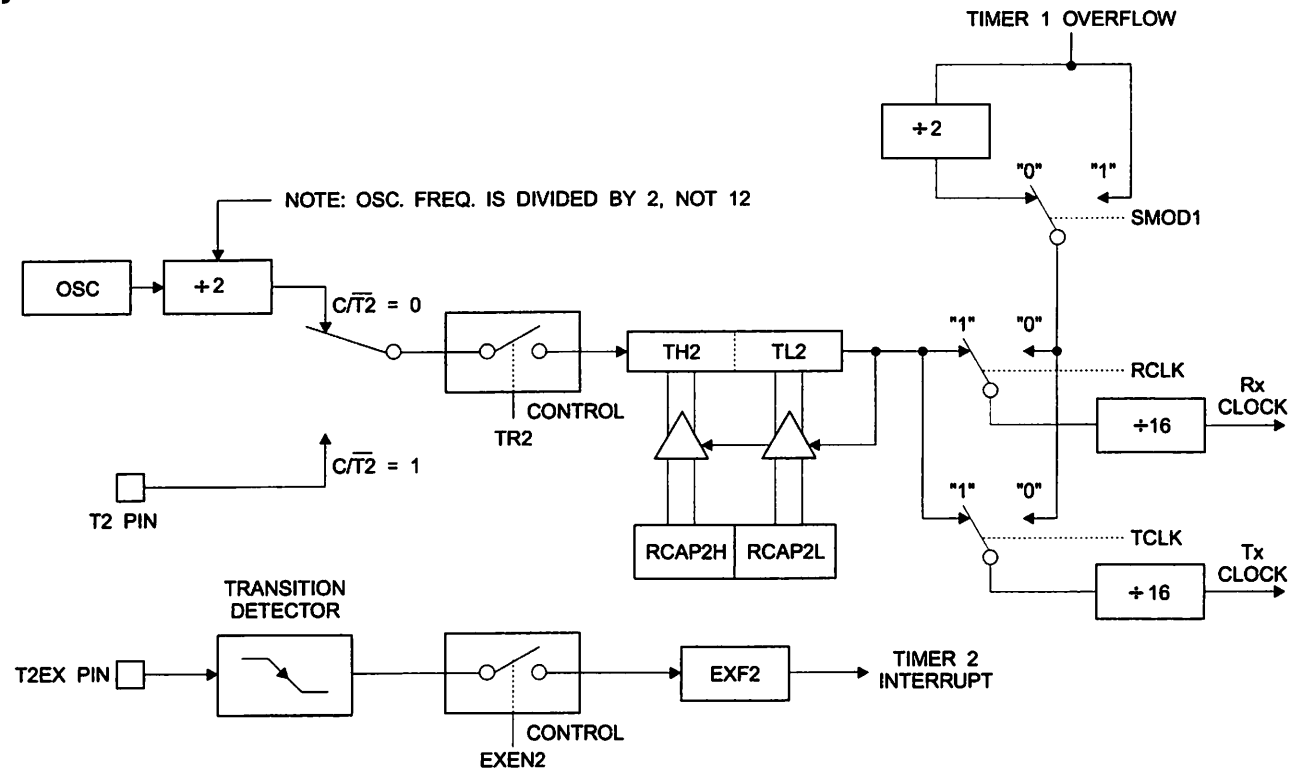
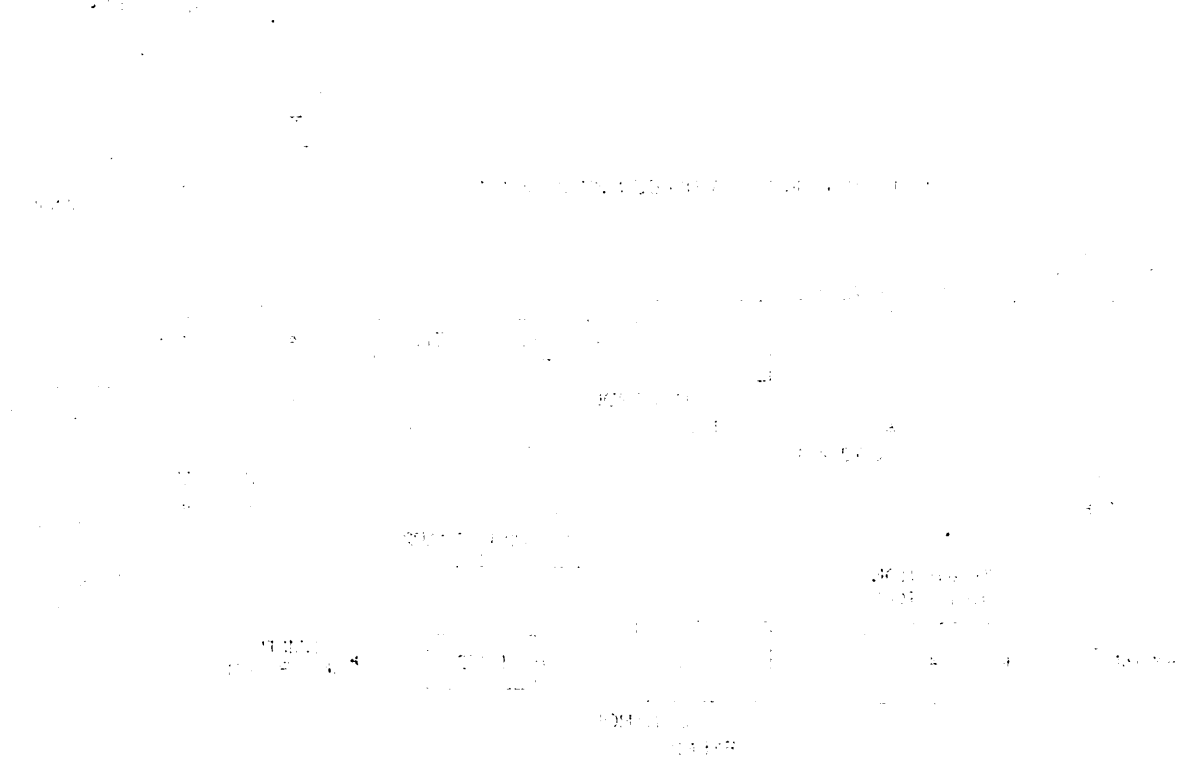
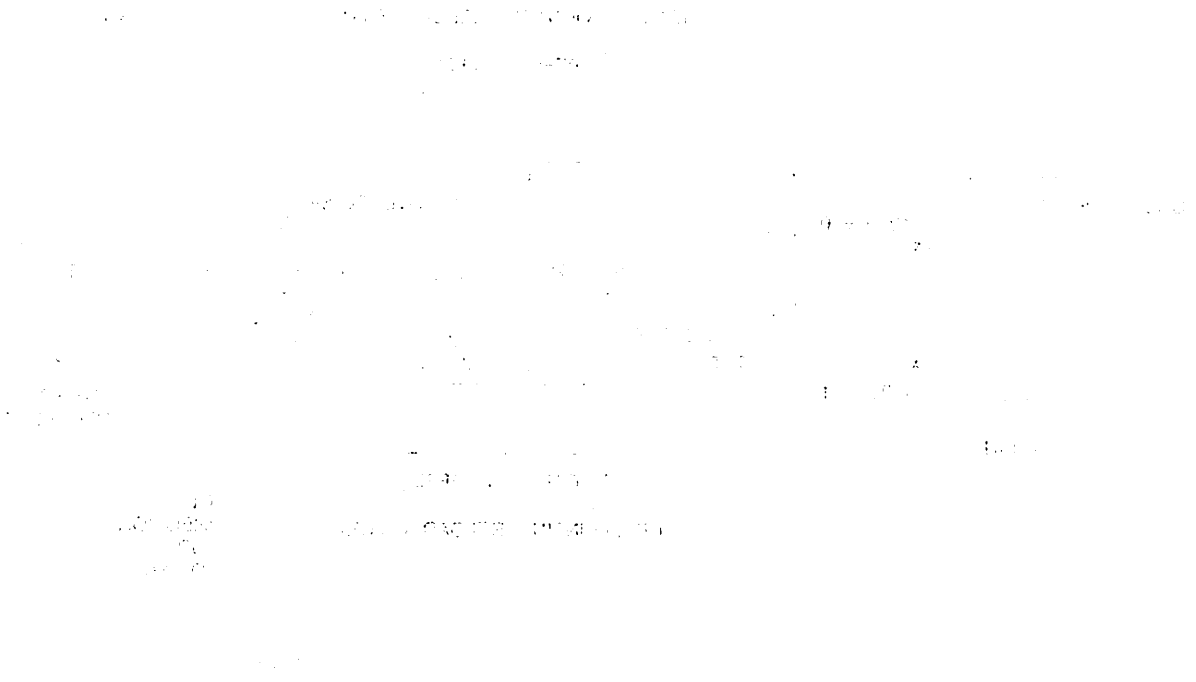
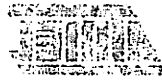


Figure 4. Timer 2 in Baud Rate Generator Mode





Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting RCLK and/or TCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($CP/T2 = 0$). The timer operation is different from Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at $1/12$ the oscillator frequency). As a baud rate generator, however, it increments every state time (at $1/2$ the oscillator frequency). The baud rate formula is given below.

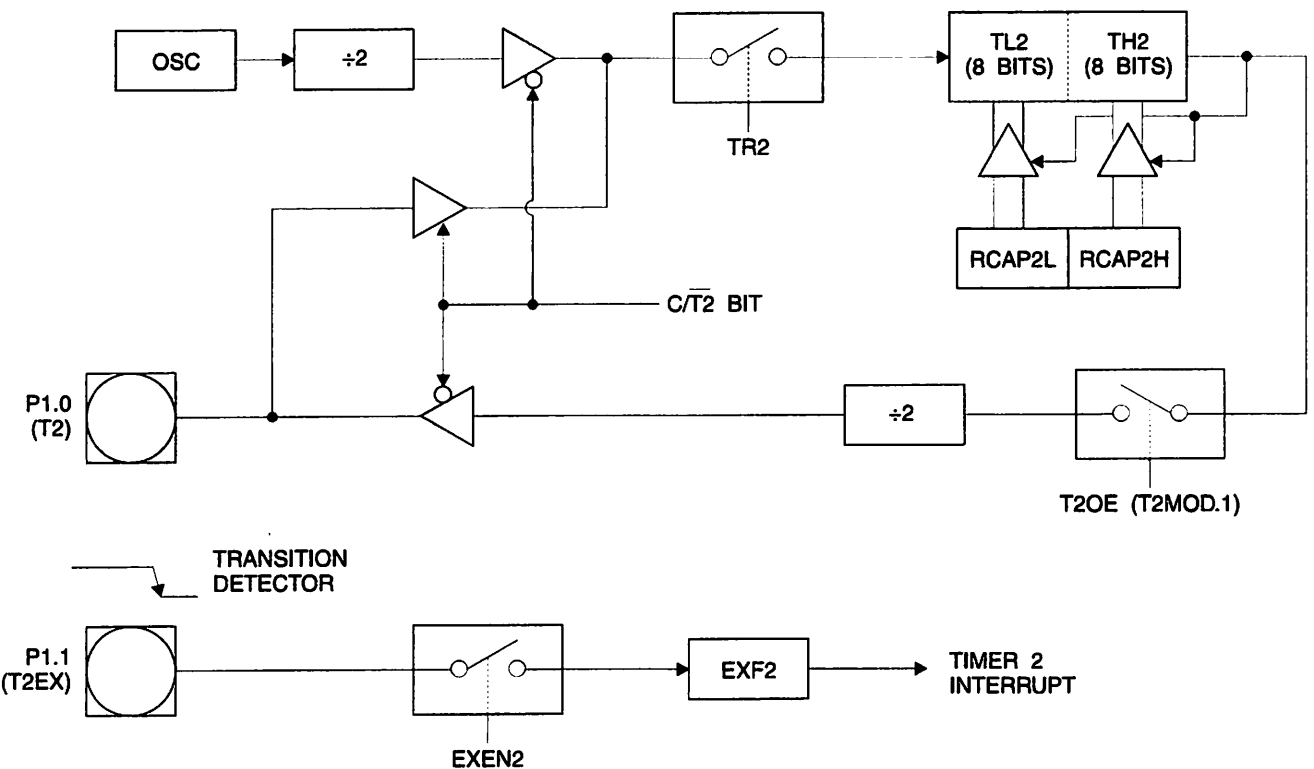
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

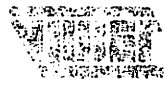
where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running ($TR2 = 1$) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-Out Mode





130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

130-11

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on pin 1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or output a 50% duty cycle clock ranging from 61 Hz to 4 kHz at a 16 MHz operating frequency.

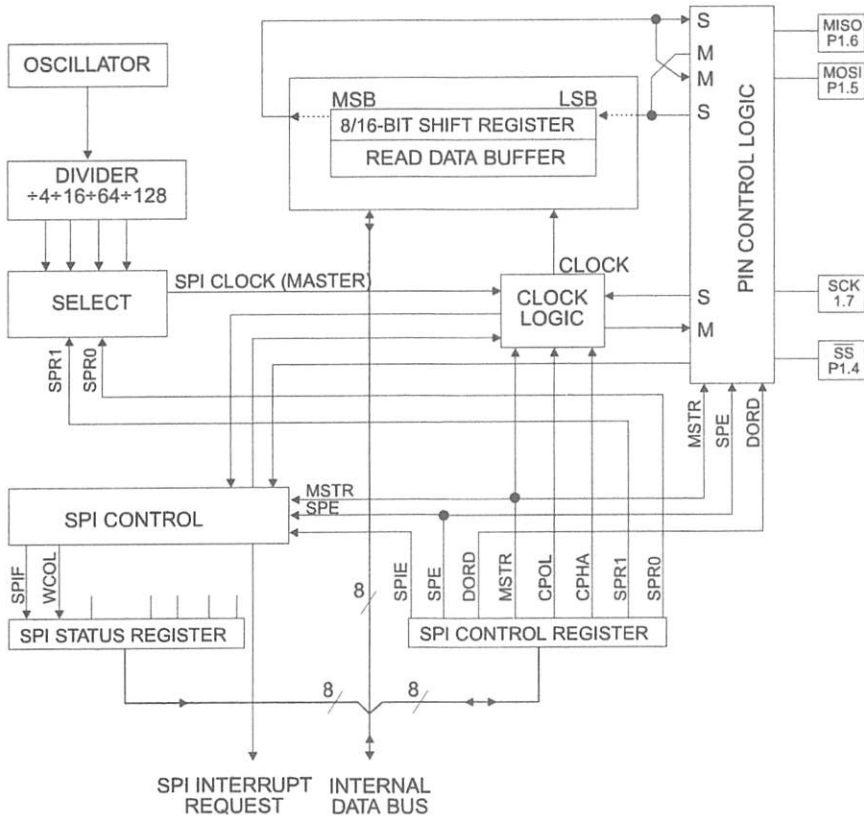
To configure the Timer/Counter 2 as a clock generator, bit T2IF (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \cdot (5536 - \text{RCAP2H}, \text{RCAP2L})}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Figure 6. SPI Block Diagram



UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5-MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

18888888

SECRET

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

SECRET

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

SECRET

SECRET

SECRET

- SECRET
- CONFIDENTIAL
- CONFIDENTIAL
- CONFIDENTIAL
- CONFIDENTIAL
- CONFIDENTIAL
- CONFIDENTIAL
- CONFIDENTIAL

SECRET

SECRET



SECRET

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

Figure 7. SPI Master-Slave Interconnection

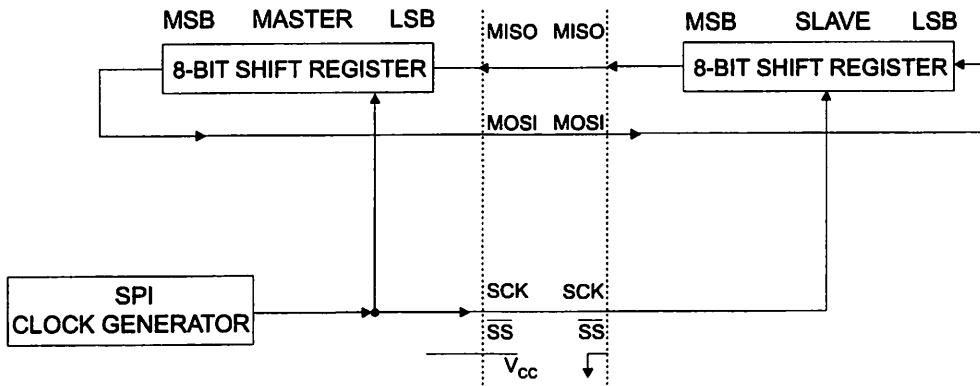
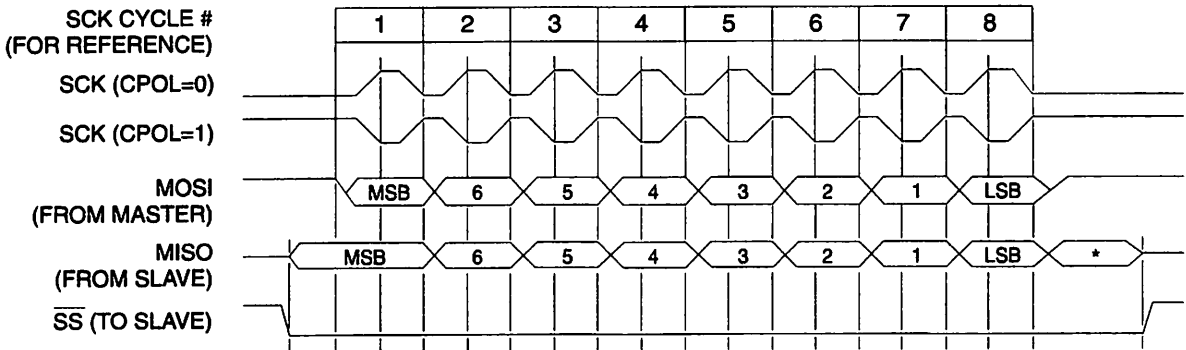


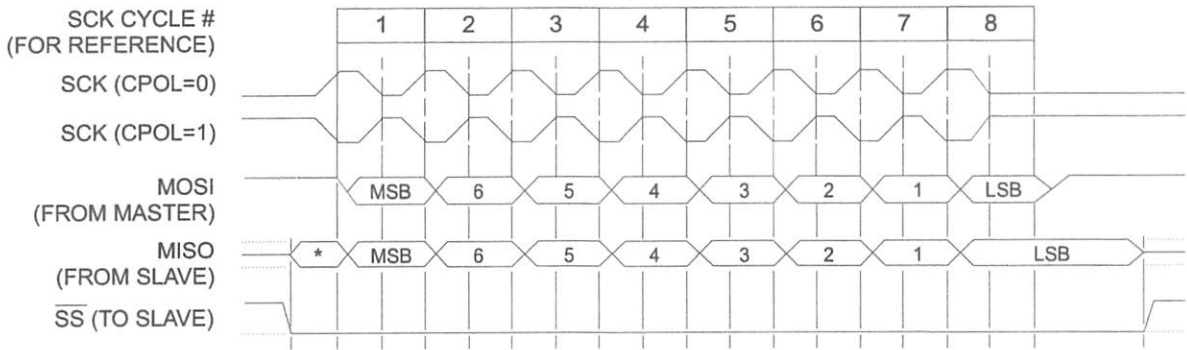
Figure 8. SPI transfer Format with CPHA = 0



not defined but normally MSB of character just received



Figure 9. SPI Transfer Format with CPHA = 1



not defined but normally LSB of previously transmitted character

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 10. Interrupt Enable (IE) Register

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Figure 10. Interrupt Sources

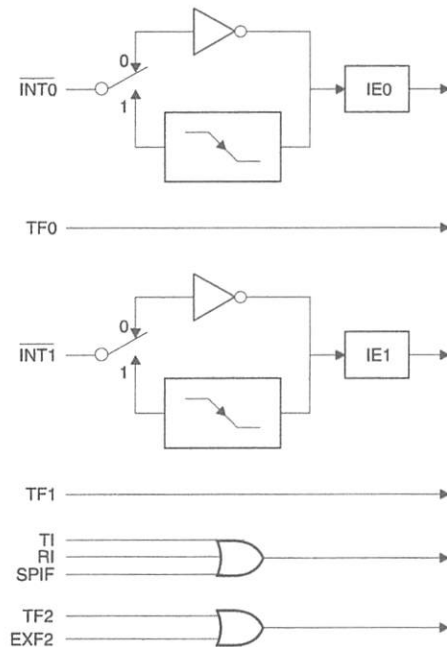
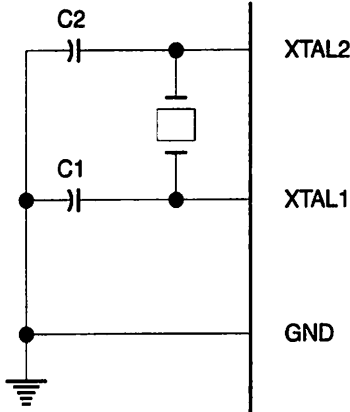


Figure 11. Oscillator Connections



Note: C1, C2 = 30 pF □ 10 pF for Crystals
 = 40 pF □ 10 pF for Ceramic Resonators

Oscillator Characteristics

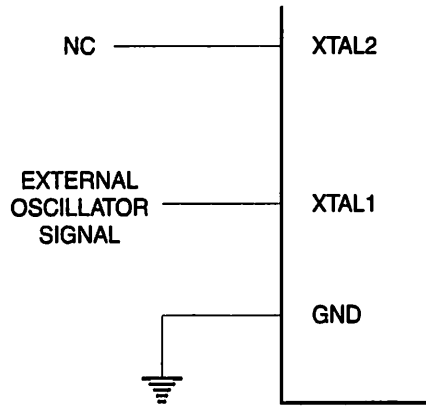
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the

Figure 12. External Clock Drive Configuration



internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power Down Mode

In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data





Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random

value and holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Lock Bit Protection Modes⁽¹⁾⁽²⁾

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory. \overline{EA} is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

- Notes:
1. U = Unprogrammed
 2. P = Programmed

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to perform the Chip Erase operation first to erase both arrays.

Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set \overline{PSEN} pin to "L"
 - ALE pin to "H"
 - \overline{EA} pin to "H" and all other pins to "H".
3. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
 - Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise \overline{EA}/V_{PP} to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/\overline{PROG} once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.

Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.

Power-off sequence:

Set XTAL1 to "L".

Set RST and \overline{EA} pins to "L".

Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

DATA Polling

The AT89S8252 features \overline{DATA} Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P3.7 (parallel mode), and on the MSB of the serial output on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next write cycle may begin. \overline{DATA} Polling may begin any time after a write cycle has been initiated.

RDY/BSY

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate \overline{BUSY} . P3.4 is pulled High again when programming is done to indicate READY.

Lock Verify

Lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase

Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/ \overline{PROG} low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

Serial Programming Fuse

A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming Mode enabled.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

(030H) = 1EH indicates manufactured by Atmel

(031H) = 72H indicates 89S8252

Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.





Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

Power-up sequence:

Apply power between V_{CC} and GND pins.

Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.

The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is

written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

- Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
- At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn V_{CC} power off.

Serial Programming Instruction

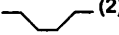


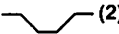

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	$\overline{\text{LB}}_3 \overline{\text{LB}}_2 \overline{\text{LB}}_1 \times 111$	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

- Notes:
- $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
 - "aaaaa" = high order address.
 - "x" = don't care.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h ⁽¹⁾	h ⁽¹⁾	x						
Chip Erase	H	L	 (2)	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	 (2)	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 (2)	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

- Notes:
1. "h" = weakly pulled "High" internally.
 2. Chip Erase and Serial Programming Fuse require a 10-ms $\overline{\text{PROG}}$ pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
 3. P3.4 is pulled Low during programming to indicate RDY/ $\overline{\text{BSY}}$.
 4. "X" = don't care



Figure 14. Programming the Flash/EEPROM Memory

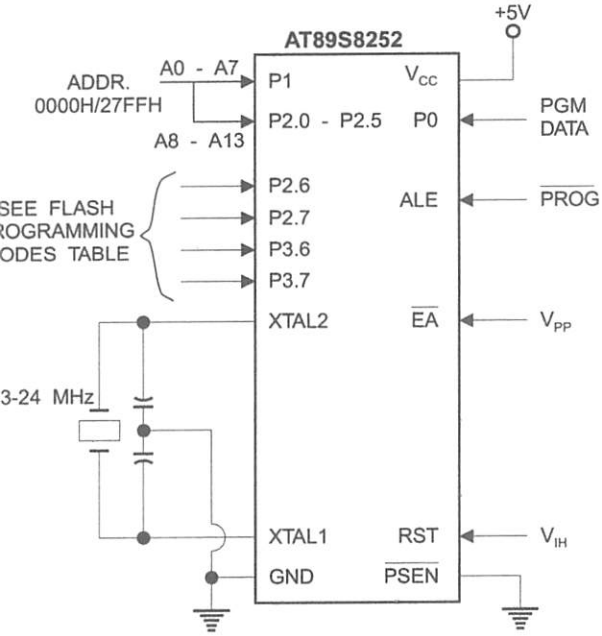


Figure 15. Flash/EEPROM Serial Downloading

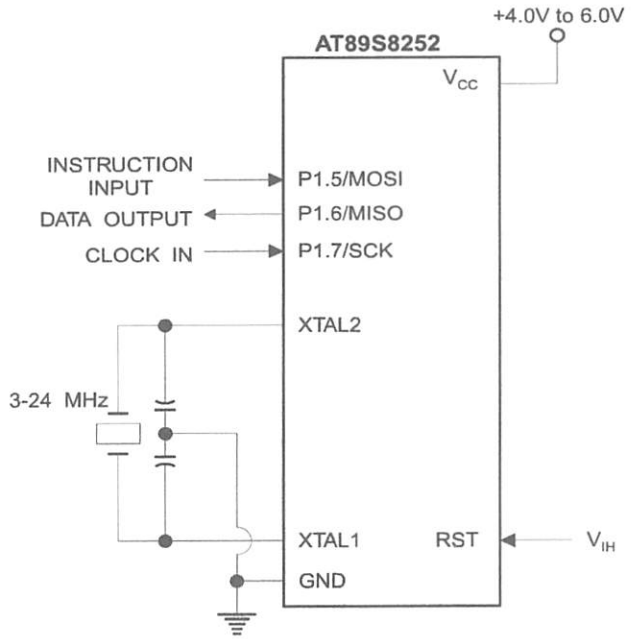
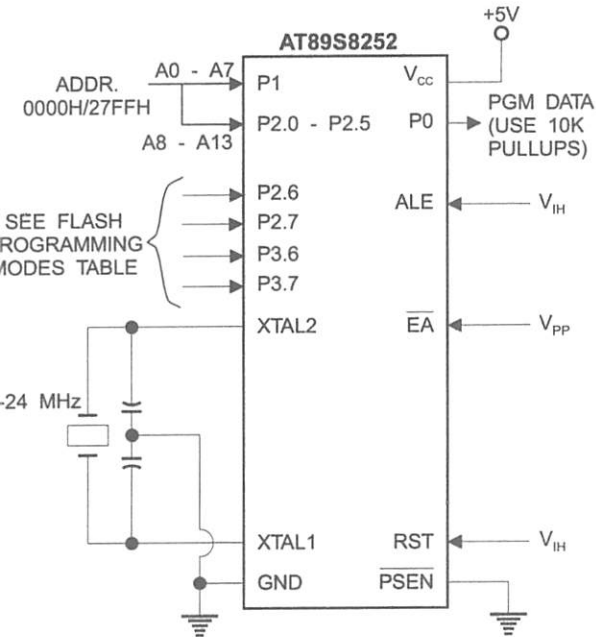


Figure 16. Verifying the Flash/EEPROM Memory



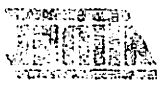


Figure 14. Programming the FlashEPROM Memory



Figure 15. FlashEPROM Serial Flashing

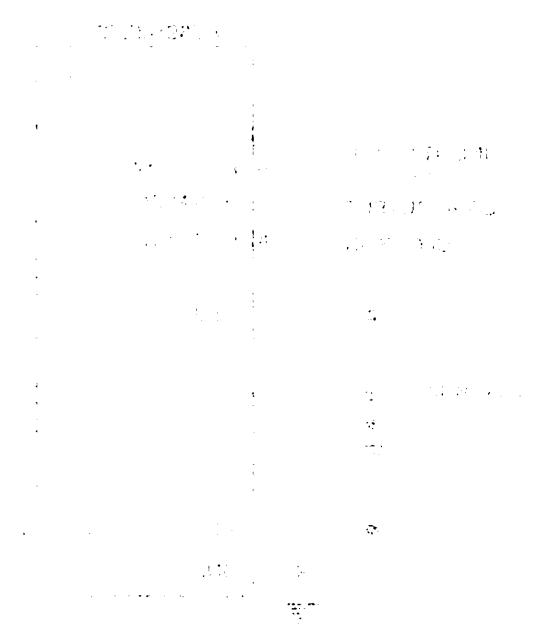
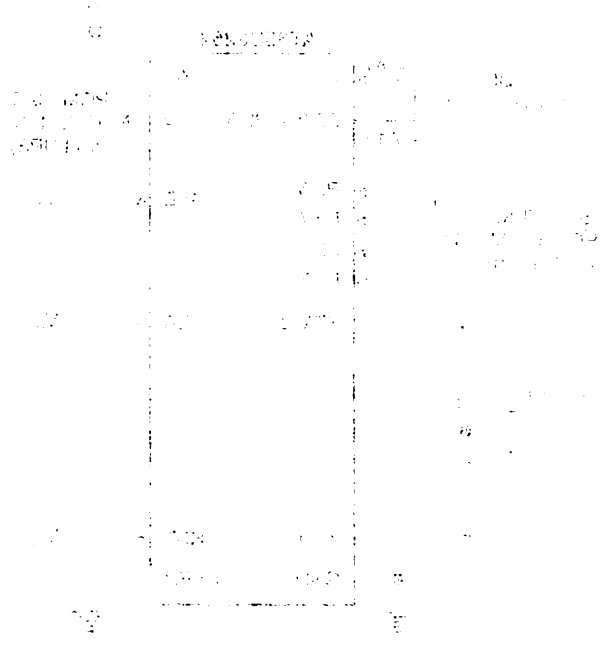


Figure 16. Reading the FlashEPROM Memory



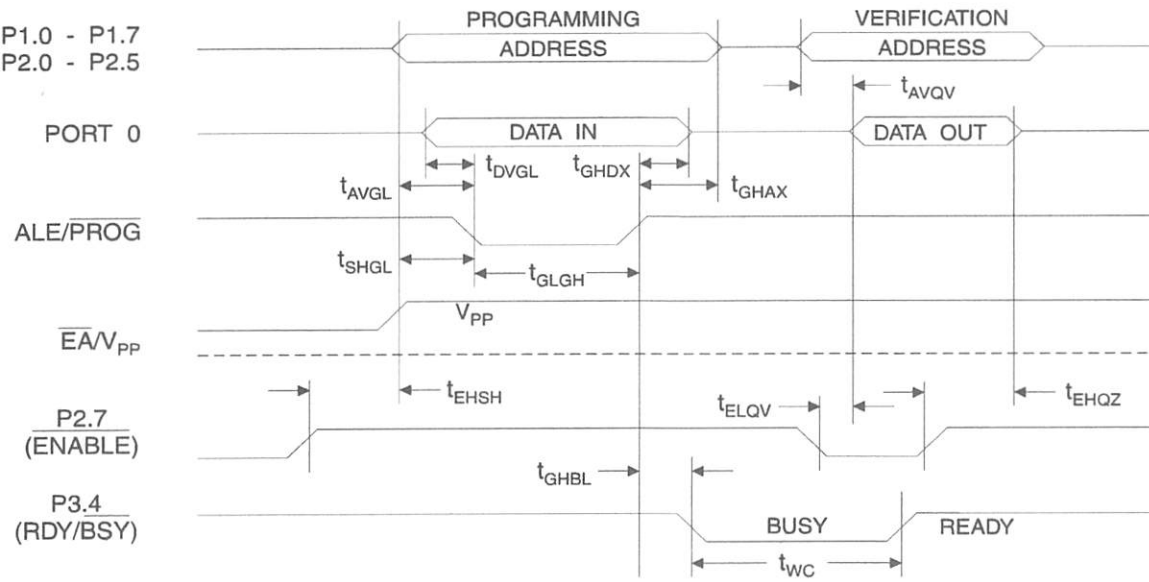
Flash Programming and Verification Characteristics-Parallel Mode

= 0°C to 70°C, V_{CC} = 5.0V □10%

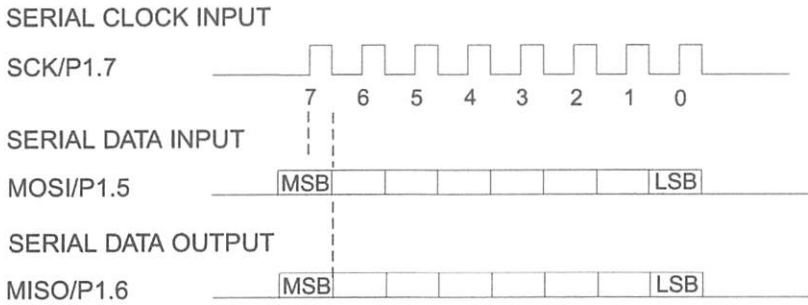
Symbol	Parameter	Min	Max	Units
V _{PP}	Programming Enable Voltage	11.5	12.5	V
I _{PP}	Programming Enable Current		1.0	mA
f _{CLCL}	Oscillator Frequency	3	24	MHz
t _{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{HAX}	Address Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
t _{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{DHDX}	Data Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
t _{HSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V _{PP}	48t _{CLCL}		
t _{HGL}	V _{PP} Setup to $\overline{\text{PROG}}$ Low	10		ns
t _{PLGH}	$\overline{\text{PROG}}$ Width	1	110	ns
t _{AVQV}	Address to Data Valid		48t _{CLCL}	
t _{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		48t _{CLCL}	
t _{DHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	48t _{CLCL}	
t _{HBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	ns
t _{WC}	Byte Write Cycle Time		2.0	ms

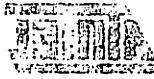


Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



Serial Downloading Waveforms





Parallel ROM Programming and Verification Waveforms - Parallel Mode



Initial Downloading Waveforms



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
IO Output Current.....	15.0 mA

***NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

C Characteristics

Values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage (\overline{EA})		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, \overline{PSEN})	$I_{OL} = 3.2 \text{ mA}$		0.5	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, \overline{PSEN})	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_L	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{IL}	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
R_{RST}	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8-bit port:
 Port 0: 26 mA
 Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2V





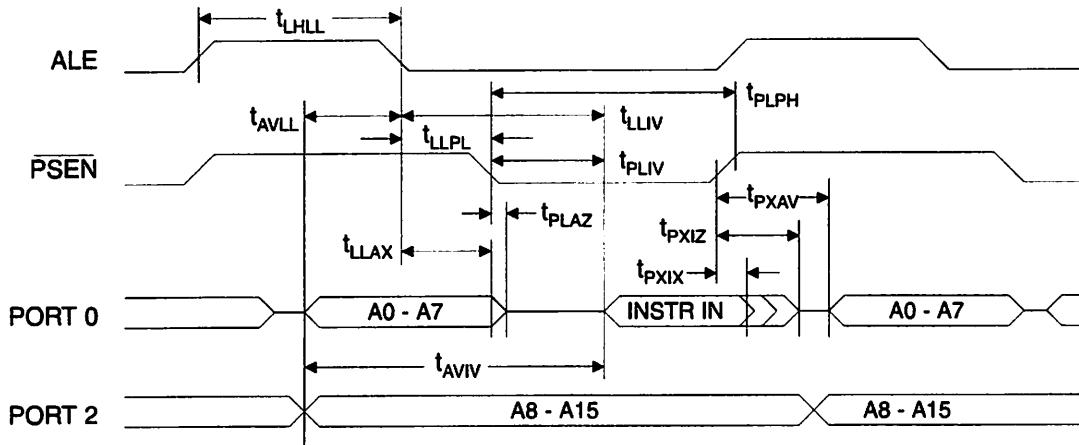
C Characteristics

Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

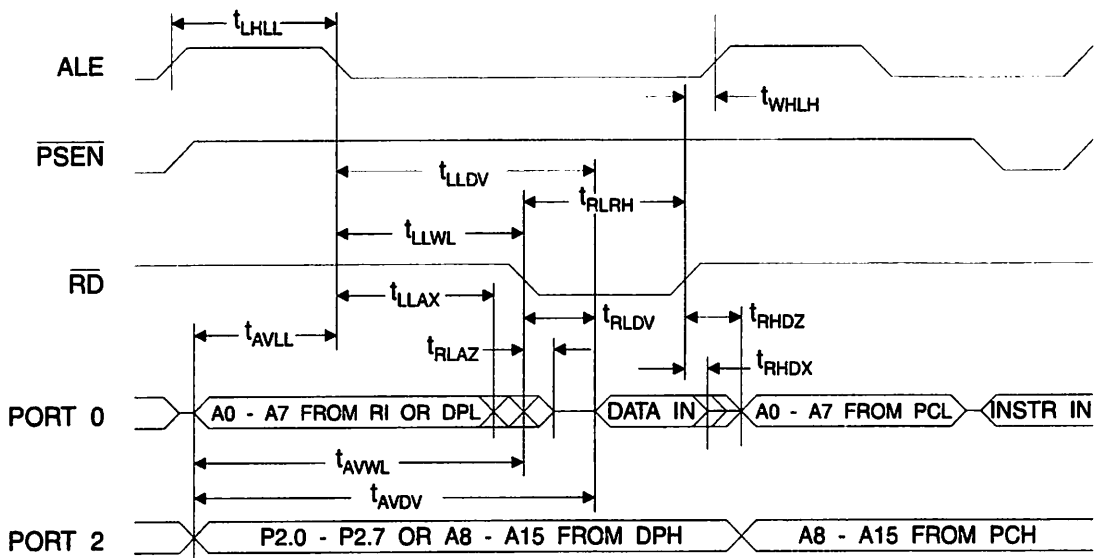
External Program and Data Memory Characteristics

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency	0	24	MHz
t_{LHLL}	ALE Pulse Width	$2t_{\text{CLCL}} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	$t_{\text{CLCL}} - 13$		ns
t_{LLAX}	Address Hold After ALE Low	$t_{\text{CLCL}} - 20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		$4t_{\text{CLCL}} - 65$	ns
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	$t_{\text{CLCL}} - 13$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	$3t_{\text{CLCL}} - 20$		ns
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		$3t_{\text{CLCL}} - 45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		ns
t_{PXIZ}	Input Instruction Float After $\overline{\text{PSEN}}$		$t_{\text{CLCL}} - 10$	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	$t_{\text{CLCL}} - 8$		ns
t_{AVIV}	Address to Valid Instruction In		$5t_{\text{CLCL}} - 55$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	$6t_{\text{CLCL}} - 100$		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	$6t_{\text{CLCL}} - 100$		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		$5t_{\text{CLCL}} - 90$	ns
t_{RHDX}	Data Hold After $\overline{\text{RD}}$	0		ns
t_{RHDZ}	Data Float After $\overline{\text{RD}}$		$2t_{\text{CLCL}} - 28$	ns
t_{LLDV}	ALE Low to Valid Data In		$8t_{\text{CLCL}} - 150$	ns
t_{AVDV}	Address to Valid Data In		$9t_{\text{CLCL}} - 165$	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	$3t_{\text{CLCL}} - 50$	$3t_{\text{CLCL}} + 50$	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	$4t_{\text{CLCL}} - 75$		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	$t_{\text{CLCL}} - 20$		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	$7t_{\text{CLCL}} - 120$		ns
t_{WHQX}	Data Hold After $\overline{\text{WR}}$	$t_{\text{CLCL}} - 20$		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	$t_{\text{CLCL}} - 20$	$t_{\text{CLCL}} + 25$	ns

External Program Memory Read Cycle



External Data Memory Read Cycle



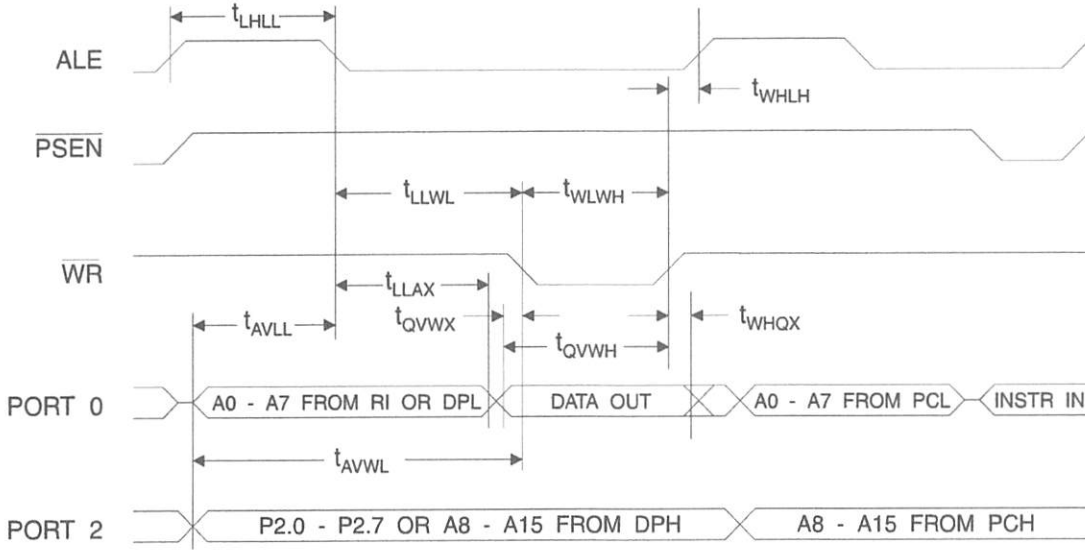
External Program Memory Read Cycle



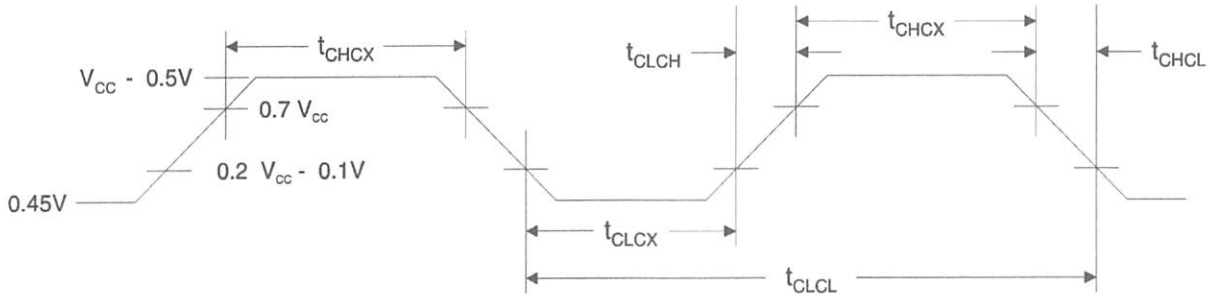
External Data Memory Read Cycle



External Data Memory Write Cycle

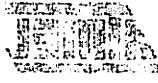


External Clock Drive Waveforms

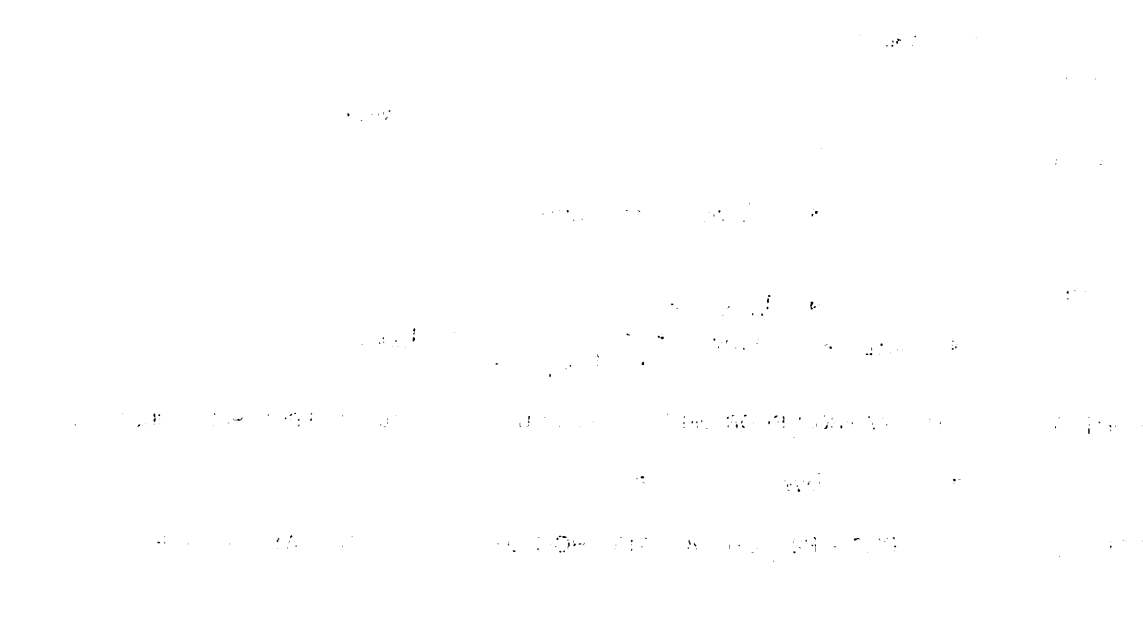


External Clock Drive

Symbol	Parameter	$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

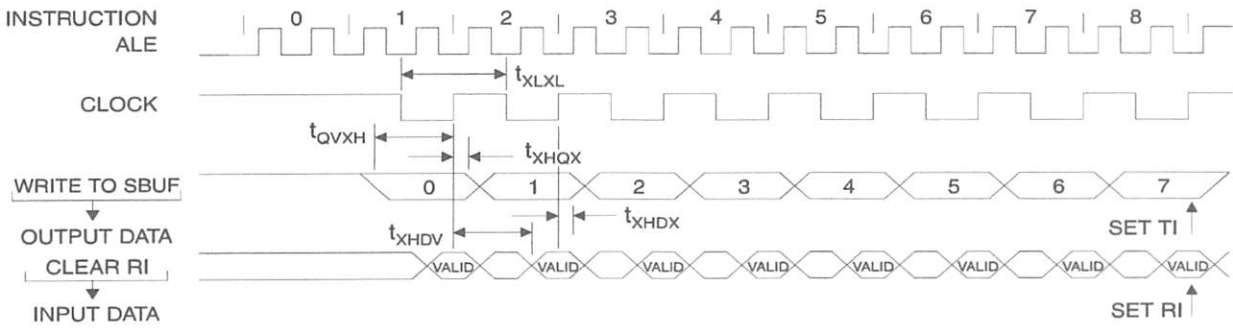
Signal	Level	Time	Notes
CS	0	0	Active Low
WE	0	0	Active Low
Data Bus	0	0	Low Level
CS	1	0	High Level
WE	1	0	High Level
Data Bus	1	0	High Level
CS	0	0	Active Low
WE	0	0	Active Low
Data Bus	0	0	Low Level
CS	1	0	High Level
WE	1	0	High Level
Data Bus	1	0	High Level

Serial Port Timing: Shift Register Mode Test Conditions

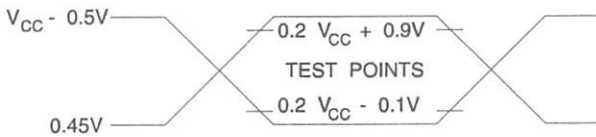
The values in this table are valid for $V_{CC} = 4.0V$ to $6V$ and Load Capacitance = 80 pF .

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	$10t_{CLCL} - 133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	$2t_{CLCL} - 117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		ns
t_{XHDX}	Clock Rising Edge to Input Data Valid		$10t_{CLCL} - 133$	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾

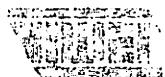


Notes: 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms⁽¹⁾



Notes: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.



... ..
... ..
... ..

... ..
... ..
... ..

... ..
... ..
... ..

... .. (1)

... .. (1)

... ..
... ..
... ..
... ..
... ..
... ..
... ..
... ..

... ..

...
...
...
...
...

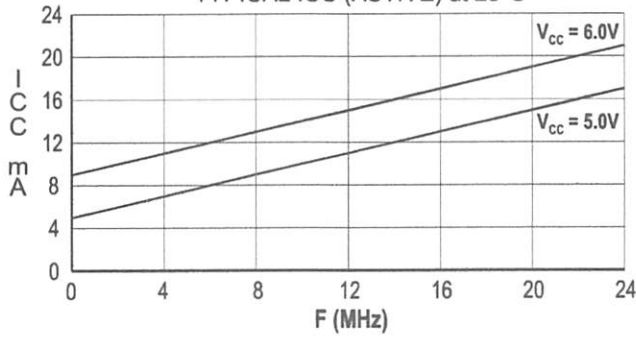
... ..

... ..



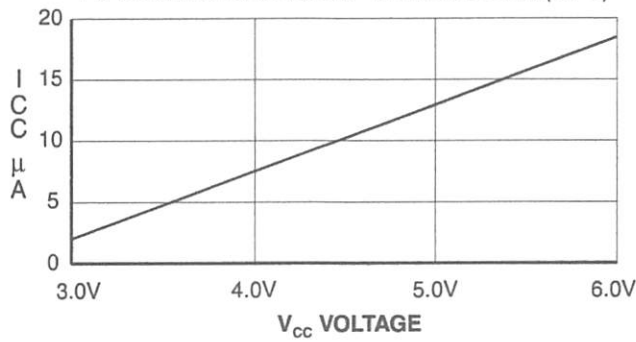
AT89S8252

TYPICAL ICC (ACTIVE) at 25°C



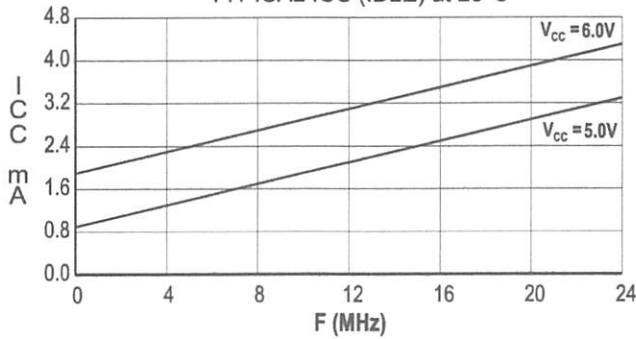
AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



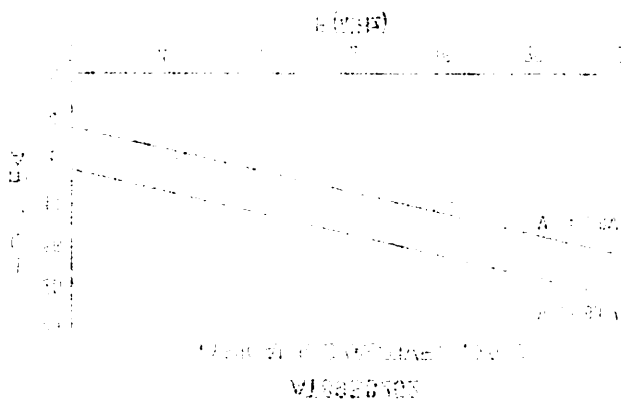
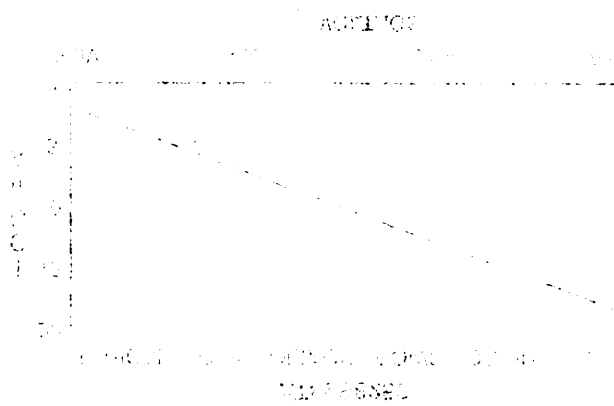
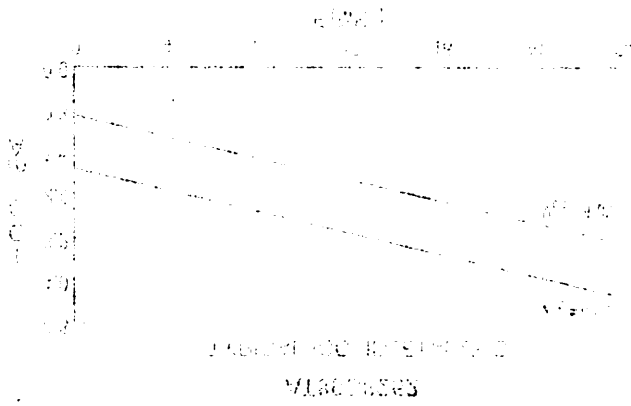
AT89S8252

TYPICAL ICC (IDLE) at 25°C



- Notes:
1. XTAL1 tied to GND for I_{CC} (power down)
 2. Lock bits programmed

- 2) розр. на відбиттях
- 3) довж. вол. в фазі довж. (howa howa)



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
16	4.0V to 6.0V	AT89S8252-16AA	44A	Automotive (-40°C to 105°C)
		AT89S8252-16JA	44J	
		AT89S8252-16PA	40P6	
		AT89S8252-16QA	44Q	
24	4.0V to 6.0V	AT89S8252-24AC	44A	Commercial (0°C to 70°C)
		AT89S8252-24JC	44J	
		AT89S8252-24PC	40P6	
		AT89S8252-24QC	44Q	
	4.0V to 6.0V	AT89S8252-24AI	44A	Industrial (-40°C to 85°C)
		AT89S8252-24JI	44J	
		AT89S8252-24PI	40P6	
		AT89S8252-24QI	44Q	
33	4.5V to 5.5V	AT89S8252-33AC	44A	Commercial (0°C to 70°C)
		AT89S8252-33JC	44J	
		AT89S8252-33PC	40P6	
		AT89S8252-33QC	44Q	

= Preliminary Information

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)



LIQUID CRYSTAL DISPLAY MODULE

M 1 6 3 2

USER MANUAL

Seiko Instruments Inc.

PREFACE

This manual describes technical informations on functions and instructions of M1632 from Seiko Instruments Inc. Please read this instruction manual carefully to understand all the module functions and make the best use of them. Description details may be changed without notice.

Revision Record

<u>Edition</u>	<u>Revision</u>	<u>Date</u>
1	Original	April 1985
2	Completely revised	Jan. 1987

© Seiko Instruments Inc. 1987

Printed in Japan

1. GENERAL

1.1 General

The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view TN LCD panel and a CMOS LCD drive controller built in. The controller has a built-in character generator ROM/RAM, and display data RAM. All the display functions are controlled by instructions and the module can easily be interfaced with an MPU. This makes the module applicable to a wide range of purposes including terminal display units for microcomputers and display units for measuring gages.

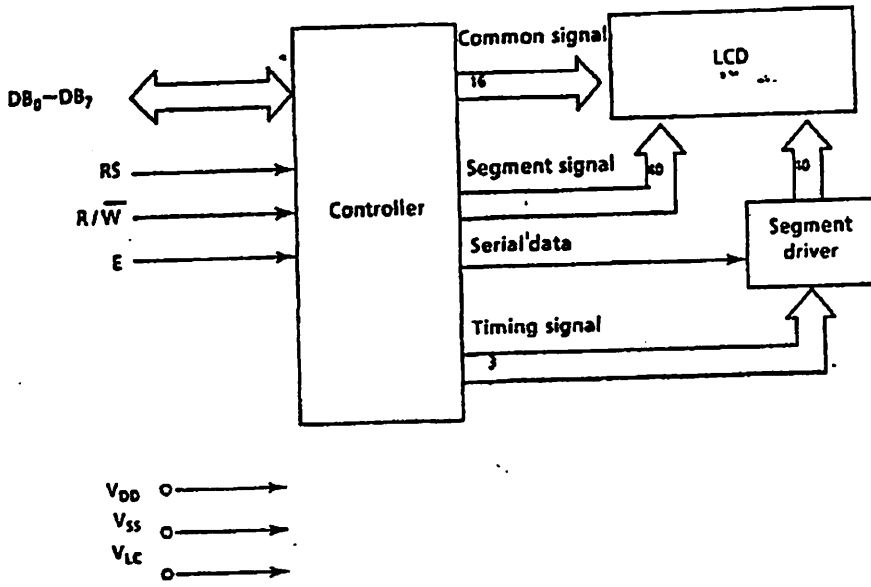
1.2 Features

- 16-character, two-line TN liquid crystal display of 5 x 7 dot matrix + cursor
- Duty ratio: 1/16
- Character generator ROM for 192 character types.
(character font: 5 x 7 dot matrix)
- Character generator RAM for eight character types (program write)
(character font: 5 x 7 dot matrix)
- 80 x 8 bit display data RAM (80 characters maximum)
- Interface with four-bit and eight-bit MPUs possible
- Display data RAM and character generator RAM readable from MPU
- Many instruction functions

**Display Clear, Cursor Home, Display ON/OFF, Cursor ON/OFF, Display
Character Blink, Cursor Shift, and Display Shift**

- Built-in oscillator circuit
- +5 V single power supply
- Built-in automatic reset circuit at power-on
- CMOS process
- Operating temperature range: 0°C to 50°C

1.4 Block Diagram



1.5 Absolute Maximum Ratings

$V_{SS} = 0V$

Item	Symbol	Standard	Unit	Remarks
Power supply voltage	V_{DD}	-0.3 to +7.0	V	
	V_{LC}	$V_{DD} - 13.5$ to $V_{DD} + 0.3$	V	
Input voltage	V_{in}	-0.3 to $V_{DD} + 0.3$	V	
Operating temperature	T_{opr}	0 to +50	°C	
Storage temperature	T_{stg}	-20 to +60	°C	At 50% RH

1.6 Electrical Characteristics

$V_{DD} = 5V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ C$ to $50^\circ C$

Item		Symbol	Conditions	Standard			Unit
				Min.	Typ.	Max.	
Input voltage	High	V_{IH1}		2.2	-	V_{DD}	V
	Low	V_{IL1}		0	-	0.6	V
Output voltage (TTL)	High	V_{OH1}	$-I_{OH} = 0.205$ mA	2.4	-	-	V
	Low	V_{OL1}	$I_{OL} = 1.2$ mA	-	-	0.4	V
Output voltage (CMOS)	High	V_{OH2}	$-I_{OH} = 0.04$ mA	$0.9V_{DD}$	-	-	V
	Low	V_{OL2}	$I_{OL} = 0.04$ mA	-	-	$0.1V_{DD}$	V
Power supply voltage		V_{DD}		4.75	5.00	5.25	V
		$-V_{LC}$	$V_{DD} = 5V$, $T_A = 25^\circ C$	-	0.25	-	V
Current consumption		I_{DD}		-	2.0	3.0	mA
		I_{LC}	$V_{LC} = 0.25V$	-	-	1.0	mA
Clock oscillation freq.		f_{osc}	Resistance oscillation	190	270	350	kHz

1.7 Optical Characteristics

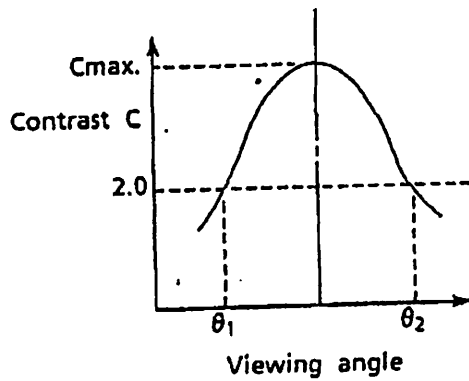
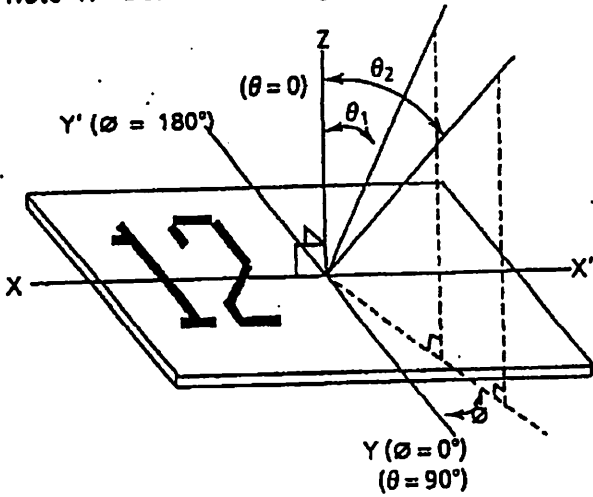
1.7.1 Optical characteristics

Maximum viewing angle: 6 o'clock ($\varnothing = 0^\circ$)
 $T_A = 25^\circ\text{C}$, $V_{opr} = 4.75\text{V}$

Item	Symbol	Conditions	Min.	Typ.	Max.	Remarks
Viewing angle	$\theta_2 - \theta_1$	$C \geq 2.0$, $\varnothing = 0^\circ$	35	-	-	See Notes 1 and 2.
Contrast	C	$\theta = 25^\circ$, $\varnothing = 0^\circ$	5	8	-	See Note 3.
Rise time	t_{on}	$\theta = 25^\circ$, $\varnothing = 0^\circ$	-	60 ms	70 ms	See Note 4.
Fall time	t_{off}	$\theta = 25^\circ$, $\varnothing = 0^\circ$	-	150 ms	170 ms	See Note 4.

Note 1: Definition of angles \varnothing and θ

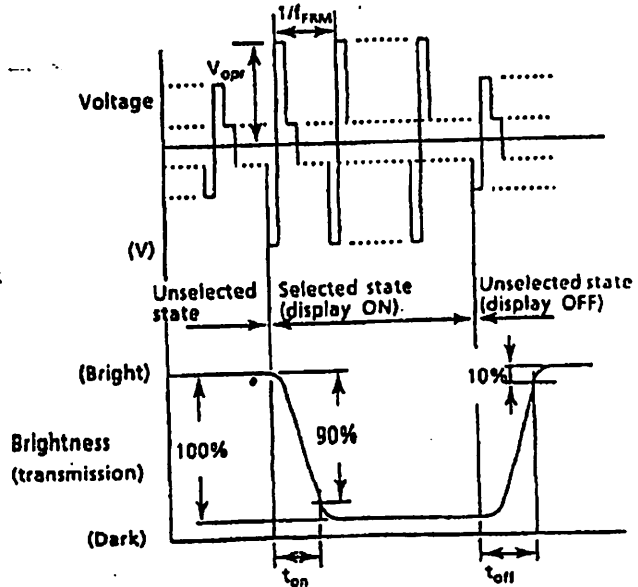
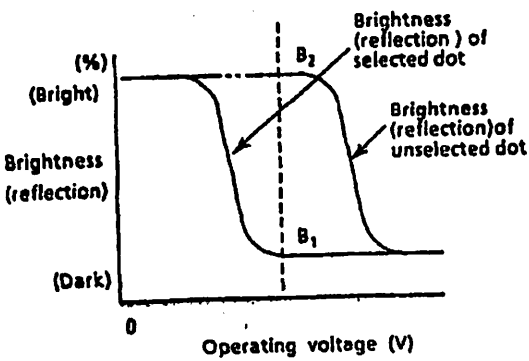
Note 2: Definition of viewing angles θ_1 and θ_2



Note 3: Definition of contrast C

Note 4: Definition of response time

$$C = \frac{\text{Brightness (reflection) of unselected dot (B2)}}{\text{Brightness (reflection) of selected dot (B1)}}$$



V_{opr} : Operating voltage (V)
 f_{FRM} : Frame frequency (Hz)
 t_{on} : Response time (rise)(ms)
 t_{off} : Response time (fall)(ms)

1.7.2 Recommended operating voltage

The viewing angle and screen contrast of the LCD panel can be varied by changing the liquid crystal operating voltage (V_{opr}), that is V_{LC} .

The optical characteristics is influenced by an ambient temperature. The recommended value of V_{opr} for an ambient temperatures are shown below.

Temperature (°C)	0	10	25	40	50
Voltage V_{opr} (V)	5.00	4.90	4.75	4.60	4.50

$$V_{opr} = V_{DD} - V_{LC}$$

2. OPERATING INSTRUCTIONS

2.1 Terminal Functions

Table 1 Terminal functions

Signal name	No. of terminals	I/O	Destination	Function
DB ₀ to DB ₃	4	I/O	MPU	Tristate bidirectional lower four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. If the interface data is 4 bits, the signals are not used.
DB ₄ to DB ₇	4	I/O	MPU	Tristate bidirectional upper four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. DB ₇ is also used as a busy flag.
E	1	Input	MPU	Operation start signal: The signal activates data write or read.
\overline{RW}	1	Input	MPU	Read (R) and Write (W) selection signals 0: Write 1: Read
RS	1	Input	MPU	Register selection signals 0: Instruction register (Write) Busy flag and address counter (Read) 1: Data register (Write and Read)
V _{LC}	1	-	Power supply	Power supply terminal for driving liquid crystal display: The screen contrast can be varied by changing V _{LC} .
V _{DD}	1	-	Power supply	+5V
V _{SS}	1	-	Power supply	Ground terminal: 0V

2.2 Basic Operations

2.2.1 Registers

The controller has two kinds of eight-bit registers: the instruction register (IR) and the data register (DR). They are selected by the register select (RS) signal as shown in Table 2.

The IR stores instruction codes such as Display Clear and Cursor Shift, and the address information of display data RAM (DD RAM) and character generator RAM (CG RAM). They can be written from the MPU, but cannot be read to the MPU.

The DR temporarily stores data to be written into DD RAM or CG RAM, or read from DD RAM or CG RAM. When data is written into DD RAM or CG RAM from the MPU, the data in the DR is automatically written into DD RAM or CG RAM by internal operation. However, when data is read from DD RAM or CG RAM, the necessary data address is written into the IR. The specified data is read out to the DR and then the MPU reads it from the DR. After the read operation, the next address is set and DD RAM or CG RAM data at the address is read into the DR for the next read operation.

Table 2 Register selection

RS	R/W	Operation
0	0	IR selection, IR write. Internal operation : Display clear
0	1	Busy flag (DB ₇) and address counter (DB ₀ to DB ₆) read
1	0	DR selection, DR write. Internal operation : DR to DD RAM or CG RAM
1	1	DR selection, DR read. Internal operation : DD RAM or CG RAM to DR

2.2.2 Busy flag (BF)

The flag indicates whether the module is ready to accept the next instruction. As shown in Table 2, the signal is output to DB₇ if RS = 0 and R/W = 1. If the value is 1, the module is working internally and the instruction cannot be accepted. If the value is 0, the next instruction can be written. Therefore, the flag status needs to be checked before executing an instruction. If an instruction is executed without checking the flag status, wait for more than the execution time shown by 2.4 Instruction Outline.

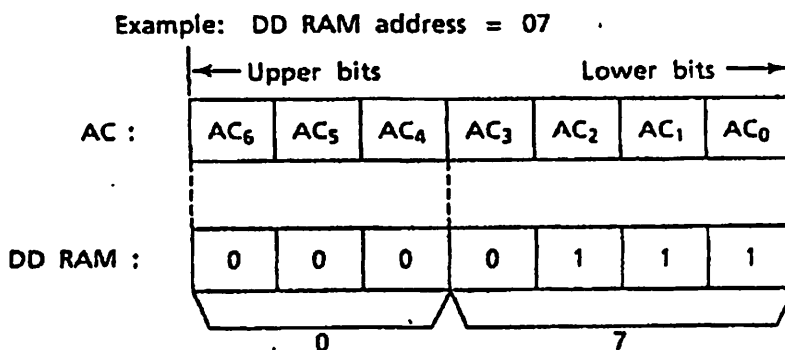
2.2.3 Address counter (AC)

The counter specifies an address when data is written into DD RAM or CG RAM and the data stored in DD RAM or CG RAM is read out. If an Address Set instruction (for DD RAM or CG RAM) is written in the IR, the address information is transferred from the IR to the AC. When display data is written into or read from DD RAM or CG RAM, the AC is automatically incremented or decremented by one according to the Entry Mode Set. The contents of the AC are output to DB₀ to DB₆ as shown in Table 2 if RS = 0 and $\overline{R/W} = 1$.

2.2.4 Display data RAM (DD RAM)

DD RAM has a capacity of up to 80 × 8 bits and stores display data of 80 eight-bit character codes. Some storage areas of DD RAM which are not used for display can be used as general data RAM.

A DD RAM address to be set in the AC is expressed in hexadecimal form as follows.



00H to 0FH of the DD RAM address is set in the line 1, and 40H to 4FH in the line 2.

Note : The addresses in the digit 16 of line 1 and the digit 1 of line 2 are not consecutive.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DD RAM address
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

If the display is shifted, DD RAM address 00H to 27H are displayed in line 1 and 40H to 67H in line 2. The following figures are examples of display shifts.

*Left shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	DD RAM address
Line 2	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	

*Right shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	DD RAM address
Line 2	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	

2.2.5 Character generator ROM (CG ROM)

Character generator ROM generates 192 types of 5 x 7 dot-matrix character patterns from eight-bit character codes.

Table 3 shows the correspondence between the CG ROM character codes and character patterns.

2.2.6 Character generator RAM (CG RAM)

CG RAM is used to create character patterns freely by programming. Eight types of character patterns can be written.

Table 4 shows the character patterns created from CG RAM addresses and data. To display a created character pattern, the character code in the left column of the table is written into DD RAM corresponding to the display position (digit). The areas not used for display are available as general data RAM.

Table 3 Correspondence between character codes and character patterns

Upper bit 4 bit Lower bit 4 bit	0	2	3	7	8	6	7	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	a	P	.	p	-	o	e		o	o
xxxx0001	(2)	!	l	A	O	a	q	u	7	*	△	ä	o
xxxx0010	(3)	"	2	B	R	b	r	r	4	w	x	è	è
xxxx0011	(4)	#	3	C	S	c	s	u	9	7	8	é	é
xxxx0100	(5)	*	4	D	T	d	t	u	I	A	†	W	o
xxxx0101	(6)	+	5	E	U	e	u	.	7	†	1	o	o
xxxx0110	(7)	o	6	F	V	f	v	9	n	2	3	o	z
xxxx0111	(8)	.	7	G	W	g	w	7	†	7	9	o	π
xxxx1000	(1)	c	8	H	X	h	x	4	o	*	u	7	π
xxxx1001	(2))	9	I	Y	i	y	9	7	1	u	"	u
xxxx1010	(3)	*	;	J	Z	j	z	3	o	1	v	i	π
xxxx1011	(4)	+	;	K	I	k	i	*	7	e	o	*	π
xxxx1100	(5)	.	<	L	*	l	l	†	9	o	o	o	π
xxxx1101	(6)	...	=	N	1	n	1	3	z	o	o	o	+
xxxx1110	(7)	;	>	N	^	n	^	3	3	9	o	o	
xxxx1111	(8)	/	?	O	_	o	_	+	u	u	o	o	π

Table 4 Relationships between CG RAM addresses and character codes (DD RAM) and character patterns (CG RAM data)

Character code (DD RAM data)		CG RAM address				Character pattern (CG RAM data)										
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
← Upper bit Lower bit →		← Upper bit		Lower bit →		← Upper bit		Lower bit →								
0 0 0 0 * 0 0 0		0 0 0		0 0 0		0 0 0		* * *								Example of character pattern (R)
0 0 0 0 * 0 0 1		0 0 1		0 1 1		0 1 1		* * *								Example of character pattern (Y)
0 0 0 0 * 1 1 1		1 1 1		1 0 0		1 0 0		* * *								

Notes: • In CG RAM data, 1 corresponds to Selection and 0 to Non-selection on the display.

- Character code bits 0 to 2 and CG RAM address bits 3 to 5 correspond with each other (three bits, eight types).
- CG RAM address bits 0 to 2 specify a line position for a character pattern. Line 8 of a character pattern is the cursor position where the logical sum of the cursor and CG RAM data is displayed. Set the data of line 8 to 0 to display the cursor. If the data is changed to 1, one bit lights, regardless of the cursor.

The character pattern column positions correspond to CG RAM data bits 0 to 4 and bit 4 comes to the left end. CG RAM data bits 5 to 7 are not displayed but can be used as general data RAM.

When reading a character pattern from CG RAM, set to 0 all of character code bits 4 to 7. Bits 0 to 2 determine which pattern will be read out. Since bit 3 is not valid, 00H and 08H select the same character.

2.3 Timing Characteristics

2.3.1 Write timing characteristics

$V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ C$ to $50^\circ C$

Item	Symbol	Standard		Unit	
		Min.	Max.		
Enable cycle time	t_{CYCE}	1000	-	ns	
Enable pulse width	High level	PW_{EH}	450	-	ns
Enable rise and fall time	t_{Er}, t_{Ef}	-	25	ns	
Setup time	RS, $\overline{RW} - E$	t_{AS}	140	-	ns
Address hold time	t_{AH}	10	-	ns	
Data setup time	t_{DSW}	195	-	ns	
Data hold time	t_H	10	-	ns	

Write operation

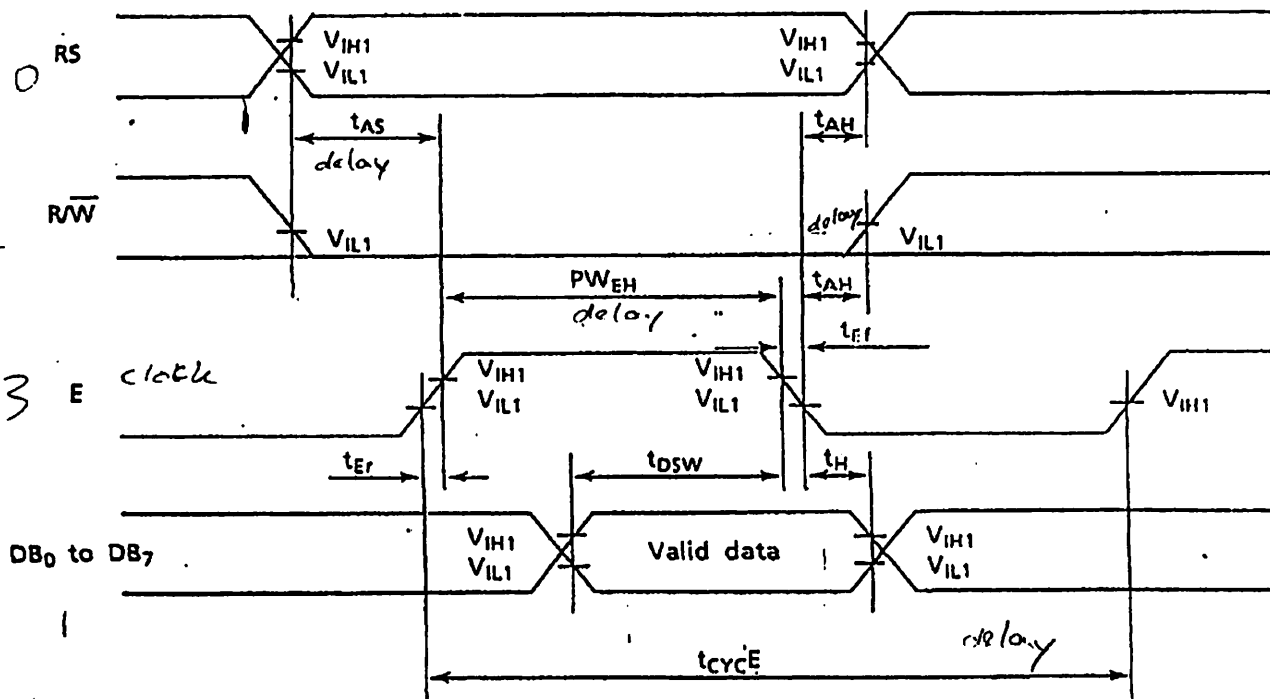


Figure 3 Data write from MPU to module

2.3.2 Read timing characteristics

$V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$; $T_A = 0^\circ C$ to $50^\circ C$

Item	Symbol	Standard		Unit
		Min.	Max.	
Enable cycle time	t_{CYCE}	1000	-	ns
Enable pulse width	High level	PW_{EH}	-	ns
Enable rise and fall time	t_{Er}, t_{Ef}	-	25	ns
Setup time	t_{AS}	140	-	ns
Address hold time	t_{AH}	10	-	ns
Data delay time	t_{DDR}	-	320	ns
Data hold time	t_{H}	20	-	ns

Read operation

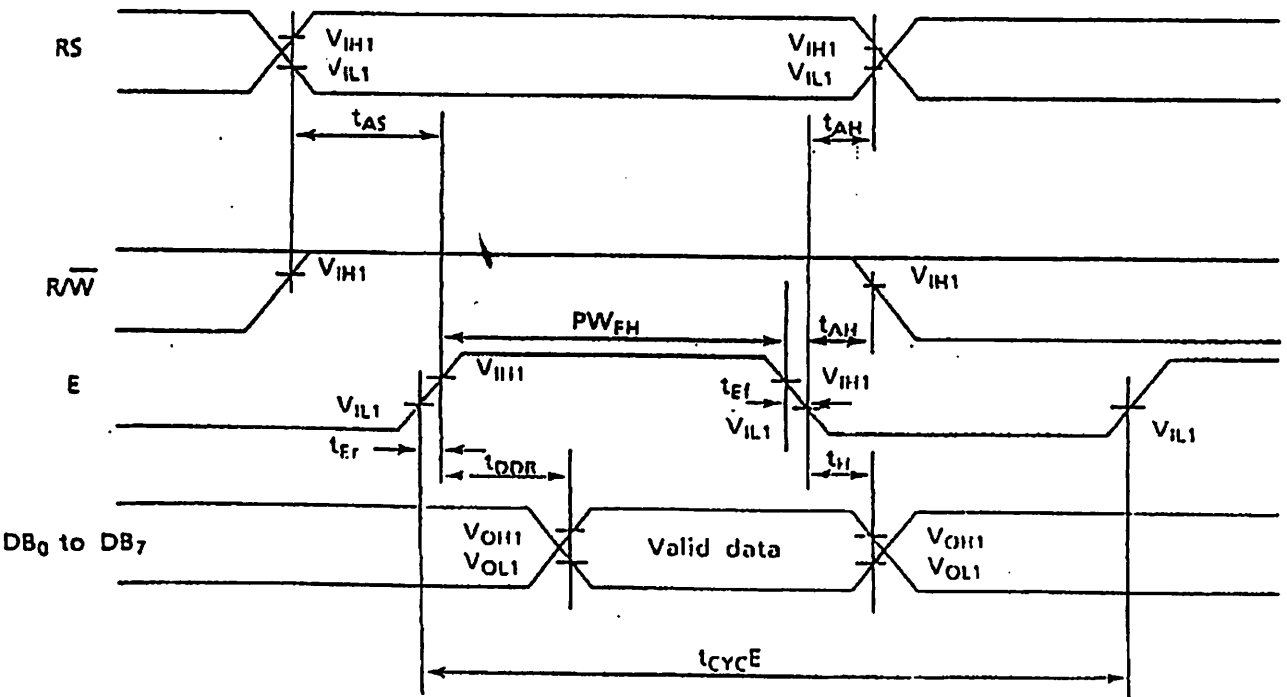


Figure 4 Data read from module to MPU

Instruction Outline

Table 5 List of instructions

Instruction	Code											Function	Execution time
	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀			
1) Display clear ✓	0	0	0	0	0	0	0	0	0	0	1	Clears all display and returns cursor to home position (address 0)	1.64 ms
2) Cursor Home ✓	0	0	0	0	0	0	0	0	0	1	0	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
3) Entry Mode Set ✓	0	0	0	0	0	0	0	0	1	RD	S	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 μs
4) Display ON / OFF control	0	0	0	0	0	0	0	1	B	C	D	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 μs
5) Cursor/Display Shift ✓	0	0	0	0	0	1	S/C	R/L	0	0	0	Moves cursor and shifts display without changing DD RAM contents	40 μs
6) Function Set ✓	0	0	0	0	1	DL	1	0	0	0	0	Sets interface data length (DL)	40 μs
7) CG RAM Address Set	0	0	0	1	Acc							Sets CG RAM address to start transmitting or receiving CG RAM data	40 μs
8) DD RAM Address Set	0	0	1	ADD							Sets DD RAM address to start transmitting or receiving DD RAM data	40 μs	
9) BF/Address Read	0	1	BF	AC							Reads BF indicating module in internal operation and AC contents (used for both CG RAM and DD RAM)	0 μs	
10) Data Write to CG RAM or DD RAM	1	0	Write Data									Writes data into DD RAM or CG RAM	40 μs
11) Data Read from CG RAM or DD RAM	1	1	Read Data									Reads data from DD RAM or CG RAM	40 μs

0 : Invalid bit

Acc : CG RAM address

ADD : DD RAM address

I/D = 1 : Increment

I/D = 0 : Decrement

S = 1 : Display shift

S = 0 : No display shift

D = 1 : Display ON

D = 0 : Display OFF

C = 1 : Cursor ON

C = 0 : Cursor OFF

B = 1 : Blink ON

B = 0 : Blink OFF

S/C = 1 : Display

shift

S/C = 0 : Cursor movement

R/L = 1 : Right shift

R/L = 0 : Left shift

DL = 1 : 8 bits

DL = 0 : 4 bits

BF = 1 : Internal operation in progress

BF = 0 : Instruction can be accepted

2.5 Instruction Details

(1) Display Clear

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	0	0	0	1

Display Clear clears all display and returns cursor to home position (address 0).

Space code 20 (hexadecimal) is written into all the addresses of DD RAM, and DD RAM address 0 is set to the AC. If shifted, the display returns to the original position. After execution of the Display Clear instruction, the entry mode is incremented.

Note: When executing the Display Clear instruction, follow the restrictions listed in Table 6.

(2) Cursor Home

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	0	0	1	*

* : Invalid bit

Cursor Home returns cursor to home position (address 0).

DD RAM address 0 is set to the AC. The cursor returns to the home position. If shifted, the display returns to the original position. The DD RAM contents do not change. If the cursor or blinking is ON, it returns to the left side.

Note: When executing the Cursor Home instruction, follow the restrictions listed in Table 6.

Table 6 Restrictions on execution of Display Clear and Cursor Home instructions

Conditions of use	Restrictions
When executing the Display Clear or Cursor Home instruction when the display is shifted (after execution of Display Shift instruction)	The Cursor Home instruction should be executed again immediately after the Display Clear or Cursor Home instruction is executed. Do not leave an interval of a multiple of $400/f_{osc}$ * second after the first execution. Example: 1.5 ms, 3 ms, 4.5 ms for $f_{osc} = 270$ kHz * f_{osc} : Oscillation frequency
When 23 ₁₁ , 77 ₁₁ , 63 ₁₁ , or 67 ₁₁ is used as a DD RAM address to execute Cursor Home instruction	Before executing the Cursor Home instruction, the data of the four DD RAM addresses given at the left should be read and saved. After execution, write the data again in DD RAM. (This restriction is necessary to prevent the contents of the DD RAM addresses from being destroyed after the Cursor Home instruction has been executed.)

(3) Entry Mode Set

	RS	R/W	DB ₇						DB ₀	
Code	0	0	0	0	0	0	0	1	I/D	S

Entry Mode Set sets the direction of cursor movement and whether display will be shifted.

I/D : The DD RAM address is incremented or decremented by one when a character code is written into or read from DD RAM. This is also true for writing into or reading from CG RAM.

When I/D = 1, the address is incremented by one and the cursor or blink moves to the right.

When I/D = 0, the address is decremented by one and the cursor or blink moves to the left.

S : If S = 1, the entire display is shifted either to the right or left for writing into DD RAM. The cursor position does not change, only the display moves. There is no display shift for reading from DD RAM.

When S = 1 and I/D = 1, the display shifts to the left.

When S = 1 and I/D = 0, the display shifts to the right.

If S = 0, the display does not shift.

(4) Display ON/OFF Control

	RS	R/W	DB ₇						DB ₀	
Code	0	0	0	0	/0	0	1	D	C	B

Display ON/OFF Control turns the total display and the cursor ON and OFF, and makes the cursor position start blinking. Cursor ON/OFF and blinking is done at the column indicated by the specified DD RAM address by the AC.

D : When D = 1, the display is turned ON.

When D = 0, the display is turned OFF.

If D = 0 is used, display data remains in DD RAM. Change 0 to 1 to display data.

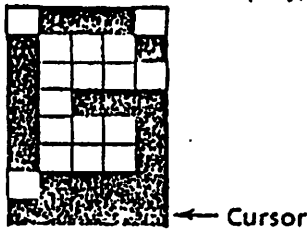
C : When $C = 1$, the cursor is displayed.
 When $C = 0$, the cursor is not displayed.

The cursor is displayed in the dot line below the 5 x 7 dot-matrix character fonts. If the cursor is OFF, display data is written into DD RAM in the order specified by I/D.

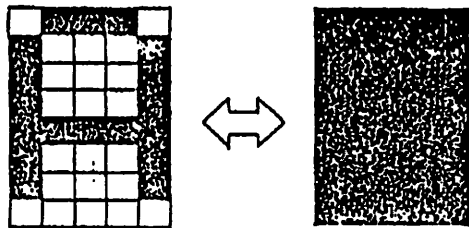
B : When $B = 1$, the character at the cursor position starts blinking.
 When $B = 0$, it does not blink.

For blinking, all-black dots and the character are switched about every 0.4 seconds. The cursor and blinking can be set at the same time.

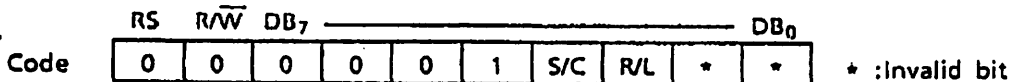
Example: $C = 1$ (cursor display)



$B = 1$ (blinking)



(5) Cursor/Display Shift



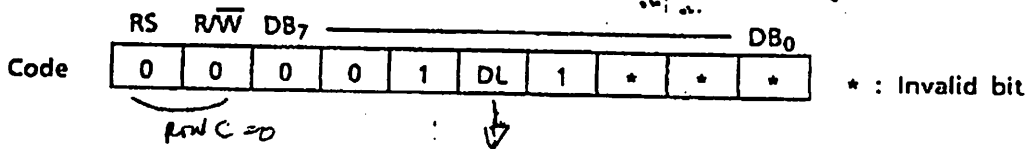
Cursor/Display Shift moves the cursor and shifts the display without changing the DD RAM contents.

The cursor position and the AC contents match. This instruction is available for display correction and retrieval because the cursor position or display can be shifted without writing or reading display data. Since the DD RAM capacity is 40-character and two lines, the cursor is shifted from digit 40 of line 1 to digit 1 of line 2. Displays of lines 1 and 2 are shifted at the same time. Therefore, the display pattern of line 2 is not shifted to line 1.

S/C	R/L	Operation
0	0	The cursor position is shifted to the left (the AC decrements one).
0	1	The cursor position is shifted to the right (the AC increments one).
1	0	The entire display is shifted to the left with the cursor.
1	1	The entire display is shifted to the right with the cursor.

Note: If only display shift is done, the AC contents do not change.

(6) Function Set



Function Set sets the interface data length.

DL : Interface data length

When DL = 1, the data length is set at eight bits (DB₇ to DB₀).

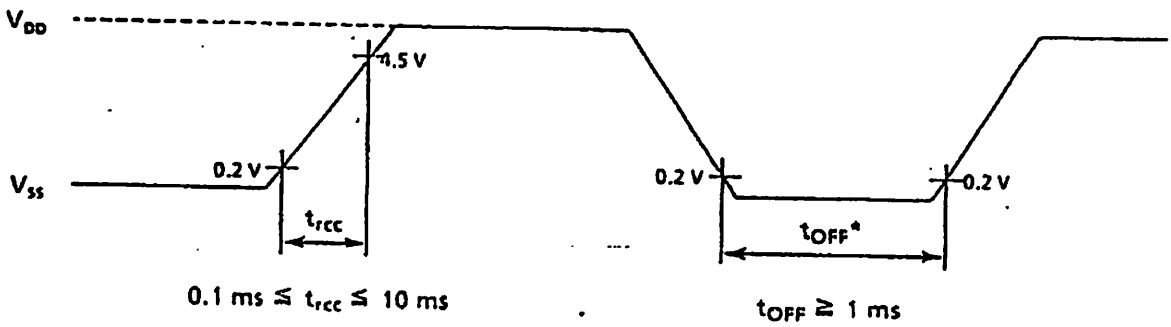
When DL = 0, the data length is set at four bits (DB₇ to DB₄).

The upper four bits are transferred first, then the lower four bits follow.

The Function Set instruction must be executed prior to all other instructions except for Busy Flag/Address Read. If another instruction is executed first, no function instruction except changing the interface data length can be executed.

Remarks: Initialization

The system is automatically initialized at power-on if the following power supply conditions are satisfied.



* t_{off} : Time when power supply is OFF if cut instantaneously or turned ON and OFF repeatedly

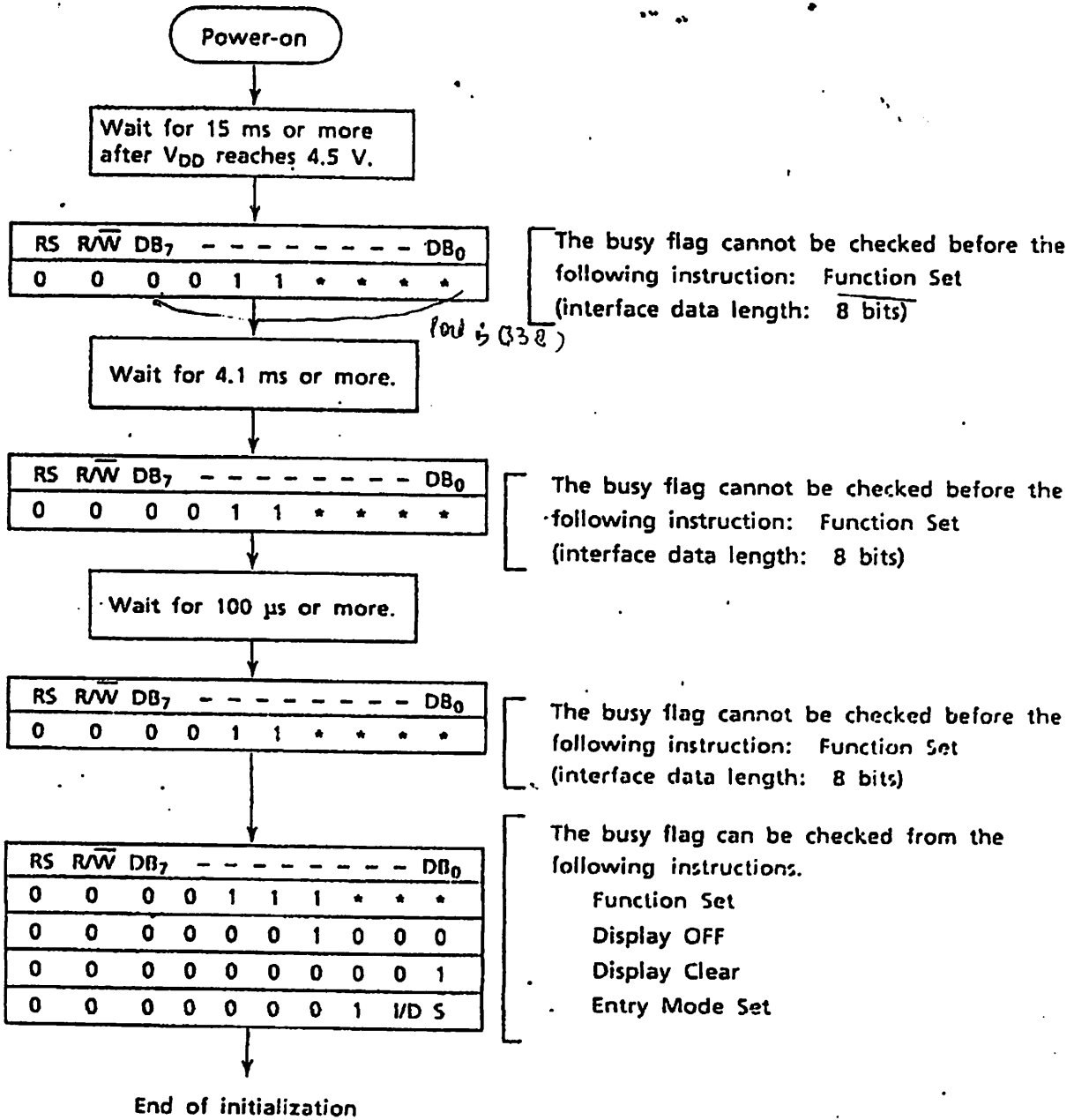
The following instructions are executed for initialization.

- 5 x 7 dot-matrix character font: 1/8 duty
- Display clear
- Function Set DL = 1: Interface data length: 8 bits
- Display ON/OFF Control D = 0: Display OFF
 C = 0: Cursor OFF
 B = 0: Blink OFF
- Entry mode I/O = 1: Increment
 S = 0: No display shift

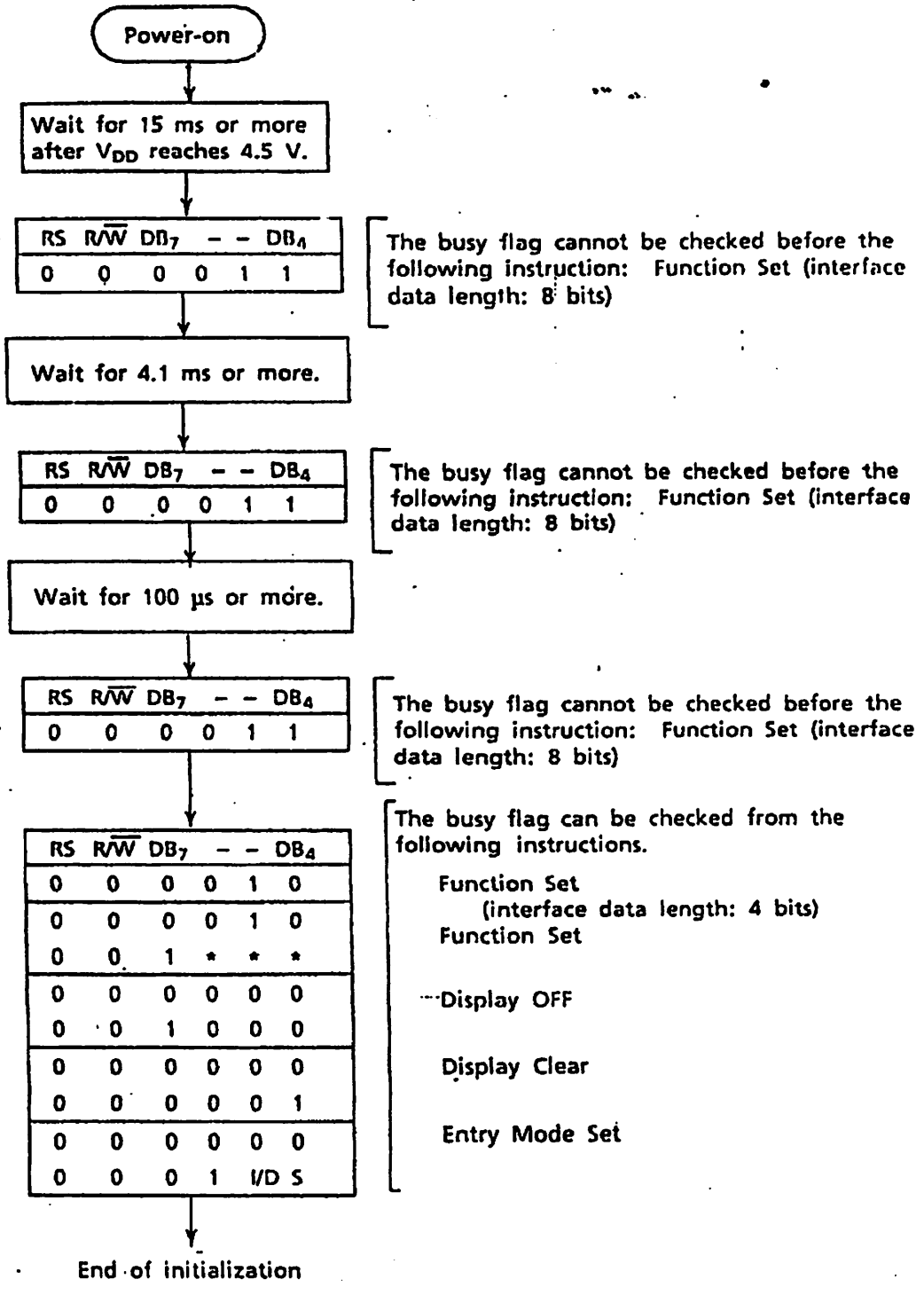
Since the condition is not suitable for the M1632, further function setting is necessary.

If automatic initialization is not executed because the above power supply conditions are not satisfied, use the instruction from next page on.

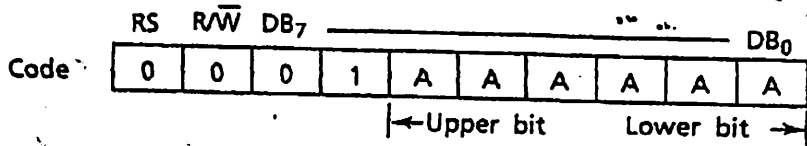
(a) Interface data length : Eight bits



(b) Interface data length: Four bits

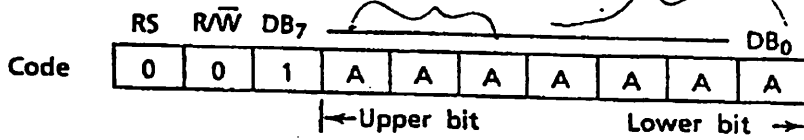


(7) CG RAM Address Set



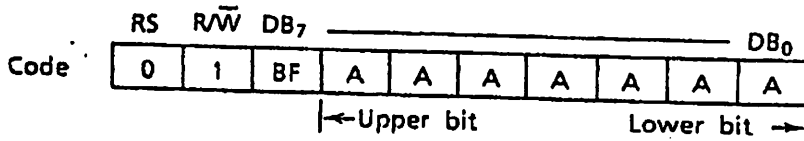
CG RAM addresses expressed as binary AAAAAA are set to the AC. Then data in CG RAM is written from or read to the MPU.

(8) DD RAM Address Set



DD RAM addresses expressed as binary AAAAAAA are set to the AC. Then data in DD RAM is written from or read to the MPU. The addresses used for display in line 1 (AAAAAAA) are 00H to 27H and those for line 2 (AAAAAAA) are 40H to 67H.

(9) Busy Flag/Address Read



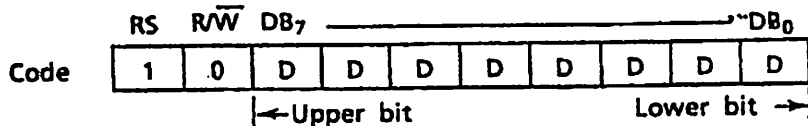
The BF signal is read out, indicating that the module is working internally because of the previous instruction.

When BF = 1, the module is working internally and the next instruction cannot be accepted until the BF value becomes 0.

When BF = 0, the next instruction can be accepted.

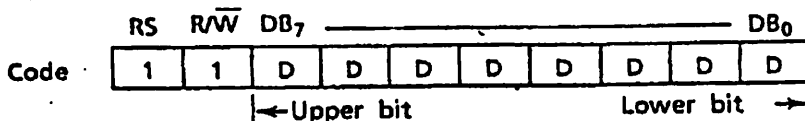
Therefore, make sure that BF = 0 before writing the next instruction. The AC values of binary AAAAAAA are read out at the same time as reading the busy flag. The AC addresses are used for both CG RAM and DD RAM but the address set before execution of the instruction determines which address is to be used.

(10) Data Write to CG RAM or DD RAM



Binary eight-bit data DDDDDDDD is written into CG RAM or DD RAM. The CG RAM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) before this instruction selects either RAM. After the write operation, the address and display shift are determined by the entry mode setting.

(11) Data Read from CG RAM or DD RAM



Binary eight-bit data DDDDDDDD is read from CG RAM or DD RAM. The CG RAM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) before this instruction selects either RAM. In addition, either instruction (7) or (8) must be executed immediately before this instruction. If no address set instruction is executed before a read instruction, the first data read becomes invalid. If read instructions are executed consecutively, data is normally read from the second time. However, if the cursor is shifted by the Cursor Shift instruction when reading DD RAM, there is no need to execute an address set instruction because the Cursor Shift instruction does this.

After the read operation, the address is automatically incremented or decremented by one according to the entry mode, but the display is not shifted.

Note : The AC is automatically incremented or decremented by one according to the entry mode after a write instruction is executed to write data in CG RAM or DD RAM. However, the data of the RAM selected by the AC are not read out even if a read instruction is executed immediately afterwards.

Correct data is read out under the following conditions.

- An address set instruction is executed immediately before readout.
- For DD RAM, the Cursor Shift instruction is executed immediately before readout.
- The second, or later, instruction is executed in consecutive execution of read instructions.

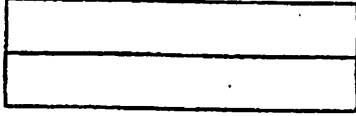
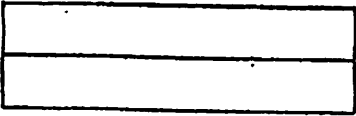
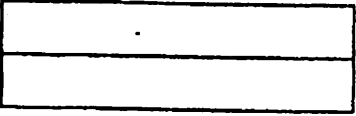
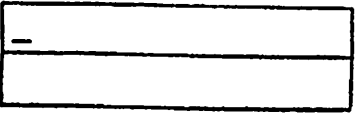
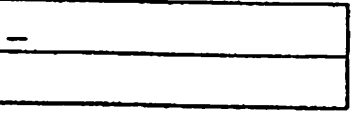
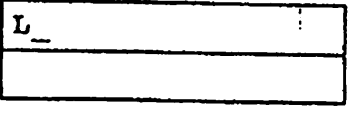
2.6 Examples of Instruction Use

(1) Interface data length: Eight bits

No.	Instruction	Display	Operation										
1	<p>Power-on</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>/</td> <td>/</td> <td colspan="3">/</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	/	/	/				The built-in reset circuit initializes the module.
RS	R/W	DB ₇	—	DB ₀									
/	/	/											
2	<p>Function Set ✓</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 1 1</td> <td>*</td> <td>*</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 1 1 1	*	*		The interface data length is set to 8 bits. The character format becomes 5 x 7 dot-matrix at 1/16 duty cycle.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 1 1 1	*	*									
3	<p>Display ON/OFF Control</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 1 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 1 1 1 0				The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 0 0 1 1 1 0											
4	<p>Entry Mode Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 0 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 0 1 1 0				The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 0 0 0 1 1 0											
5	<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 1 1 0 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 1 1 0 0				L is written. The AC is incremented by one and the cursor shifts to the right.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 1 0 0 1 1 0 0											
6	<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 0 0 1 1</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 0 0 1 1				C is written.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 1 0 0 0 0 1 1											
7													
8	<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 0 1 1 0 0 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 0 1 1 0 0 1 0				2 is written in digit 16. Cursor disappears.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 0 1 1 0 0 1 0											

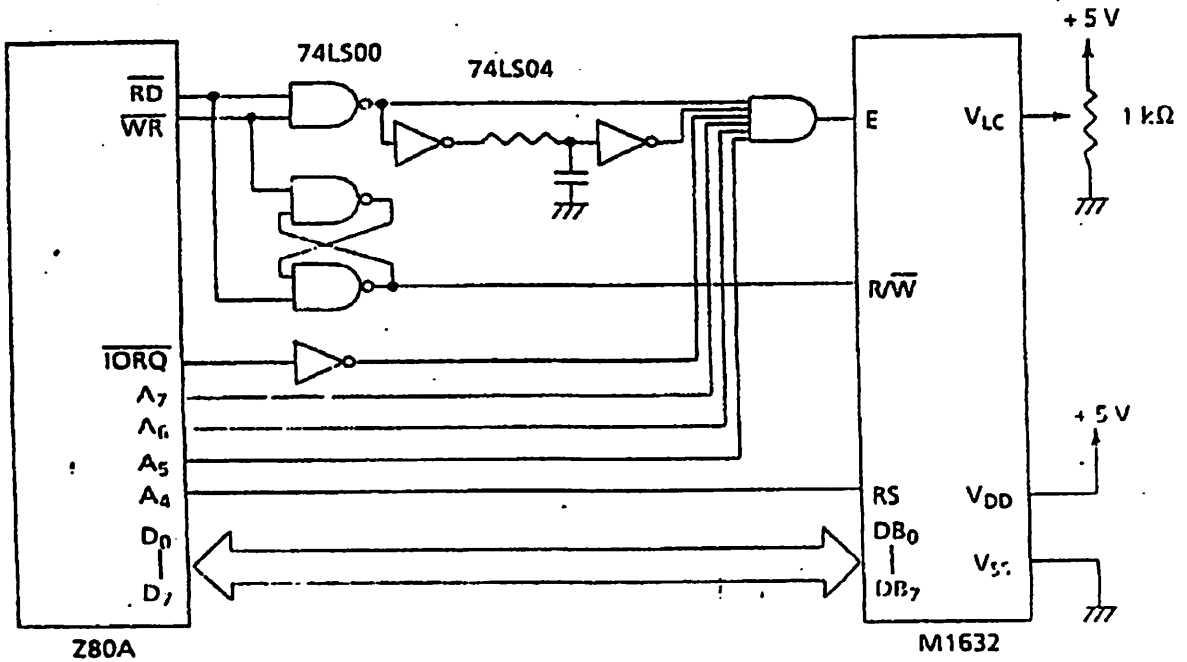
No.	Instruction	Display	Operation																				
9	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	1	1	0	0	0	0	0	0	LCD MODULE M1632 _	The DD RAM address is set so that the cursor appears at digit 1 of line 2.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	1	1	0	0	0	0	0	0														
10	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	0	1	1	0	0	0	1	LCD MODULE M1632 1_	1 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	0	1	1	0	0	0	1														
11	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	0	1	1	0	1	1	0	LCD MODULE M1632 16_	6 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	0	1	1	0	1	1	0														
12																							
13	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	1	0	1	0	0	1	1	LCD MODULE M1632 16DIGITS, 2LINES	5 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	1	0	1	0	0	1	1														
14	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	1	0	0	0	0	0	0	0	LCD MODULE M1632 16DIGITS, 2LINES	The cursor returns to the home position.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	1	0	0	0	0	0	0	0														
15	Display clear <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	0	0	0	0	0	0	0	1	_	All the display disappears and the cursor remains at the home position.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	0	0	0	0	0	0	0	1														
16																							

(2) Interface data length: Four bits

No.	Instruction	Display	Operation									
1	Power-on <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	/	/	/		The built-in reset circuit initializes the module.			
RS	R/W	DB ₇ — DB ₄										
/	/	/										
2	Function Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	/	/	/		Four-bit operation mode is set. *Eight-bit operation mode is set by initialization, and the instruction is executed only once.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 1 0										
/	/	/										
3	Function Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 * * *</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	0	0	1 * * *		The 4-bit operation mode, 1/16 duty cycle, and 5 x 7 dot-matrix character format are selected. Then 4-bit operation mode starts.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 1 0										
0	0	1 * * *										
4	Display ON/OFF Control <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	1 1 1 0		The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 0 0										
0	0	1 1 1 0										
5	Entry Mode Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	0 1 1 0		The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 0 0										
0	0	0 1 1 0										
6	Write to CG RAM or DD RAM. <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1 1 0 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	1	0	0 1 0 0	1	0	1 1 0 0		L is written. the AC is incremented by one and the cursor shifts to the right.
RS	R/W	DB ₇ — DB ₄										
1	0	0 1 0 0										
1	0	1 1 0 0										

2.7 MPU Connection Diagrams

2.7.1 Z80A



2.7.2 Z80A and 8255A

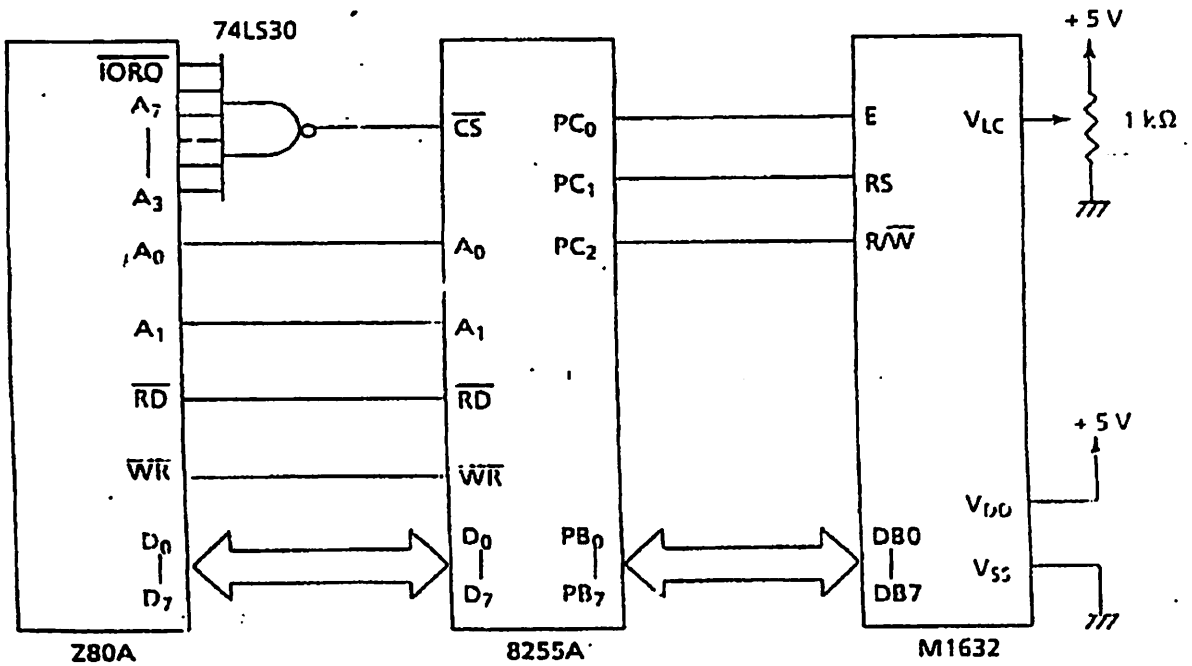


Table of Contents

ISD2560/75/90/120 Products

Single-Chip Voice Record/Playback Devices
60-, 75-, 90-, and 120-Second Durations

DETAILED DESCRIPTION	1
Speech/Sound Quality	1
Duration	1
EEPROM Storage	1
Microcontroller Interface	1
Programming	1
PIN DESCRIPTIONS	2
Voltage Inputs (V_{CCA} , V_{CCD})	2
Ground Inputs (V_{SSA} , V_{SSD})	2
Power Down Input (PD)	2
Chip Enable Input (\overline{CE})	2
Playback/Record Input (P/R)	3
End-Of-Message / RUN Output (\overline{EOM})	3
Overflow Output (\overline{OVF})	3
Microphone Input (MIC)	3
Microphone Reference Input (MIC REF)	3
Automatic Gain Control Input (AGC)	3
Analog Output (ANA OUT)	3
Analog Input (ANA IN)	4
External Clock Input (XCLK)	4
Speaker Outputs (SP+/SP-)	4
Auxiliary Input (AUX IN)	4
Address/Mode Inputs (Ax/Mx)	5
OPERATIONAL MODES	5
OPERATIONAL MODES DESCRIPTION	6
M0 — Message Cueing	6
M1 — Delete \overline{EOM} Markers	6
M2 — Unused	6
M3 — Message Looping	6
M4 — Consecutive Addressing	6
M5 — \overline{CE} -Level Activated	6
M6 — Push-Button Mode	6
\overline{CE} Pin (START/PAUSE)	7
PD Pin (STOP/RESET)	7
\overline{EOM} Pin (RUN)	7
Good Audio Design Practices	8
ISD1000A COMPATIBILITY	8
Addressing	8
Overflow	8

SD2560/75/90/120 Products

Push-Button Mode	8
Looping Mode	8
TIMING DIAGRAMS	9
TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (PACKAGED PARTS)	13
TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (DIE)	17
EXPLANATION	19
PUSH-BUTTON TIMING DIAGRAMS	22
DEVICE PHYSICAL DIMENSIONS	23
ORDERING INFORMATION	30

FIGURES, CHARTS, AND TABLES IN THE ISD2560/75/90/120 PRODUCTS DATASHEET

Figure 1:	ISD2560/75/90/120 Device Pinouts	2
Figure 2:	Record	9
Figure 3:	Playback	9
Figure 4:	ISD2560/75/90/120 Application Example—Design Schematic	18
Figure 5:	ISD2560/75/90/120 Application Example—Microcontroller/ISD2500 Interface	20
Figure 6:	ISD2500 Application Example—Push-Button	20
Figure 7:	Push-Button Mode Record	22
Figure 8:	Push-Button Mode Playback	22
Figure 9:	28-Lead 8x13.4mm Plastic Thin Small Outline Package (TSOP) Type I (E)	23
Figure 10:	28-Lead 0.600-Inch Plastic Dual Inline Package (PDIP) (P)	24
Figure 11:	32-Lead 8x20mm Plastic Thin Small Outline Package (TSOP) Type I (T)	26
Figure 12:	ISD2560/75/90/120 Products <i>Current</i> Bonding Physical Layout (Unpackaged Die)	27
Figure 13:	ISD2560/75/90/120 Products <i>Future</i> Bonding Physical Layout (Unpackaged Die)	29
Chart 1:	Record Mode Operating Current (I_{CC})	13
Chart 2:	Total Harmonic Distortion	13
Chart 3:	Standby Current (I_{SB})	13
Chart 4:	Oscillator Stability	13
Chart 5:	Record Mode Operating Current (I_{CC})	17
Chart 6:	Total Harmonic Distortion	17
Chart 7:	Standby Current (I_{SB})	17
Chart 8:	Oscillator Stability	17
Table 1:	External Clock Sample Rates	4
Table 2:	Operational Modes Table	5
Table 3:	Alternate Functionality in Pins	6
Table 4:	Absolute Maximum Ratings (Packaged Parts)	10
Table 5:	Operating Conditions (Packaged Parts)	10
Table 6:	DC Parameters (Packaged Parts)	10
Table 7:	AC Parameters (Packaged Parts)	11
Table 8:	Absolute Maximum Ratings (Die)	14
Table 9:	Operating Conditions (Die)	14
Table 10:	DC Parameters (Die)	14
Table 11:	AC Parameters (Die)	15
Table 12:	Application Example—Basic Device Control	18
Table 13:	Application Example—Passive Component Functions	19
Table 14:	Application Example—Push-Button Control	21
Table 15:	Application Example—Passive Component Functions	21
Table 16:	Push-Button Parameters	21
Table 17:	Plastic Thin Small Outline Package (TSOP) Type I (E) Dimensions	23
Table 18:	Plastic Dual Inline Package (PDIP) (P) Dimensions	24
Table 19:	Plastic Thin Small Outline Package (TSOP) Type I (T) Dimensions	26
Table 20:	ISD2560/75/90/120 Products <i>Current</i> PIN/PAD Designations	28
Table 21:	ISD2560/75/90/120 Products <i>Future</i> PIN/PAD Designations	30

ISD2560/75/90/120 Products

Single-Chip Voice Record/Playback Devices

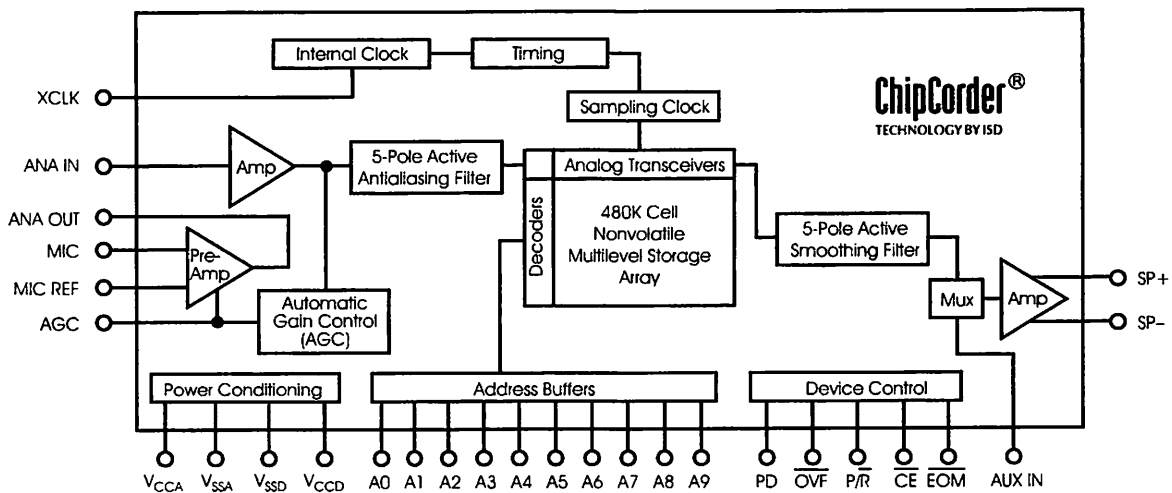
60-, 75-, 90-, and 120-Second Durations

GENERAL DESCRIPTION

Information Storage Devices' ISD2500 ChipCorder® Series provides high-quality, single-chip record/playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multilevel storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved.

Recordings are stored in on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through ISD's patented multilevel storage technology. Voice and audio signals are stored directly into memory in their natural form, providing high-quality, solid-state voice reproduction.

Figure i: ISD2560/75/90/120 Device Block Diagram



FEATURES

- Easy-to-use single-chip voice record/playback solution
 - High-quality, natural voice/audio reproduction
 - Manual switch or microcontroller compatible playback can be edge- or level-activated
 - Single-chip durations of 60, 75, 90, and 120 seconds
 - Directly cascadable for longer durations
 - Automatic Power-Down (Push-Button Mode)
 - Standby current 1 μ A (typical)
 - Zero-power message storage
 - Eliminates battery backup circuits
 - Fully addressable to handle multiple messages
 - 100-year message retention (typical)
 - 100,000 record cycles (typical)
 - On-chip clock source
 - Programmer support for play-only applications
 - Single +5 volt power supply
 - Available in die form, DIP, SOIC, and TSOP packaging
-

Table i: ISD2560/75/90/120 Product Summary

Part Number	Duration (Seconds)	Input Sample Rate (KHz)	Typical Filter Pass Band (KHz)
ISD2560	60	8.0	3.4
ISD2575	75	6.4	2.7
ISD2590	90	5.3	2.3
ISD25120	120	4.0	1.7

DETAILED DESCRIPTION

SPEECH/SOUND QUALITY

The ISD2500 series includes devices offered at 4.0, 5.3, 6.4, and 8.0 KHz sampling frequencies, allowing the user a choice of speech quality options. Increasing the duration within a product series decreases the sampling frequency and bandwidth, which affects sound quality. Please refer to the ISD2560/75/90/120 Product Summary table on page *ii* to compare filter pass band and product durations.

The speech samples are stored directly into on-chip nonvolatile memory without the digitization and compression associated with other solutions. Direct analog storage provides a very true, natural sounding reproduction of voice, music, tones, and sound effects not available with most solid-state digital solutions.

DURATION

To meet end system requirements, the ISD2500 series offers single-chip solutions at 60, 75, 90, and 120 seconds. Parts may also be cascaded together for longer durations.

EEPROM STORAGE

One of the benefits of ISD's ChipCorder technology is the use of on-chip nonvolatile memory, providing zero-power message storage. The message is retained for up to 100 years typically without power. In addition, the device can be re-recorded typically over 100,000 times.

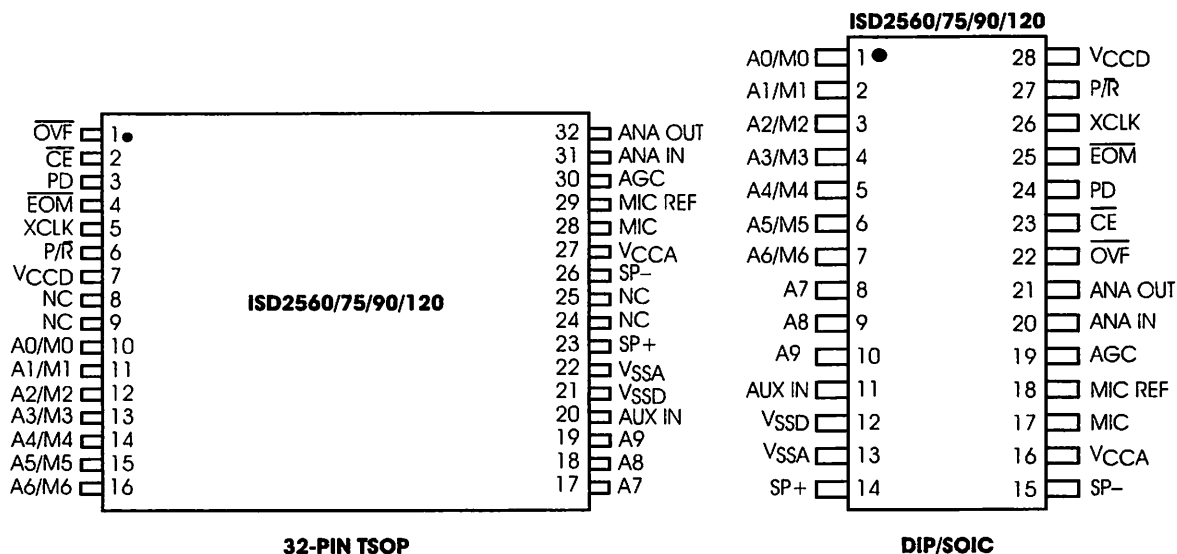
MICROCONTROLLER INTERFACE

In addition to its simplicity and ease of use, the ISD2500 series includes all the interfaces necessary for microcontroller-driven applications. The address and control lines can be interfaced to a microcontroller and manipulated to perform a variety of tasks, including message assembly, message concatenation, predefined fixed message segmentation, and message management.

PROGRAMMING

The ISD2500 series is also ideal for playback-only applications, where single or multiple messages are referenced through buttons, switches, or a microcontroller. Once the desired message configuration is created, duplicates can easily be generated via an ISD programmer.

Figure 1: ISD2560/75/90/120 Device Pinouts



PIN DESCRIPTIONS

VOLTAGE INPUTS (V_{CCA} , V_{CCD})

To minimize noise, the analog and digital circuits in the ISD2500 series devices use separate power busses. These voltage busses are brought out to separate pins and should be tied together as close to the supply as possible. In addition, these supplies should be decoupled as close to the package as possible.

GROUND INPUTS (V_{SSA} , V_{SSD})

The ISD2500 series of devices utilizes separate analog and digital ground busses. These pins should be connected separately through a low-impedance path to power supply ground.

POWER DOWN INPUT (PD)

When not recording or playing back, the PD pin should be pulled HIGH to place the part in a very low power mode (see I_{SB} specification). When overflow (\overline{OVF}) pulses LOW for an overflow condition, PD should be brought HIGH to reset the address pointer back to the beginning of the record/playback space. The PD pin has additional functionality in the M6 (Push-Button) Operational Mode described later in the Operational Mode section.

CHIP ENABLE INPUT (\overline{CE})

The \overline{CE} pin is taken LOW to enable all playback and record operations. The address inputs and playback/record input (P/R) are latched by the falling edge of \overline{CE} . \overline{CE} has additional functionality in the M6 (Push-Button) Operational Mode described later in the Operational Mode section.

PLAYBACK/RECORD INPUT ($\overline{P/R}$)

The $\overline{P/R}$ input is latched by the falling edge of the \overline{CE} pin. A HIGH level selects a playback cycle while a LOW level selects a record cycle. For a record cycle, the address inputs provide the starting address and recording continues until PD or \overline{CE} is pulled HIGH or an overflow is detected (i.e. the chip is full). When a record cycle is terminated by pulling PD or \overline{CE} HIGH, an End-Of-Message (EOM) marker is stored at the current address in memory. For a playback cycle, the address inputs provide the starting address and the device will play until an EOM marker is encountered. The device can continue past an EOM marker in an Operational Mode, or if \overline{CE} is held LOW in address mode. (See page 5 for more Operational Modes).

END-OF-MESSAGE / RUN OUTPUT (\overline{EOM})

A nonvolatile marker is automatically inserted at the end of each recorded message. It remains there until the message is recorded over. The \overline{EOM} output pulses LOW for a period of T_{EOM} at the end of each message.

In addition, the ISD2500 series has an internal V_{CC} detect circuit to maintain message integrity should V_{CC} fall below 3.5 V. In this case, \overline{EOM} goes LOW and the device is fixed in playback-only mode.

When the device is configured in Operational Mode M6 (Push-Button Mode), this pin provides an active-HIGH RUN signal, indicating the device is currently recording or playing. This signal can conveniently drive an LED for a visual indicator of a record or playback operation in process.

OVERFLOW OUTPUT (\overline{OVF})

This signal pulses LOW at the end of memory space, indicating the device has been filled and the message has overflowed. The \overline{OVF} output then follows the \overline{CE} input until a PD pulse has reset the device. This pin can be used to cascade several ISD2500 devices together to increase record/playback durations.

MICROPHONE INPUT (MIC)

The microphone input transfers its signal to the on-chip preamplifier. An on-chip Automatic Gain Control (AGC) circuit controls the gain of this preamplifier from -15 to 24 dB. An external microphone should be AC coupled to this pin via a series capacitor. The capacitor value, together with the internal 10 K Ω resistance on this pin, determines the low-frequency cutoff for the ISD2500 series passband. See Application Information for additional information on low-frequency cutoff calculation.

MICROPHONE REFERENCE INPUT (MIC REF)

The MIC REF input is the inverting input to the microphone preamplifier. This provides a noise-canceling or common-mode rejection input to the device when connected to a differential microphone.

AUTOMATIC GAIN CONTROL INPUT (AGC)

The AGC dynamically adjusts the gain of the preamplifier to compensate for the wide range of microphone input levels. The AGC allows the full range of whispers to loud sounds to be recorded with minimal distortion. The "attack" time is determined by the time constant of a 5 K Ω internal resistance and an external capacitor (C2 on the schematic on page 18) connected from the AGC pin to V_{SSA} analog ground. The "release" time is determined by the time constant of an external resistor (R2) and an external capacitor (C2) connected in parallel between the AGC Pin and V_{SSA} analog ground. Nominal values of 470 K Ω and 4.7 μ F give satisfactory results in most cases.

ANALOG OUTPUT (ANA OUT)

This pin provides the preamplifier output to the user. The voltage gain of the preamplifier is determined by the voltage level at the AGC pin.

ANALOG INPUT (ANA IN)

The analog input pin transfers its signal to the chip for recording. For microphone inputs, the ANA OUT pin should be connected via an external capacitor to the ANA IN pin. This capacitor value, together with the 3.0 K Ω input impedance of ANA IN, is selected to give additional cutoff at the low-frequency end of the voice passband. If the desired input is derived from a source other than a microphone, the signal can be fed, capacitively coupled, into the ANA IN pin directly.

EXTERNAL CLOCK INPUT (XCLK)

The external clock input for the ISD2500 devices has an internal pull-down device. These devices are configured at the factory with an internal sampling clock frequency centered to ± 1 percent of specification. The frequency is then maintained to a variation of ± 2.25 percent over the entire commercial temperature and operating voltage ranges. The internal clock has a ± 5 percent tolerance over the industrial temperature and voltage range. A regulated power supply is recommended for industrial temperature range parts. If greater precision is required, the device can be clocked through the XCLK pin as follows:

Table 1: External Clock Sample Rates

Part Number	Sample Rate	Required Clock
ISD2560	8.0 KHz	1024 KHz
ISD2575	6.4 KHz	819.2 KHz
ISD2590	5.3 KHz	682.7 KHz
ISD25120	4.0 KHz	512 KHz

These recommended clock rates should not be varied because the antialiasing and smoothing filters are fixed, and aliasing problems can occur if the sample rate differs from the one recommended. The duty cycle on the input clock is not critical, as the clock is immediately divided by two. **If the XCLK is not used, this input must be connected to ground.**

SPEAKER OUTPUTS (SP+ /SP-)

All devices in the ISD2500 series include an on-chip differential speaker driver, capable of driving 50 mW into 16 Ω from AUX IN (12.2 mW from memory).

The speaker outputs are held at V_{SSA} levels during record and power down. It is therefore not possible to parallel speaker outputs of multiple ISD2500 devices or the outputs of other speaker drivers.

NOTE Connection of speaker outputs in parallel may cause damage to the device.

A single output may be used alone (including a coupling capacitor between the SP pin and the speaker). These outputs may be used individually with the output signal taken from either pin. Using the differential outputs results in a 4 to 1 improvement in output power.

NOTE Never ground or drive an unused speaker output.

AUXILIARY INPUT (AUX IN)

The Auxiliary Input is multiplexed through to the output amplifier and speaker output pins when \overline{CE} is HIGH, P/\overline{R} is HIGH, and playback is currently not active or if the device is in playback overflow. When cascading multiple ISD2500 devices, the AUX IN pin is used to connect a playback signal from a following device to the previous output speaker drivers. For noise considerations, it is suggested that the auxiliary input not be driven when the storage array is active.

ADDRESS/MODE INPUTS (AX/MX)

The Address/Mode Inputs have two functions depending on the level of the two Most Significant Bits (MSB) of the address (A8 and A9).

If either or both of the two MSBs are LOW, the inputs are all interpreted as address bits and are used as the start address for the current record or playback cycle. The address pins are inputs only and do not output internal address information as the operation progresses. Address inputs are latched by the falling edge of \overline{CE} .

If both MSBs are HIGH, the Address/Mode Inputs are interpreted as Mode bits according to the Operational Mode table. There are six Operational Modes (M0..M6) available as indicated in the table. It is possible to use multiple Operational Modes simultaneously. Operational Modes are sampled on each falling edge of \overline{CE} , and thus Operational Modes and direct addressing are mutually exclusive.

OPERATIONAL MODES

The ISD2500 series is designed with several built-in Operational Modes that provide maximum functionality with minimum additional components. These are described in detail below. The Operational Modes use the address pins on the ISD2500 devices, but are mapped outside the valid address range. When the two Most Significant Bits (MSBs) are HIGH (A8 and A9), the remaining address signals are interpreted as mode bits and not as address bits. Therefore, Operational Modes and direct addressing are not compatible and cannot be used simultaneously.

There are two important considerations for using Operational Modes. First, all operations begin initially at address 0, which is the beginning of the ISD2500 address space. Later operations can begin at other address locations, depending on the Operational Mode(s) chosen. In addition, the address pointer is reset to 0 when the device is changed from record to playback, playback to record (except M6 mode), or when a Power-Down cycle is executed.

Second, Operational Modes are executed when \overline{CE} goes LOW and the two MSBs are HIGH. This Operational Mode remains in effect until the next LOW-going \overline{CE} signal, at which point the current address/mode levels are sampled and executed.

Table 2: Operational Modes Table

Mode Control	Function	Typical Use	Jointly Compatible ¹
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete EOM markers	Position EOM marker at the end of the last message	M3, M4, M5, M6
M2	Not applicable	Reserved	N/A
M3	Looping	Continuous playback from Address 0	M1, M5, M6
M4	Consecutive addressing	Record/play multiple consecutive messages	M0, M1, M5
M5	\overline{CE} level-activated	Allows message pausing	M0, M1, M3, M4
M6	Push-button control	Simplified device interface	M0, M1, M3

¹. Additional Operational Modes can be used simultaneously with the given mode.

OPERATIONAL MODES DESCRIPTION

The Operational Modes can be used in conjunction with a microcontroller, or they can be hard-wired to provide the desired system operation.

M0 — MESSAGE CUEING

Message Cueing allows the user to skip through messages, without knowing the actual physical addresses of each message. Each $\overline{\text{CE}}$ LOW pulse causes the internal address pointer to skip to the next message. This mode should be used for playback only, and is typically used with the M4 Operational Mode.

M1 — DELETE EOM MARKERS

The M1 Operational Mode allows sequentially recorded messages to be combined into a single message with only one EOM marker set at the end of the final message. When this Operational Mode is configured, messages recorded sequentially are played back as one continuous message.

M2 — UNUSED

When Operational Modes are selected, the M2 pin should be LOW.

M3 — MESSAGE LOOPING

The M3 Operational Mode allows for the automatic, continuously repeated playback of the message located at the beginning of the address space. A message can completely fill the SD2500 device and will loop from beginning to end without $\overline{\text{OVF}}$ going LOW.

M4 — CONSECUTIVE ADDRESSING

During normal operations, the address pointer will reset when a message is played through to an EOM marker. The M4 Operational Mode inhibits the address pointer reset on EOM, allowing messages to be played back consecutively.

M5 — $\overline{\text{CE}}$ -LEVEL ACTIVATED

The default mode for ISD2500 devices is for $\overline{\text{CE}}$ to be edge-activated on playback and level-activated on record. The M5 Operational Mode causes the $\overline{\text{CE}}$ pin to be interpreted as level-activated as opposed to edge-activated during playback. This is specifically useful for terminating playback operations using the $\overline{\text{CE}}$ signal.

In this mode, $\overline{\text{CE}}$ LOW begins a playback cycle, at the beginning of the device memory. The playback cycle continues as long as $\overline{\text{CE}}$ is held LOW. When $\overline{\text{CE}}$ goes HIGH, playback will immediately end. A new $\overline{\text{CE}}$ LOW will restart the message from the beginning unless M4 is also HIGH.

M6 — PUSH-BUTTON MODE

The ISD2500 series of devices contain a Push-Button Operational Mode. The Push-Button mode is used primarily in very low-cost applications and is designed to minimize external circuitry and components, thereby reducing system cost. In order to configure the device in Push-Button Operational Mode, the two most significant address bits must be HIGH, and the M6 mode pin must also be HIGH. A device in this mode always powers down at the end of each playback or record cycle after $\overline{\text{CE}}$ goes HIGH.

When this Operational Mode is implemented, several of the pins on the device have alternate functionality:

Table 3: Alternate Functionality in Pins

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{\text{CE}}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse activated)
EOM	Active-HIGH Run Indicator

$\overline{\text{CE}}$ PIN (START/PAUSE)

In Push-Button Operational Mode, $\overline{\text{CE}}$ acts as a LOW-going pulse-activated START/PAUSE signal. If no operation is currently in progress, a LOW-going pulse on this signal will initiate a playback or a record cycle according to the level on the $\text{P}/\overline{\text{R}}$ pin. A subsequent pulse on the $\overline{\text{CE}}$ pin, before an End-Of-Message is reached in playback or an overflow condition occurs, will cause the device to pause. The address counter is not reset, and another $\overline{\text{CE}}$ pulse will cause the device to continue the operation from the place where it was paused.

PD PIN (STOP/RESET)

In push-button Operational Mode, PD acts as a HIGH-going pulse-activated STOP/RESET signal. When a playback or record cycle is in progress and a HIGH-going pulse is observed on PD, the current cycle is terminated and the address pointer is reset to address 0, the beginning of the message space.

 $\overline{\text{EOM}}$ PIN (RUN)

In Push-Button Operational Mode, $\overline{\text{EOM}}$ becomes an active-HIGH RUN signal which can be used to drive an LED or other external device. It is HIGH whenever a record or playback operation is in progress.

Recording in Push-Button Mode

1. The PD pin should be LOW, usually using a pull-down resistor.
2. The $\text{P}/\overline{\text{R}}$ pin is taken LOW.
3. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording starts, $\overline{\text{EOM}}$ goes HIGH to indicate an operation in progress.
4. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording pauses, $\overline{\text{EOM}}$ goes back LOW. The internal address pointers are not cleared, but an EOM marker is stored in memory to point to the message end. The $\text{P}/\overline{\text{R}}$ pin may be taken HIGH at this time. Any subsequent $\overline{\text{CE}}$ would start a playback at address 0.

5. The $\overline{\text{CE}}$ pin is pulsed LOW. Recording starts at the next address after the previous set EOM marker. $\overline{\text{EOM}}$ goes back HIGH.

NOTE *If the M1 Operational Mode pin is also HIGH, the just previously written EOM bit is erased, and recording starts at that address.)*

6. When the recording sequences are finished, the final $\overline{\text{CE}}$ pulse LOW will end the last record cycle, leaving a set $\overline{\text{EOM}}$ marker at the message end. Recording may also be terminated by a HIGH level on PD, which will leave a set EOM marker.

Playback in Push-Button Mode

1. The PD pin should be LOW.
2. The $\text{P}/\overline{\text{R}}$ pin is taken HIGH.
3. The $\overline{\text{CE}}$ pin is pulsed LOW. Playback starts, $\overline{\text{EOM}}$ goes HIGH to indicate an operation in progress.
4. If the $\overline{\text{CE}}$ pin is pulsed LOW or an EOM marker is encountered during an operation, the part will pause. The internal address pointers are not cleared, and $\overline{\text{EOM}}$ goes back LOW. The $\text{P}/\overline{\text{R}}$ pin may be changed at this time. A subsequent record operation would not reset the address pointers and the recording would begin where playback ended.
5. $\overline{\text{CE}}$ is again pulsed LOW. Playback starts where it left off, with $\overline{\text{EOM}}$ going HIGH to indicate an operation in progress.
6. Playback continues as in steps 4 and 5 until PD is pulsed HIGH or overflow occurs.
7. If in overflow, pulling $\overline{\text{CE}}$ LOW will reset the address pointer and start playback from the beginning. After a PD pulse, the part is reset to address 0.

NOTE *Push-button mode can be used in conjunction with modes M0, M1, and M3.*

GOOD AUDIO DESIGN PRACTICES

SD products are very high-quality single-chip voice recording and playback systems. To ensure the highest quality voice reproduction, it is important that good audio design practices on layout and power supply decoupling be followed. See the ISD Application Notes in this book for details.

SD1000A COMPATIBILITY

The ISD2500 series of devices is designed to provide upward compatibility with the ISD1000A family. When designing with the ISD2500 series, the following differences should be noted.

ADDRESSING

The ISD2560/75/90/120 devices have 480K storage cells designed to provide 60 seconds of storage at a sampling rate of 8.0 KHz. This is approximately four times the storage of the SD1000A family. To enable the same addressing resolution, two additional address pins have been added. The address space of each device is divisible into 600 increments with valid addressing from 00 to 257 Hex. Some higher addresses are mapped into the Operational Modes. All other addresses are invalid.

OVERFLOW

The ISD1000A series combined two functions on the $\overline{\text{EOM}}$ pin: end-of-message indication and overflow. The ISD2500 separates these two functions. Pin 25 (PDIP package) remains as $\overline{\text{EOM}}$, but outputs only the EOM signal indication. Pin 22 (PDIP package) becomes $\overline{\text{OVF}}$ and pulses LOW only when the device reaches its end of memory, or is "full." This change allows easy message cueing and addressability across device boundaries. This also means that the M2 Operational Mode found in the ISD1000A family is not implemented in the ISD2500 series.

PUSH-BUTTON MODE

The ISD2500 series includes an additional Operational Mode called Push-Button mode. This provides an alternative interface to the record and playback functions of the part. The $\overline{\text{CE}}$ and PD pins become redefined as edge-activated "push-buttons." A pulse on $\overline{\text{CE}}$ initiates a cycle, and if triggered again, pauses the current cycle without resetting the address pointer (i.e., a Start or Pause function). PD stops any current cycle and resets the address pointer to the beginning of the message space (i.e., a Stop and Reset function). Additionally, the $\overline{\text{EOM}}$ pin functions as an active-HIGH run indicator, and can be used to drive an LED indicating a record or playback operation is in progress. Devices in the Push-Button mode cannot be cascaded.

LOOPING MODE

The ISD2500 series can loop with a message that completely fills the memory space.

NOTE Additional descriptions of ISD2500 device functionality and application examples are provided in the ISD Application Notes in this book.

TIMING DIAGRAMS

Figure 2: Record

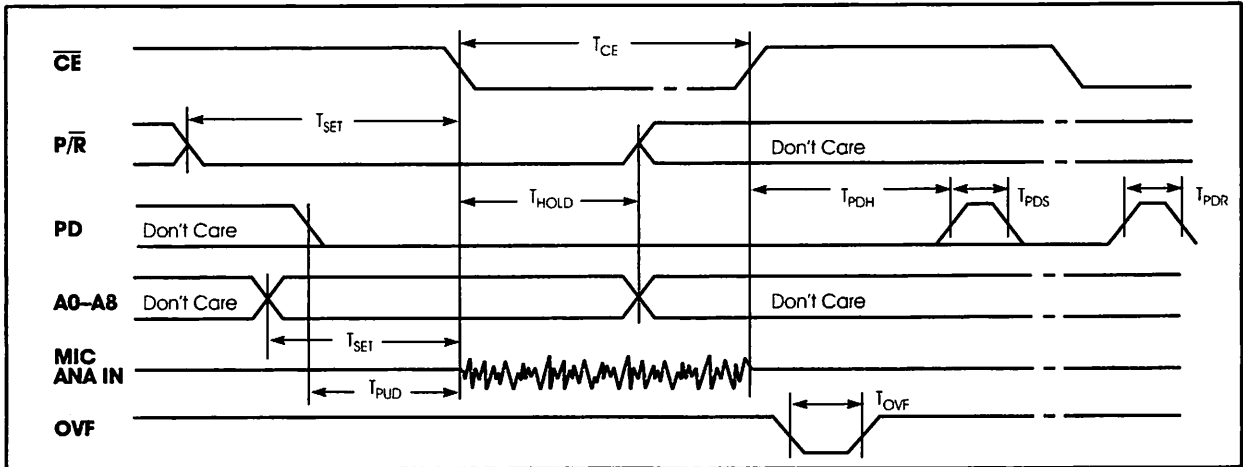


Figure 3: Playback

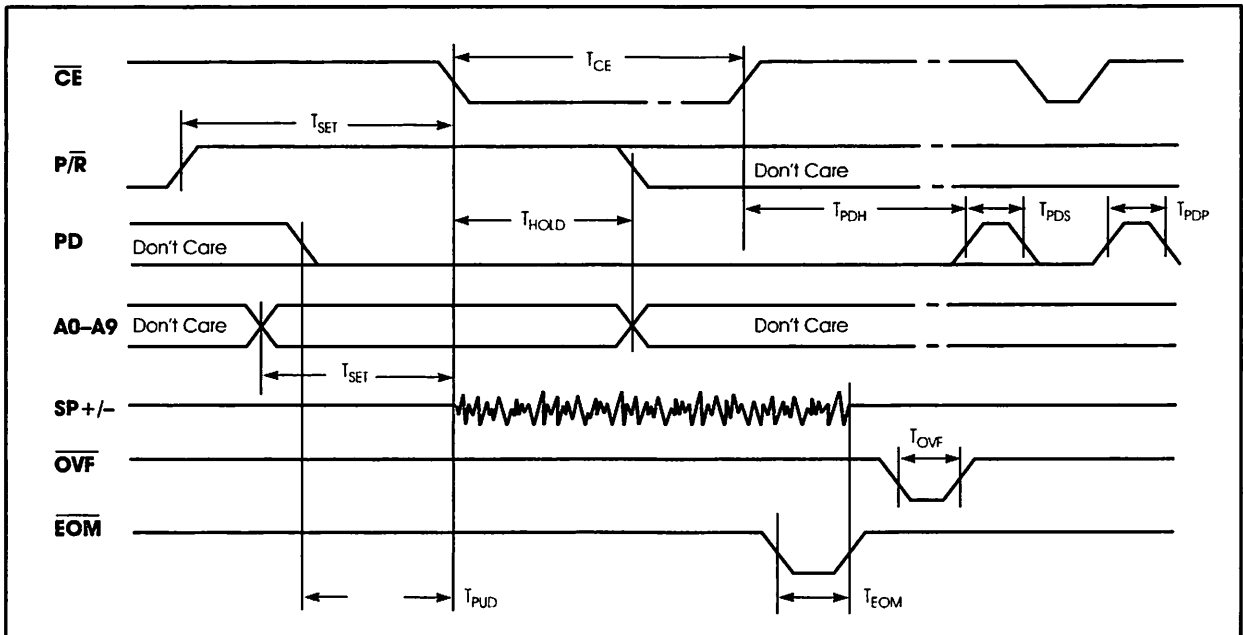


Table 4: Absolute Maximum Ratings (Packaged Parts)⁽¹⁾

Condition	Value
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pin	(V _{SS} - 0.3 V) to (V _{CC} + 0.3 V)
Voltage applied to any pin (Input current limited to ±20 mA)	(V _{SS} - 1.0 V) to (V _{CC} + 1.0 V)
Lead temperature (soldering - 10 seconds)	300°C
V _{CC} - V _{SS}	-0.3 V to +7.0 V

Table 5: Operating Conditions (Packaged Parts)

Condition	Value
Commercial operating temperature range ⁽¹⁾	0°C to +70°C
Industrial operating temperature range ⁽¹⁾	
Supply voltage (V _{CC}) ⁽²⁾	+4.5 V to +5.5 V
Ground voltage (V _{SS}) ⁽³⁾	0 V

1. Case temperature.
2. V_{CC} = V_{CCA} = V_{CCD}.
3. V_{SS} = V_{SSA} = V_{SSD}.
4. Consult factory.

1. Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability. Functional operation is not implied at these conditions.

Table 6: DC Parameters (Packaged Parts)

Symbol	Parameters	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions
V _{IL}	Input Low Voltage			0.8	V	
V _{IH}	Input High Voltage	2.0			V	
V _{OL}	Output Low Voltage			0.4	V	I _{OL} = 4.0 mA
V _{OH}	Output High Voltage	V _{CC} -0.4			V	I _{OH} = -10 μA
V _{OH1}	OVF Output High Voltage	2.4			V	I _{OH} = -1.6 mA
V _{OH2}	EOM Output High Voltage	V _{CC} - 1.0	V _{CC} - 0.8		V	I _{OH} = -3.2 mA
I _{CC}	V _{CC} Current (Operating)		25	30	mA	R _{EXT} = ∞ ⁽³⁾
I _{SB}	V _{CC} Current (Standby)		1	10	μA	⁽³⁾
I _{IL}	Input Leakage Current			±1	μA	
I _{ILPD}	Input Current HIGH w/Pull Down			130	μA	Force V _{CC} ⁽⁴⁾
R _{EXT}	Output Load Impedance	16			Ω	Speaker Load
R _{MIC}	Preamp In Input Resistance	4	9	15	KΩ	MIC and MIC REF Pins
R _{AUX}	AUX INPUT Resistance	5	11	20	KΩ	

Table 6: DC Parameters (Packaged Parts)

Symbol	Parameters	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions
R _{ANA IN}	ANA IN Input Resistance	2.3	3	5	K Ω	
A _{PRE1}	Preamp Gain 1	21	24	26	dB	AGC = 0.0 V
A _{PRE2}	Preamp Gain 2		-15	5	dB	AGC = 2.5 V
A _{AUX}	AUX IN/SP+ Gain		0.98	1.0	V/V	
A _{ARP}	ANA IN to SP+/- Gain	21	23	26	dB	
R _{AGC}	AGC Output Resistance	2.5	5	9.5	K Ω	

1. Typical values @ $T_A = 25^\circ\text{C}$ and 5.0 V.
2. All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.
3. V_{CCA} and V_{CCD} connected together.
4. XCLK pin only.

Table 7: AC Parameters (Packaged Parts)

Symbol	Characteristic	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions	
F _S	Sampling Frequency	ISD2560	8.0		KHz	(7)	
		ISD2575	6.4		KHz	(7)	
		ISD2590	5.3		KHz	(7)	
		ISD25120	4.0		KHz	(7)	
F _{CF}	Filter Pass Band	ISD2560	3.4		KHz	3 dB Roll-Off Point ^{(3) (8)}	
		ISD2575	2.7		KHz	3 dB Roll-Off Point ^{(3) (8)}	
		ISD2590	2.3		KHz	3 dB Roll-Off Point ^{(3) (8)}	
		ISD25120	1.7		KHz	3 dB Roll-Off Point ^{(3) (8)}	
T _{REC}	Record Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation ⁽⁷⁾
		ISD2560	56.5	60.0	63.8	sec	Industrial Operation ⁽⁷⁾
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation ⁽⁷⁾
		ISD2575	70.7	75.0	79.7	sec	Industrial Operation ⁽⁷⁾
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation ⁽⁷⁾
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation ⁽⁷⁾
T _{PLAY}	Playback Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation
		ISD2560	56.5	60.0	63.8	sec	Industrial Operation
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation
		ISD2575	70.7	75.0	79.7	sec	Industrial Operation
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation
T _{CE}	$\overline{\text{CE}}$ Pulse Width		100		nsec		
T _{SET}	Control/Address Setup Time		300		nsec		
T _{HOLD}	Control/Address Hold Time		0		nsec		

Table 7: AC Parameters (Packaged Parts)

Symbol	Characteristic	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions	
T _{PUD}	Power-Up Delay	ISD2560	24.1	25.0	27.8	msec	Commercial Operation
		ISD2560	23.5		28.5	msec	Industrial Operation
		ISD2575	30.2	31.3	34.3	msec	Commercial Operation
		ISD2575	29.3	31.3	35.2	msec	Industrial Operation
		ISD2590	36.2	37.5	40.8	msec	Commercial Operation
		ISD25120	48.2	50.0	53.6	msec	Commercial Operation
T _{PDR}	PD Pulse Width Record	ISD2560		25		msec	
		ISD2575		31.25		msec	
		ISD2590		37.5		msec	
		ISD25120		50.0		msec	
T _{PDP}	PD Pulse Width Play	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T _{PDS}	PD Pulse Width Static		100		nsec	(6)	
T _{PDH}	Power Down Hold		0		nsec		
T _{EOM}	EOM Pulse Width	ISD2560		12.5		msec	
		ISD2575		15.625		msec	
		ISD2590		18.75		msec	
		ISD25120		25.0		msec	
T _{OVF}	Overflow Pulse Width		6.5		μsec		
THD	Total Harmonic Distortion		1	2	%	@ 1 KHz	
P _{OUT}	Speaker Output Power		12.2	50	mW	R _{EXT} = 16 Ω ⁽⁴⁾	
V _{OUT}	Voltage Across Speaker Pins			2.5	V p-p	R _{EXT} = 600 Ω	
V _{IN1}	MIC Input Voltage			20	mV	Peak-to-Peak ⁽⁵⁾	
V _{IN2}	ANA IN Input Voltage			50	mV	Peak-to-Peak	
V _{IN3}	Aux Input Voltage			1.25	V	Peak-to-Peak; R _{EXT} = 16 Ω	

1. Typical values @ T_A = 25°C and 5.0 V.

2. All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.

3. Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).

4. From AUX IN; if ANA IN is driven at 50 mV p-p, the P_{OUT} = 12.2 mW, typical.

5. With 5.1 KΩ series resistor at ANA IN.

6. T_{PDS} is required during a static condition, typically overflow.

7. Sampling Frequency and playback Duration can vary as much as ±2.25 percent over the commercial temperature range and voltage range and ±5 percent over the industrial temperature and voltage range. For greater stability, an external clock can be utilized (see Pin Descriptions).

8. Filter specification applies to both the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 6 dB drop by nature of passing through both filters.

TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (PACKAGED PARTS)

Chart 1: Record Mode Operating Current (I_{CC})

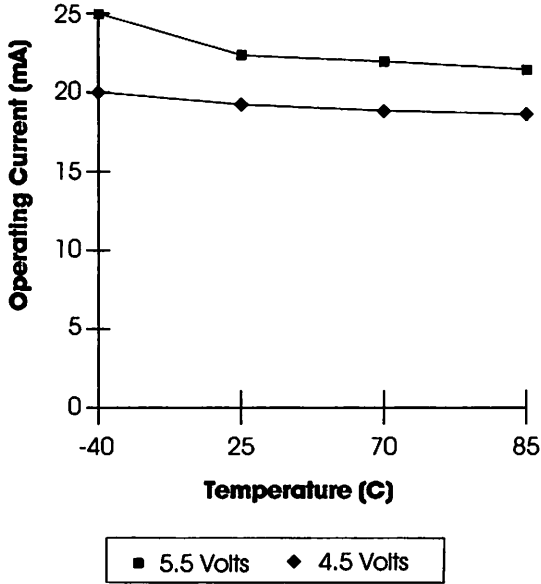


Chart 3: Standby Current (I_{SB})

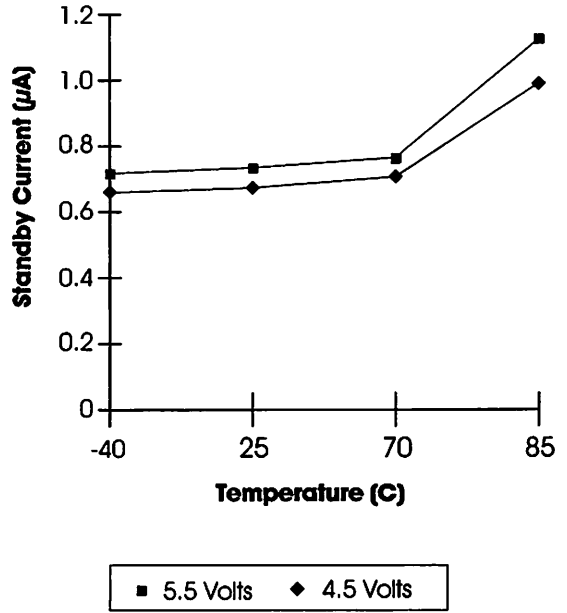


Chart 2: Total Harmonic Distortion

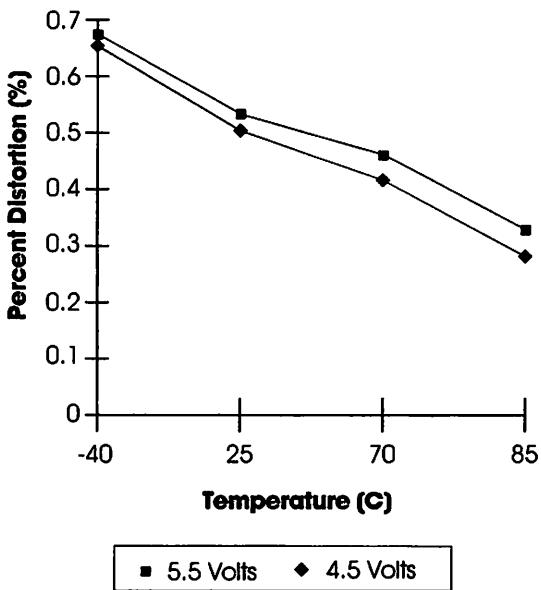


Chart 4: Oscillator Stability

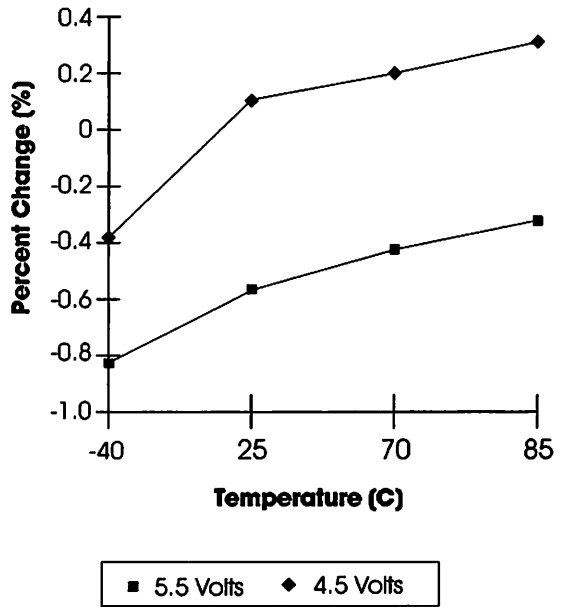


Table 8: Absolute Maximum Ratings (Die)⁽¹⁾

Condition	Value
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pad	(V _{SS} - 0.3 V) to (V _{CC} + 0.3 V)
Voltage applied to any pad (Input current limited to ±20 mA)	(V _{SS} - 1.0 V) to (V _{CC} + 1.0 V)
V _{CC} - V _{SS}	-0.3 V to +7.0 V

Table 9: Operating Conditions (Die)

Condition	Value
Commercial operating temperature range	0°C to +50°C
Supply voltage (V _{CC}) ⁽¹⁾	+4.5 V to +6.5 V
Ground voltage (V _{SS}) ⁽²⁾	0 V

1. V_{CC} = V_{CCA} = V_{CCD}.

2. V_{SS} = V_{SSA} = V_{SSD}.

1. Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability. Functional operation is not implied at these conditions.

Table 10: DC Parameters (Die)

Symbol	Parameters	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions
V _{IL}	Input Low Voltage			0.8	V	
V _{IH}	Input High Voltage	2.0			V	
V _{OL}	Output Low Voltage			0.4	V	I _{OL} = 4.0 mA
V _{OH}	Output High Voltage	V _{CC} - 0.4			V	I _{OH} = -10 μA
V _{OH1}	OVF Output High Voltage	2.4			V	I _{OH} = -1.6 mA
V _{OH2}	EOM Output High Voltage	V _{CC} - 1.0	V _{CC} - 0.8		V	I _{OH} = -3.2 mA
I _{CC}	V _{CC} Current (Operating)		25	30	mA	R _{EXT} = ∞ ⁽³⁾
I _{SB}	V _{CC} Current (Standby)		1	10	μA	⁽²⁾
I _{IL}	Input Leakage Current			±1	μA	
I _{ILPD}	Input Current HIGH with Pull Down			130	μA	Force V _{CC} ⁽⁴⁾
R _{EXT}	Output Load Impedance	16			Ω	Speaker Load
R _{MIC}	Preamp In Input Resistance	4	9	15	KΩ	MIC and MIC REF Pads
R _{AUX}	AUX INput Resistance	5	11	20	KΩ	
R _{ANA IN}	ANA IN Input Resistance	2.3	3	5	KΩ	
A _{PRE1}	Preamp Gain 1	21	24	26	dB	AGC = 0.0 V

Table 10: DC Parameters (Die)

Symbol	Parameters	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions
A _{PRE2}	Preamplifier Gain 2		-15	5	dB	AGC = 2.5 V
A _{AUX}	AUX IN/SP+ Gain		0.98	1.0	V/V	
A _{ARP}	ANA IN to SP+/- Gain	21	23	26	dB	
R _{AGC}	AGC Output Resistance	2.5	5	9.5	KΩ	

1. Typical values @ $T_A = 25^\circ\text{C}$ and 5.0 V.
2. All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.
3. V_{CCA} and V_{OCD} connected together.
4. XCLK pad only.

Table 11: AC Parameters (Die)

Symbol	Characteristic	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions	
F _S	Sampling Frequency	ISD2560	8.0		KHz	(7)	
		ISD2575	6.4		KHz	(7)	
		ISD2590	5.3		KHz	(7)	
		ISD25120	4.0		KHz	(7)	
F _{CF}	Filter Pass Band	ISD2560	3.4		KHz	3 dB Roll-Off Point (3) (8)	
		ISD2575	2.7		KHz	3 dB Roll-Off Point (3) (8)	
		ISD2590	2.3		KHz	3 dB Roll-Off Point (3) (8)	
		ISD25120	1.7		KHz	3 dB Roll-Off Point (3) (8)	
T _{REC}	Record Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation ⁽⁷⁾
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation ⁽⁷⁾
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation ⁽⁷⁾
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation ⁽⁷⁾
T _{PLAY}	Playback Duration	ISD2560	58.1	60.0	62.0	sec	Commercial Operation ⁽⁷⁾
		ISD2575	72.6	75.0	77.5	sec	Commercial Operation ⁽⁷⁾
		ISD2590	87.1	90.0	93.0	sec	Commercial Operation ⁽⁷⁾
		ISD25120	116.1	120.0	123.9	sec	Commercial Operation ⁽⁷⁾
T _{CE}	CE Pulse Width		100		nsec		
T _{SET}	Control/Address Setup Time		300		nsec		
T _{HOLD}	Control/Address Hold Time		0		nsec		
T _{PUD}	Power-Up Delay	ISD2560	24.1	25.0	27.8	msec	Commercial Operation
		ISD2575	30.2	31.3	34.3	msec	Commercial Operation
		ISD2590	36.2	37.5	40.8	msec	Commercial Operation
		ISD25120	48.2	50.0	53.6	msec	Commercial Operation

Table 11: AC Parameters (Die)

Symbol	Characteristic	Min ⁽²⁾	Typ ⁽¹⁾	Max ⁽²⁾	Units	Conditions
T _{PDR}	PD Pulse Width Record	ISD2560	25		msec	
		ISD2575	31.25		msec	
		ISD2590	37.5		msec	
		ISD25120	50.0		msec	
T _{PDP}	PD Pulse Width Play	ISD2560	12.5		msec	
		ISD2575	15.625		msec	
		ISD2590	18.75		msec	
		ISD25120	25.0		msec	
T _{PDS}	PD Pulse Width Static		100		nsec	(6)
T _{PDH}	Power Down Hold		0		nsec	
T _{EOM}	EOM Pulse Width	ISD2560	12.5		msec	
		ISD2575	15.625		msec	
		ISD2590	18.75		msec	
		ISD25120	25.0		msec	
T _{OVF}	Overflow Pulse Width		6.5		μsec	
THD	Total Harmonic Distortion		1	3	%	@ 1 KHz
P _{OUT}	Speaker Output Power		12.2	50	mW	R _{EXT} = 16 Ω ⁽⁴⁾
V _{OUT}	Voltage Across Speaker Pins			2.5	V p-p	R _{EXT} = 600 Ω
V _{IN1}	MIC Input Voltage			20	mV	Peak-to-Peak ⁽⁵⁾
V _{IN2}	ANA IN Input Voltage			50	mV	Peak-to-Peak
V _{IN3}	Aux Input Voltage			1.25	V	Peak-to-Peak; R _{EXT} = 16 Ω

1. Typical values @ T_A = 25°C and 5.0 V.

2. All Min/Max limits are guaranteed by ISD via electrical testing or characterization. Not all specifications are 100 percent tested.

3. Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).

4. From AUX IN; if ANA IN is driven at 50 mV p-p, the P_{OUT} = 12.2 mW, typical.

5. With 5.1 KΩ series resistor at ANA IN.

6. T_{PDS} is required during a static condition, typically overflow.

7. Sampling Frequency and playback Duration can vary as much as ±2.25 percent over the commercial temperature range and voltage range. For greater stability, an external clock can be utilized (see Pin Descriptions).

8. Filter specification applies to the antialiasing filter and the smoothing filter.

TYPICAL PARAMETER VARIATION WITH VOLTAGE AND TEMPERATURE (DIE)

Chart 5: Record Mode Operating Current (I_{CC})

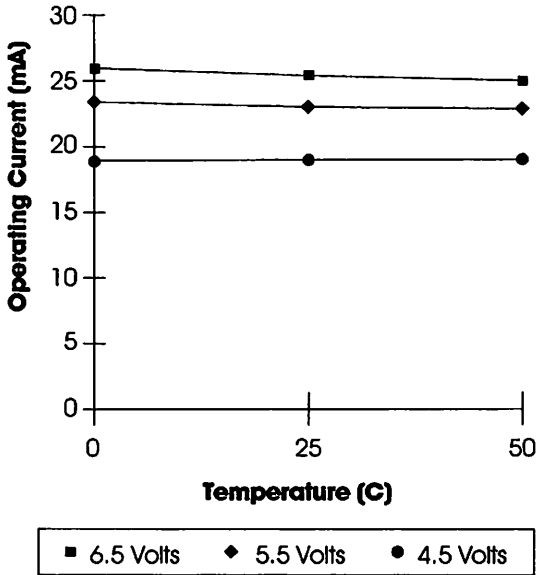


Chart 7: Standby Current (I_{SB})

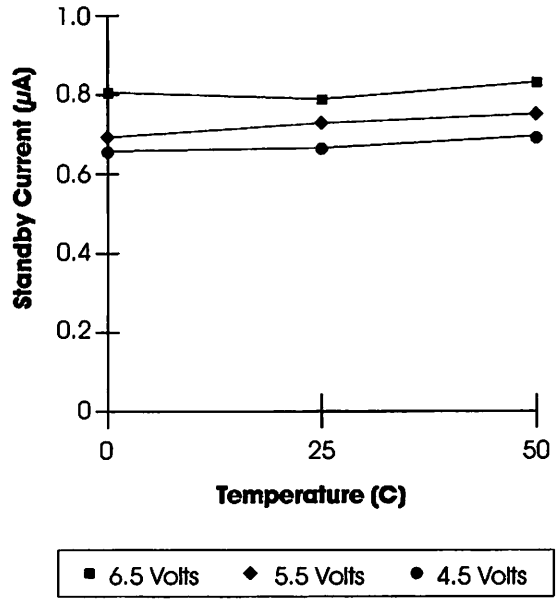


Chart 6: Total Harmonic Distortion

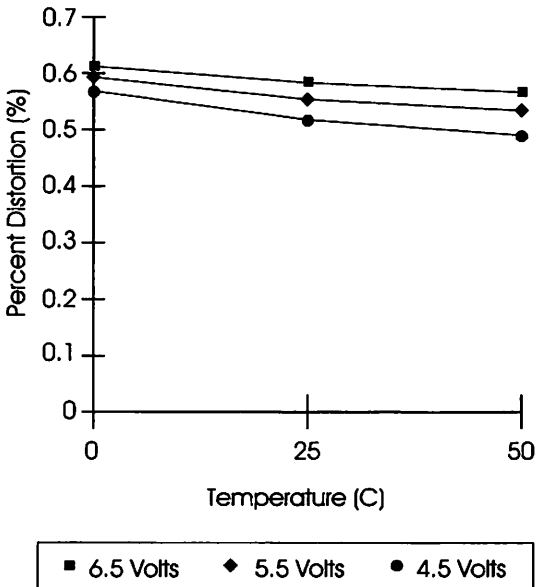


Chart 8: Oscillator Stability

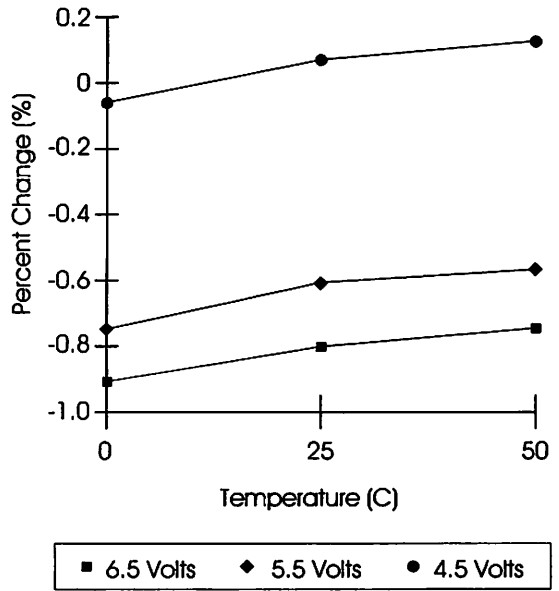
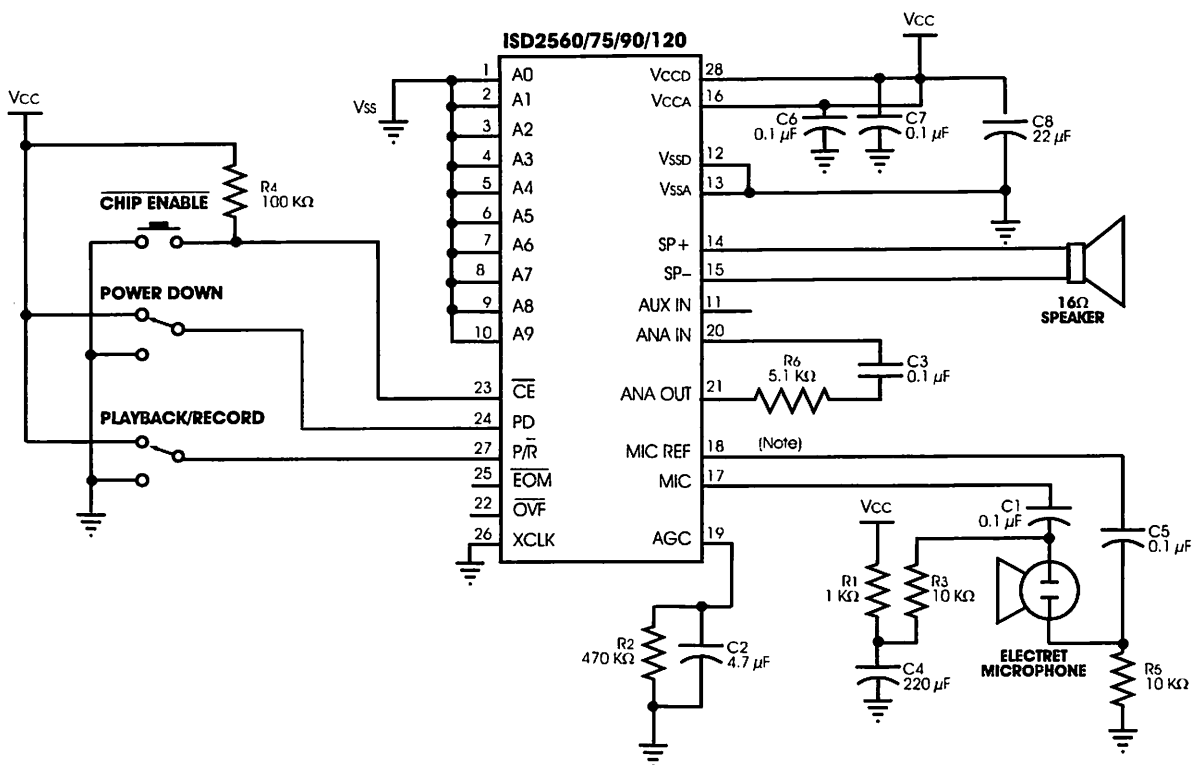


Figure 4: ISD2560/75/90/120 Application Example—Design Schematic



NOTE: If desired, pin 18 (PDIP package) may be left unconnected (microphone preamplifier noise will be higher). In this case, pin 18 must not be tied to any other signal or voltage. Additional design example schematics are provided in the Application Notes in this book.

Table 12: Application Example—Basic Device Control

Control Step	Function	Action
1	Power up chip and select record/playback mode	(1.) PD = LOW, (2.) P/R = As desired
2	Set message address for record/playback	Set addresses A0–A9
3A	Begin playback	P/R = HIGH, CE = Pulsed LOW
3B	Begin record	P/R = LOW, CE = LOW
4A	End playback	Automatic
4B	End record	PD or CE = HIGH

Table 13: Application Example—Passive Component Functions

Part	Function	Comments
R1	Microphone power supply decoupling	Reduces power supply noise
R2	Release time constant	Sets release time for AGC
R3, R5	Microphone biasing resistors	Provides biasing for microphone operation
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages.
R6	Series limiting resistor	Reduces level to high supply voltages
C1, C5	Microphone DC-blocking capacitor Low-frequency cutoff	Decouples microphone bias from chip. Provides single-pole low-frequency cutoff and common mode noise rejection.
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff
C4	Microphone power supply decoupling	Reduces power supply noise
C6, C7, C8	Power supply capacitors	Filter and bypass of power supply

EXPLANATION

In this simplified block diagram of a microcontroller application, the Push-Button mode and message cueing are used. The microcontroller is a 16-pin version with enough port pins for buttons, an LED, and the ISD2500 series device. The software can be written to use three buttons: one each for play and record, and one for message selection. Because the microcontroller is interpreting the buttons and commanding the ISD2500 device, software can be written for any functions desired in a particular application.

NOTE *ISD does not recommend connecting address lines directly to a microprocessor bus. Address lines should be externally latched.*

Figure 5: ISD2560/75/90/120 Application Example—Microcontroller/ISD2500 Interface

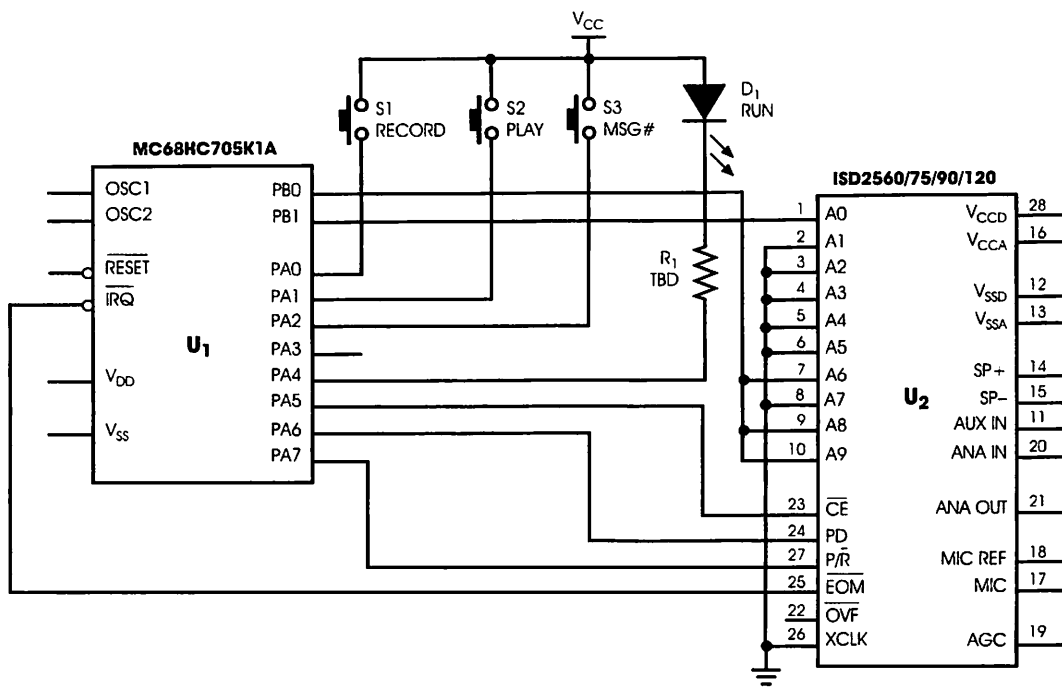
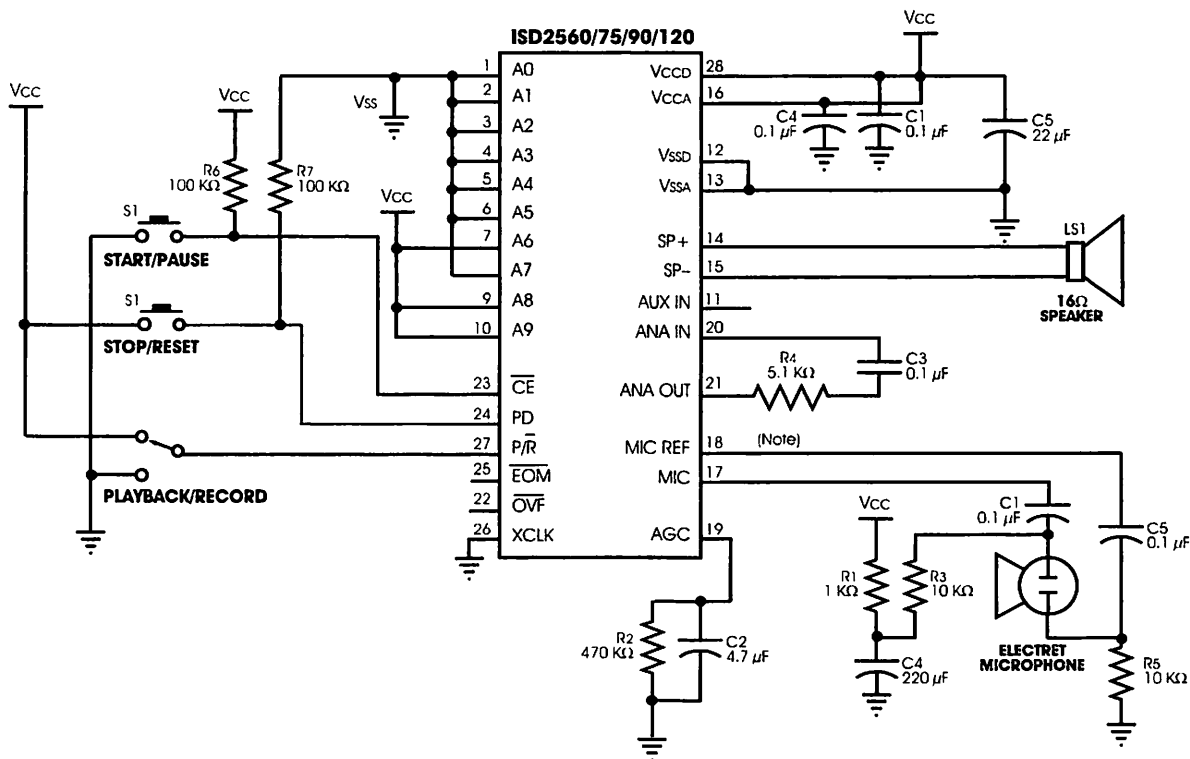


Figure 6: ISD2500 Application Example—Push-Button



NOTE: Please refer to Application Information.

Table 14: Application Example—Push-Button Control

Control Step	Function	Action
1	Select record/playback mode	P/\bar{R} = As desired
2A	Begin playback	P/\bar{R} = HIGH, $\bar{C}E$ = Pulsed LOW
2B	Begin record	P/\bar{R} = LOW, $\bar{C}E$ = Pulsed LOW
3	Pause record or playback	$\bar{C}E$ = Pulsed LOW
4A	End playback	Automatic at EOM marker or PD = Pulsed HIGH
4B	End record	PD = Pulsed HIGH

Table 15: Application Example—Passive Component Functions

Part	Function	Comments
R2	Release time constant	Sets release time for AGC
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages
R6, R7	Pull-up and pull-down resistors	Defines static state of inputs
C1, C4, C5	Power supply capacitors	Filters and bypass of power supply
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff

Table 16: Push-Button Parameters

Symbol	Characteristic	Min	Typ (1)	Max	Units	Conditions
T_{CE}	$\bar{C}E$ Pulse Width [Start/Pause]		300		nsec	
T_{SET}	Control/Address Setup Time		300		nsec	
T_{PUD}	Power-Up Delay					
	ISD2560		25		msec	
	ISD2575		31.25		msec	
	ISD2590		37.25		msec	
	ISD25120		50.0		msec	
T_{PD}	PD Pulse Width [Stop/Reset]		300		nsec	
T_{RUN}	$\bar{C}E$ to EOM HIGH	25		400	nsec	
T_{PAUSE}	$\bar{C}E$ to EOM LOW	50		400	nsec	
T_{DB}	$\bar{C}E$ HIGH Debounce					
	ISD2560	70		105	msec	
	ISD2575	85		135	msec	
	ISD2590	105		160	msec	
	ISD25120	135		215	msec	

PUSH-BUTTON TIMING DIAGRAMS

Figure 7: Push-Button Mode Record

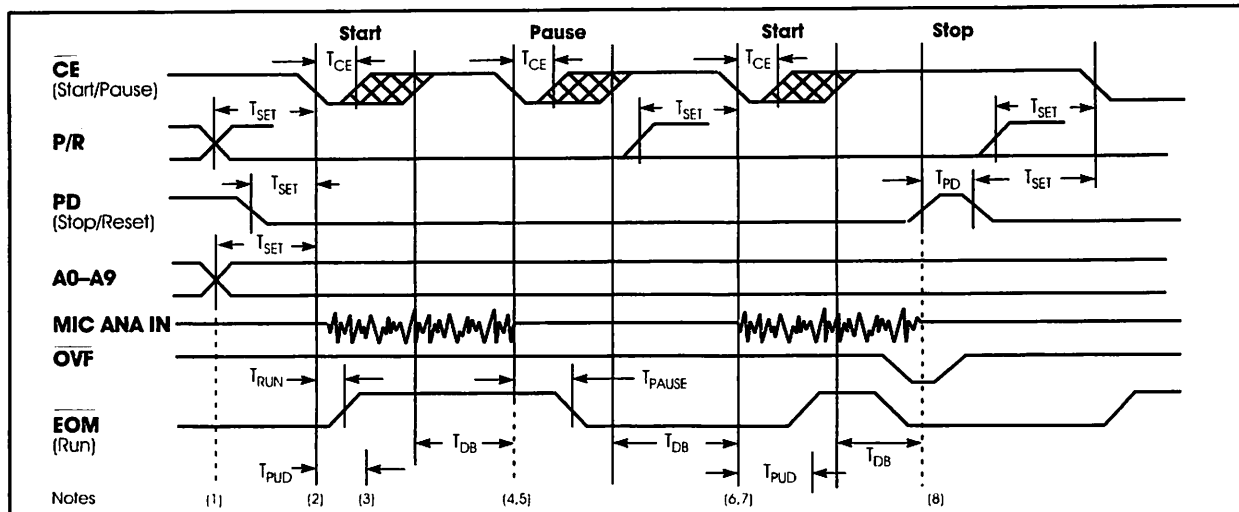
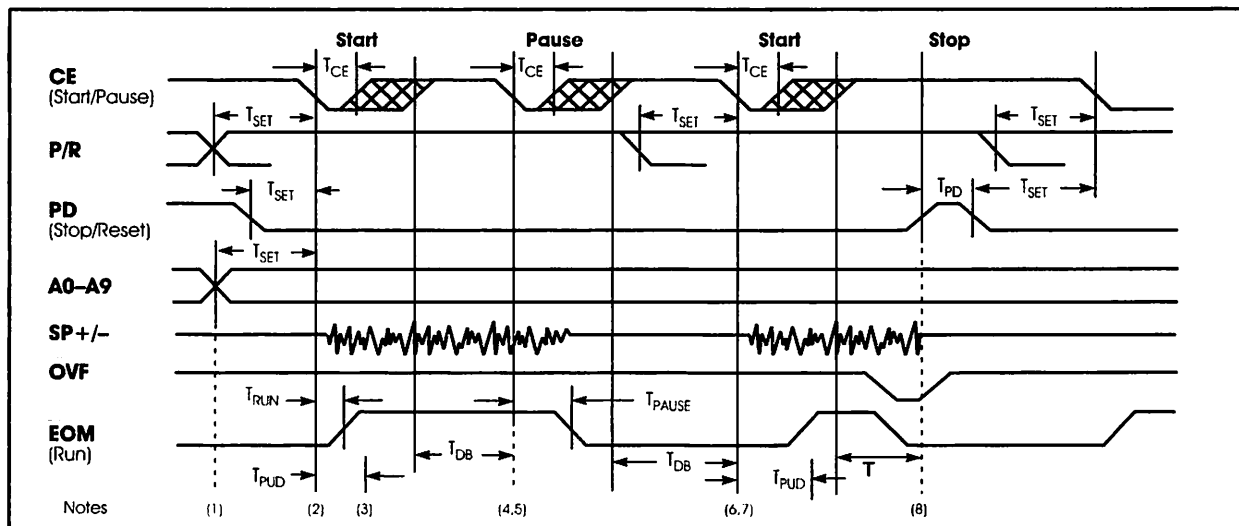


Figure 8: Push-Button Mode Playback



1. A9, A8, and A6 = 1 for push-button operation.
2. The first \overline{CE} LOW pulse performs a Start function.
3. The part will begin to play or record after a power-up delay T_{PUD} .
4. The part must have \overline{CE} HIGH for a debounce period T_{DB} before it will recognize another falling edge of \overline{CE} and pause.
5. The second \overline{CE} LOW pulse, and every even pulse thereafter, performs a Pause function.
6. Again, the part must have \overline{CE} HIGH for a debounce period T_{DB} before it will recognize another falling edge of \overline{CE} , which would restart an operation. In addition, the part will not do an internal power down until \overline{CE} is HIGH for the T_{DB} time.
7. The third \overline{CE} LOW pulse, and every odd pulse thereafter, performs a Resume function.
8. At any time, a HIGH level on PD will stop the current function, reset the address counter, and power down the device.

DEVICE PHYSICAL DIMENSIONS

Figure 9: 28-Lead 8x13.4mm Plastic Thin Small Outline Package (TSOP) Type I (E)

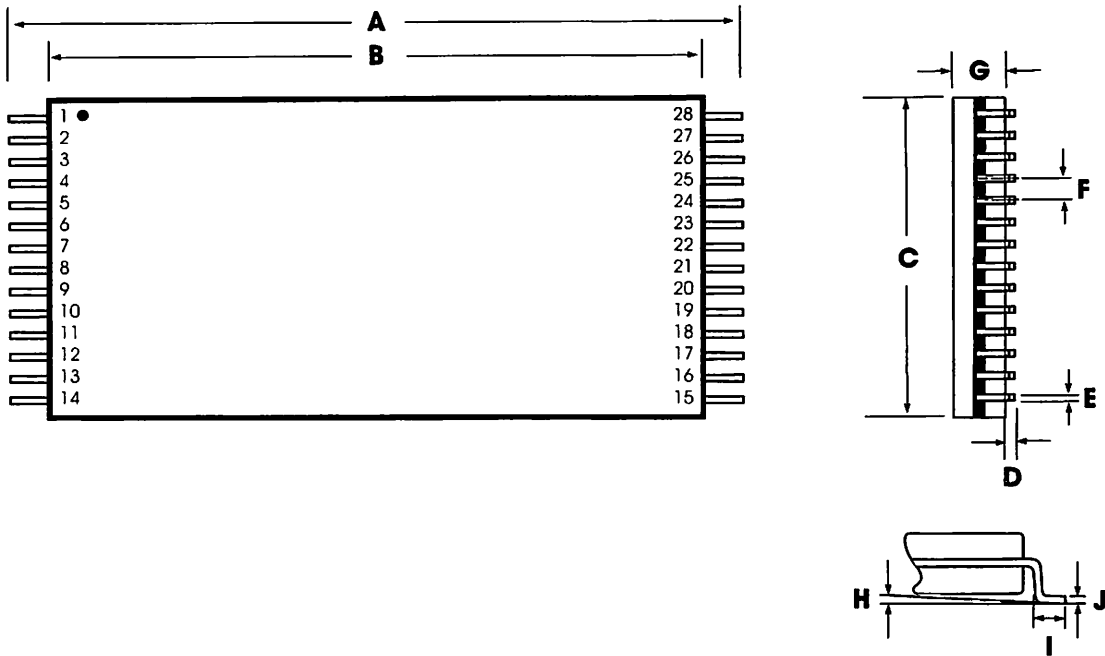


Table 17: Plastic Thin Small Outline Package (TSOP) Type I (E) Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.520	0.528	0.535	13.20	13.40	13.60
B	0.461	0.465	0.469	11.70	11.80	11.90
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.007	0.009	0.011	0.17	0.22	0.27
F		0.0217			0.55	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0°	3°	6°	0°	3°	6°
I	0.020	0.022	0.028	0.50	0.55	0.70
J	0.004		0.008	0.10		0.21

NOTE: Lead coplanarity to be within 0.004 inches.

Figure 10: 28-Lead 0.600-Inch Plastic Dual Inline Package (PDIP) (P)

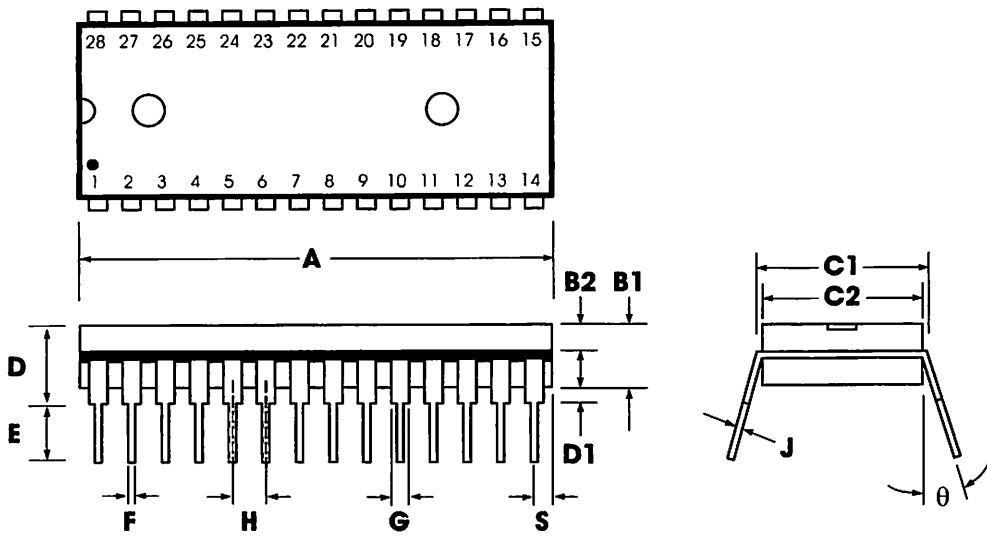


Table 18: Plastic Dual Inline Package (PDIP) (P) Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	1.445	1.450	1.455	36.70	36.83	36.96
B1		0.150			3.81	
B2	0.065	0.070	0.075	1.65	1.78	1.91
C1	0.600		0.625	15.24		15.88
C2	0.530	0.540	0.550	13.46	13.72	13.97
D			0.19			4.83
D1	0.015			0.38		
E	0.125		0.135	3.18		3.43
F	0.015	0.018	0.022	0.38	0.46	0.56
G	0.055	0.060	0.065	1.40	1.52	1.65
H		0.100			2.54	
J	0.008	0.010	0.012	0.20	0.25	0.30
S	0.070	0.075	0.080	1.78	1.91	2.03
q	0°		15°	0°		15°

NOTE: Lead coplanarity to be within 0.004 inches.

Figure 11: 32-Lead 8x20mm Plastic Thin Small Outline Package (TSOP) Type I (T)

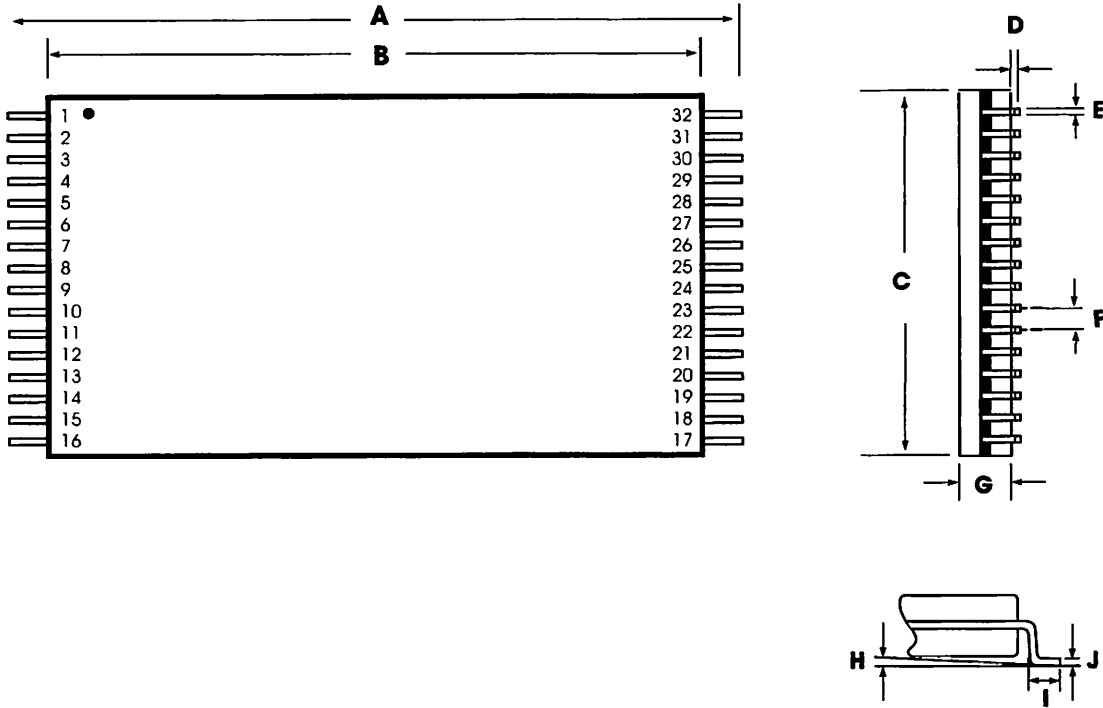


Table 19: Plastic Thin Small Outline Package (TSOP) Type I (T) Dimensions

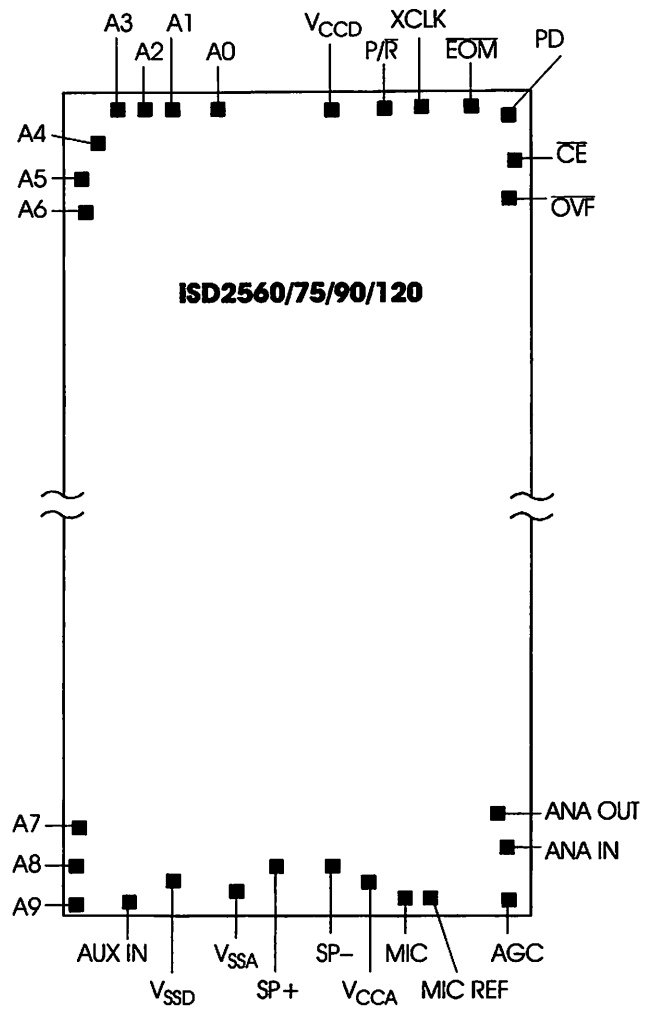
	INCHES			MILLIMETERS		
	Min	Nom	Max	Min		Max
A	0.780	0.787	0.795	19.80	20.00	20.20
B	0.720	0.724	0.728	18.30	18.40	18.50
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.006	0.009	0.011	0.17	0.22	0.27
F		0.0197			0.50	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0°	3°	5°	0°	3°	5°
I	0.020	0.024	0.028	0.50	0.60	0.70
J	0.004		0.008	0.10		0.21

NOTE: Lead coplanarity to be within 0.002 inches.

Figure 12: ISD2560/75/90/120 Products **Current Bonding Physical Layout**¹ (Unpackaged Die)

ISD2560/75/90/120²

- I. Die Dimensions
 X: 187 ± 1 mils
 Y: 399 ± 1 mils
- II. Die Thickness²
 17.5 ± 1 mils
- III. Pad Opening
 109 x 109 microns
 4.3 x 4.3 mils



- 1. The backside of die is internally connected to V_{SS}. It **MUST NOT** be connected to any other potential or damage may occur.
- 2. Die thickness is subject to change, please contact ISD factory for status.

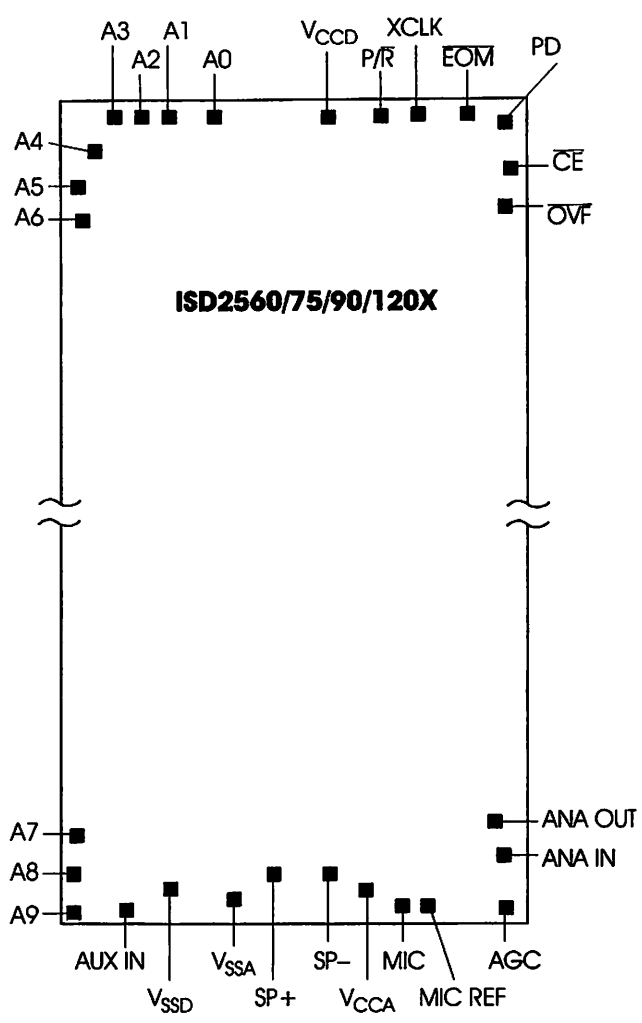
Table 20: ISD2560/75/90/120 Products Current PIN/PAD Designations, with Respect to Die Center (μm)

Pin	Pin Name	X Axis	Y Axis
A0	Address 0	-1148.9	4898.2
A1	Address 1	-1406.9	4898.2
A2	Address 2	-1661.9	4898.2
A3	Address 3	-1916.9	4898.2
A4	Address 4	-2069.9	4608.2
A5	Address 5	-2194.9	4358.2
A6	Address 6	-2194.9	4108.2
A7	Address 7	-2194.9	-4212.3
A8	Address 8	-2194.9	-4456.3
A9	Address 9	-2076.4	-4897.3
AUX IN	Auxiliary Input	-1607.9	-4868.3
V _{SSD}	V _{SS} Digital Power Supply	-1343.9	-4850.8
V _{SSA}	V _{SS} Analog Power Supply	-551.9	-4884.8
SP+	Speaker Output +	-111.4	-4790.8
SP-	Speaker Output -	425.6	-4790.8
V _{CCA}	V _{CC} Analog Power Supply	865.1	-4848.32
MIC	Microphone Input	1320.7	-4897.3
MIC REF	Microphone Reference	1605.1	-4897.3
AGC	Automatic Gain Control	1877.6	-4871.3
ANA IN	Analog Input	2202.11	-4269.8
ANA OUT	Analog Output	2123.1	-3910.8
OVF	Overflow Output	2142.6	4154.7
CE	Chip Enable Input	2202.1	4558.7
PD	Power Down Input	2048.1	4898.2
EOM	End of Message	1648.1	4865.7
XCLK	No Connect (optional)	1221.1	4898.2
P/ \bar{R}	Playback/Record	965.6	4898.2
V _{CCD}	V _{CC} Digital Power Supply	646.1	4895.7

Figure 13: ISD2560/75/90/120 Products Future Bonding Physical Layout¹ (Unpackaged Die)

ISD2560/75/90/120X²

- I. Die Dimensions
 X: 149.5 ± 1 mils
 Y: 262.0 ± 1 mils
- II. Die Thickness²
 11.8 ± .4 mils
- III. Pad Opening
 111 x 111 microns
 4.4 x 4.4 mils



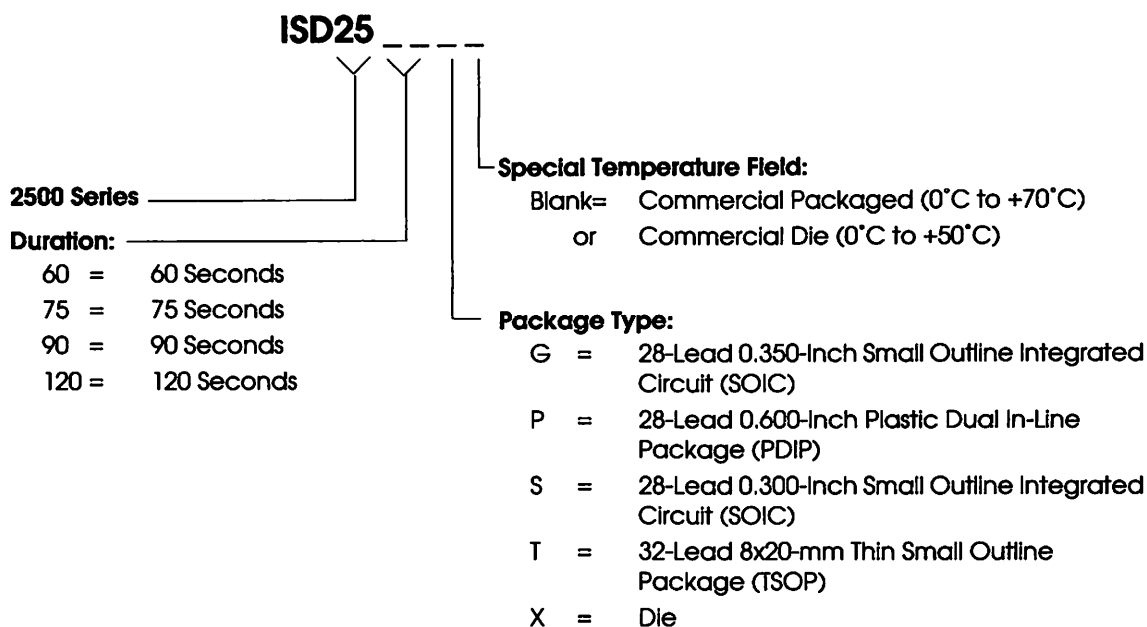
- 1. The backside of die is internally connected to V_{SS}. It **MUST NOT** be connected to any other potential or damage may occur.
- 2. Die thickness is subject to change, please contact ISD factory for status and availability.

Table 21: ISD2560/75/90/120 Products *Future* PIN/PAD Designations, with Respect to Die Center (μm)

Pin	Pin Name	X Axis	Y Axis
A0	Address 0	-897.9	3135.2
A1	Address 1	-1115.4	3135.2
A2	Address 2	-1331.0	3135.2
A3	Address 3	-1544.0	3135.2
A4	Address 4	-1640.4	2888.9
A5	Address 5	-1698.2	2671.0
A6	Address 6	-1698.2	2441.5
A7	Address 7	-1731.2	-2583.2
A8	Address 8	-1731.2	-2768.4
A9	Address 9	-1731.2	-3050.8
AUX IN	Auxiliary Input	-1410.1	-3115.7
V _{SSD}	V _{SS} Digital Power Supply	-1112.8	-3096.2
V _{SSA}	V _{SS} Analog Power Supply	-407.8	-3138.5
SP+	Speaker Output +	-47.4	-3067.7
SP-	Speaker Output -	386.9	-3067.7
V _{CCA}	V _{CC} Analog Power Supply	746.5	-3110.4
MIC	Microphone Input	1101.2	-3146.0
MIC REF	Microphone Reference	1294.7	-3146.0
AGC	Automatic Gain Control	1666.4	-3130.3
ANA IN	Analog Input	1728.6	-2654.0
ANA OUT	Analog Output	1700.9	-2411.0
OVF	Overflow Output	1340.9	3121.7
CE	Chip Enable Input	1726.7	2824.4
PD	Power Down Input	1730.5	3094.0
EOM	End of Message	1340.9	3121.7
XCLK	No Connect (optional)	986.5	3160.7
P/ \bar{R}	Playback/Record	807.2	3163.4
V _{CCD}	V _{CC} Digital Power Supply	544.7	3159.2

ORDERING INFORMATION

Product Number Descriptor Key



When ordering ISD2560/75/90/120 products, please refer to the following valid part numbers.

Part Number	Part Number	Part Number	Part Number
ISD2560G	ISD2575G	ISD2590G	ISD25120G
ISD2560P	ISD2575P	ISD2590P	ISD25120P
ISD2560S	ISD2575S	ISD2590S	ISD25120X
ISD2560T	ISD2575T	ISD2590T	
ISD2560X	ISD2575X	ISD2590X	

For the latest product information, access ISD's worldwide website at <http://www.isd.com>.

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 from 20 to 200.

The inputs are ground referenced while the output automatically biases 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.

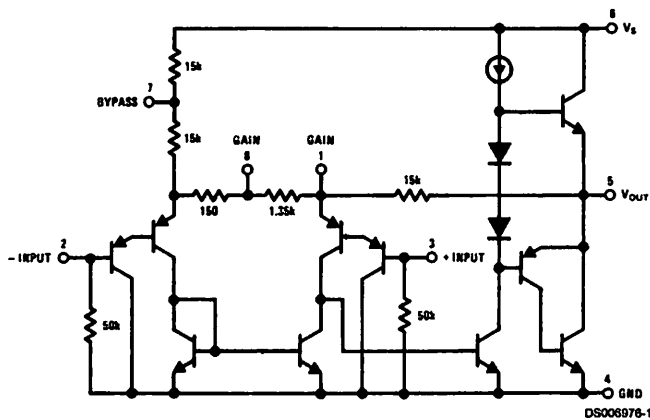
Features

- Battery operation
- Minimum external parts
- Wide supply voltage range: 4V–12V or 5V–18V
- Low quiescent current drain: 4mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion: 0.2% ($A_V = 20$, $V_S = 6V$, $R_L = 8\Omega$, $P_O = 125mW$, $f = 1kHz$)
- Available in 8 pin MSOP package

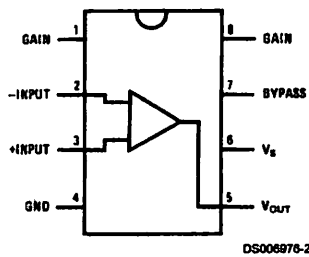
Applications

- 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



Small Outline,
Molded Mini Small Outline,
and Dual-In-Line Packages



Top View
 Order Number LM386M-1,
 LM386MM-1, LM386N-1,
 LM386N-3 or LM386N-4
 See NS Package Number
 M08A, MUA08A or N08E

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (LM386N-1, -3, LM386M-1)	15V
Supply Voltage (LM386N-4)	22V
Package Dissipation (Note 3) (LM386N)	1.25W
(LM386M)	0.73W
(LM386MM-1)	0.595W
Input Voltage	±0.4V
Storage Temperature	-65°C to +150°C
Operating Temperature	0°C to +70°C
Junction Temperature	+150°C
Soldering Information	

Dual-In-Line Package

Soldering (10 sec)

+260°C

Small Outline Package

(SOIC and MSOP)

Vapor Phase (60 sec)

+215°C

Infrared (15 sec)

+220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Thermal Resistance

 θ_{JC} (DIP)

37°C/W

 θ_{JA} (DIP)

107°C/W

 θ_{JC} (SO Package)

35°C/W

 θ_{JA} (SO Package)

172°C/W

 θ_{JA} (MSOP)

210°C/W

 θ_{JC} (MSOP)

56°C/W

Electrical Characteristics (Notes 1, 2) $T_A = 25^\circ\text{C}$

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage (V_S) LM386N-1, -3, LM386M-1, LM386MM-1 LM386N-4		4 5		12 18	V V
Quiescent Current (I_Q)	$V_S = 6V, V_{IN} = 0$		4	8	mA
Output Power (P_{OUT}) LM386N-1, LM386M-1, LM386MM-1 LM386N-3 LM386N-4	$V_S = 6V, R_L = 8\Omega, THD = 10\%$ $V_S = 9V, R_L = 8\Omega, THD = 10\%$ $V_S = 16V, R_L = 32\Omega, THD = 10\%$	250 500 700	325 700 1000		mW mW mW
Voltage Gain (A_V)	$V_S = 6V, f = 1\text{ kHz}$ 10 μF from Pin 1 to 8		26 46		dB dB
Bandwidth (BW)	$V_S = 6V, \text{Pins 1 and 8 Open}$		300		kHz
Total Harmonic Distortion (THD)	$V_S = 6V, R_L = 8\Omega, P_{OUT} = 125\text{ mW}$ $f = 1\text{ kHz, Pins 1 and 8 Open}$		0.2		%
Power Supply Rejection Ratio (PSRR)	$V_S = 6V, f = 1\text{ kHz, } C_{BYPASS} = 10\ \mu\text{F}$ Pins 1 and 8 Open, Referred to Output		50		dB
Input Resistance (R_{IN})			50		k Ω
Input Bias Current (I_{BIAS})	$V_S = 6V, \text{Pins 2 and 3 Open}$		250		nA

Note 1: All voltages are measured with respect to the ground pin, unless otherwise specified.

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and 1) a thermal resistance of 107°C/W junction to ambient for the dual-in-line package and 2) a thermal resistance of 170°C/W for the small outline package.

Application Hints

GAIN CONTROL

To make the LM386 a more versatile amplifier, two pins (1 and 8) are provided for gain control. With pins 1 and 8 open the 1.35 k Ω resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35 k Ω resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitively coupling a resistor (or FET) from pin 1 to ground.

Additional external components can be placed in parallel with the internal feedback resistors to tailor the gain and frequency response for individual applications. For example, we can compensate poor speaker bass response by frequency shaping the feedback path. This is done with a series RC from pin 1 to 5 (paralleling the internal 15 k Ω resistor). For 6 dB effective bass boost: $R \approx 15$ k Ω , the lowest value for good stable operation is $R = 10$ k Ω if pin 8 is open. If pins 1 and 8 are bypassed then R as low as 2 k Ω can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

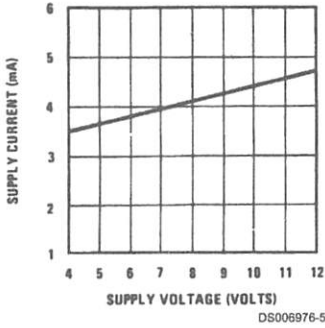
INPUT BIASING

The schematic shows that both inputs are biased to ground with a 50 k Ω resistor. The base current of the input transistors is about 250 nA, so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than 250 k Ω it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than 10 k Ω , then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

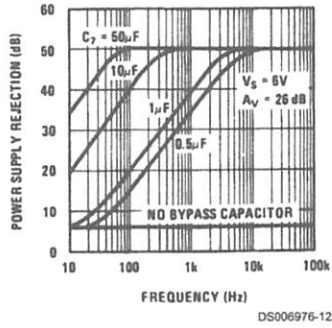
When using the LM386 with higher gains (bypassing the 1.35 k Ω resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1 μ F capacitor or a short to ground depending on the dc source resistance on the driven input.

Typical Performance Characteristics

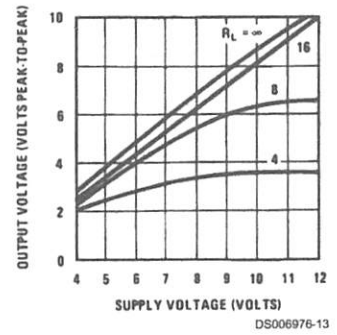
Quiescent Supply Current vs Supply Voltage



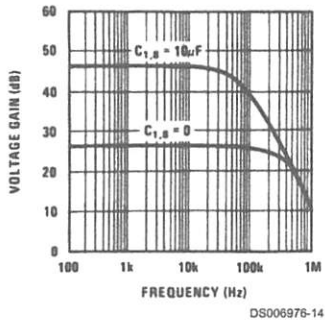
Power Supply Rejection Ratio (Referred to the Output) vs Frequency



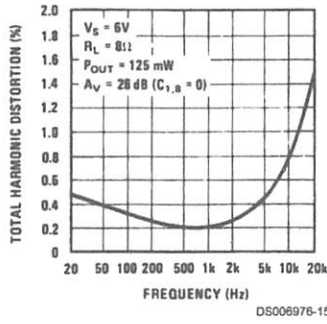
Peak-to-Peak Output Voltage Swing vs Supply Voltage



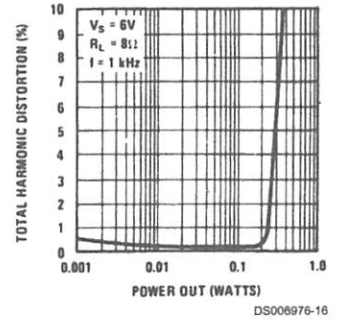
Voltage Gain vs Frequency



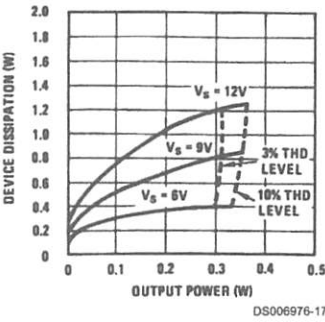
Distortion vs Frequency



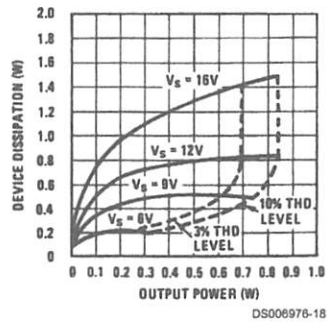
Distortion vs Output Power



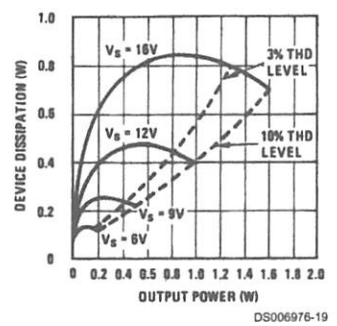
Device Dissipation vs Output Power—4Ω Load



Device Dissipation vs Output Power—8Ω Load

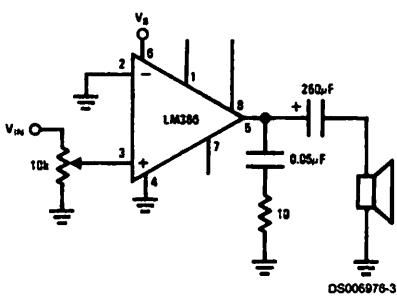


Device Dissipation vs Output Power—16Ω Load

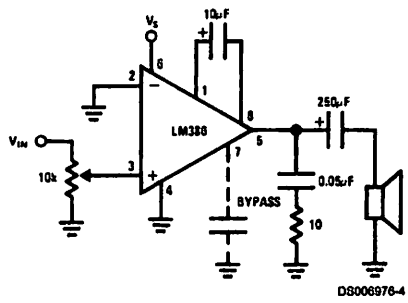


Typical Applications

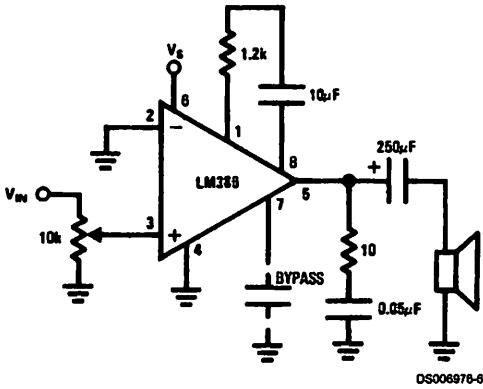
**Amplifier with Gain = 20
Minimum Parts**



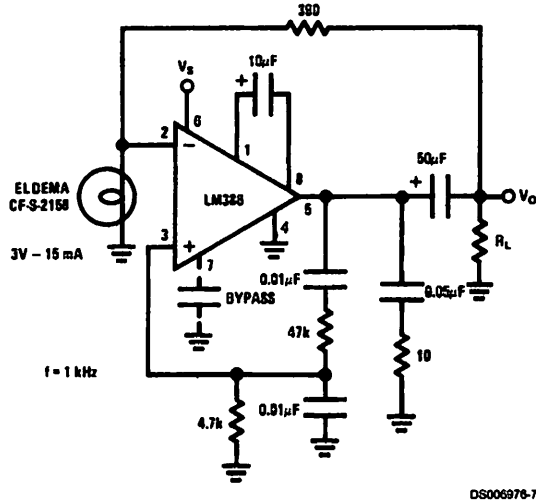
Amplifier with Gain = 200



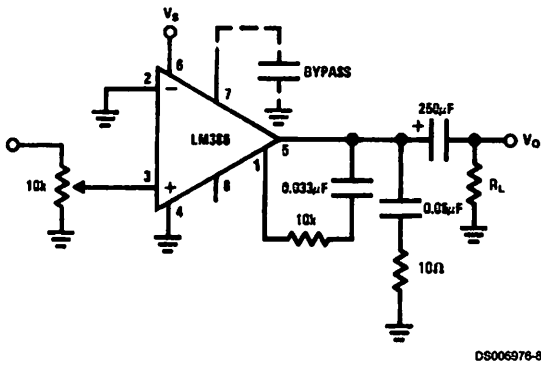
Amplifier with Gain = 50



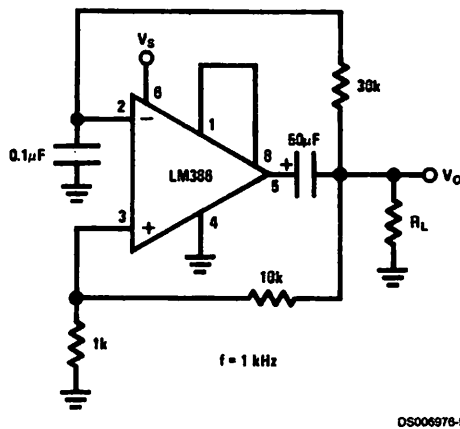
Low Distortion Power Wienbridge Oscillator



Amplifier with Bass Boost

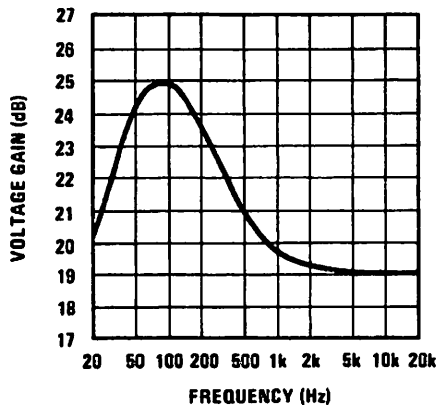


Square Wave Oscillator



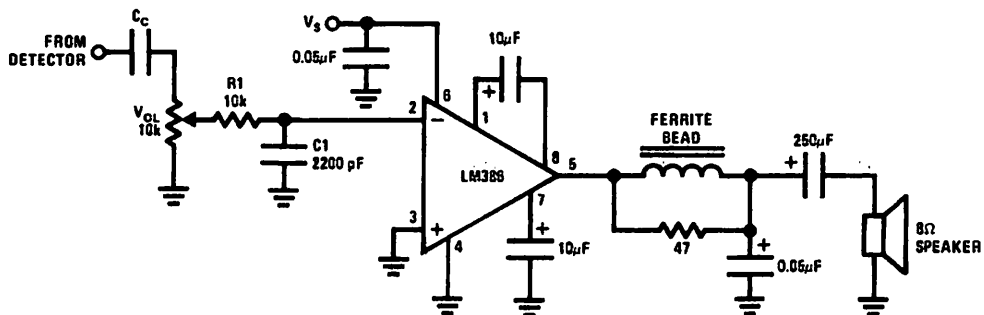
Typical Applications (Continued)

Frequency Response with Bass Boost



DS006976-10

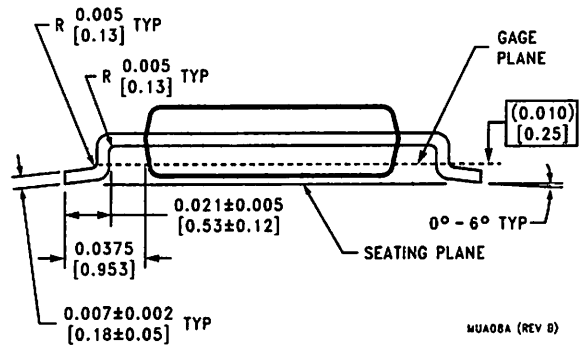
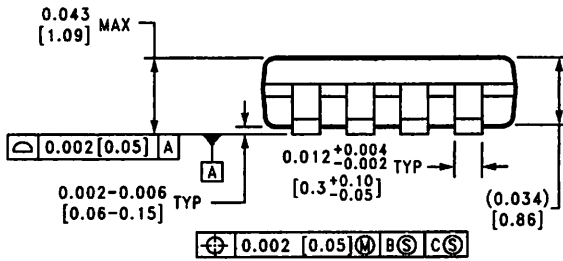
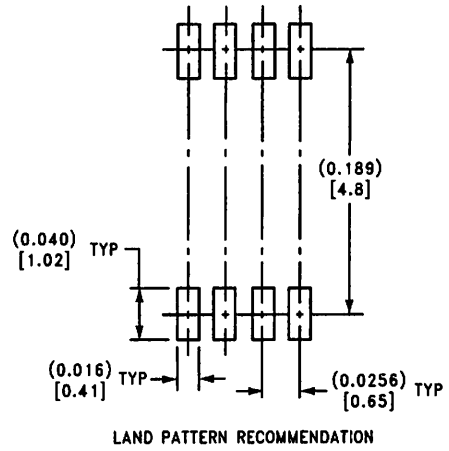
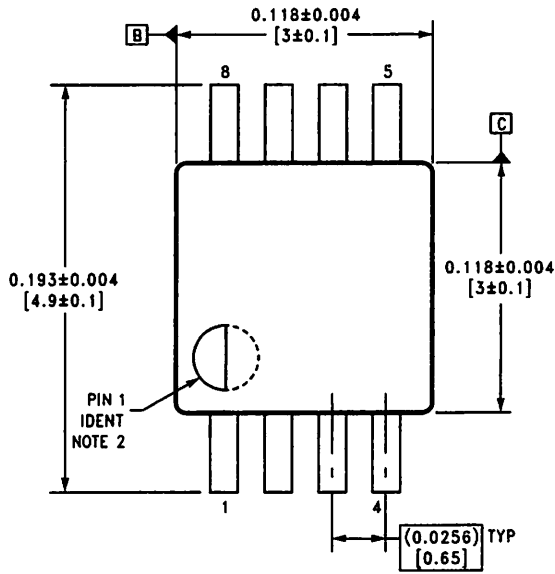
AM Radio Power Amplifier



DS006976-11

- Note 4:** Twist Supply lead and supply ground very tightly.
- Note 5:** Twist speaker lead and ground very tightly.
- Note 6:** Ferrite bead in Ferroxcube K5-001-001/3B with 3 turns of wire.
- Note 7:** R1C1 band limits input signals.
- Note 8:** All components must be spaced very closely to IC.

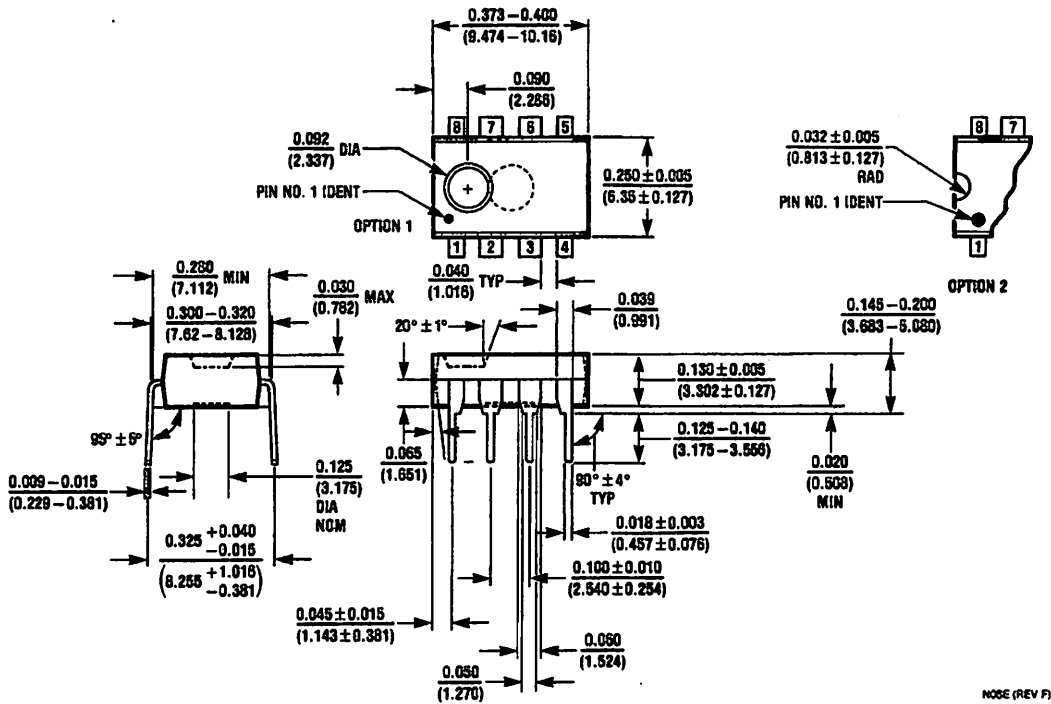
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



MUA08A (REV B)

8-Lead (0.118" Wide) Molded Mini Small Outline Package
Order Number LM386MM-1
NS Package Number MUA08A

Physical Dimensions Inches (millimeters) unless otherwise noted (Continued)



Dual-In-Line Package (N)
Order Number LM386N-1, LM386N-3 or LM386N-4
NS Package Number N08E

NO8E (REV F)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
 Americas
 Tel: 1-800-272-9959
 Fax: 1-800-737-7018
 Email: support@nsc.com
 www.national.com

National Semiconductor Europe
 Fax: +49 (0) 180-530 85 86
 Email: europe.support@nsc.com
 Deutsch Tel: +49 (0) 69 9508 6208
 English Tel: +44 (0) 870 24 0 2171
 Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
 Tel: 65-2544466
 Fax: 65-2504466
 Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
 Tel: 81-3-5639-7560
 Fax: 81-3-5639-7507