SKRIPSI

PERANCANGAN DAN PEMBUATAN KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA BERBASIS MIKROKONTROLLER AT 89S8252

Disusun Oleh:

ANDI PRASETYO W
02.17.055

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

SEPTEMBER 2007

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA BERBASIS MIKROKONTROLLER AT89S8252

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Pada Jurusan Teknik Elektro Strata Satu (S-1)

Disusun oleh :

ANDI PRASETYO WIBOWO

NIM : 02.17.055

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
I Komang Somawirata,ST, MT	**M. Ashar, ST, MT**
NIP.P 1030100361	**NIP.P. 1030500408**

Mengetahui

Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT

NIP.Y. 1039500274

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2007

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Andi Prasetyo Wibowo
NIM : 02.17.055
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : "PERANCANGAN DAN PEMBUATAN KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA BERBASIS MIKROKONTROLLER AT 89S8252"

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada:
Hari : Kamis
Tanggal : 6 September 2007
Dengan Nilai : 82,5 (A)

Panitia Ujian Skripsi

Ketua
(Ir. Mochtar Asroni, MSME)
NIP.Y. 1018100036

Sekretaris
(Ir.F.Yudi Limpraptono, MT)
NIP.Y. 1039500274

Anggota Penguji

Penguji Pertama
(Ir.F.Yudi Limpraptono, MT)
NIP.Y. 1039500274

Penguji Kedua
(Ir. Teguh Herbasuki, MT)
NIP. Y. 1038900209

lembar persembahan

For The God

Allah SWT, yang telah memberikan segala kemudahan sejak awal kuliah hingga sampai de

Papa, ma ecil sampai
dengan seka **a'ku** yang s ayang

Makasih yach bua na ini kamu
lakuin

Thank's yach se gas kuliah

teri a ini.

Buat **HENDRA** n skripsi...

For All

Buat **MAS YUDI** makasih udah banyak bantuin aku dalam skripsiku
dan buat semua temen-temenku yang belum sempat disebutkan semua thank's yach
buat dukungannya selama ini.

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN
KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA
BERBASIS MIKROKONTROLLER AT89S8252

(Andi Prasetyo Wibowo, 02.17.055, Teknik Elektro S.1/Elektronika)
(Dosen Pembimbing : I Komang Somawirata, ST,MT dan M. Ashar, ST, MT)

Kata Kunci : Mikrokontroller AT89S8252, ISD, EEPROM, *Keyboard*

Umumnya sudah banyak kamus yang beredar dimasyarakat saat ini, tetapi kamus yang beredar tidak disertai dengan cara pengucapannya tetapi hanya menampilkan tulisannya saja. Sehingga biasanya orang hanya membaca tulisannya saja, padahal belum tentu tulisan dengan cara pengucapannya itu sama. Untuk itu maka dibuatlah alat yang dapat menterjemahkan bahasa Indonesia – inggris atau sebaliknya dengan keluaran suara. Alat ini dapat menyimpan maksimal 256 kata beserta terjemahannya didalam memori EEPROM Dan lama pencarian datanya rata – rata adalah sekitar 0.5 s. Untuk kapasitas penyimpanan suaranya, ISD memiliki maksimal 120 detik untuk menyimpan suara. Oleh karena itu alat ini dapat digunakan untuk mempermudah seseorang dalam mempelajari bahasa Inggris atau bahasa Indonesia.

KATA PENGANTAR

Alhamdulillah, dengan memanjatkan puji syukur kehadirat Allah SWT, yang telah memberikan rahmat, hidayah serta segala karunia-Nya, akhirnya penyusun dapat menyelesaikan skripsi ini yang berjudul **"Perancangan Dan Pembuatan Kamus Teknik Digital Dengan Output Suara Berbasis Mikrokontroller AT 89S8252"**. Laporan skripsi ini merupakan salah satu persyaratan kelulusan Strata satu Jurusan Teknik Elektro Program Studi Elektronika, Institut Teknologi Nasional Malang.

Keberhasilan penyusunan laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Allah SWT, yang telah memberikan segala kemudahan sehingga laporan skripsi ini dapat terselesaikan dengan baik.
2. Ayah dan Ibu dan keluarga besar atas dukungan moril, spiritual dan biaya serta do'a restunya.
3. Bapak Prof. DR. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
4. Bapak Ir. Mochtar Asroni, MSME, selaku Dekan Institut Teknologi Nasional Malang.
5. Bapak Ir. F. Yudi Limpraptono, MT, selaku Kajur Teknik Elektro S-1
6. Bapak I Komang Somawirata, ST, MT, selaku Dosen Pembimbing I.
7. Bapak M Ashar, ST, MT, selaku Dosen Pembimbing II
8. Puspitasari R yang dengan sabar telah memberikan semangat dan motivasi dalam penyelesaian skripsi ini.
9. Sahabat-sahabatku yang telah banyak membantu terselesainya skripsi ini.

Karena keterbatasan waktu dan faktor lain yang dihadapi sehingga menyebabkan laporan skripsi ini tidak lepas dari banyaknya kekurangan. Semoga laporan skripsi dari pemikiran sederhana ini akan menjadi cikal bakal dari karya yang lebih inovatif dan dapat bermanfaat untuk semua orang.

Malang, September 2007

Penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi yang pesat diperlukan adanya kemampuan dalam menghadapi tantangan di era globalisasi ini. Salah satu kemampuan tersebut adalah kemampuan penguasaan bahasa asing. Bahasa Inggris merupakan salah satu bahasa asing yang sering digunakan dalam berkomunikasi di segala aspek kehidupan.

Tetapi didalam kenyataan terdapat perbedaan antara lisan dan tulisannya.sehingga orang biasanya salah didalam pengucapanya karena hanya membaca tulisannya sedangkan pada pengucapannya, biasanya tidak sama dengan tulisan yang tertera. Maka dari itu dibutuhkan kamus terjemahan bahasa teknik yang lebih dari hanya dapat menerjemahkan bahasa teknik saja, akan tetapi dibutuhkan kamus bahasa teknik yang dapat menerjemahkan bahasa teknik yang dilengkapi dengan cara pengucapannya, sehingga orang dapat mengerti benar arti dan cara pengucapan dari kata yang diterjemahkan.

Hal di atas tentunya memerlukan waktu yang lama apabila harus mencari terjemahan dalam kamus secara manual, meskipun saat ini banyak dijumpai kamus Bahasa Inggris Umum dalam bentuk semacam kalkukator. Akan tetapi hal tersebut masih belum dapat menjadi solusi dalam penyelesaian terjemahan dalam bidang teknik. Apalagi pada kamus Bahasa Inggris Umum tidak dilengkapi dengan cara

pengucapan yang baik dan benar. Dalam skripsi ini penulis akan merancang dan membuat suatu alat yang berisikan terjemahan Bahasa Teknik berbasis mikrokontroller AT 89S8252. Diharapkan alat ini dapat memudahkan penggunanya untuk mencari terjemahan yang diinginkan. Karena hanya dengan memasukkan kata yang dimaksud maka hasil terjemahannya akan ditampilkan dalam layar LCD dan cara pengucapannya pada speaker.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah ini adalah :

1. Bagaimana merancang dan membuat kamus teknik digital (bahasa Inggris / bahasa Indonesia) dan perangkat lunak yang memiliki fasilitas pencarian data (*search engine*),
2. Bagaimana merancang dan membuat agar suara dapat keluar dari *data base* ke speaker sesuai dengan cara pengucapan bahasa Inggris atau bahasa Indonesia.

1.3 Batasan Masalah

Mengacu pada permasalahan yang ada, maka diperlukan adanya batasan-batasan masalah dalam pembahasannya, yaitu :

1. Memori eksternal yang digunakan sebesar 8 kB tetapi memori tersebut dapat ditambah sampai dengan 64 kB sehingga hanya mampu menampung kapasitas sebesar 8.000 karakter.

2. Kata yang dimasukkan hanya mempunyai maksimal karakter sebanyak 16 karakter.
3. Data yang akan disebutkan cara pengucapannya akan dipecah menjadi beberapa suku kata

1.4 Tujuan

Untuk merancang dan membuat suatu alat yang berfungsi sebagai data base terjemahan Bahasa Teknik dengan output berupa suara berbasiskan AT 89S8252.

1.5 Metodologi

Metodologi yang dipakai dalam pembuatan skripsi ini adalah:

1. Studi Literatur

   Dengan mencari referensi-referensi yang berhubungan dengan perencanaan dan pembuatan alat yang akan dibuat.

2. *Field Research*

   Dengan melakukan penelitian secara langsung mengenai objek-objek yang berhubungan langsung dengan perencanaan alat yang akan dibuat.

3. *Design* dan Pembuatan Alat
4. Yaitu meliputi pembuatan PCB, perakitan komponen serta penyolderan dan pembuatan perangkat lunak.
5. Pengujian Alat

   Dengan melakukan pengujian perblok rangkaian dan kerja seluruh sistem pada alat tersebut.

6. Penyusunan Laporan Skripsi

Membuat laporan yang terdiri dari: Pendahuluan, Landasan Teori, Perencanaan dan Pembuatan Alat, Pengujian Alat dan Penutup.

1.6. Sistematika Pembahasan

Sistematika penyusunan skripsi dibagi menjadi beberapa bagian utama, yaitu:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Teori Penunjang

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Perencanaan dan Pembuatan Alat

Perancangan dan perealisasian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

BAB IV Pengujian Alat

Memuat hasil pengujian terhadap alat yang telah direalisasikan.

BAB V Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran

BAB II

TINJAUAN PUSTAKA

Landasan teori sangat membantu untuk dapat memahami suatu sistem. Selain dari pada itu dapat juga dijadikan sebagai bahan acuan didalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya.

2.1 Mikrokontroller AT89S8252

2.1.1 Pendahuluan

Perbedaan mendasar antara mikrokontroller dengan mikroprosesor adalah mikrokontroller selain memiliki *Central Processing Unit (CPU)* juga dilengkapi dengan memori, input-output yang merupakan kelengkapan sebagai minimum sistem mikrokomputer sehingga sebuah mikrokontroller dapat dikatakan sebagai mikrokomputer dalam keping tunggal (*Single Chip Microcomputer*) yang dapat berdiri sendiri.

Mikrokontroller AT89S8252 adalah mikrokrokontroller ATMEL yang kompatibel penuh dengan mikrokontroller keluarga MCS-51, membutuhkan daya yang rendah, memiliki *performance* yang tinggi dan merupakan mikrokomputer 8 *bit* yang dilengkapi 8K byte flash PEROM (Programmable and eraseable Read Only Memory) yaitu ROM yang dapat ditulis menggunakan programmer. Serta 2 Kbyte EEPROM (*Electrical Eraseable Programmable Read Only Memory*) internal. Program memori dapat diprogram ulang dalam sistem atau dengan

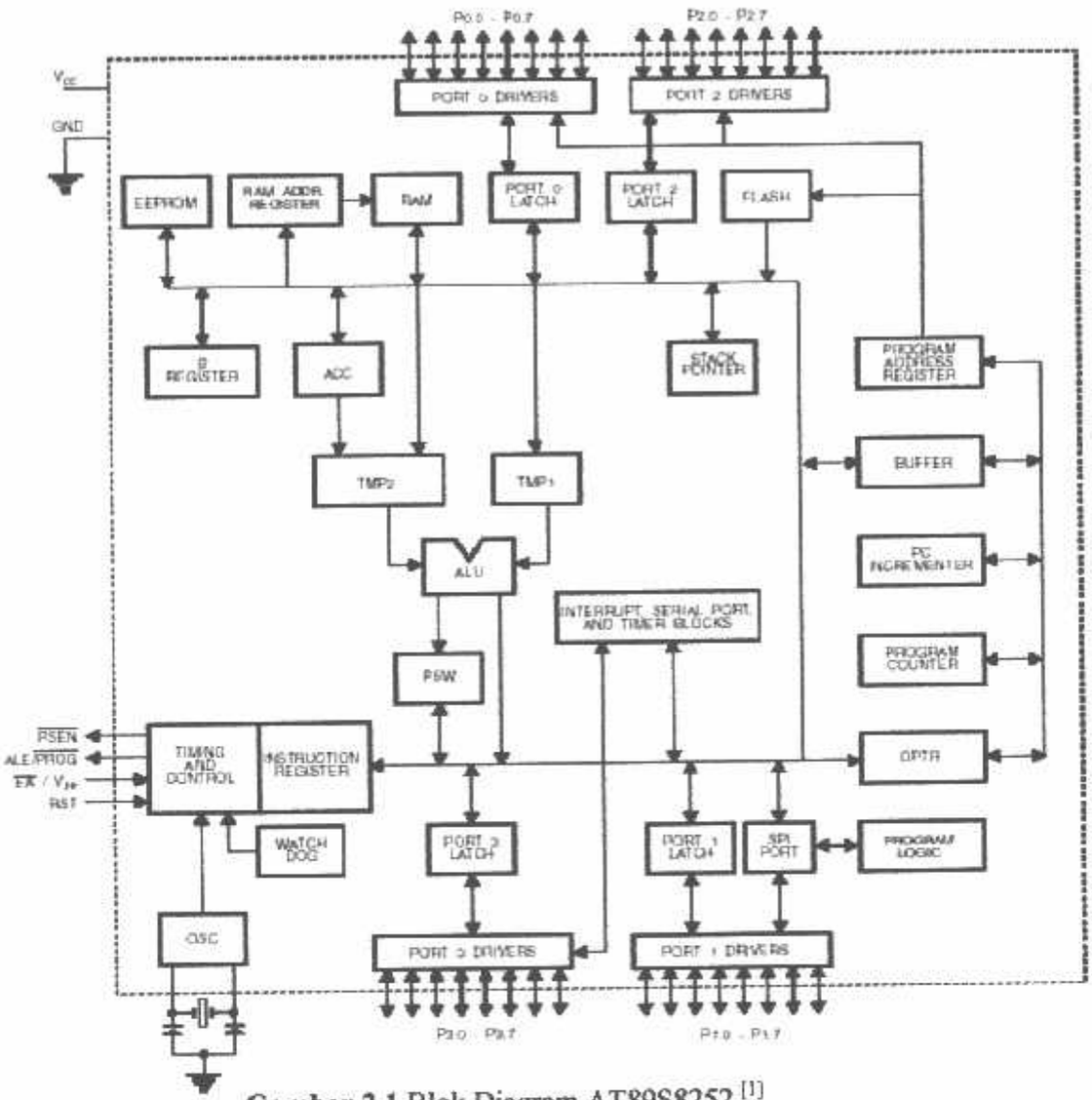
menggunakan *programmer Nonvolately* yang mempunyai kemampuan untuk ditulis ulang hingga 1000 kali dan berisikan perintah standart MCS-51.

2.1.2 Arsitektur AT89S8252

Sebagai *single chip* yaitu suatu sistem mikroprosesor yang terintegrasi, mikrokontroller AT89S8252 mempunyai konfigurasi sebagai berikut:

1. *Central Processing Unit* (CPU) 8 *bit* dengan *register* A (*Accumultor*) dan *register* B
2. 16 *bit Program Counter* (PC) dan *Data Pointer* (DPTR)
3. 8 *bit Program Status Word* (PSW)
4. 8 *bit Stack Pointer* (SP)
5. 2 K *byte internal* EEPROM
6. *256 byte internal* RAM
7. 32 pin input-output tersusun atas P0-P3, masing-masing 8 bit
8. 3 (TO&T1) dengan masing-masing 16 bit *timer / counter*
9. *Receiver register data serial full duplex. Serial Buffer*(SBUF)
10. *Central Register* yaitu TCON, TMOP, SCON, PCON, IP dan IE
11. 9 buah sumber interupsi (2 buah *interrupt* eksternal dan 3 buah *interrupt internal*)
12. *Oscillator* dan *clock internal*

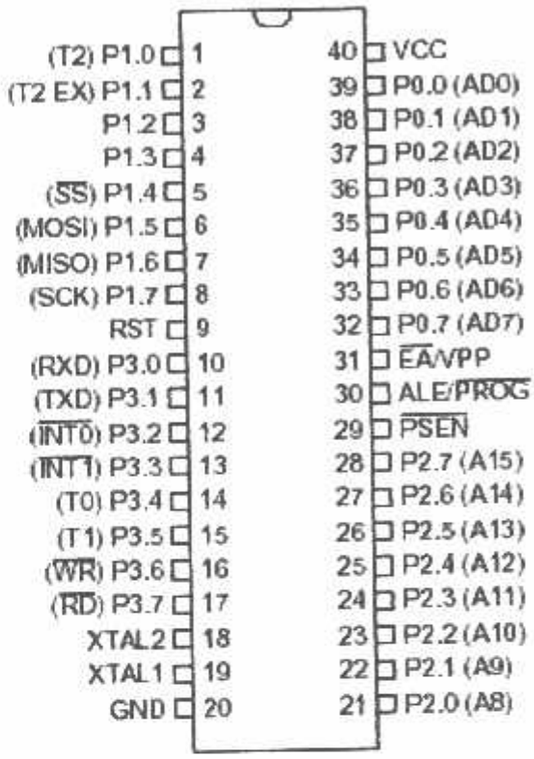
Arsitektur dasar dari mikrokontroller AT89S8252 seperti blok diagram berikut:



Gambar 2.1 Blok Diagram AT89S8252 [1]

2.1.3 Konfigurasi pin pada mikrokontroller AT89S8252

Susunan pin-pin mikrokontroller terdiri dari 40 pin terlihat pada gambar



Gambar 2.2 Konfigurasi Pin AT 89S8252 [1]

Penjelasan dari fungsi tiap-tiap pin adalah sebagaiberikut:

1. VCC berada pin 40 (catu daya)
2. GND (*Ground*) berapa pada pin 20
3. *Port* 0

   Port 0 berada pada pin 32-39 merupakan *port* input-output dua arah dan dikonfigurasi sebagai multipleks *data bus* alamat rendah (A0-A7)dan data selama pengaksesan program memori dan data eksternal

4. *Port* 1

Port 1 berada pada pin 1-8 merupakan *port* input-output dua arah dengan internal *pull up*

5. *Port* 2

Port 2 berada pada pin 21-28 merupakan *port* input-output dengan internal *pull up* mengeluarkan *address* tinggi selama pengambilan program memori eksternal dan selama pengaksesan data ke memori eksternal. Selama pengaksesan ke eksternal data memori port 2 mengeluarkan isi P2 SFR (*Special Function Register*) menerima *address* tinggi dan beberapa sinyal kontrol selama pemrogramman dan verifikasi.

6. *Port* 3

Port 3 berada pada pin 10-17 merupakan input-output dengan internal *pull up*, *port* 3 juga memiliki fungsi khusus yaitu:

- Pin 10 (P3.0) RXD adalah pin input *serial* atau penerima data pada pin *serial* terletak pada *bit address* B0 H.
- Pin 11 (P3.1) TXD adalah pin output serial atau pemancar data pada pin serial terletak pada bit address B1 H.
- Pin 12 (P3.2) INT0 adalah *interrupt* 0 eksternal terletak pada *bit address* B2 H.
- Pin 13 (P3.3) INT1 adalah *interrupt* 1 eksternal terletak pada *bit address* B3 H
- Pin 14 (P3.4) T0 adalah input eksternal timer 0 terletak pada *bit address* B4 H

- Pin 15 (P3.5) T1 adalah input eksternal timer 1 terletak pada *bit address* B5 H
- Pin 16 (P3.6) WR adalah *strobe* tulis data memori eksternal, terletak pada *bit address* B6 H
- Pin 17 (P3.7) RD adalah *strobe* baca data memori eksternal, terletak pada *bit address* B7 H

7. Reset (RST)

   Input reset pada pin 9 adalah reset master untuk mikrokontroller AT89S8252, perubahan tegangan rendah ke tinggi akan mereset mikrokontroller AT89S8252.

8. ALE/ PROG (Pin 30)

   Pulsa output ALE (*Address Latch Enable*) digunakan untuk proses latching byte address rendah (A0-A7) selama pengaksesan ke eksternal memori. Pin ini juga digunakan untuk memasukkan pulsa program (PROG) selama pemrogramman.

9. PSEN (*Program Strobe Enable*)

   Merupakan *strobe* baca ke program memori eksternal yang ada pada pin 29.

10. EA

    EA (*External Address Enable*) terdapat pada pin 31 digunakan untuk mengakses memori eksternal, untuk mengakses memori internal maka EA dihubungkan ke VCC(+5V).

11. XTall 1 dan XTall 2 (pin18-19)

Pin ini dihubungkan dengan kristal apabila menggunakan *oscillator* internal XTall 1 merupakan input *inverting oscillator amplifier* sedangkan XTall 2 merupakan output *inverting amplifier*.

2.1.4 Memori Program (ROM)

Memori program merupakan tempat penyimpan data yang permanen. Memori program merupakan memori yang hanya dapat dibaca atau lebih dikenal dengan *Read Only Memory* (ROM). Data dalam ROM tidak akan terhapus meskipun catu daya dimatikan bersifat *nonvolatile*.

Mikkrokontroller AT89S8252 memiliki program internal 8 K *byte* dengan ruang alamat 0000H-0FA0H. Jika alamat-alamat program lebih tinggi dari 0FA0H yang melebihi kapasitas ROM internal, menyebabkan mikrokontroller AT89S8252 secara otomatis mengambil *byte* code dari program memori eksternal, kode *byte* juga dapat diambil hanya dari eksternal memori dengan alamat 0000H-FFFFH dengan cara menghubungkan pin EA ke ground.

Ada beberapa tipe ROM, diantaranya adalah ROM murni yaitu memori yang sudah di program oleh pabrik, PROM, EPROM dan EEPROM, PROM merupakan memori yang dapat di program oleh pemakai tetapi tidak dapat di program ulang, EPROM merupakan PROM yang dapat di program ulang, pada EPROM ditandai dengan adanya jendela kaca pada konstruksinya yang digunakan untuk menghapus program yang akan diisi program baru (di program ulang), EEPROM prinsipnya sama dengan EPROM perbedaannya terletak pada pengisian dan penghapusan program untuk EPROM menggunakan sinar ultraviolet

sedangkan pada EEPROM penghapusan / pengisian secara langsung dengan tegangan sehingga penggunaan EEPROM lebih fleksibel dibanding EPROM.

2.1.5 Memori Data (RAM)

Memori data adalah tempat untuk menyimpan data yang sifatnya sementara, sehingga pada memori data bersifat *volatile*, yaitu data akan hilang jika tidak diberi catu daya. Memori data lebih dikenal dengan nama *Random Access Memory* (RAM)

Ruang memori data (RAM) *internal* kapasitas 256 *byte* yaitu 00H-FFH terbagi menjadi 3 daerah yaitu:

1. *4 Bank Register*

setiap *bank* terdiri dari 8 *register* (R0-R7) sehingga jumlah *register* untuk keempat *bank register* (bank0-bank3) menjadi 32 buah *register* yang menempati ruang alamat 00H-1FH. Cara mengaktifkan salah satu *bank register* yaitu dengan mengatur RS0-RS1 pada *Program Status Word (PSW)*

Tabel 2.1 Pengaturan RS0-RS1Untuk *Register Bank*

RS1	RS0	Selek *Register Bank*	Address
0	0	*Bank* 0	00H-07H
0	1	*Bank* 1	08H-0FH
1	0	*Bank* 2	10H-17H
1	1	*Bank* 3	18H-1FH

2. *Bit Address Area*

Terdiri dari 16 *byte* yang dimulai dari alamat 20H-2FH masing-masing 208 *bit* lokasi yang dapat dialamati secara langsung.

3. *Stratch Pad Area*

Terdiri dari 208 *byte* terletak pada alamat 30H-FFH, yang dapat dialamati secara langsung dan digunakan untuk keperluan umum (*General Purpose*) misalnya digunakan untuk lokasi *stack*.

2.1.6 SFR *(Special Functional Register)*

Untuk operasi AT89S8252 yang menggunakan alamat internal RAM (00H-FFH), tetapi tidak semua alamat tersebut digunakan SFR_s. Berikut ini adalah contoh dari vektor alamat pada SFR_s.

- *Accumulator* (ACC) atau *register* A dan *register* B, kedua *register* ini digunakan untuk operasi perkalian dan pembagian
- *Program Status Word*, *register* ini meliputi : CY (*Carry*), AC (*Auxcilliary carry*), FO(*Flag*), RS0 dan RS1 (untuk pemilihan *register bank*), OV (*overflow*) dan *Parity* (*parity flag*).
- *Stack Pointer* merupakan *register* yang digunakan untuk menunjuk alamat, *register* ini digunakan bila terdapat suatu *routine* pada program utama.
- *Data Pointer Two Byte Register* (DPTR), DPTR adalah suatu *register* yang digunakan untuk pengalamatan yang tidak langsung, *register* ini digunakan untuk mengakses memori program baik secara eksternal maupun internal yang digunakan untuk alamat eksternal data, DPTR ini dikontrol oleh 2 buah *register* 8 bit yaitu *data pointer low* (DPL) dan *data pointer high* (DPH).
- *Port0, Port1, Port2* dan *Port3,* masing-masing dapat dialamati baik secara *byte* maupun bit, masing-masing merupakan *port bi directional* yang artinya dapat berjalan dua arah (input dan output), *port0* dan *port2* digunkan untuk

pengalamatan dari luar, *port1* untuk I/O dari mikrokontroller, sedangkan *port3* berisi sinyal-sinyal kontrol seperti *interrupt*, *serial*, WR dan RD.

- *Register Priority Interrupt*, merupakan *register* yang berisi bit-bit untuk mengaktifkan prioritas dari suatu *interrupt* yang ada pada mikrokontroller pada taraf yang diinginkan.
- *Interrupt Enable Register*, merupakan register yang berisi bit-bit untuk menghidupkan atau mematikan sumber *interrupt*
- *Timer/Counter Control* (TCON) *Register*, TCON merupakan suatu *register* yang berisi bit-bit untuk memulai atau menghentikan pencacah / pewaktu
- *Serial Control Buffer* (SBUF), register ini digunakan untuk menampung data masukan (SBUF IN) atau keluaran (SBUF OUT) dari *serial port*

Berikut adalah *address* yang digunakan SFR,

Tabel 2.2 *Special Function Register* (SFR) [2]

Simbol	Nama	*Address*
ACC	*Accumulator*	E0H
B	B *register*	F0H
PSW	*Program Status Word*	D0H
SP	*Stack Pointer*	81H
DPTR	*Data Pointer Two Byte Register*	-
DPL	*Data Pointer Low Byte*	82H
DPH	*Data pointer High Byte*	83H
P0	*Port*0	80H

P1	*Port1*	90H
P2	*Port2*	A0H
P3	*Port3*	B0H
IP	*Interrupt Parity Control*	B8H
IE	*Interrup Enable Control*	ABH
TMOD	*Timer / Counter Mode Control*	89H
TCON	*Timer / Counter Control*	88H
T2CON	*Timer / Counter 2 Control*	C8H
TH0	*Timer / Counter 0 High Control*	8CH
TL0	*Timer / Counter 0 Low Control*	8AH
TH1	*Timer / Counter 1 High Control*	8DH
TL1	*Timer / Counter 1 Low Control*	8BH
TH2	*Timer / Counter 2 High Control*	CDH
TL2	*Timer / Counter 2 Low Control*	CCH
RCAP2H	T/C *Capture Register High Byte*	CBH
RCAP2L	T/C *Capture Register Low Byte*	CAH
SCON	*Serial Control*	98H
SBUF	*Serial Data Buffer*	99H
PCON	*Power Control*	87H

2.1.7 *Program Status Word* (PSW)

Cara mendefenisikannya *register* ini ditunjukkan dalam tabel 2.3 :

Tabel 2.3 Pengaturan RS0-RS1 *Bank Register* [3]

Data	Simbol	Posisi	Fungsi /Art-1
D0	P	PSW.0	*Parity flag*
D1	-	PSW.1	*Flag* didefinisikan oleh pemakai.
D2	OV	PSW.2	*Overflow Flag*
D3	RS0	PSW.3	Bit pemilih *bank register.*
D4	RS1	PSW.4	Bit pemilih *bank register.*
D5	F0	PSW.5	*Flag 0*
D6	AC	PSW.6	*Auxiliary CarryFlag*
D7	CY	PSW.7	*Carry Flag*

2.1.8 *Power Control* (PCON)

Untuk *register* cara mendefenisikannya ditunjukkan dalam tabel 2.3 :

Tabel 2.4 Skema Medefinisikan PCON [3]

Data	Simbol	Fungsi /Arti
D0	IDL	*Idle* mode bit
D1	PD	*Power Down* bit
D2	GF0	*Bit flag* serbaguna.

D3	GF1	Bit *flag* serbaguna.
D4	-	Tidak dipakai.
D5	-	Tidak dipakai.
D6	-	Tidak dipakai.
D7	SMOD	Digunakan untuk menghasilkan *baudrate* dan SMOD_1, maka *baudrate* akan *double* baik mode 0,1,2 atau 3.

2.1.9 Sistem *Interrupt*

Mikrokontroller AT89S8252 mempunyai 9 buah sumber *interrupt* yang dapat membangkitkan permintaan *interrupt* yaitu INT0, INT1, T0, T1 *Port serial* dan beberapa port lainnya. Saat terjadinya *interrupt*, mikrokontroller secara otomatis akan menuju ke *subroutine* pada alamat tersebut. Setelah *interrupt service* selesai dikerjakan mikrokontroller akan mengerjakan program semula. Dua sumber *interrupt* eksternal adalah INT0 dan INT1, dimana kedua interruptsi eksternal akan aktif pada transisi rendah selain itu juga ada *Timer/Counter* 0, *Timer/Counter* 0 dan interupsi dari port serial(*receiver*). Interupsi serial dibangkitkan dengan melakukan operasi OR pada R1 dan T1 tiap-tiap interupsi dapat enable atau disable secara *soft ware*. Tingkat prioritas semua sunber *interrupt* dapat diprogram sendiri-sendiri dengan *set* atau *clear* bit pada SFRS IP(*Interrupt Priority*)

Tabel 2.5 Alamat Sumber Interrupt [3]

Sumber Interrupt	Alamat Awal
Power on reset	0000h
Interrupt luar 0 (INT 0)	0003h
Pewaktu/Pencacah 0 (T0)	000Bh
Interrupt Luar 1 (INT 1)	0013h
Pewaktu/Pencacah 1 (T1)	001Bh
Port 110 *Serial*	0023h

2.1.10 *Timer / Counter*

Pengendalian kerja *timer / counter* dilakukan dengan pengaturan *register* yang berhubungan dengan kerja *timer / counter* yaitu melalui sebuah *timer counter mode control*.Untuk menaktifkan *timer / counter* yang meliputi penentuan fungsi sebagai *timer* atau sebagai *counter* serta pemilihan mode operasi dapat diatur melalui TMOD. Konfigurasi dari register TMOD seperti yang ditunjukkan dalam table berikut :

Tabel 2.6 Register TMOD [3]

Data	Simbol	Posisi	Fungsi/Arti
D0	IT0	TCON.0	*Interrupt 0 type control bit*
D1	IE0	TCON.1	*External interrupt 0 edge flag*
D2	IT1	TCON.2	*Interrupt type 1 control bit*. Diatur oleh *software* untuk menentukan aktif *low* atau *high trigger* dari external

D3	IE1	TCON.3	*External interrupt 1 edge flag.* Diatur oleh *hardware* ketika *external interrupt terditeksi* dan nol-kan melalui *software* ketika interrupt diproses
D4	TR0	TCON.4	*Timer 0 control bit. Diatur oleh software ketika timer counter 0*
D5	TF0	TCON.5	*Timer 0 overlow flag control bit.* Diatur oleh *software ketika timer/counter 0 overlow*
D6	TR1	TCON.6	*Timer 1 control bit.* Diatur oleh *software ketika timer counter 0*
D7	TF1	TCON.7	*Timer 1 overlow flag control bit.* Diatur oleh *software ketika timer/counter 0 overlow*

Pengaturan *timer / counter* dilakukan pertama kali dengan mengatur TMOD. Disini akan menentukan fungsi sebagai *timer* atau *counter* lengkap dengan spesifikasinya. Demikian juga dengan pengaktifan interrupsi yang berhubungan dengan penggunaan mode ini.

Tabel 2.7 Mode Operasi *Timer / Counter* [3]

M1	M2	Operation Mode
0	0	*Timer / counter* 13 bit
0	1	*Timer / counter* 16 bit
1	0	8 bit *auto reload timer / counter*
1	1	TL0 dari *timer* adalah 8 bit *timer / counter* dikendalikan oleh

		kontrol bit *timer*0, TH0 adalah *timer* 8 bit yang dikendalikan oleh *timer*1 kontrol bit.

2.1.11 Metode Pengalamatan

Bentuk data dari operand sumber ditentukan oleh jenis mode pengalamatannya. Instruksi-instruksi MCU AT89S8252 dibagi menjadi 4 mode pengalamatan yaitu :

1. Pengalamatan bit (Direct Bit Addressing) :

   Pengalamtan langsung tiap bit ini hanya dilakukan pada lokasi RAM internal yaitu 20H-2FH, dan sebagian SFR yaitu port 0, port 1, port 2, port 3, TCON register, SCON register, IE register, PSW register, ACC dan B register.

2. Pengalamatan tak langsung (Indirect Bit Addressing) :

   Pada pengalamatan tak langsung, intruksi menunjukkan suatu register yang isinya adalah alamat dari operand, eksternal dan internal RAM dapat dialamati secara tidak langsung. Register alamat untuk data dengan lebar 8 bit dapat berupa R0 dan R1 yang digunakan untuk memilih angka register atau stack pointer. Register alamat untuk data, dengan lebar 16 bit digunakan data pointer (DPTR).

3. Pengalamatan ber indeks :

   Yang dapat diakses dengan pengalamatan berindeks hanya memory program. Mode ini dimaksudkan untuk membaca look-up table program.

4. Konstanta immediet :

   Pengalamatan langsung dilakukan dengan memberikan nilai ke register secara langsung, dilakukan dengan menggunakan tanda #,(Contoh : Mov A, #100).

2.1.12 Rangkaian *Oscillator*

Jantung dari mikrokontroller AT89S8252 terletak pada rangkaian yang membangkitkan pulsa *clock*. Pin Xtal1 dan Xtal2 disediakan untuk disambungkan dengan jaringan resonan untuk membentuk sebuah *oscillator*. Sedangkan untuk jaringan resonannya menggunakan kristal, karena dari beberapa jenis kristal yang ditemukan dialam menunjukkan efek piezoelektrik bila penerapannya tegangan ac melintasi bahan bahan ini maka kristal tersebut bergetar dengan frekwensi yang sama dengan frekwensi tegangan yang diterapkan, sebaliknya bila dipaksa untuk bergetar maka akan membangkitkan tegangan ac.

Rumus untuk frekwensi dasar sebuah kristal adalah $F = \frac{K}{t}$

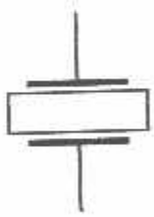
Dimana :

K : tetapan yang tergantung dari jenis potongan dan unsur lainnya

t : ketebalan kristal

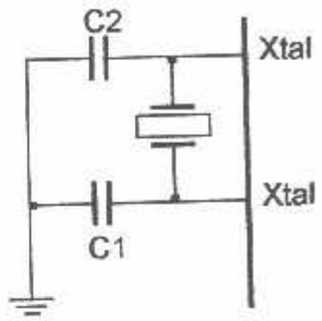
F : frekwensi

Adapun simbol dari kristal adalah seperti gambar 2.3 berikut ini.



Gambar 2.3 Simbol Kristal

AT89S8252 dirancang untuk running pada frekwensi 3 MHz sampai 24 MHz. penambahan rangkaian oscillator ditunjukkan pada gambar 2.4 berikut ini.



Gambar 2.4 Rangkaian Oscilator [4]

2.2 LCD (Liquid Crystal Display)

LCD (liquid Crystal Display) adalah suatu jenis piranti output yang menggunakan daya rendah dengan pengontrol kontras dan kecerahan. Pengontrol utamanya dan karakter ada pada ROM (Read Only Memory) generator dan display data RAM (Random Access Memory) yang akan menghasilkan extended key codes (kode tombol/keyboard standart internasional dalam Hexsa) jika padanya diberikan inputan. Untuk mendapatkan fungsi dengan baik maka perlu diperhatikan proses inisialisasi yang telah ditentukan oleh pabrik pembuatnya. Timming penginisialisasian sangat perlu dipertimbangkan, karena jika meleset sampai orde millisecond, maka dapat dipastikan LCD itu tidak dapat berfungsi.

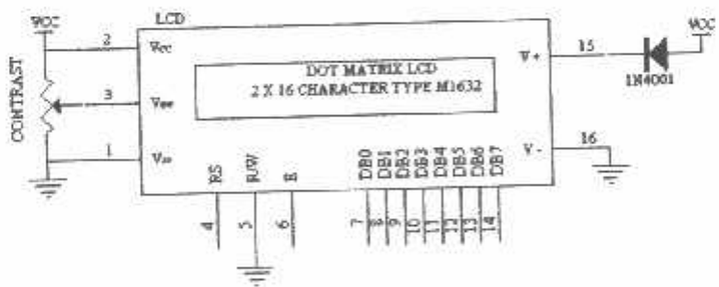
Ada dua jenis register yang terdapat dalam LCD M1632 ini, yaitu data register dan instruction register. Dengan menggunakan pin RS (Register Selact) pada LCD, pemakaian kedua register dapat dipilih. Pemilihan register dapat dipilih. Pemilihan register pada LCDditunjukkan dalam table berikut ini :

Tabel 2.8 Pemilihan Register Pada LCD M1632 [5]

Nama Sinyal	No. Terminal	I/O	Tujuan	Keterangan Sinyal
RS	4	Input	MPU	0 : Instruction Register 0 : Data Register

Jika bagian yang dipilih adalah instruction registermaka output yang dihasilkan adalah meliputi operasional dari LCD, misalnya fungsi display clear, cursor home, entry mode set, display on/off, cursor shift, dan sejenisnya. Sebaliknya, jika bagian yang dipilih adalah data register, output yang dihasilkan adalah meliputi karakter yang tabelnya terdapat pada lampiran data sheet LCD.

Berikut adalah gambar dari LCD dengan pin- pin yang terhubung dengan mikrokontroller AT89S8252 :



Gambar 2.5 Modul LCD 2 ×16 karakter [5]

LCD M1632 mempunyai spesifikasi sebagai berikut :

- 16 karakter 2 baris dalam bentuk dot matrik 5 × 7 dan kursor.
- *Duty ratio* 1/16.

- Memiliki ROM pembangkit karakter untuk 192 jenis karakter.
- RAM untuk data *display* sebanyak 80 × 8 bit (80 karakter maksimum).
- Dapat dirangkai dengan MPU (*Mikroprocessor Unit*) 8 bit atau 4 bit.
- RAM data *display* dan RAM pembangkit karakter dibaca oleh MPU.
- Memiliki fungsi intruksi : *display ON/OFF, cursor ON/OFF, display character blink, cursor shift dan display shift.*
- Memiliki rangkaian oscillator sendiri.
- Sumber tegangan tunggal +5 volt.
- Memiliki rangkaian reset otomatis pada catu daya dihidupkan.
- Temperature operasi 0^{0}-50^{0} C.

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

Tabel 2.9 Fungsi Pin – Pin LCD [5]

No. PIN	Nama PIN	Fungsi
1	Vss	Terminal Ground
2	Vcc	Tegangan Catu + 5 volt
3	Vee	Mengendalikan kecerahan LCD
4	RS	Sinyal pemilihan register 0 = Tulis 1 = Baca
5	R/W	Sinyal seleksi tulis atau baca 0 = Tulis 1 = Baca
6	E	Sinyal operasi awal yang mengaktifkan data tulis atau baca

7 – 14	**DB0 – DB7**	Merupakan saluran data berisi perintah data yang akan ditampilkan
15	**V + BL**	Back Light Supply 4 - 4,2 (Volt)
16	**V – BL**	Back Ligth Supply 0 (Ground)

Pada LCD juga terdapat instruksi – instruksi sebagai berikut :

- *Display clear* : membersihkan tampilan yang ada pada LCD serta menyimpan, sedangkan kursor kembali ke posisi semula.
- *Cursor home* : hanya membersihkan tampilan dan kursor kembali ke semula.
- *Empty mode Set* : layar beraksi sebagai tampilan tulis.

  S : 1/0 = menggeser layar.

  1/0 : 1 = kursor bergerak ke kanan dan layar bergerak ke kiri.

  1/0 : 0 = kursor bergerak ke kiri dan layar bergerak ke kanan

- *Display On/Off* kontrol.

  D : 1 = layar on

  D : 0 = layar off

  C : 1 = kursor on

  C : 0 = kursor off

  B : 1 = kursor berkedip-kedip

  B : 0 = kursor tidak berkedip – kedip

- *Cursor Display Shift*

  S/C : 1 = LCD diidentifikasikan sebagai layar

  S/C : 0 – LCD diidentifikasikan sebagai kursor

  R/L : 1 = menggeser satu spasi ke kanan

  R/L : 0 = menggeser satu spasi ke kiri

- *Fuction Set*

  DL : 1 = panjang data LCD pada 8 bit

  DL : 0 = panjang data LCD pada 4 bit

  Bit upper ditransfer terlebih dahulu kemudian diikuti dengan 4 bit lower.

  N : 1/0 = LCD menggunakan 2 atau 1 baris karakter

  P : 1/0 = LCD menggunakan 5 x 10 dot matrik

- *CG RAM address set* : menulis alamat RAM ke karakter

- *DD RAM address set* : menulis alamat RAM ke tampilan

- *BF/address set* : BF = 1/0, LCD dalam keadaan sibuk atau tidak sibuk.

- *Data write to CG RAM or DD RAM* : membaca byte dari alamat terakhir RAM yang dipilih.

2.2.1 Sinyal Interface M1632

Untuk berhubungan dengan mikrokontroler pemakai, M1632 dilengkapi dengan 8 jalur data (**DB0..DB7**) yang dipakai untuk menyalurkan kode ASCII

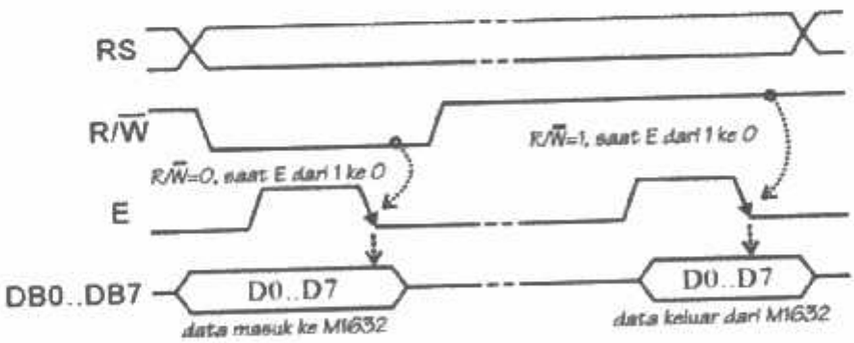
maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor.

Kombinasi lainya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas Intel dengan kombinasi sinyal **WR** dan **RD.**.

RS, singkatan dari *Register Select*, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

Proses mengirim/mengambil data ke/dari M1632 digambarkan dalam gambar 2-26 bisa dijabarkan sebagai berikut :



Gambar 2.6 Mengirim/Mengambil Data Ke/Dari M1632 [7]

1. **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.

2. **R/W** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB0..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB0 .. DB7** diterima oleh M1632.

3. Untuk mengambil data dari M1632 sinyal **R/W** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saatu **E** menjadi 1, M1632 akan meletakkan datanya di **DB0 .. DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

2.3. Keyboard PC

2.3.1 Tata Kerja Keyboard PC

Setiap kali salah satu tombol *keyboard* ditekan atau dilepas, *Keyboard* akan mengirim kode ke *host* (*host* adalah komputer jika *keyboard* dihubungkan ke PC, atau berupa mikrokontroller jika *keyboard* dihubungkan ke peralatan berbasis mikrokontroller).

Komunikasi antar *keyboard* dan host adalah komunikasi dua arah, *keyboard* mengirim scan code ke *host*, *host* bisa mengirim perintah untuk mengatur kerja dari *keyboard*. Kode perintah untuk *keyboard* tidak sebanyak *scan code*, berikut ini daftar kode perintah untuk *keyboard* (dalam heksadesimal) adalah:

- **ED**, perintah untuk menyalakan/memadamkan lampu indikator di *keyboard*, setelah menerima perintah **ED** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** –

acknowledge) dan menunggu 1 *byte* perintah lagi dari *host* untuk menentukan lampu indikator mana yang perlu di-nyala/padam-kan. 1 *byte* perintah susulan tersebut akan diartikan sebagai berikut : bit 0 dipakai untuk mengatur lampu indikator ***scroll lock***, bit 1 untuk ***Num Lock*** dan bit 2 untuk ***Capc Lock***, bit-bit lainya diabaikan.

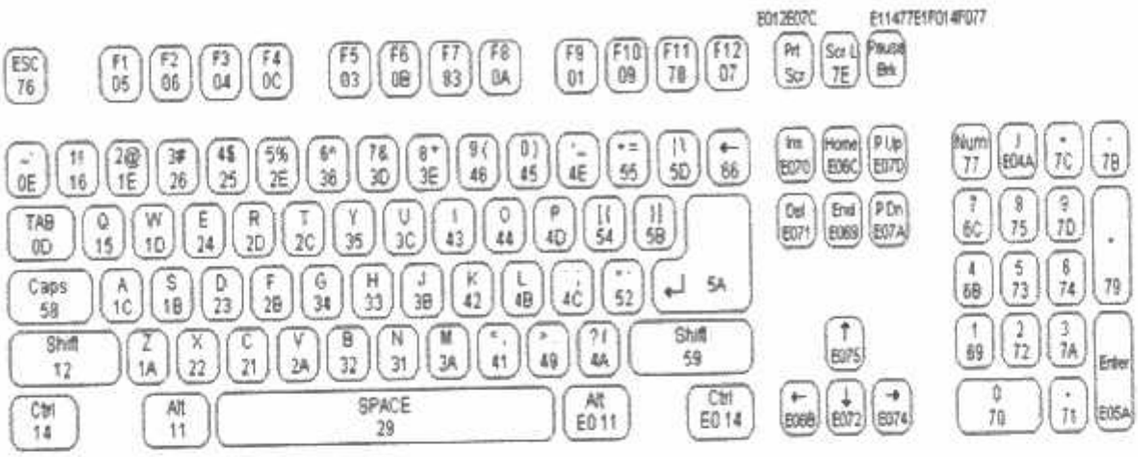
- **EE,** dipakai *host* untuk memeriksa apakah *keyboard* masih aktif. Setelah menerima perintah **EE** dari *host*, *keyboard* akan menjawab dengan **EE** pula, menandakan dirinya masih aktif.
- **FO,** ada keyboard yang dilengkapi 3 set *scan code*, perintah ini dipakai untuk memilih *scan code* yang dipakai. Setelah menerima perintah **FO** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – *acknowledge*) dan *host* menjawab 1 *byte* lagi (nilainya 1, 2 atau 3)untuk memilih set *scan code*. Jika *byte* yang dikirimkan nilainya 0, *keyboard* akan menjawab dengan nomor set *scan code* yang dipakai.
- **F3,** dipakai untuk mengatur kecepatan tanggapan *keyboard* (*Typematic Repeat Rate*), setelah menerima perintah **F3** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – *acknowledge*) dan *host* menjawab 1 *byte* nilai kecepatan tanggapan *keyboard* yang dikehendaki.
- **F4,** dipakai untuk meng-aktif-kan kembali *keyboard*, setelah menerima perintah ini *keyboard* akan menjawab dengan **FA** (**ACK** – *acknowledge*).

- **F5,** dipakai untuk me-nonaktif-kan *keyboard*, setelah menerima perintah ini *keyboard* akan menjawab dengan **FA (ACK** – *acknowledge*).
- **FE,** dipakai meminta *keyboard* mengirim ulang *scan code* terakhir yang dikirim.
- **FF, perintah untuk me-reset *keyboard***

Selain perintah dari *host*, *keyboard* juga mempunyai kode-kode lain selain *scan code* yang dikirimkan ke *host*, sebagai berikut :

- **FA,** berarti **ACK** (*acknowledge*), yaitu jawaban dari *keyboard* bahwa perintah dari *host* sudah dikenali dengan baik.
- **AA,** berarti *keyboard* selesai memeriksa diri dan siap bekerja setelah diberi catu daya.
- **EE,** identik dengan perintah **EE** diatas.
- **FE,** artinya minta *host* mengulang perintah terakhir yang dikirim.
- **FF / 00,** berarti terjadi kesalahan di *keyboard*.

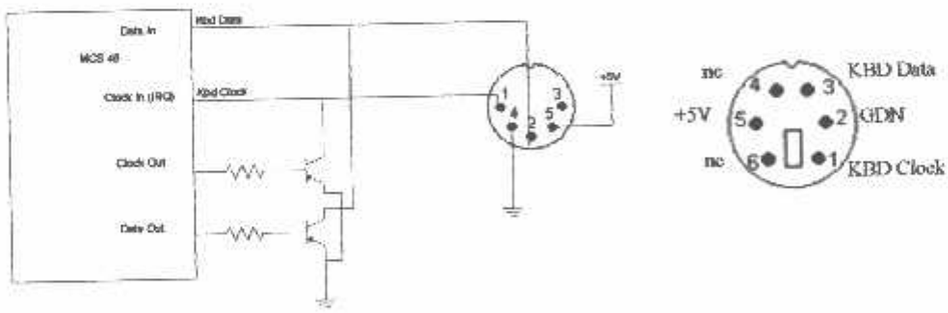
Gambar 2.17 menjelaskan *scan code* masing-masing tombol *keyboard* PC. Terlihat pada gambar tersebut, scan code tidak berupa kode ASCII yang biasa dipakai mewakili huruf, dan ditentukan secara acak juga. Sehingga setelah diterima *host*, scan code harus dirubah menjadi kode ASCII dengan memakai cara 'pencarian tabel'.



Gambar 2.7 *Keyboard PC* dan *Scancode*

2.3.2 Rangkaian Penghubung

Keyboard PC dibangun dengan mikrokontroler MCS48, yang merupakan sudara tua MCS51 tapi jau lebih sederhana. Untuk keperluan membentuk rangkaian penghubung tidak perlu diketahui bagaimana cara kerja mikrokontroler dalam *keyboard*, tapi cukup meninjau rangkaian elektronik bagian penghubung pada gambar 2.18.



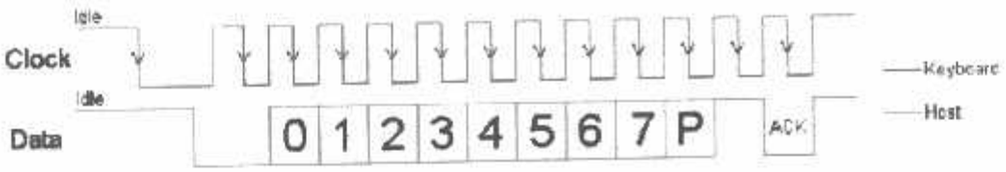
Gambar 2.8 Penghubung dalam *Keyboard* PC

Kbd Clock dibangkitkan oleh MCS 48, merupakan sinyal pendorong **Kbd Data** yang biasa bersumber dari *keyboard* maupun bersumber dari PC. Level tegangan pada kedua sinyal ini memenuhi standart sinyal TTL biasa, jadi bisa langsung dihubungkan ke mikrokontroler. Sumber daya untuk *keyboard* dicatu dari luar, harus diperhatikan kebutuhan arusnya cukup besar bisa sampai 300 mA.

2.3.3 Sinyal Pengiriman Data dari Keyboard

Saat tidak ada pengiriman data, sinyal **Kbd Clock** dan **Kbd Data** dalam keadaan '1'. Sinyal pengiriman data dari *keyboard* dalam gambar 2.17 dijelaskan sebagai berikut:

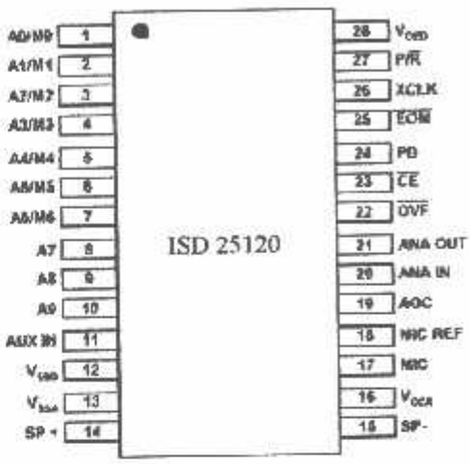
- Data mulai dikirimkan dengan me-nol-kan **Kbd Data** sebagai tanda mulai pengiriman (*start bit*), berapa saat kemudian setelah **Kbd Data** stabil disusul **Kbd Clock** berubah menjadi '0' dan kembali '1' lagi, ini berarti selesai mengirimkan data 1 bit.
- Setelah mengirim *'start bit'*, dikirimkan bit 0, bit 1 dan seterusnya sampai it 7.
- Menyusul dikirim *'parity bit'*, yaitu bit kontrol yang berguna bagi host penerima data untuk memastikan data yang diterima tidak ada kesalahan. Jika banyaknya bit 1' yang terdapat di bit 0 sampai 7 ganjil, *'parity bit'* akan bernilai '1'.
- Sebagai penutup (*stop bit*) **Kbd Data** dikembalikan ke keadaan normalnya, yaitu '1'.



Gambar 2.9 Sinyal Pengiriman Data *Keyboard*

2.4 Pemutar/Perekam Suara ISD25120

Peralatan penyimpan informasi suara, ISD seri 25120 mempunyai kualitas bagus, dengan durasi penyimpanan 60 sampai 120 detik. Peralatan CMOS yang ada didalamnya adalah *Chip Oscilator, Microphone preamplifier, Automatic gain control, antiliasing filter, smouthing filter, speaker amplifier.* ISD25120 adalah kompatibel dengan mikrokontroler AT89C8252. Alamat dan jalur kendali dapat dihubungkan dengan mikrokontroler, sehingga mengijinkan penyimpanan dan pengalamatan yang komplek. Perekam disimpan dalam suatu chip yang tidak mudah berubah dalam cell memori. Sinyal suara dan audio disimpan secara langsung ke memori pada tempat naturalnya dengan kualitas suara yang bagus. Adapun susunan kaki-kaki ISD25120 ditunjukkan dalam Gambar 2.3. berikut:



Gambar 2.10 Susunan kaki ISD25120[18]

Penjelasan dari fungsi spesifik masing-masing kaki dari ISD25120 adalah sebagai berikut:

- *Voltage Input* (Vcca, Vccd), untuk mengurangi *noise*, rangkaian analog dan digital dalam IC ISD25120 digunakan power suplay tersendiri dalam pak.
- *Ground Input (*Vssa, Vssd), pin ini harus dihubungkan dalam power suplay ground dengan impedansi rendah.
- *Power Down Input* (PD), ketika sedang tidak digunakan untuk merekam atau menyalakan, pin PD harus diberi logika tinggi. Ketika pulsa *overflow* (OVF) rendah, PD harus dibawa ke logika tinggi untuk me-reset addres pointer kembali ke awal perekam atau playback.
- *Chip Enable Input* (CE), pin CE diberi logika rendah untuk membolehkan operasi perekam atau *playback*. Alamat input dan *playback/record input* (P/R) ditahan dengan adanya transisi turun dari CE.
- *Playback/Record Input* (P/R) input ditahan dengan adanya transisi turun dari pin CE. Logika tinggi akan memilih *playback cycle* dan logika rendah

untuk memilih *record cycle*. Untuk *Record Cycle*, alamat input menetapkan mulainya alamat dan merekam secara kontinyu sampai PD atau CE ke transisi tinggi atau dengan adanya *overflow*. Ketika *Record Cycle* berhenti dengan adanya transisi tinggi dari PD atau CE, Sinyal *End-Of-Message* (EOM) disimpan dialamat terakhir di memori. Untuk *Playback cycle*, alamat input menetapkan mulainya dan akan memutar suara perekam sampai alamat EOM ditemukan.

- *End-Of-Message/Run Output* (EOM), sebuah alamat akan dimasukkan secara otomatis diakhir masing-masing perekam. Pulsa Output EOM akan rendah untuk setiap periode dari T_{EOM} diakhir masing-masing perekam.
- *Microphone Input* (MIC), digunakan untuk mentransfer sinyal suara ke *on-chip preamplifier*. Rangkaian *On-Chip Automatic Gain Control* (AGC) mengontrol penguatan dari penguat mula dari –15 sampai 24 dB.
- *Micriphone Reference Input* (MIC REF), adalah input pembalik ke *microphone preamplifier*. Ini untuk memberikan *noise-canceling* atau *common-mode rejection input* ke IC ini ketika dihubungkan ke sebuah mikrofon diferensial.
- *Overflow Output* (OVF), sinyal pulsa rendah pada akhir tempat memori, mengindiaksikan bahwa IC ini telah terpenuhi dan pesan telah melebihi kapasitas. Keluaran OVF kemudian diikuti masukan CE sampai pulsa PD telah mereset. Pin ini juga dapat untuk menambah beberapa IC ISD25120 untuk menambah durasi *record/playback*.

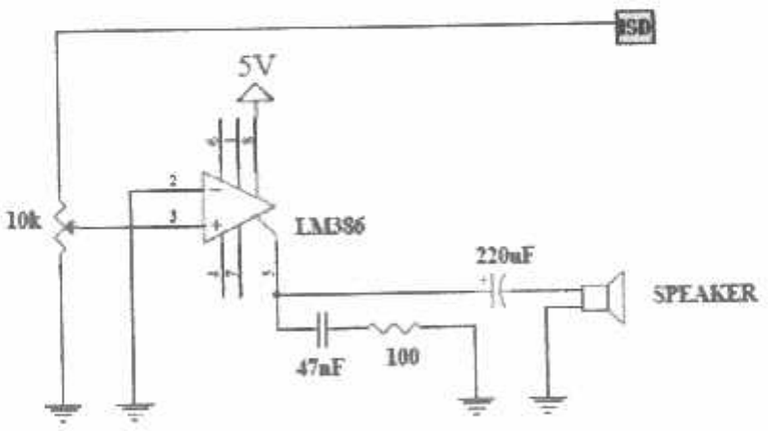
- *Automatic Gain Control* (AGC), agc secara dinamik mengubah penguatan dari penguat mula untuk mengimbangi dari lebar jarak dari level mikrofon input. AGC memberikan jarak secara penuh dari suara rendah ke tinggi untuk direkam dengan *distorsi* minimal.
- *Analog Output* (ANA OUT), pin ini memberikan keluaran penguat mula ke pengguna. Teganagan penguatan dari penguat mula ditentukan oleh level tegangan dari pin AGC.
- *Analog Input* (ANA IN), pin masukan analog akan mentransfer sinyal kedalam chip untuk perekaman. Untuk masukan mikrofon, pin ANA OUT harus dihubungkan ke kapasitor eksternal ke pin ANA IN. Jika permintaan input diperoleh dari sumber lain dari mikrofon, sinyal ini dapat langsung dikopel oleh kapasitor ke pin ANA IN.
- *Eksternal Clock Input* (XCLK), untuk ISD25120 mempunyai sebuah *pull-down* internal. Frekuensi *clock sampling* internal kurang lebih 1% dari spesifikasi. Frekuensi ini berfariasi dari $\pm$ 2,25% berada pada suhu kamar dan dalam range tegangan operasi. Internal clock mempunyai toleransi $\pm$ 5% pada temperatur dan tegangan kerja. Jika power suplay yang digunakan mempunyai presisi tinggi,maka alat ini dapat diclock langsung pada pin XCLK. Duty Cycle pada masukan *clock* tidak perlu dipermasalahkan karena *clock* dengan segera dibagi dua. Jika pin XCLK tidak digunakan, harus dihubungkan ke ground.
- *Speaker Output* (SP+/SP-), ISD25120 telah mempunyai *driver on-chip diferensial speaker*, sanggup memikul beban 50 mW dalam 16 Ω dari

AUX IN. Speaker output berada pada level Vssa selama proses *record* dan *power down.*

- *Auxiliry Input* (AUX IN), dihubungkan langsung ke kaki keluaran amplifier dan keluaran speaker ketika CE, PR berada pada logika tinggi dan playback tidak aktif.
- *Addres/Mode Input (AX/MX),* mempunyai dua fungsi tergantung pada level dari dua *most significant Bit (MSB)* dari alamat tersebut (A8 dan A9). Jika salah satu atau keduanya dari dua MSB berlogika rendah, semua input dianggap sebagai bit alamat dan digunakan sebagai awal alamat untuk proses *record* atau *playback cycle* terbaru. Alamat input ditahan oleh transisi turun dari CE. Jika kedua MSB, berlogika tinggi, masukan *adress/mode* dianggap sebagai bit mode ISD25120

2.5 Penguat Sinyal AC

Di dalam beberapa aplikasi dibutuhkan untuk memberikan penguatan kepada sebuah sinyal ac sementara sinyal DC lainnya yang muncul harus diblok.



Gambar 2.11 Gambar penguat sinyal AC[9]

Tegangan keluaran Vo sebagai sebuah fungsi variabel komplek s didapat dari persamaan rangkaian pada gambar 2.11 menjadi:

$$Vo = EdxAOL$$

Vo = tegangan keluaran

Ed = tegangan pada masukan (+) – tegangan pada masukan (-)

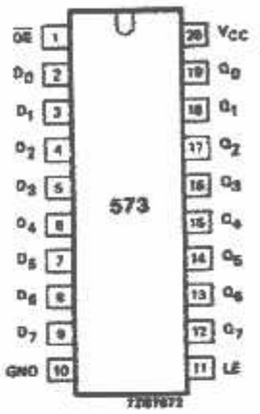
AoL = gain tegangan untaian terbuka (200.000)

2.6 Latch

Latch digunakan untuk memisahkan antara bit – bit data dengan bit – bit alamat yang dikeluarkan pada port 0 oleh mikrokontroller AT 89S8252. Tipe yang digunakan adalah 74HCT573.Latch ini merupakan latch berdelapan yang didalamnya terdapat rangkaian logika Flip-Flop D yang berarti bahwa saat enable dan LE diberikan logika high '1' maka keluaran Q akan memiliki nilai yang sama dengan masukan D.Sedangkan bila LE diberikan logika low '0' maka keluaran akan di kunci pada level data yang telah diset sebelumnya (Q0).seperti pada tabel berikut :

Tabel 2.10 Tabel fungsi Latch 74HCT573

INPUTS			INTERNAL LATCHES	OUTPUTS
$\overline{OE}$	LE	D_N		Q_0 to Q_7
L	H	L	L	L
L	H	H	H	H
L	L	l	L	L
L	L	h	H	H
H	L	l	L	Z
H	L	h	H	Z



Gambar 2.12 Konfigurasi Pin- Pin 74HCT573

Fungsi dari tiap – tiap pin adalah sebagai berikut :

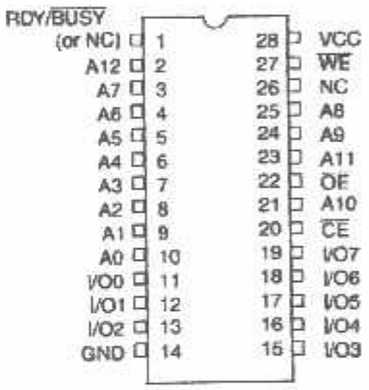
- OE adalah output enable berrfungsi untuk mengaktifkan output latch dan harus diberikan logika rendah '0'.
- LE adalah Latch Enable berfungsi untuk membedakan input apakah sebagai data atau sebagai address yang dikontrol oleh mikrokontroller.
- D0 – D7 adalah input 8 bit.
- Q0 – Q7 adalah output 8 bit.
- VCC adalah sumber tegangan +5volt.
- GND adalah Ground 0 volt

2.7 EEPROM AT28C64

AT28C64 merupakan 8K-byte non-volatile Electrically Erassable and Programmable Read Only Memory dengan feture penggunaan yang mudah. IC memory ini dapat diakses seperti mengakses static RAM bias pada saat siklus baca dan tulis tanpa memerlukan komponen tambahan lain. Selama proses

penulisan byte, alamat dan data di latch secara internal sehingga alamat dan data mikrokontroller dapat bebas untuk melakukan operasi lain.

Sedangkan selama siklus tulis, IC ini akan berada pada state busy dan secara otomatis menghapus dan menulis data yang di latch dengan menggunakan internal control timer. Penditeksian akhir dari siklus tulis dapat dilakukan dengan melakukan proses polling data pada I/O_7. Pada saat akhir dari siklus tulis dideteksi, maka akses untuk melakukan siklus tulis dideteksi, maka akses untuk melakukan siklus baca dan tulis yang baru dapat dilakukan.



Gambar 2.13 Konfigurasi Pin- Pin AT28C64

A0-A12 :Alamat Input untuk akses lokasi data pada memori sesuai dengan yang dibutuhkan.

$\overline{CE}$:Chip Enable untuk mengaktifkan memori agar dapat dilakukan proses baca dan tulis, dan memiliki logika aktif low.

$\overline{OE}$:Output Enable

$\overline{WE}$:Write Enable

I/O0-I/O7 :Data Input dan Output.

VCC :+5 volt tegangan sumber.

VSS :Ground.

Mode Operasi AT28C64 :

- READ MODE

  AT28C64 memiliki dua fungsi control, kedua fungsi tersebutharus memiliki logika yang sesuai agar didapat output yang diinginkan.Saat pin CE dan OE low dan WE high, data yang tersimpan pada memori dengan alamat yang ditunjuk oleh pin address akan dikeluarkan.

- STANDBY MODE

  AT28C64 berada pada kondisi standbay dengan memberikan logika high pada masukan CE. Pada saat standbay mode output berada pada kondisi impedansi tinggi dan tidak dipengaruhi oleh masukan OE.

- DATA PROTECTION

  Untuk memastikan integritas data, terutama pada saat tegangan kritis naik-turun maka data dapat dilindungi sebagai berikut : Pertama, pada saat VCC mendeteksi tegangan 3,3 volt maka data program tidak akan terhapus,dan pada saat VCC dibawah 3,3V sirkuit masih bisa berjalan. Kedua, pada saat WE menyaring dan mencegah WE berada pada pulsa dibawah 10ns untuk waktu siklus tulis. Ketiga, menahan WE atau CE pada kondisi high dan OE low, untuk melindungi siklus tulis selama power on dan power off(Vcc)

- WRITE MODE

  Penulisan data AT28C64 juga sama seperti penulisan pada RAM.Pulsa rendah dari WEatau CE input dengan OE tinggi dan CE atau WE rendah secara berturut-turut akan memberikan sinyal awal penulisan data. Lokasi alamat ini di latch pada saat transisi rendah dari WE (atau CE), dan data akan di latch pada saat transisi naik. Secara internal device akan self-clear sebelum dilakukan penulisan.

- DATA POLLING

  AT28C64 menyediakan data polling untuk memberikan sinyal bahwa proses penulisan data telah berakhir. Selama terjadi siklus tulis, jika dilakukan proses pembacaan tehadap data yang sedang ditulis maka I/O_7 akan mengeluarkan nilai komplemen dari data tersebut (output yang lain tidak didefinisikan). Jika siklus tulis selesai dilakukan, maka semua data akan muncul di semua output.

- ELECTRONIC SIGNATURE FOR DEVICE IDENTIFICATION

  Suatu baris ekstra 32byte pada memori EEPROM adalah berguna untuk identifikasi alat. Dengan menggunakan alamat penunjuk 7E0 sampai 7FF, Penambahan byte dapat dibaca dan tulis sama seperti prosese memori regular.

- CHIP CLEAR

  Semua data mungkin dapat dihapus dalam waktu 1 detik dengan memberikan tegangan 12 volt pada pin OE dan memberikan tegangan low pada WE dan CE.Prosedur ini akan membersihkan semua data pada memori.

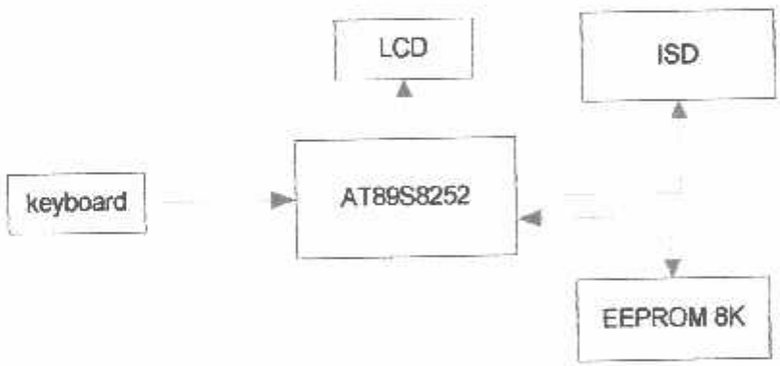
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini akan dibahas mengenai perancangan dan pembuatan alat yang meliputi perangkat keras (hardware) , perangkat lunak (software), dan perancangan database EEPROM. Pembahasan dalam bab ini akan dilakukan perblok seperti perancangan berikut ini :

3.1. Perancangan Perangkat Keras (Hardware)

Perancangan hardware utamanya adalah berdasarkan blok sistem dari seluruh proses seperti yang ditunjukkan oleh gambar 3.1 berikut ini:



Gambar 3.1 Diagram Blok Keseluruhan Sistem

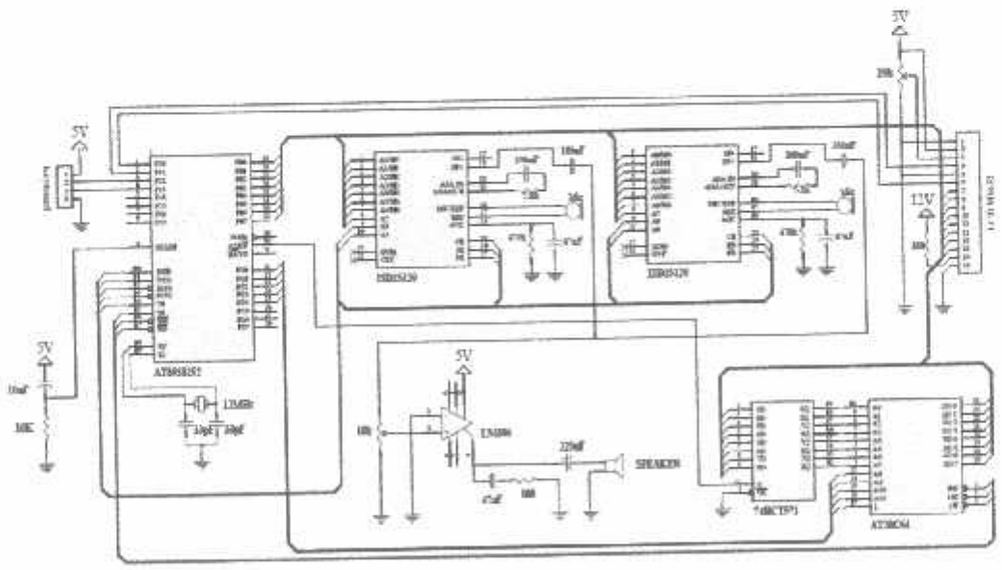
Fungsi dari tiap-tiap blok diagram dijelaskan sebagai berikut :

1. Unit system MC 89S8252 merupakan unit pengolah data.
2. Unit LCD berfungsi untuk menampilkan data/output data.
3. Unit keyboard berfungsi untuk memasukkan data/input data.
4. Unit EEPROM berfungsi untuk melakukan proses penyimpanan data.

3.1.1. Sistem Mikrokontroller AT89S8252

mikrokontroller AT89S8252 berfungsi sebagai pengolah data yang dihasilkan oleh penekanan keypad yang berfungsi sebagai inputan dan menampilkan data tersebut ke dalam LCD yang berfungsi sebagai outputan. Mikrokontroller juga akan menampilkan data identitas penulis ke dalam LCD pada awal penyalaan sistem.

Pengaturan jalur *input* dan *output* pada rangkaian mikrokontroller untuk sebuah rancangan terprogram, sangat berkaitan erat dengan program yang kita buat. Agar tidak terjadi kesalahan saat pembacaan data, mikrokontroller menyediakan jalur-jalur 32 *input-output* yang dapat digunakan secara berkelompok atau bersamaan untuk tiap kelompok terisi 8 bit. Untuk lebih jelasnya, hubungan mikrokontroller dengan perangkat pendukungnya dapat dilihat dalam gambar 3.2 di bawah ini :

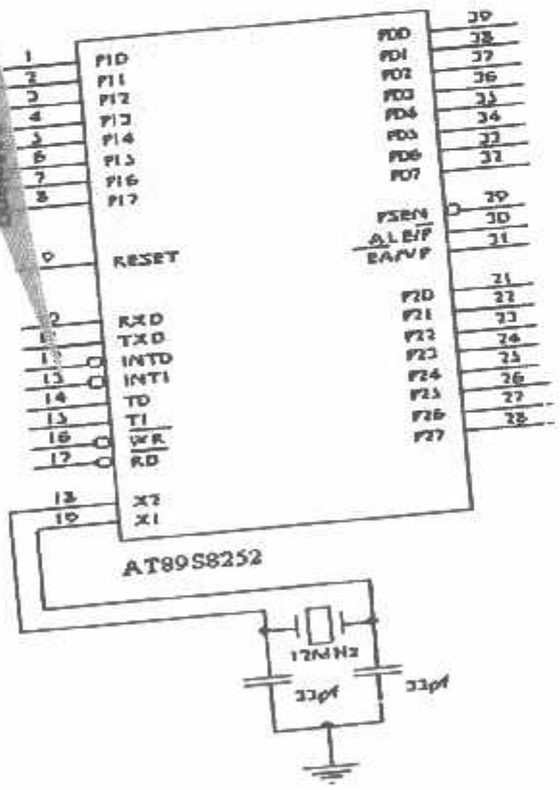


Gambar 3.2 Perancangan Sistem Mikrokontroller AT89S8252

- **lock**

cepatan proses yang dilakukan oleh mikrokontroller ditentukan oleh sumber *k* (pewaktuan) yang mengendalikan mikrokontroller tersebut. Sistem yang akan ancang ini akan menggunakan osilator internal yang sudah tersedia dalam chip rokontroller AT89S8252. Untuk menentukan frekuensi osilatornya cukup denga ara menghubungkan kristal pada pin 19 (XTAL 1) dan pin 18 (XTAL 2) ser a buah kapasitor ke pentanahan (ground).

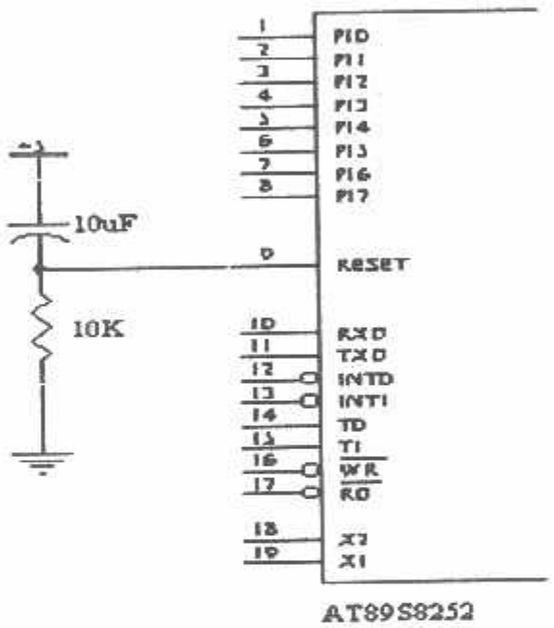
Besar sitansi, disesuaikan dengan spesifikasi pada lembar data AT89S8252 yait 3 pF. Kristal yang digunakan adalah 11,059 MHz. Gambar 3.3 memperlihatkan ra gkaian *clock* yang digunakan.



Gambar 3.3 Perancangan Rangkaian Clock

- **Reset**

Untuk m*ereset* mikrokontroller AT89S8252, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam gambar 3.4 sebagai berikut :



Gambar 3.4 Perancangan Rangkaian Reset

Karena kristal yang digunakan mempunyai frekuensi sebesar 12 MHz, maka satu periode membutuhkan waktu sebesar :

$$\tau = \frac{1}{f_{XTAL}} = \frac{1}{12\,MHz}\,S = 8{,}333\times10^{-8}\,s$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk *mereset* mikrokontroller adalah :

Reset (min) = τ × periode yang dibutuhkan

$= 8{,}333 \times 10^{-8} \times 24 = 1{,}999\ \mu s$

Jadi mikrokontroller membutuhkan waktu minimal 1,999 µs untuk *mereset*. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dari persamaan konstanta waktu $\tau = R \times C$ (William H Hyat, 1998, h132 [1]) dan jika nilai R ditentukan sebesar 10 kΩ, maka nilai C adalah :

$$C = \frac{\tau}{R}$$

$$= \frac{1{,}999 \times 10^{-6}}{10 \times 10^{3}}$$

$$= 199{,}9 \times 10^{-12}\ F$$

Kapasitor minimal yang dibutuhkan adalah 199,9 pF. Dengan menggunakan kapasitor sebesar 10 μF, maka akan menjamin waktu reset di atas nilai minimal waktu yang dibutuhkan untuk mereset mikrokontroller.

3.1.2. LCD M1632

LCD Display Module M1632 buatan Seiko Instrument Inc. adalah komponen *display* yang paling umum digunakan saat ini. LCD M1632 merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing-masing baris bisa menampung 16 huruf/angka.

Untuk berhubungan dengan mikrokontroller, pemakai LCD M1632 dilengkapi dengan 8 jalur data (**DB0..DB7**) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W*** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor.

RS, singkatan dari Register Select, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

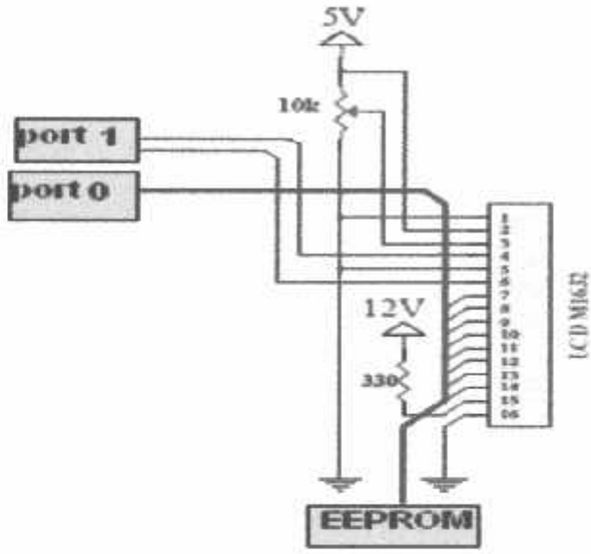
Proses mengirim/mengambil data ke/dari M1632 bisa dijabarkan sebagai berikut :

- **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
- **R/W*** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB0..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB0 .. DB7** diterima oleh M1632.
- Untuk mengambil data dari M1632 sinyal **R/W*** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saat **E** menjadi 1, M1632 akan meletakkan

datanya di **DB0 .. DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali cursor pada baris huruf pertama baris pertama, menghidup/matikan tampilan dan lain sebagainya, semua itu dibahas secara terperinci dalam Lembar Data M1632.

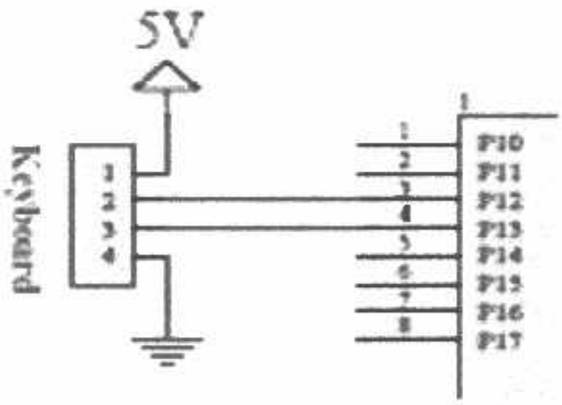
Untuk tampilan dipergunakan LCD Dot Matrik 2 x 16 karakter. Sinyal-sinyal yang diperlukan oleh LCD adalah RS dan Enable, sinyal RS dan Enable dipergunakan sebagai input yang outputnya dipakai untuk mengaktifkan LCD. LCD akan aktif apabila mikrokontroller memberikan instruksi tulis pada LCD. Saat kondisi RS don't care dan Enable 0 maka LCD tetap pada kondisi semula, pengiriman data ke LCD dilakukan saat RS berlogika 1 dan enable berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 0 dan Enable 1. Pin LCD ini untuk data terkoneksi pada *Port 0* mikrokontoller. Kemudian untuk RS dihubungkan pada *Port 2.7*, tulis/baca *(Read/Write)* diberikan logika *low* karena disini LCD bersifat menulis data, dan yang terakhir *Enable* (E) dikendalikan dengan *Port 2.6*. Gambar rangkaian LCD ditunjukkan pada gambar 3.5 sebagai berikut :



Gambar 3.5 rangkaian LCD

3.1.3. Keyboard

Setiap kali salah satu tombol *keyboard* ditekan atau dilepas, *Keyboard* akan mengirim kode ke *host* (*host* adalah komputer jika *keyboard* dihubungkan ke PC, atau berupa mikrokontroller jika *keyboard* dihubungkan ke peralatan berbasis mikrokontroller).



Gambar 3.6 Rangkaian Keyboard

Komunikasi antar *keyboard* dan host adalah komunikasi dua arah, *keyboard* mengirim scan code ke *host*, *host* bisa mengirim perintah untuk mengatur kerja dari *keyboard*. Kode perintah untuk *keyboard* tidak sebanyak *scan code*, berikut ini daftar kode perintah untuk *keyboard* (dalam heksadesimal) adalah:

- **ED,** perintah untuk menyalakan/memadamkan lampu indikator di *keyboard*, setelah menerima perintah **ED** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – *acknowledge)* dan menunggu 1 *byte* perintah lagi dari *host* untuk menentukan lampu indikator mana yang perlu di-nyala/padam-kan. 1 *byte* perintah susulan tersebut akan diartikan sebagai berikut : bit 0 dipakai untuk mengatur lampu indikator ***scroll lock***, bit 1 untuk ***Num Lock*** dan bit 2 untuk ***Capc Lock***, bit-bit lainya diabaikan.
- **EE,** dipakai *host* untuk memeriksa apakah *keyboard* masih aktif. Setelah menerima perintah **EE** dari *host*, *keyboard* akan menjawab dengan **EE** pula, menandakan dirinya masih aktif.
- **FO,** ada keyboard yang dilengkapi 3 set *scan code*, perintah ini dipakai untuk memilih *scan code* yang dipakai. Setelah menerima perintah **FO** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – *acknowledge*) dan *host* menjawab 1 *byte* lagi (nilainya 1, 2 atau 3)untuk memilih set *scan code*. Jika *byte* yang dikirimkan nilainya 0, *keyboard* akan menjawab dengan nomor set *scan code* yang dipakai.

- **F3,** dipakai untuk mengatur kecepatan tanggapan *keyboard* (*Typematic Repeat Rate*), setelah menerima perintah **F3** dari *host*, *keyboard* akan menjawab dengan **FA** sebagai tanda perintah itu telah dikenali (**ACK** – *acknowledge*) dan *host* menjawab 1 *byte* nilai kecepatan tanggapan *keyboard* yang dikehendaki.
- **F4,** dipakai untuk meng-aktif-kan kembali *keyboard*, setelah menerima perintah ini *keyboard* akan menjawab dengan **FA** (**ACK** – *acknowledge*).
- **F5,** dipakai untuk me-nonaktif-kan *keyboard*, setelah menerima perintah ini *keyboard* akan menjawab dengan **FA** (**ACK** – *acknowledge*).
- **FE,** dipakai meminta *keyboard* mengirim ulang *scan code* terakhir yang dikirim.
- **FF,** perintah untuk me-reset *keyboard*

Selain perintah dari *host*, *keyboard* juga mempunyai kode-kode lain selain *scan code* yang dikirimkan ke *host*, sebagai berikut :

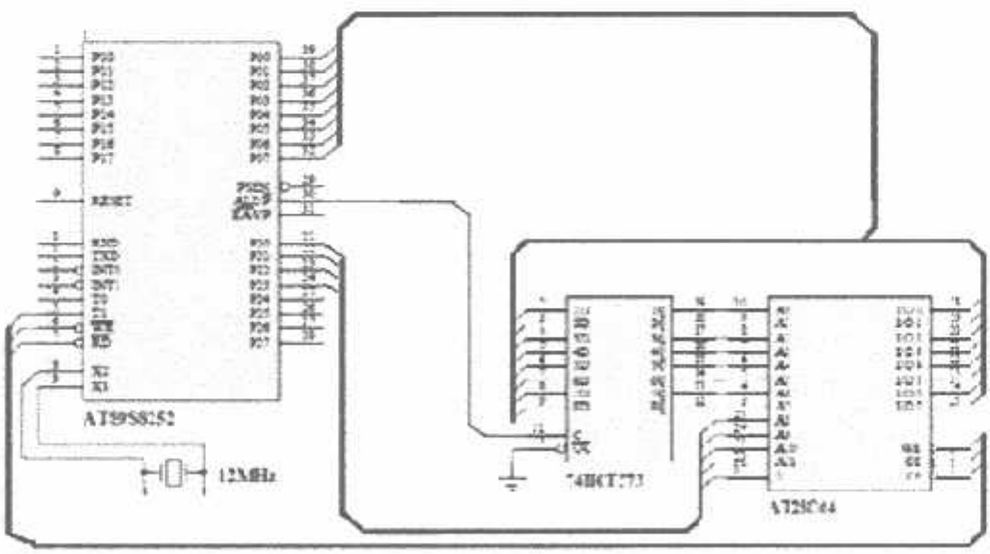
- **FA,** berarti **ACK** (*acknowledge*), yaitu jawaban dari *keyboard* bahwa perintah dari *host* sudah dikenali dengan baik.
- **AA,** berarti *keyboard* selesai memeriksa diri dan siap bekerja setelah diberi catu daya.
- **EE,** identik dengan perintah **EE** diatas.
- **FE,** artinya minta *host* mengulang perintah terakhir yang dikirim.
- **FF / 00,** berarti terjadi kesalahan di *keyboard*.

3.1.4. EEPROM

Memori data eksternal merupakan memori baca/tulis dan untuk proses pembacaan dan penulisan melibatkan sinyal *RD* dan *RW*. Pengaksesan memori eksternal dengan menggunakan intruksi MOVX, dan menggunakan data pointer 16 bit (DPTR), R0 atau R1 sebagai register alamat. Interface mikrokontroller menggunakan jalur *RD(pin 17)* yang dihubungkan dengan pin *output enable(OE)* EEPROM dan *WR(pin 16)* dihubungkan dengan *write enable(WE)* EEPROM.

Sedangkan Port 0 dan Port 2 pada mikrokontroller digunakan sebagai jalur penghubung EEPROM untuk bus alamat dan bus data. Port 0 dapat digunakan sebagai bus data dan alamat yang dapat diatur oleh *IC* Latch(74HCT573). Sedangkan pada Port 2 hanya mengirim sinyal untuk bus alamat saja.

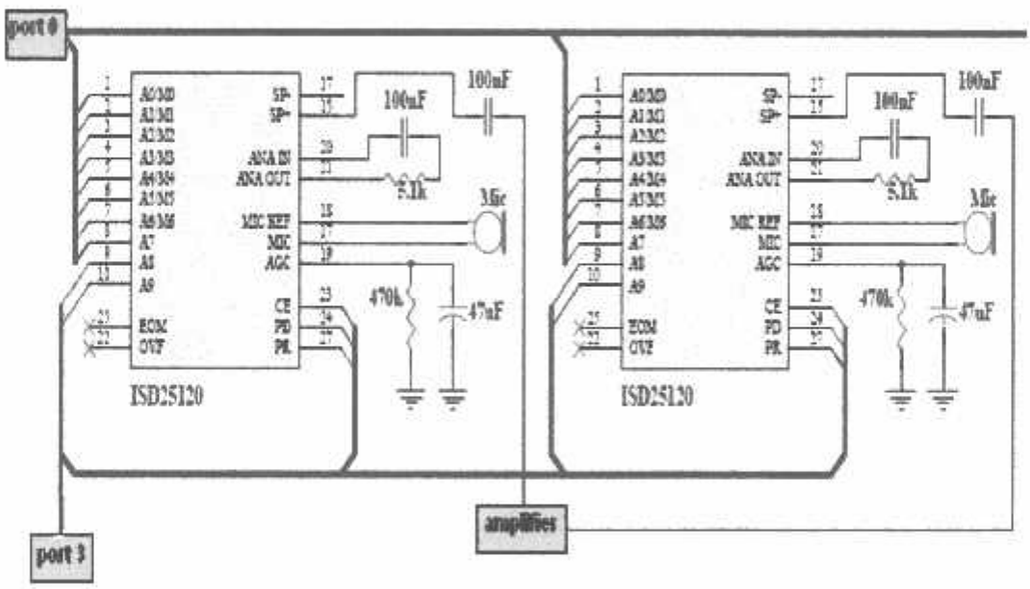
Gambar 3.7 merupakan rangkaian mikrokontroller dengan 2 buah EEPROM.



Gambar 3.7 Rangkaian Mikrokontroller dengan EEPROM

3.1.5. ISD25120

Pada alat pengubah teks menjadi suara ini pemutar/perekam suara ISD25120 digunakan untuk menyimpan rekaman suara yang berupa gabungan dua buah fonem. Pemutar/perekam suara ISD25120 ini kompatibel dengan keluarga MCS-51, schingga mudah untuk dihubungkan dengan mikrokontroler yang digunakan yaitu AT89S8252. Alasan penggunaan pemutar/perekam suara ISD25120 adalah karena pemutar/perekam suara ini memiliki kapasitas yang cukup besar yaitu 120 detik yang terbagi menjadi 600 ruang alamat. Sebagai penyimpan rekaman suara, pin-pin ISD25120 dihubungkan pada rangkaian mikrokontroler membentuk suatu sistem minimum seperti dalam Gambar 3.8



Gambar 3.8 Rangkaian ISD25120

Pemutar/perekam suara mempunyai 10 jalur alamat dan 2 buah pin kontrol. Pada perancangan ini pin-pin yang digunakan adalah:

A0-A7 : Alamat ISD yang dihubungkan ke mikrokontroler AT89S8252 port 0.

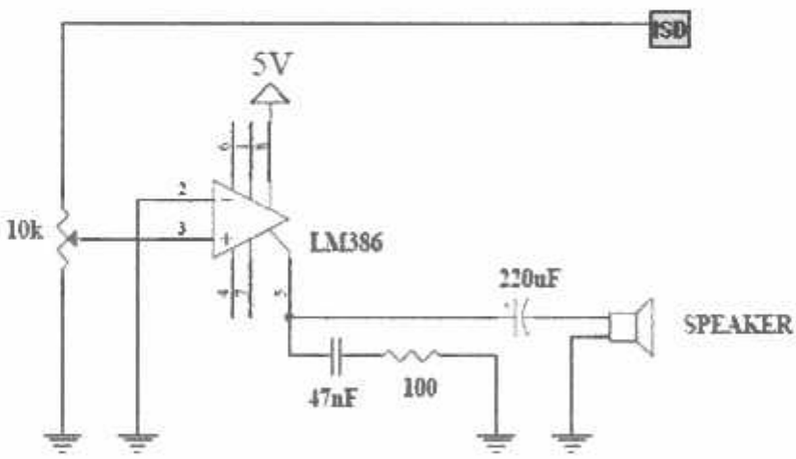
CE : digunakan untuk mengaktifkan ISD25120 dan dihubungkan ke port3.0.

PD : digunakan untuk merekam ke ISD25120 dan dihubungkan ke port3.1.

Pada pembuatan alat pengubah teks menjadi suara, karena pemutar/perekam suara ISD25120 memiliki durasi waktu 120 detik yang terbagi menjadi 600 ruang alamat (alamat 000h sampai dengan alamat 257h) sehingga tiap ruang alamat memiliki durasi 0,5 detik. Karena durasi waktu yang dibutuhkan untuk mengucapkan 1 suku kata adalah 0,5 detik maka dibutuhkan 2 ruang alamat pada ISD25120 untuk menyimpan rekaman satu suku kata. Dengan demikian ISD25120 dapat menampung 300 suku kata.

3.1.6. Penguat Audio

Penguat Audio digunakan untuk menguatkan tegangan keluaran dari ISD25120 sehingga suara yang dihasilkan menjadi lebih keras. Komponen utama dari rangkaian penguat audio ini adalah IC Op-amp LM386. Adapun rangkaian dari penguat audio ini dapat ditunjukkan pada gambar 3.9 berikut:



Gambar 3.9 Rangkaian Penguat Audio

Dari gambar di atas maka dapat diketahui besarnya tegangan pada V_o yaitu sebesar:

$$Vo = EdxAoL$$

Vo = tegangan keluaran

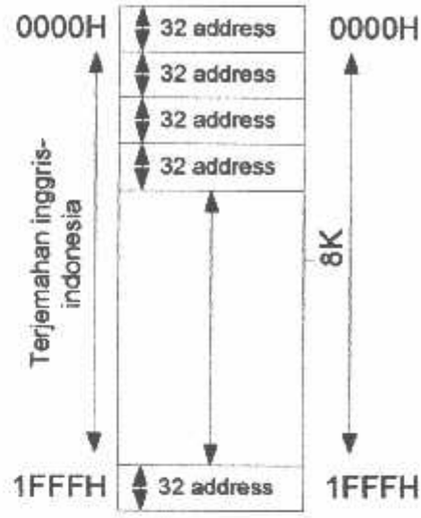
Ed = tegangan pada masukan (+) – tegangan pada masukan (-)

AoL = gain tegangan untaian terbuka (200.000)

3.2. Perancangan Pengalamatan EEPROM

Perancangan database dalam EEPROM dilakukan dengan melakukan pengisian data informasi terjemahan yang berbeda pada EEPROM dengan kapasitas masing-masing 8K. Sehingga dapat diketahui bahwa semua EEPROM menempati alamat mulai dari 0000H-1FFFH dan mempunyai kapasitas total memori sebesar 8K

Pada perancangan alat ini memori yang digunakan sebagai media penyimpanan adalah memori EEPROM, memori EEPROM digunakan untuk menyimpan data terjemahan bahasa inggris dan bahasa indonesia. Pada memori EEPROM akan dibagi menjadi beberapa blok yang disediakan untuk proses pengisian dan kemudian ditampilkan pada LCD, tiap blok tersebut memiliki lebar alamat 32 address yang digunakan menyimpan data terjemahan yang kita cari. Dari penjelasan diatas dapat digambarkan mulai dari memori pertama hingga yang ketiga seperti gambar berikut :



Gambar 3.10 Pembagian Alamat memori untuk terjemahan inggris-indonesia

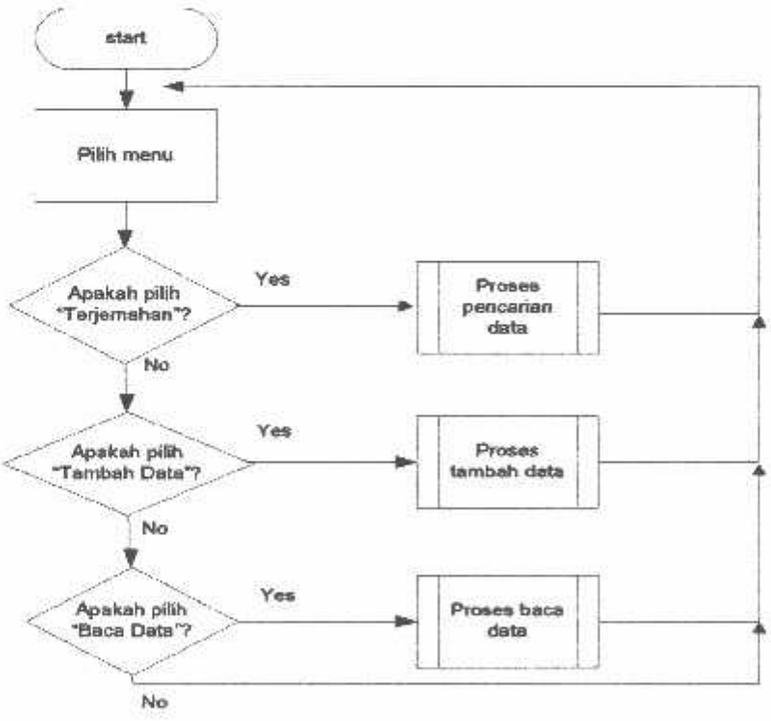
3.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak sangat diperlukan oleh *programmer* dalam mempermudah menentukan langkah-langkah atau alur dari program. Selain mempermudah langkah-langkah pemrograman, diagram alir juga difungsikan supaya program sesuai dan sinkron dengan kerja perangkat keras (*hardware*), sehingga sesuai dengan apa yang direncanakan. Mikrokontroller AT89S8252 menggunakan bahasa *assembly* MCS-51 karena mikrokontroller ini adalah masih keluarga MCS-51. Untuk *compiler* menggunakan SDCC *(Small Device C Compiler)* dan untuk *software downloading programmer*-nya menggunakan *Pony-Prog 2000 - Serial Device Programmer,* kedua software ini dapat *didownload* di internet secara *freeware*.

3.3.1. Diagram Alir (*Flowchart*)

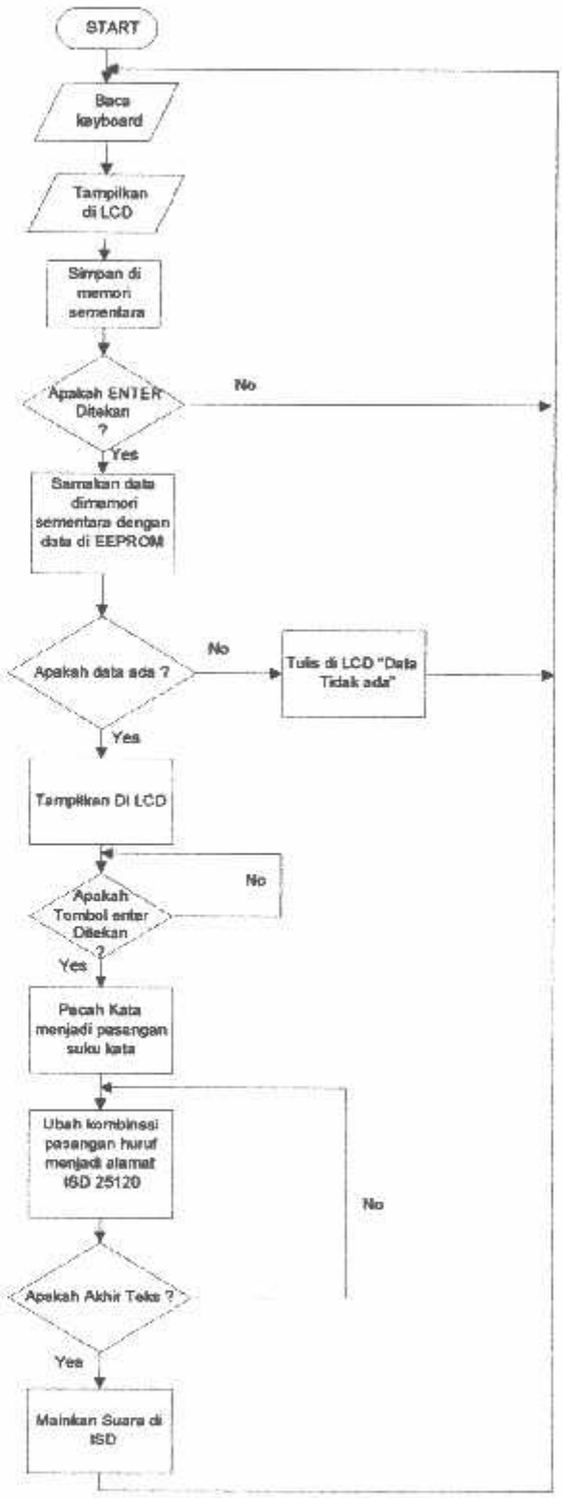
Mendesain *flowchart* sebelum membuat suatu program sangatlah penting, sehingga program yang akan dirancang sedemikian rupa, dapat memaksimalkan bahasa *assembler* yang disediakan oleh mikrokontroller. Selain itu juga memudahkan dalam penganalisaan bila terjadi kesalahan saat membuat program, dengan demikian akan menghasilkan suatu program yang waktu mengeksekusi bahasa atau menjalankan program sangat cepat. Agar perancangan perangkat lunak ini dalam pemahamannya lebih mudah, maka ditunjukkan seperti yang dalam gambar di bawah ini :

3.3.1.1. Diagram Alir (*Flowchart*) Utama



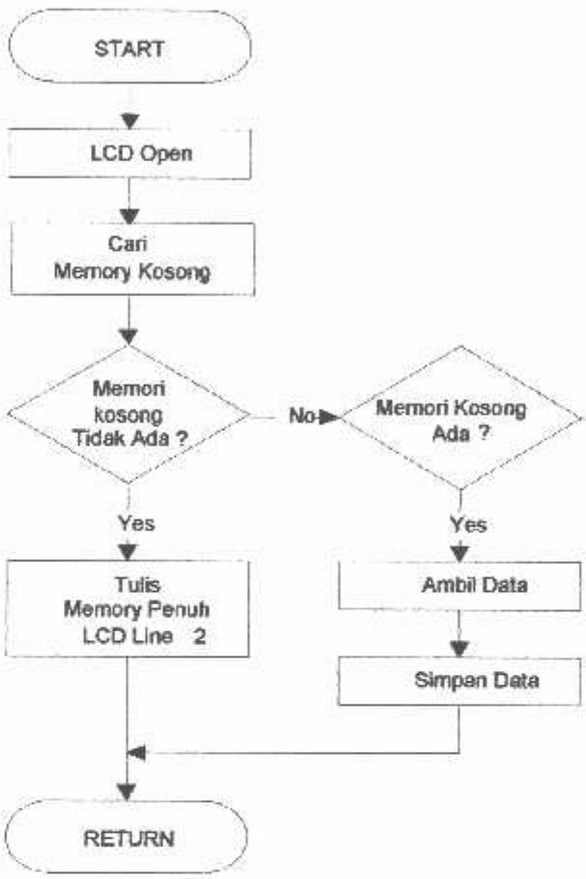
Gambar 3.11 flowchart utama

3.3.1.2. Diagram Alir (*Flowchart*) Untuk *Searching* / Pencarian Data



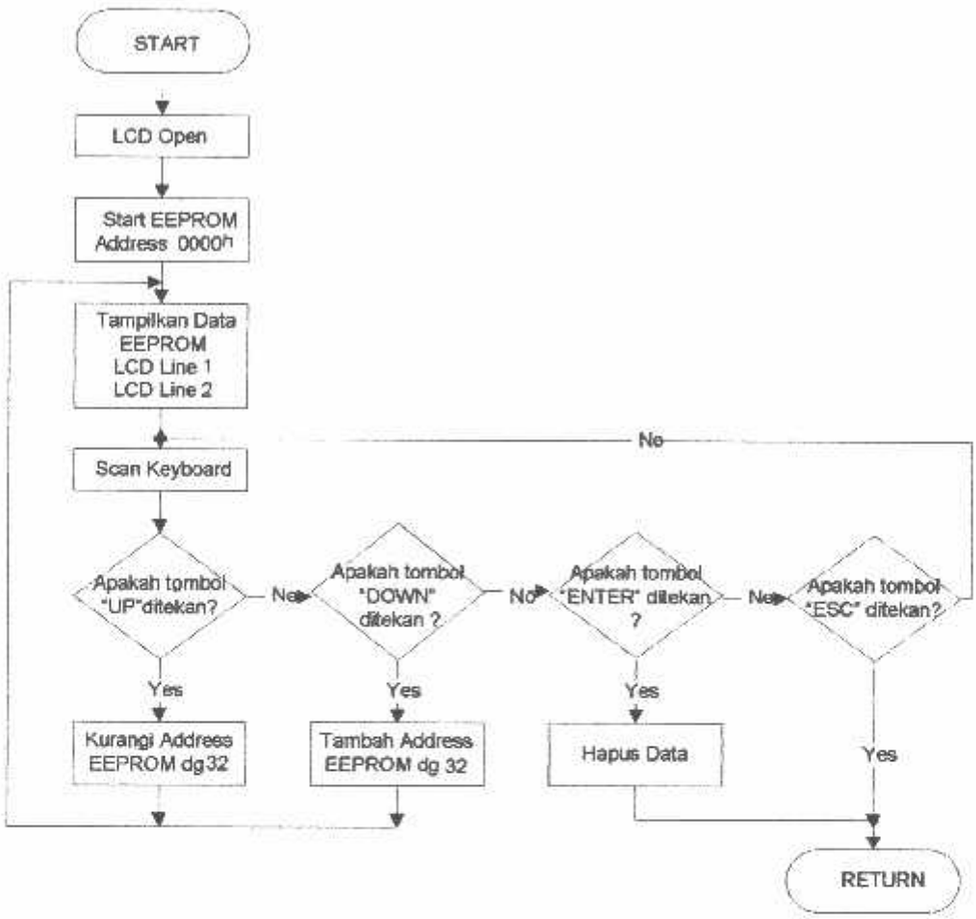
Gambar 3.12 Flowchart Untuk Proses Pencarian Data

3.3.1.3. Diagram Alir (*Flowchart*) Untuk Menambah Data



Gambar 3.13 Flowchart Untuk Proses Menambah Data

3.3.1.4. Diagram Alir (*Flowchart*) Untuk Baca Data



Gambar 3.14 Flowchart Untuk Proses Baca Data

BAB IV

PENGUJIAN DAN ANALISA DATA

Dalam bab ini akan dibahas mengenai pengujian dan analisis alat yang telah dibuat. Secara umum, pengujian ini bertujuan untuk mengetahui piranti yang direlisasikan dapat bekerja sesuai dengan spesifikasi perancangan yang telah ditetapkan. Pengujian ini dilakukan pada sistem secara keseluruhan dengan mengintegrasikan perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk mengetahui unjuk kerja sistem.

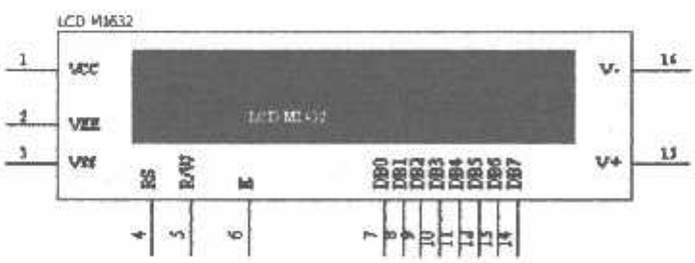
4.1. Pengujian LCD M1632

4.1.1. Tujuan

Untuk mengetahui LCD yang digunakan dalam keadaan baik.

4.1.2. Pelaksanaan Pengujian

Berikut adalah gambar dari LCD M1632 yang akan diuji :



Gambar 4.1 LCD M1632

Di bawah ini adalah tabel dari pengujian instruksi dari LCD M1632.

Tabel 4.1
Hasil Pengujian Instruksi

Instruksi	Kode										Fungsi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Display Clear	0	0	0	0	0	0	0	0	0	1	Menghapus semua tampilan

											layar dan mengembalikan *cursor* ke posisi semula
Cursor Home	0	0	0	0	0	0	0	0	1	*	*Cursor* kembali ke posisi semula dan geser layar ke posisi semula
Empty Mode Set	0	0	0	0	0	0	0	1	I/D	S	Mengatur arah gerakan *cursor* dan menggeser layar ketika data sudah ditulis/dibaca
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Menyalakan/memstikan semua layar (D) dan *cursor* (C), dan membuat kolom *cursor* pada posisi awal kosong
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Memindahkan *cursor* dan menggeser layar
Function Set	0	0	0	0	1	DL	1	*	*	*	Mengatur penambahan banyaknya data
CG RAM Address Set	0	0	0	1	A_{CG}						Menulis alamat RAM ke karakter
DD RAM Address Set	0	0	1	A_{DD}							Menulis alamat RAM ke tampilan
BF/address read	0	1	BF	AC							LCD dalam keadaan sibuk atau tidak sibuk
Data Write to CG RAM or DD RAM	1	0	Write data								Menulis data ke DD RAM atau CG RAM
Data Write from CG RAM or DD RAM	1	1	Read data								Membaca data dari DD RAM atau CG RAM

Keterangan :

1. Empty mode Set : layar beraksi sebagai tampilan tulis.

   S : 1/0 = menggeser layar.

   I/D : 1 = kursor bergerak ke kanan dan layar bergerak ke kiri.

   I/D : 0 = kursor bergerak ke kiri dan layar bergerak ke kanan

2. Display On/Off kontrol.

   D : 1 = layar on

D : 0 = layar off

C : 1 = kursor on

C : 0 = kursor off

B : 1 = kursor berkedip-kedip

B : 0 = kursor tidak berkedip – kedip

3. Cursor Display Shift

S/C : 1 = LCD diidentifikasikan sebagai layar

S/C : 0 = LCD diidentifikasikan sebagai kursor

R/L : 1 = menggeser satu spasi ke kanan

R/L : 0 = menggeser satu spasi ke kiri

4. Fuction Set

DL : 1 = panjang data LCD pada 8 bit

DL : 0 = panjang data LCD pada 4 bit

Bit upper ditransfer terlebih dahulu kemudian diikuti dengan 4 bit lower.

5. CG RAM Address Set :

A_{CG} = alamat CG RAM

6. DD RAM address set :

A_{DD} = alamat DD RAM

7. BF/address read :

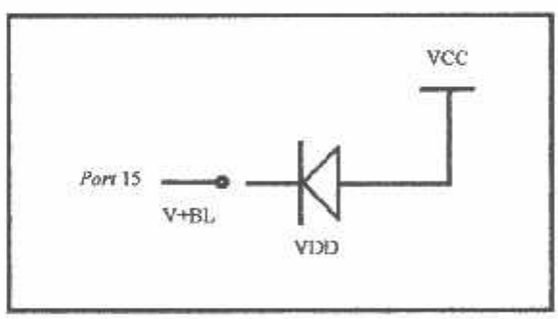
BF : 1 = pengoperasian instruksi sedang dijalankan

BF : 0 = instruksi dapat diterima

Setelah dilakukan pengujian seperti tabel di atas, kemudian dibandingkan dengan *data sheet* LCD M1632 ternyata hasilnya sesuai. Berarti dapat disimpulkan bahwa LCD dalam keadaan baik.

4.1.2.1 Pengujian Tegangan LCD

Selain hal di atas, pengujian LCD juga dapat dilakukan dengan mengukur besar tegangan yang masuk melalui *port* 15 pada LCD. Pengujian dapat dilakukan dengan rangkaian seperti gambar 4.2 di bawah ini :



Gambar 4.2 Rangkaian Pengujian Tegangan LCD

4.1.2.1.1 Langkah-langkah Pengujian

1. Dalam keadaan menyala, ukur power supply sebagai VCC
2. Dalam keadaan menyala, ukur dioda sebagai VDD
3. Kurangi nilai VCC dengan nilai VDD, hasilnya sebagai V+BL
4. Nilai V+BL inilah yang masuk ke *port* 15 pada LCD, yang akan menyalakan lampu pada LCD

4.1.2.1.2 Analisa Hasil Pengujian

perhitungan	pengukuran	error
5 V	4,9 V	2,04 %

5 V	4,89 V	2,24 %
5 V	4,91 V	1,83 %
5 V	4,87 V	2,66 %
5 V	4,9 V	2,04 %
	Error Rata-Rata	2,1%

Tabel 4.2 Pengujian Tegangan VCC Pada LCD

Dari hasil pengujian didapatkan rata-rata nilai VCC adalah 4,89 volt dan nilai VDD adalah 0,7 volt. Dengan mengurangi nilai VCC dengan nilai VDD, maka didapatkan nilai V+BL sebesar 4,2 volt. Maka dengan nilai V+BL sebesar 4,2 volt LCD sudah dapat menyala. (*Module M1632 User Manual,* [5])

4.2 Pengujian ISD 25120

4.2.1. Tujuan

Pengujian terhadap ISD25120 dilakukan untuk memeriksa apakah ISD25120 dapat melakukan perekaman dan memainkan ulang hasil rekaman sesuai dengan perancangan. Pengujian dilakukan dengan cara melakukan perekaman suara dan memainkan ulang hasil rekaman tersebut.

4.2.2 Peralatan yang digunakan

1. Rangkaian unit ISD25120.

4.2.3 Langkah-Langkah Pengujian

1. Mengatur rangkaian seperti terlihat dalam Gambar 4.3
2. Menghidupkan catu daya
3. Memasukkan rekaman suara lewat mikrofon.
4. Mendengarkan hasil rekaman suara lewat speaker.

4.2.4 Analisa

Dari pengujian yang dilakukan, diperoleh hasil pengujian seperti yang terlihat dalam Tabel 4.3

Tabel 4.3
Hasil Pengujian Rangkaian ISD25120

Alamat 000H	Suara yang direkam ab	Keluaran suara ab
005H	so	so
010H	lut	lut
015H	ano	ano
020H	de	de

Dari tabel hasil pengujian dapat dilihat bahwa rangkaian ISD25120 telah dapat bekerja sesuai dengan yang diharapkan pada perancangan. Rangkaian ini dapat melakukan proses perekaman dan pemutaran ulang suara.

4.3. Pengujian Alat Keseluruhan

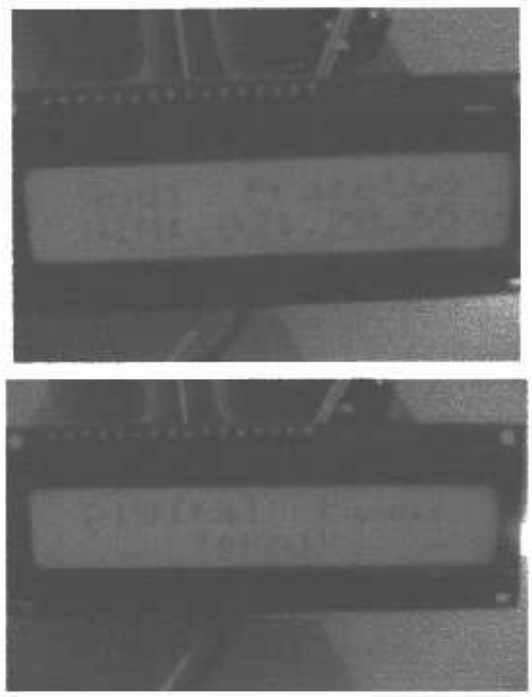
4.3.1. Tujuan

Untuk mengetahui alat yang dibuat sesuai dengan yang telah dirancangkan.

4.3.2 Cara Pengoperasian Alat :

1. Hidupkan saklar ke posisi ON.

2. setelah alat menyala, maka muncul nama pembuat alat dan nama alat bergantian. Seperti gambar dibawah ini:



Gambar 4.3 Tampilan LCD pada saat saklar di ON-kan

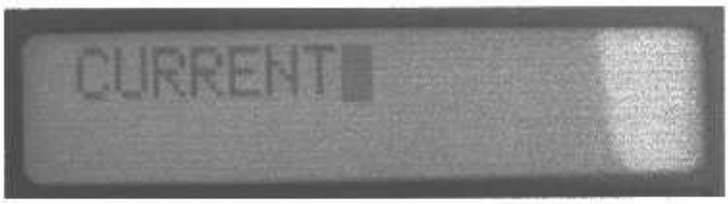
3. Tekan "F1 – F3" untuk memunculkan tampilan menu pada LCD.
4. Tekan "ENTER" untuk menentukan menu yang dipilih.

4.3.2.1 Mencari Data

Gambar 4.4 Tampilan LCD pada saat mencari data

1. Pada menu ini tekan "ENTER".

2. Tuliskan data yang dicari.



Gambar 4.5 Menuliskan kata yang dicari

3. Tekan "ENTER"

Gambar 4.6 Kata yang dicari dan terjemahannya

4. Jika kata yang dicari tidak ada, maka pada LCD akan muncul pesan "*Data not found*".
5. Tekan "ESC" untuk kembali ke menu awal.

4.3.2.2 Tambah Data

Gambar 4.7 Tampilan LCD pada saat tambah data

1. Pada menu ini, tekan "ENTER".
2. Masukkan kata yang akan diterjemahkan, tekan "ENTER".
3. Kemudian masukkan kata terjemahan, tekan "ENTER".
4. Jika memori penuh, maka pada LCD akan muncul pesan "*Memory full*".

4.3.2.3 Baca Data

Gambar 4.8 Tampilan LCD pada saat Baca data

1. Pada menu ini, tekan "ENTER".
2. Tekan "UP" atau "DOWN" untuk melihat data yang disimpan di memori.
3. Apabila ada data yang ingin dihapus, tekan "DEL" untuk menghapus.

4.3.2.4 Cara Pengucapan

1. Setelah data yang dicari ditampilakan di LCD, tekan "ENTER"
2. Suara akan keluar melalui speaker

Tabel 4.4
Hasil Pengujian Pengucapan

Kata dicari	Arti terjemahan	Penggalan suku kata	Hasil pengucapan
CATHODE	KUTUB NEGATIF	/KU//TUB// //NE//GA//TIF/	/KU//TUB// //NE//GA//TIF/
ANTENNA	ANTENA	/AN//TE//NA/	/AN//TE//NA/

ARRESTOR	PENANGKAL PETIR	/PE//NANG//KAL// //PE//TIR/	/PE//NANG//KAL// //PE//TIR/
BUFFER	PENYANGGA	/PE//NYANG//GA/	/PE//NYANG//GA/
INDUCTANCE	INDUKTANSI	/IN//DUK//TAN//SI/	/IN//DUK//TAN//SI/
FILTER	SARINGAN	/SA//RI//NGAN/	/SA//RI//NGAN/
INDUCTOR	INDUKTOR	/IN//DUK//TOR/	/IN//DUK//TOR/
BATTERY	BATERAI	/BA//TE//RAI/	/BA//TE//RAI/
INPUT	MASUKAN	/MA//SU//KAN/	/MA//SU//KAN/
ANODE	KUTUB POSITIF	/KU//TUB// //PO//SI//TIF/	/KU//TUB// //PO//SI//TIF/

4.3.2.5 Pengujian Searching Data

Tabel 4.5

Pengujian Searching Data

No	kata yang dicari	waktu pencarian
1	VOLTAGE	10 ms
2	RESISTOR	21 ms
3	DIODE	33 ms
4	CATHODE	44 ms
5	ANODE	52 ms
6	ELECTRODE	63 ms
7	ELECTRON	72 ms
8	ELECTRO	84 ms
9	TRANSMITTER	94 ms
10	AMPLIFIER	104 ms
11	WIRE	115 ms
12	GROUND	124 ms
13	GAIN	133 ms
14	FUSE	142 ms
15	LAMP	153 ms
16	ANTENNA	164 ms
17	ARRESTOR	174 ms
18	BUFFER	185 ms
19	OSCILLATOR	192 ms
20	FILAMENT	203 ms

no	kata yang dicari	waktu pencarian
21	CIRCUIT	214 ms
22	INDUCTANCE	225 ms
23	RESISTANCE	236 ms
24	AMPEREMETER	243 ms
25	VOLTMETER	255 ms
26	SWITCH	263 ms
27	INPUT	272 ms
28	OUTPUT	283 ms
29	ON	295 ms
30	OFF	303 ms
31	POWER	315 ms
32	FILTER	323 ms
33	CONDUCTOR	335 ms
34	INDUCTOR	345 ms
35	ELECTRIC	354 ms
36	SHORT	365 ms
37	FEEDBACK	372 ms
38	LOOP	383 ms
39	CABLE	393 ms
40	SIGNAL	405 ms

no	kata yang dicari	waktu pencarian
41	NOISE	413 ms
42	FLUX	424 ms
43	SERVER	434 ms
44	PORT	446 ms
45	REGISTER	453 ms
46	HOLE	463 ms
47	FREEZER	475 ms
48	HEATER	483 ms
49	CYCLE	496 ms
50	BUTTON	506 ms
51	BATTERY	514 ms
52	TUBE	526 ms
53	SOLDER	534 ms
54	CONNECTOR	546 ms
55	SOCKET	556 ms
56	COMPONENT	564 ms
57	PLUG	577 ms
58	AMPLITUDE	584 ms
59	FREQUENCY	596 ms
60	CURRENT	607 ms
61	DISTANCE	614 ms
62	DYNAMO	622 ms
63	DETECTOR	635 ms
64	EFFECTIVE	646 ms
65	ELECTROMAGNETIC	654 ms
66	EMIT	666 ms
67	EXPANSI	677 ms
68	CAPACITY	684 ms
69	IMPEDANCE	697 ms
70	INDUCTIVE	706 ms

no	kata yang dicari	waktu pencarian
71	LIQUID	713 ms
72	ERROR	724 ms
73	BIPOLAR	736 ms
74	AMPERE	744 ms
75	ABSOLUTE	754 ms
76	ACTIVATE	767 ms
77	ACCU	774 ms
78	GRAVITY	784 ms
79	MAGNET	794 ms
80	NUCLEAR	804 ms
81	PARALEL	814 ms
82	PARTIAL	823 ms
83	PARITY	834 ms
84	POSITIVE	846 ms
85	PORTAL	854 ms
86	POWER SUPPLY	865 ms
87	PROTON	874 ms
88	NEUTRON	884 ms
89	RADIO	894 ms
90	RESONANCE	903 ms
91	ROTOR	916 ms
92	STATOR	924 ms
93	SPEAKER	935 ms
94	SPACE	945ms
95	TEMPERATURE	954 ms
96	TRANSFER	963 ms
97	VELOCITY	975 ms
98	UTILITY	984 ms
99	VARIABEL	995 ms
100	VERIFY	1010 ms

Rata - rata pencarian data adalah

$$= \frac{jumlahperhitungan}{100}$$

$$= \frac{50248}{100} = 502{,}48 \text{ ms} = 0{,}5 \text{ s}$$

BAB V

PENUTUP

5.1. Kesimpulan

Dari perancangan dan pembuatan kamus teknik digital dengan output suara berbasis AT 89S8252 ini, maka dapat diambil kesimpulan sebagai berikut :

- Dari hasil pengujian didapatkan error rata-rata nilai VCC sebesar 2,1%
- Alamat yang dapat dicari pada ISD hanya sebanyak 70 address saja.
- Lama waktu rata-rata pencarian data pada EEPROM adalah sekitar 0,5 s

5.2. Saran

Dalam perancangan dan pembuatan kamus teknik digital dengan output suara berbasis AT 89S8252 ini, masih didapatkan hal-hal yang bisa menjadikan fungsi alat lebih optimal dengan adanya penambahan atau pengembangan. Hal tersebut antara lain :

1. Untuk dapat menyimpan informasi yang lebih banyak, sebaiknya ditambahkan EEPROM *eksternal* sesuai dengan kebutuhan.
2. Untuk mempermudah pengisian database ke dalam alat ini, hendaknya ditambahkan suatu rangkaian pengisian EEPROM yang dapat diprogram melalui *PC*.

DAFTAR PUSTAKA

[1] ATMEL Data Manual, 2002

[2] Bereksperimen Dengan Mikrokotroller 8031, 1997

[3] Belajar Mikrokontroller ATMEL AT89S8252

[4] Data Sheet ATMEL, Tanpa Tahun

[5] Seiko Instrument Inc, *liquid Crystal Display* Modul M1632 *User Manual*, japan: jan. 1987

[6] Alds. STTS. Edu, Rubrik Analog. STTS Surabaya

[7] Data Sheet ISD 25120, 1998:2

[8] Jacob Millman, 1983 : 542

LAMPIRAN

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

ERSERO) MALANG
NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 April 2007

Nomor : ITN-0258/7/TA /2007
Lampiran :
Perihal : *Bimbingan Skripsi*

Kepada : Yth. Sdr. **I. KOMANG SOMAWIRATA, ST, MT**
Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skipsi untuk mahasiswa:

Nama	: **ANDI PRASETYO WIBOWO**
Nim	: **0217055**
Fakultas	: **Teknologi Industri**
Jurusan	: **Teknik Elektro S-1**
Konsentrasi	: **Teknik Elektronika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal:

17 APRIL 2007 S/D 17 OKTOBER 2007

Sebagai satu syarat untuk menempuh Ujian sarjana.
Demikian atas perhatian serta kerjasama yang baik kami ucapkan terima kasih

Ketua Jurusan
Teknik Elektro S-1

INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI JURUSAN TEKNIK ELEKTRO STRATA-1

Ir. F. Yudi Limpraptono, MT
NIP. Y. 1039500274

Form S-4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PERSERO) MALANG
NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 23 April 2007

Nomor : **ITN-0259/7/TA /2007**
Lampiran :
Perihal : ***Bimbingan Skripsi***

Kepada : Yth. Sdr. **M. ASHAR, ST, MT**
Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skipsi untuk mahasiswa:

Nama	: **ANDI PRASETYO WIBOWO**
Nim	: **0217055**
Fakultas	: **Teknologi Industri**
Jurusan	: **Teknik Elektro S-1**
Konsentrasi	: **Teknik Elektronika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal:

17 APRIL 2007 S/D 17 OKTOBER 2007

Sebagai satu syarat untuk menempuh Ujian sarjana.
Demikian atas perhatian serta kerjasama yang baik kami ucapkan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
NIP. Y. 1039500274

Form S-4a

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-gura No. 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Andi Prasetyo Wibowo
NIM : 02.17.055
Masa Bimbingan : 17 April 2007 s/d 17 Oktober 2007
Judul Skripsi : Perancangan Dan Pembuatan Kamus Teknik Digital Dengan Output Suara Berbasis Mikrokontroler AT89S8252

No.	Tanggal	Uraian	Paraf Pembimbing
1	8/08 07	Revisi Bab I & III.	
2	9/08 07	Acc Bab I & Bab III	
3	3/09 07	Revisi Bab - 5.1 Kesimpulan di ubah sesuai dgn prog.	
4		Acc Bab IV	
5	4/09 07	Acc Mahalah Skripsi	
6			
7			
8			
9			
10			

Malang, 4-09-2007
Dosen Pembimbing I

I Komang Somawirata, ST, MT
NIP. P.1030100361

Form S-4a

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-gura No. 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Andi Prasetyo Wibowo
NIM : 02.17.055
Masa Bimbingan : 17 April 2007 s/d 17 Oktober 2007
Judul Skripsi : Perancangan Dan Pembuatan Kamus Teknik Digital Dengan Output Suara Berbasis Mikrokontroler AT89S8252

No.	Tanggal	Uraian	Paraf Pembimbing
1	2/8'07	Bab I, II, III Revisi	
2	3/8'07	Tinjauan Pustaka	
3	5/8'07	Perancangan system.	
4	15/8'07	Maluda Seminar	
5		Bab IV.	
6		Pengujian System	
7		Pengujian Software	
8		Bab V	
9		Acc Compre.	
10			

Malang,
Dosen Pembimbing II

M. Ashar, ST, MT
1030500408

Form S-4a

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA

LEMBAR PERBAIKAN SKRIPSI

Nama : Andi Prasetyo Wibowo
Nim : 02.17.055
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul : "PERANCANGAN DAN PEMBUATAN KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA BERBASIS MIKROKONTROLLER AT 89S8252"

Hari/Tgl Skripsi : Kamis, 6 September 2007

No.	Materi Perbaikan	Paraf
1.	Flowchart diperbaiki	

Diperiksa / Disetujui

Penguji I

Ir.F.Yudi Limpraptono, MT
NIP.Y. 1039500274

Mengetahui

Dosen Pembimbing I

I Komang Somawirata, ST, MT
NIP. P. 1030100361

Dosen Pembimbing II

M. Ashar, ST, MT
NIP.P. 1030500408

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA

LEMBAR PERBAIKAN SKRIPSI

Nama : Andi Prasetyo Wibowo
Nim : 02.17.055
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul : "PERANCANGAN DAN PEMBUATAN KAMUS TEKNIK DIGITAL DENGAN OUTPUT SUARA BERBASIS MIKROKONTROLLER AT 89S8252"

Hari/Tgl Skripsi : Kamis, 6 September 2007

No.	Materi Perbaikan	Paraf
1.	Ada beberapa gambar rangkaian yang perlu diperbesar karena terlalu kecil	
2.	Kesimpulan harap ditulis dari pengujian	
3.	Gambar rangkaian lengkap	

Diperiksa / Disetujui

Penguji II

Ir. Teguh Herbasuki, MT
NIP. Y. 1038900209

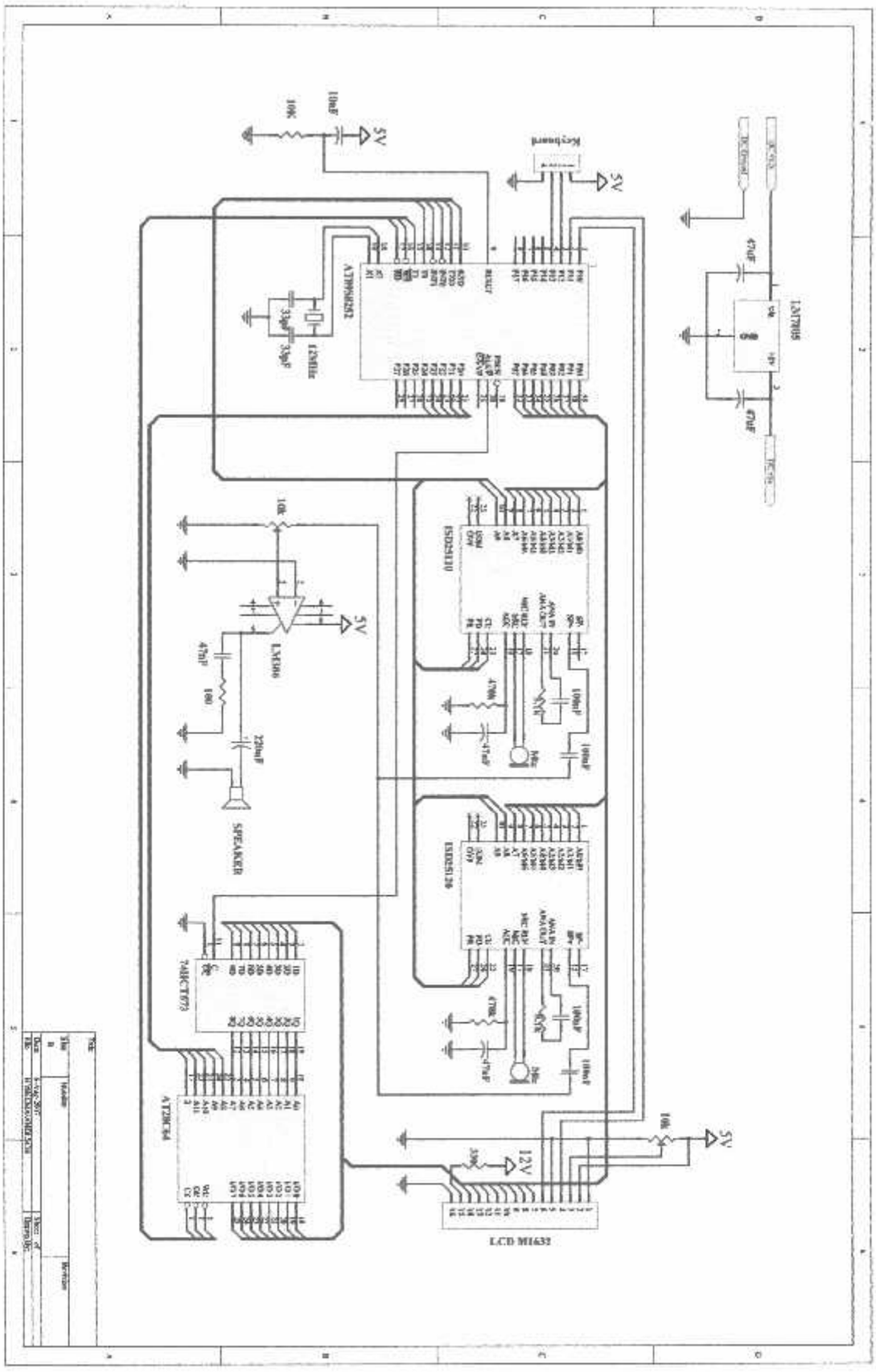
Mengetahui

Dosen Pembimbing I

I Komang Somawirata, ST, MT
NIP. P. 1030100361

Dosen Pembimbing II

M. Ashar, ST, MT
NIP.P. 1030500408


Keyboard
5V
12V
10nF
10K
47uF
LM7805
ISD25110
ISD25120
LM386
SPEAKER
74HCT573
AT28C64
LCD M1632

no	kata yang dicari	Terjemahan
1	absolute	mutlak
3	accu	pembangkit listrik
4	activate	mengaktifkan
5	ampere	satuan arus
6	ampere meter	pengukur arus
7	antenna	antena
8	anode	kutub positif
9	arerestor	penangkal petir
10	battery	baterai
11	bipolar	berkutub ganda
12	buffer	penyangga
13	capacitance	kapasitan
14	capacitor	kapasitor
15	capacity	volume
16	cathode	kutub negatif
17	conductor	konduktor
18	current	arus
19	detector	detektor
20	diode	dioda
21	distance	jarak
22	dynamo	dinamo
23	effective	berguna
24	electromagnetic	kemagnetan
25	emit	menyiarkan
26	error	kesalahan
27	expanse	ekspansi
28	filament	kawat pijar
29	filter	saringan
30	frequency	jumlah getaran
31	gravity	gaya berat
32	impedance	impedansi
33	inductive	induktive
34	input	masukan
35	liquid	zat cair
36	magnet	magnit
37	voltage	tegangan
38	resistor	hambatan

39	electrode	elektroda
40	electron	elektron
41	transmitter	pemancar
42	amplifier	penguat
43	wire	kawat logam
44	ground	pertanahan
45	gain	penguatan
46	fuse	pengaman arus
47	lamp	lampu
48	oscillator	osilator
49	circuit	rangkaian
50	resistance	daya hambat
51	voltmeter	pengukur tegangan
52	switch	saklar
53	output	keluaran
54	on	nyala
55	off	tidak nyala
56	power supply	sumber tegangan
57	inductor	induktor
58	short	hubungan singkat
59	feedback	umpan balik
60	loop	lintas melingkar
61	cable	kabel
62	signal	sinyal
63	noise	ganguan
64	fluk	arus gaya
65	server	pendistribusi
66	portal	gerbang masuk
67	register	kunci
68	hole	lubang
69	freezer	alat pendingin
70	heater	alat pemanas
71	cycle	perioda
72	button	tombol
73	tube	pipa
74	solder	pateri timah
75	connector	penghubung
76	socket	lubang masuk
77	component	komponen

78	plug	kontak tusuk
79	amplitude	lebar langkah
80	port	jalan masuk
81	power	daya
82	nuclear	nuklir
83	paralel	sejajar
84	partial	sebagian
85	parity	sifat sama
86	positve	positif
87	proton	proton
88	newton	satuan tenaga
89	radio	radio
90	resonance	resonansi
91	rotor	pemutar
92	static	stabil
93	speaker	speaker
94	space	ruang
95	temperature	suhu
96	transfer	memindahkan
97	velocity	kecepatan gerak
98	utility	kegunaan
99	variable	tidak tetap
100	verify	diteliti

Lampiran

Tabel Kode Scan Keyboard

KODE SCAN								
TOMBOL	SCAN KODE TEKAN	SCAN KODE LEPAS	TOMBOL	SCAN KODE TEKAN	SCAN KODE LEPAS	TOMBOL	SCAN KODE TEKAN	SCAN KODE LEPAS
Esc	01	81	S	1F	9F	F3	3D	BD
1	02	82	D	20	A0	F4	3E	BE
2	03	83	F	21	A1	F5	3F	BF
3	04	84	G	22	A2	F6	40	C0
4	05	85	H	23	A3	F7	41	C1
5	06	86	J	24	A4	F8	42	C2
6	07	87	K	25	A5	F9	43	C3
7	08	88	L	26	A6	F10	44	C4
8	09	89	;	27	A7	Num Lock	45	C5
9	0A	8A	'	28	A8	Scrol		
0	0B	8B	`	29	A9	Lock	46	C6
-	0C	8C	ShifKiri	2A	AA	Home	47	C7
=	0D	8D	\	2B	AB	_ <8>	48	C8
Back			Z	2C	AC	PgUp	49	C9
Space	0E	8E	X	2D	AD	<->KeyPad	4A	CA
Tab	0F	8F	C	2E	AE	<_>KeyPad	4B	CB
Q	10	90	V	2F	AF	<5>KeyPad	4C	CC
W	11	91	B	30	B0	<_>KeyPad	4D	CD
E	12	92	N	31	B1	<+>KeyPad	4E	CE
R	13	93	M	32	B2	<1>KeyPad	4F	CF
T	14	94	<	33	B3	<_>KeyPad	50	D0
Y	15	95	>	34	B4	PgDn	51	D1
U	16	96	?	35	B5	Ins	52	D2
I	17	97	Shif			Del	53	D3
O	18	98	Kanan	36	B6	F11	57	D7
P	19	99	PrtSc	37	B7	F12	58	D8
{	1A	9A	LftAlt	38	B8	RgtAlt	E038	E0B8
}	1B	9B	Space	39	B9	RgtCtrl	E01B	E09D
Enter	1C	9C	CapsLock	3A	BA	Enter	E01C	E09C
LftCtrL	1D	9D	F1	3B	BB	<KeyPad>		
A	1E	9E	F2	3C	BC			

LISTING PROGRAM

```
        org     00h
;
        Rest    Bit P1.0
        Enbl    Bit P1.1
        Kdta    Bit P1.2                ; keyboard data
        Kclk    Bit P1.3                ; keyboard clock
        ISLO    Bit P3.0                ; isd address low
        ISTR    Bit P3.1                ; isd start
        ISHI    Bit P3.2                ; isd address high
        CIS0    Bit P3.3                ; chip enable ISD0
        CIS1    Bit P3.4                ; chip enable ISD1
        CMEM    Bit P3.5                ; chip enable memory
        Ste0    Bit 20h.0               ; status enter 0
        Ste1    Bit 20h.1               ; status enter 1
        Ste2    Bit 20h.2               ; status enter 2
        Ste3    Bit 20h.3               ; status enter 3
        Sttp    Bit 20h.4               ; status tampil pencarian
        Stch    Bit 20h.5               ; status character
        Ch00    Equ 30h
        Ch01    Equ 31h
        Ch02    Equ 32h
        Ch03    Equ 33h
        Ch04    Equ 34h
        Ch05    Equ 35h
        Ch06    Equ 36h
        Ch07    Equ 37h
        Ch08    Equ 38h
        Ch09    Equ 39h
        Ch0A    Equ 3Ah
        Ch0B    Equ 3Bh
        Ch0C    Equ 3Ch
        Ch0D    Equ 3Dh
        Ch0E    Equ 3Eh
        Ch0F    Equ 3Fh
        Ch10    Equ 40h
        Ch11    Equ 41h
        Ch12    Equ 42h
        Ch13    Equ 43h
        Ch14    Equ 44h
        Ch15    Equ 45h
        Ch16    Equ 46h
        Ch17    Equ 47h
        Ch18    Equ 48h
        Ch19    Equ 49h
        Ch1A    Equ 4Ah
        Ch1B    Equ 4Bh
        Ch1C    Equ 4Ch
        Ch1D    Equ 4Dh
        Ch1E    Equ 4Eh
        Ch1F    Equ 4Fh
        Ch20    Equ 50h
        Ch21    Equ 51h
        Ch22    Equ 52h
        Ch23    Equ 53h
        Ch24    Equ 54h
        Ch25    Equ 55h
        Ch26    Equ 56h
        Ch27    Equ 57h
        Ch28    Equ 58h
        Ch29    Equ 59h
        Ch2A    Equ 5Ah
        Ch2B    Equ 5Bh
        Ch2C    Equ 5Ch
        Ch2D    Equ 5Dh
        Ch2E    Equ 5Eh
        Ch2F    Equ 5Fh
        Hurf    Equ 60h
        Ucap    Equ 61h
        Ckb0    Equ 62h
        Ckb1    Equ 63h
        Cnt0    Equ 64h                 ; counter 0
        Cnt1    Equ 65h                 ; counter 1
        Cnt2    Equ 66h                 ; counter 2
        Cnt3    Equ 67h                 ; counter 3
        Dly0    Equ 68h
        Dly1    Equ 69h
        Dly2    Equ 6Ah
```

```
;
init:   lcall   lcd_in
;
mulai:  mov     DPTR,#nama
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
        lcall   line2
        mov     Hurf,#16
        lcall   tulis
        lcall   delay1
        mov     DPTR,#judul
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
        lcall   line2
        mov     Hurf,#16
        lcall   tulis
        lcall   delay1
        ljmp    mulai
;
menu00: lcall   lcdclr
        mov     DPTR,#elckms
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
menu01: lcall   scnkbd
        cjne    R0,#118,menu02          ; tekan esc
        mov     SP,#07h
        ljmp    mulai
menu02: cjne    R0,#090,menu01          ; tekan enter
        ljmp    kamus
;
menu10: lcall   lcdclr
        mov     DPTR,#mskdta
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
menu11: lcall   scnkbd
        cjne    R0,#118,menu12          ; tekan esc
        mov     SP,#07h
        ljmp    mulai
menu12: cjne    R0,#090,menu11          ; tekan enter
        ljmp    isdkms                  ; isi data kamus
;
menu20: lcall   lcdclr
        mov     DPTR,#bcadta
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
menu21: lcall   scnkbd
        cjne    R0,#118,menu22          ; tekan esc
        mov     SP,#07h
        ljmp    mulai
menu22: cjne    R0,#090,menu21          ; tekan enter
        ljmp    bcdkms
;
kamus:  lcall   lcdclr
        clr     Ste0                    ; reset status enter 0
        clr     Ste1                    ; reset status enter 1
        clr     Ste2                    ; reset status enter 2
        clr     Ste3                    ; reset status enter 3
kamus0: lcall   rstchr
        lcall   tg_tkn
        mov     Ch10,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch11,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch12,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch13,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch14,A
```

```
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch15,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch16,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch17,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch18,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch19,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1A,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1B,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1C,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1D,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1E,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch1F,A
        lcall   w_chr
kamus1: lcall   tlsscn
        lcall   caridt
        mov     SP,#07h
        ljmp    mulai
        ret
;
caridt: clr     CMEM
        clr     Sttp
        mov     Cnt1,#0
        mov     Cnt3,#255
crdta0: mov     DPTR,#0000h
        mov     A,Cnt1
        cjne    A,#0,crdta1
        ljmp    crdta3
crdta1: mov     Cnt2,Cnt1
crdta2: lcall   ctadi6
        djnz    Cnt2,crdta2
crdta3: clr     C
        movx    A,@DPTR
        mov     B,Ch10
        subb    A,B
cek00:  cjne    A,#0,cek01
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch11
        subb    A,B
cek01:  cjne    A,#0,cek02
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch12
        subb    A,B
cek02:  cjne    A,#0,cek03
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch13
        subb    A,B
cek03:  cjne    A,#0,cek04
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch14
        subb    A,B
cek04:  cjne    A,#0,cek05
        inc     DPTR
```

```
        movx    A,@DPTR
        mov     B,Ch15
        subb    A,B
cek05:  cjne    A,#0,cek06
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch16
        subb    A,B
cek06:  cjne    A,#0,cek07
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch17
        subb    A,B
cek07:  cjne    A,#0,cek08
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch18
        subb    A,B
cek08:  cjne    A,#0,cek09
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch19
        subb    A,B
cek09:  cjne    A,#0,cek0A
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1A
        subb    A,B
cek0A:  cjne    A,#0,cek0B
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1B
        subb    A,B
cek0B:  cjne    A,#0,cek0C
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1C
        subb    A,B
cek0C:  cjne    A,#0,cek0D
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1D
        subb    A,B
cek0D:  cjne    A,#0,cek0E
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1E
        subb    A,B
cek0E:  cjne    A,#0,cek0F
        inc     DPTR
        movx    A,@DPTR
        mov     B,Ch1F
        subb    A,B
cek0F:  cjne    A,#0,crdta9
        inc     DPTR
;
crdta4: jb      Sttp,crdta5
        ljmp    crdta8
crdta5: mov     DPTR,#0000h
        dec     Cnt1
        mov     A,Cnt1
        cjne    A,#0,crdta6
        ljmp    crdta8
crdta6: mov     Cnt2,Cnt1
crdta7: lcall   ctad16
        djnz    Cnt2,crdta7
;
crdta8: lcall   line2
        lcall   tlsjwb
        ljmp    crdtaB
;
crdta9: inc     Cnt1
        lcall   rbhsts
        djnz    Cnt3,crdtaA
        lcall   tlsndt
        ljmp    crdtaD
crdtaA: ljmp    crdta0
```

```
crdtaB: lcall   scnkbd
        cjne    R0,#118,crdtaC          ; tekan esc
        ljmp    crdtaD
crdtaC: cjne    R0,#090,crdtaB          ; tekan enter
        lcall   bicara
crdtaD: setb    CMEM
        mov     SP,#07h
        ljmp    mulai
;
tlsjwb: movx    A,@DPTR
        mov     Ch20,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch21,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch22,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch23,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch24,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch25,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch26,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch27,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch28,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch29,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2A,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2B,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2C,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2D,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2E,A
        lcall   cknull
        inc     DPTR
        movx    A,@DPTR
        mov     Ch2F,A
        lcall   cknull
        ret
;
isdkms: lcall   lcdclr
        clr     Ste0                    ; reset status enter 0
        clr     Ste1                    ; reset status enter 1
```

```
        clr     Ste2                    ; reset status enter 2
        setb    Ste3                    ; reset status enter 3
isdkm0: lcall   rstchr
        lcall   tg_tkn
        mov     Ch00,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch01,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch02,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch03,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch04,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch05,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch06,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch07,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch08,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch09,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0A,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0B,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0C,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0D,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0E,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch0F,A
        lcall   w_chr
        setb    Ste2
;
isdkm1: lcall   line2
        lcall   tg_tkn
        mov     Ch10,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch11,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch12,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch13,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch14,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch15,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch16,A
        lcall   w_chr
        lcall   tg_tkn
        mov     Ch17,A
```

```
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch18,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch19,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1A,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1B,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1C,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1D,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1E,A
          lcall   w_chr
          lcall   tg_tkn
          mov     Ch1F,A
          lcall   w_chr
isdkm2:   lcall   tlssvg
          lcall   tlsmem
          mov     SP,#07h
          ljmp    mulai
          ret
;
bcdkms:   mov     Cnt1,#0
bckms0:   mov     DPTR,#0000h
          mov     A,Cnt1
          cjne    A,#0,bckms1
          ljmp    bckms3
bckms1:   mov     Cnt2,Cnt1
bckms2:   lcall   ctad32
          djnz    Cnt2,bckms2
bckms3:   lcall   rd_mem
bckms4:   lcall   scnkbd
          cjne    R0,#117,bckms6          ; tekan up
          dec     Cnt1
          mov     A,Cnt1
          cjne    A,#255,bckms5
          mov     Cnt1,#0
bckms5:   ljmp    bckms0
bckms6:   cjne    R0,#114,bckms8          ; tekan down
          inc     Cnt1
          mov     A,Cnt1
          cjne    A,#100,bckms7
          mov     Cnt1,#99
bckms7:   ljmp    bckms0
bckms8:   cjne    R0,#113,bckms9          ; tekan del
          lcall   delkms
          mov     SP,#07h
          ljmp    mulai
bckms9:   cjne    R0,#118,bckms4          ; tekan esc
          mov     SP,#07h
          ljmp    mulai
;
delkms:   mov     DPTR,#deletg
          lcall   line1
          mov     Hurf,#16
          lcall   tulis
          mov     DPTR,#plwait
          lcall   line2
          mov     Hurf,#16
          lcall   tulis
          mov     DPTR,#0000h
delkm0:   mov     A,Cnt1
          cjne    A,#0,delkm1
          ljmp    delkm3
delkm1:   mov     Cnt2,Cnt1
delkm2:   lcall   ctad32
          djnz    Cnt2,delkm2
delkm3:   mov     Cnt0,#32
```

```
        clr     CMEM
delkm4: mov     A,#0FFh
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        djnz    Cnt0,delkm4
        setb    CMEM
        ret
;
rstchr: mov     Ch00,#0FFh
        mov     Ch01,#0FFh
        mov     Ch02,#0FFh
        mov     Ch03,#0FFh
        mov     Ch04,#0FFh
        mov     Ch05,#0FFh
        mov     Ch06,#0FFh
        mov     Ch07,#0FFh
        mov     Ch08,#0FFh
        mov     Ch09,#0FFh
        mov     Ch0A,#0FFh
        mov     Ch0B,#0FFh
        mov     Ch0C,#0FFh
        mov     Ch0D,#0FFh
        mov     Ch0E,#0FFh
        mov     Ch0F,#0FFh
        mov     Ch10,#0FFh
        mov     Ch11,#0FFh
        mov     Ch12,#0FFh
        mov     Ch13,#0FFh
        mov     Ch14,#0FFh
        mov     Ch15,#0FFh
        mov     Ch16,#0FFh
        mov     Ch17,#0FFh
        mov     Ch18,#0FFh
        mov     Ch19,#0FFh
        mov     Ch1A,#0FFh
        mov     Ch1B,#0FFh
        mov     Ch1C,#0FFh
        mov     Ch1D,#0FFh
        mov     Ch1E,#0FFh
        mov     Ch1F,#0FFh
        ret
;
tg_tkn: jnb     Ste0,tgtkn2
        clr     Ste0
        setb    Ste2
tgtkn0: jb      Ste3,tgtkn1
        ljmp    kamus1
tgtkn1: jnb     Ste3,tgtkn0
        ljmp    isdkm1
;
tgtkn2: jnb     Ste1,tgtkn3
        clr     Ste1
        ljmp    isdkm2
tgtkn3: lcall   scnkbd
        cjne    R0,#255,tgtkn4
        ljmp    tgtkn3
tgtkn4: cjne    R0,#090,tgtkn6
        jb      Ste2,tgtkn5
        setb    Ste0
        ljmp    tgtkn6
tgtkn5: jb      Ste1,tgtkn6
        setb    Ste1
        ljmp    tgtkn5
tgtkn6: cjne    R0,#118,tgtkn7
        mov     SP,#07h
        ljmp    mulai
tgtkn7: ret
;
tlsmem: clr     CMEM
        mov     DPTR,#0000h
tlmem0: movx    A,@DPTR
        cjne    A,#0FFh,tlmem1
        ljmp    tlmem2
tlmem1: lcall   ctad32
        ljmp    tlmem0
tlmem2: lcall   wr_mem
```

```
        setb    CMEM
        ret
;
ctad16: mov     Cnt0,#16
cad16:  inc     DPTR
        djnz    Cnt0,cad16
        ret
;
ctad32: mov     Cnt0,#32
cad32:  inc     DPTR
        djnz    Cnt0,cad32
        ret
;
rbhsts: jb      Sttp,rbhst0
        setb    Sttp
        ljmp    rbhst1
rbhst0: clr     Sttp
rbhst1: ret
;
wr_mem: mov     A,Ch00
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch01
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch02
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch03
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch04
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch05
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch06
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch07
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch08
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch09
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0A
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0B
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0C
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0D
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0E
        movx    @DPTR,A
```

```
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch0F
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch10
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch11
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch12
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch13
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch14
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch15
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch16
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch17
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch18
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch19
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1A
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1B
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1C
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1D
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1E
        movx    @DPTR,A
        lcall   wt_wr
        inc     DPTR
        mov     A,Ch1F
        movx    @DPTR,A
        lcall   wt_wr
        ret
;
rd_mem: clr     CMEM
        mov     Cnt0,#16
        lcall   line1
tsln1:  movx    A,@DPTR
        lcall   cknull
        inc     DPTR
```

```
        djnz    Cnt0,tlsln1
        mov     Cnt0,#16
        lcall   line2
tlsln2: movx    A,@DPTR
        lcall   cknull
        inc     DPTR
        djnz    Cnt0,tlsln2
        setb    CMEM
        ret
;
cknull: cjne    A,#0FFh,cknul
        mov     A,#' '
cknul:  mov     P0,A
        lcall   w_chr
        ret
;
tlsscn: mov     DPTR,#scanig
        lcall   line2
        mov     Hurf,#16
        lcall   tulis
        ret
;
tlsndt: mov     DPTR,#nodata
        lcall   line2
        mov     Hurf,#16
        lcall   tulis
        lcall   delayp
        ret
;
tlssvg: mov     DPTR,#saving
        lcall   line1
        mov     Hurf,#16
        lcall   tulis
        mov     DPTR,#plwait
        lcall   line2
        mov     Hurf,#16
        lcall   tulis
        ret
;
bicara: mov     Ucap,#0
        mov     R1,Ch20
        mov     R2,Ch21
        mov     R3,Ch22
        mov     R4,Ch23
        mov     R5,Ch24
        lcall   scnchr
        jnb     Stch,bcr00
        mov     Ucap,#5
        ljmp    bcr03
bcr00:  mov     R1,Ch20
        mov     R2,Ch21
        mov     R3,Ch22
        mov     R4,Ch23
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr01
        mov     Ucap,#4
        ljmp    bcr03
bcr01:  mov     R1,Ch20
        mov     R2,Ch21
        mov     R3,Ch22
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr02
        mov     Ucap,#3
        ljmp    bcr03
bcr02:  mov     R1,Ch20
        mov     R2,Ch21
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr03
        mov     Ucap,#2
;
bcr03:  mov     A,Ucap
```

```
        cjne    A,#2,bcr07
        mov     R1,Ch22
        mov     R2,Ch23
        mov     R3,Ch24
        mov     R4,Ch25
        mov     R5,Ch26
        lcall   scnchr
        jnb     Stch,bcr04
        mov     Ucap,#7
        ljmp    bcr07
bcr04:  mov     R1,Ch22
        mov     R2,Ch23
        mov     R3,Ch24
        mov     R4,Ch25
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr05
        mov     Ucap,#6
        ljmp    bcr07
bcr05:  mov     R1,Ch22
        mov     R2,Ch23
        mov     R3,Ch24
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr06
        mov     Ucap,#5
        ljmp    bcr07
bcr06:  mov     R1,Ch22
        mov     R2,Ch23
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr07
        mov     Ucap,#4
;
bcr07:  mov     A,Ucap
        cjne    A,#3,bcr11
        mov     R1,Ch23
        mov     R2,Ch24
        mov     R3,Ch25
        mov     R4,Ch26
        mov     R5,Ch27
        lcall   scnchr
        jnb     Stch,bcr08
        mov     Ucap,#8
        ljmp    bcr11
bcr08:  mov     R1,Ch23
        mov     R2,Ch24
        mov     R3,Ch25
        mov     R4,Ch26
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr09
        mov     Ucap,#7
        ljmp    bcr11
bcr09:  mov     R1,Ch23
        mov     R2,Ch24
        mov     R3,Ch25
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr10
        mov     Ucap,#6
        ljmp    bcr11
bcr10:  mov     R1,Ch23
        mov     R2,Ch24
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr11
        mov     Ucap,#5
;
bcr11:  mov     A,Ucap
        cjne    A,#4,bcr15
```

```
        mov     R1,Ch24
        mov     R2,Ch25
        mov     R3,Ch26
        mov     R4,Ch27
        mov     R5,Ch28
        lcall   scnchr
        jnb     Stch,bcr12
        mov     Ucap,#9
        ljmp    bcr15
bcr12:  mov     R1,Ch24
        mov     R2,Ch25
        mov     R3,Ch26
        mov     R4,Ch27
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr13
        mov     Ucap,#8
        ljmp    bcr15
bcr13:  mov     R1,Ch24
        mov     R2,Ch25
        mov     R3,Ch26
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr14
        mov     Ucap,#7
        ljmp    bcr15
bcr14:  mov     R1,Ch24
        mov     R2,Ch25
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr15
        mov     Ucap,#6
;
bcr15:  mov     A,Ucap
        cjne    A,#5,bcr19
        mov     R1,Ch25
        mov     R2,Ch26
        mov     R3,Ch27
        mov     R4,Ch28
        mov     R5,Ch29
        lcall   scnchr
        jnb     Stch,bcr16
        mov     Ucap,#10
        ljmp    bcr19
bcr16:  mov     R1,Ch25
        mov     R2,Ch26
        mov     R3,Ch27
        mov     R4,Ch28
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr17
        mov     Ucap,#9
        ljmp    bcr19
bcr17:  mov     R1,Ch25
        mov     R2,Ch26
        mov     R3,Ch27
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr18
        mov     Ucap,#8
        ljmp    bcr19
bcr18:  mov     R1,Ch25
        mov     R2,Ch26
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr19
        mov     Ucap,#7
;
bcr19:  mov     A,Ucap
        cjne    A,#6,bcr23
        mov     R1,Ch26
```

```
        mov     R2,Ch27
        mov     R3,Ch28
        mov     R4,Ch29
        mov     R5,Ch2A
        lcall   scnchr
        jnb     Stch,bcr20
        mov     Ucap,#11
        ljmp    bcr23
bcr20:  mov     R1,Ch26
        mov     R2,Ch27
        mov     R3,Ch28
        mov     R4,Ch29
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr21
        mov     Ucap,#10
        ljmp    bcr23
bcr21:  mov     R1,Ch26
        mov     R2,Ch27
        mov     R3,Ch28
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr22
        mov     Ucap,#9
        ljmp    bcr23
bcr22:  mov     R1,Ch26
        mov     R2,Ch27
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr23
        mov     Ucap,#8
;
bcr23:  mov     A,Ucap
        cjne    A,#7,bcr27
        mov     R1,Ch27
        mov     R2,Ch28
        mov     R3,Ch29
        mov     R4,Ch2A
        mov     R5,Ch2B
        lcall   scnchr
        jnb     Stch,bcr24
        mov     Ucap,#12
        ljmp    bcr27
bcr24:  mov     R1,Ch27
        mov     R2,Ch28
        mov     R3,Ch29
        mov     R4,Ch2A
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr25
        mov     Ucap,#11
        ljmp    bcr27
bcr25:  mov     R1,Ch27
        mov     R2,Ch28
        mov     R3,Ch29
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr26
        mov     Ucap,#10
        ljmp    bcr27
bcr26:  mov     R1,Ch27
        mov     R2,Ch28
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr27
        mov     Ucap,#9
;
bcr27:  mov     A,Ucap
        cjne    A,#8,bcr31
        mov     R1,Ch28
        mov     R2,Ch29
```

```
        mov     R3,Ch2A
        mov     R4,Ch2B
        mov     R5,Ch2C
        lcall   scnchr
        jnb     Stch,bcr28
        mov     Ucap,#13
        ljmp    bcr31
bcr28:  mov     R1,Ch28
        mov     R2,Ch29
        mov     R3,Ch2A
        mov     R4,Ch2B
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr29
        mov     Ucap,#12
        ljmp    bcr31
bcr29:  mov     R1,Ch28
        mov     R2,Ch29
        mov     R3,Ch2A
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr30
        mov     Ucap,#11
        ljmp    bcr31
bcr30:  mov     R1,Ch28
        mov     R2,Ch29
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr31
        mov     Ucap,#10
;
bcr31:  mov     A,Ucap
        cjne    A,#9,bcr35
        mov     R1,Ch29
        mov     R2,Ch2A
        mov     R3,Ch2B
        mov     R4,Ch2C
        mov     R5,Ch2D
        lcall   scnchr
        jnb     Stch,bcr32
        mov     Ucap,#14
        ljmp    bcr35
bcr32:  mov     R1,Ch29
        mov     R2,Ch2A
        mov     R3,Ch2B
        mov     R4,Ch2C
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr33
        mov     Ucap,#13
        ljmp    bcr35
bcr33:  mov     R1,Ch29
        mov     R2,Ch2A
        mov     R3,Ch2B
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr34
        mov     Ucap,#12
        ljmp    bcr35
bcr34:  mov     R1,Ch29
        mov     R2,Ch2A
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr35
        mov     Ucap,#11
;
bcr35:  mov     A,Ucap
        cjne    A,#10,bcr39
        mov     R1,Ch2A
        mov     R2,Ch2B
        mov     R3,Ch2C
```

```
        mov     R4,Ch2D
        mov     R5,Ch2E
        lcall   scnchr
        jnb     Stch,bcr36
        mov     Ucap,#15
        ljmp    bcr39
bcr36:  mov     R1,Ch2A
        mov     R2,Ch2B
        mov     R3,Ch2C
        mov     R4,Ch2D
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr37
        mov     Ucap,#14
        ljmp    bcr39
bcr37:  mov     R1,Ch2A
        mov     R2,Ch2B
        mov     R3,Ch2C
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr38
        mov     Ucap,#13
        ljmp    bcr39
bcr38:  mov     R1,Ch2A
        mov     R2,Ch2B
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr39
        mov     Ucap,#12
;
bcr39:  mov     A,Ucap
        cjne    A,#11,bcr43
        mov     R1,Ch2B
        mov     R2,Ch2C
        mov     R3,Ch2D
        mov     R4,Ch2E
        mov     R5,Ch2F
        lcall   scnchr
        jnb     Stch,bcr40
        mov     Ucap,#16
        ljmp    bcr43
bcr40:  mov     R1,Ch2B
        mov     R2,Ch2C
        mov     R3,Ch2D
        mov     R4,Ch2E
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr41
        mov     Ucap,#15
        ljmp    bcr43
bcr41:  mov     R1,Ch2B
        mov     R2,Ch2D
        mov     R3,Ch2D
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr42
        mov     Ucap,#14
        ljmp    bcr43
bcr42:  mov     R1,Ch2B
        mov     R2,Ch2C
        mov     R3,#' '
        mov     R4,#' '
        mov     R5,#' '
        lcall   scnchr
        jnb     Stch,bcr43
        mov     Ucap,#13
;
bcr43:  mov     A,Ucap
        cjne    A,#12,bcr46
        mov     R1,Ch2C
        mov     R2,Ch2D
        mov     R3,Ch2E
        mov     R4,Ch2F
```

```
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr44
          mov     Ucap,#16
          ljmp    bcr46
bcr44:    mov     R1,Ch2C
          mov     R2,Ch2D
          mov     R3,Ch2E
          mov     R4,#' '
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr45
          mov     Ucap,#15
          ljmp    bcr46
bcr45:    mov     R1,Ch2C
          mov     R2,Ch2D
          mov     R3,#' '
          mov     R4,#' '
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr46
          mov     Ucap,#14
;
bcr46:    mov     A,Ucap
          cjne    A,#13,bcr48
          mov     R1,Ch2D
          mov     R2,Ch2E
          mov     R3,Ch2F
          mov     R4,#' '
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr47
          mov     Ucap,#16
          ljmp    bcr48
bcr47:    mov     R1,Ch2D
          mov     R2,Ch2E
          mov     R3,#' '
          mov     R4,#' '
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr48
          mov     Ucap,#15
;
bcr48:    mov     A,Ucap
          cjne    A,#14,bcr49
          mov     R1,Ch2E
          mov     R2,Ch2F
          mov     R3,#' '
          mov     R4,#' '
          mov     R5,#' '
          lcall   scnchr
          jnb     Stch,bcr49
          mov     Ucap,#16
;
bcr49:    ret
;
scnchr:   jb      Sttp,scnch0
          lcall   scnind
          ljmp    scnch1
scnch0:   lcall   scneng
scnch1:   ret
;
scnind:   clr     Stch                    ; reset status suku kata
sci000:   mov     DPTR,#chi000
          lcall   cekchr
          jnb     Stch,sci001
          lcall   imut
          ljmp    sci999
sci001:   mov     DPTR,#chi001
          lcall   cekchr
          jnb     Stch,sci002
          lcall   ilak
          ljmp    sci999
sci002:   mov     DPTR,#chi002
          lcall   cekchr
          jnb     Stch,sci003
          lcall   ipem
```

```
        ljmp    sci999
sci003: mov     DPTR,#chi003
        lcall   cekchr
        jnb     Stch,sci004
        lcall   ibang
        ljmp    sci999
sci004: mov     DPTR,#chi004
        lcall   cekchr
        jnb     Stch,sci005
        lcall   ikit
        ljmp    sci999
sci005: mov     DPTR,#chi005
        lcall   cekchr
        jnb     Stch,sci006
        lcall   ilis
        ljmp    sci999
sci006: mov     DPTR,#chi006
        lcall   cekchr
        jnb     Stch,sci007
        lcall   itrik
        ljmp    sci999
sci007: mov     DPTR,#chi007
        lcall   cekchr
        jnb     Stch,sci008
        lcall   imeng
        ljmp    sci999
sci008: mov     DPTR,#chi008
        lcall   cekchr
        jnb     Stch,sci009
        lcall   iak
        ljmp    sci999
sci009: mov     DPTR,#chi009
        lcall   cekchr
        jnb     Stch,sci010
        lcall   itif
        ljmp    sci999
sci010: mov     DPTR,#chi010
        lcall   cekchr
        jnb     Stch,sci011
        lcall   ikan
        ljmp    sci999
sci011: mov     DPTR,#chi011
        lcall   cekchr
        jnb     Stch,sci012
        lcall   isa
        ljmp    sci999
sci012: mov     DPTR,#chi012
        lcall   cekchr
        jnb     Stch,sci013
        lcall   ituan
        ljmp    sci999
sci013: mov     DPTR,#chi013
        lcall   cekchr
        jnb     Stch,sci014
        lcall   iarus
        ljmp    sci999
sci014: mov     DPTR,#chi014
        lcall   cekchr
        jnb     Stch,sci015
        lcall   ipeng
        ljmp    sci999
sci015: mov     DPTR,#chi015
        lcall   cekchr
        jnb     Stch,sci016
        lcall   iukur
        ljmp    sci999
sci016: mov     DPTR,#chi016
        lcall   cekchr
        jnb     Stch,sci017
        lcall   ian
        ljmp    sci999
sci017: mov     DPTR,#chi017
        lcall   cekchr
        jnb     Stch,sci018
        lcall   itena
        ljmp    sci999
sci018: mov     DPTR,#chi018
```

```
        lcall   cekchr
        jnb     Stch,sci019
        lcall   iku
        ljmp    sci999
sci019: mov     DPTR,#chi019
        lcall   cekchr
        jnb     Stch,sci020
        lcall   itub
        ljmp    sci999
sci020: mov     DPTR,#chi020
        lcall   cekchr
        jnb     Stch,sci021
        lcall   ipo
        ljmp    sci999
sci021: mov     DPTR,#chi021
        lcall   cekchr
        jnb     Stch,sci022
        lcall   isi
        ljmp    sci999
sci022: mov     DPTR,#chi022
        lcall   cekchr
        jnb     Stch,sci023
        lcall   ipe
        ljmp    sci999
sci023: mov     DPTR,#chi023
        lcall   cekchr
        jnb     Stch,sci024
        lcall   inang
        ljmp    sci999
sci024: mov     DPTR,#chi024
        lcall   cekchr
        jnb     Stch,sci025
        lcall   ikal
        ljmp    sci999
sci025: mov     DPTR,#chi025
        lcall   cekchr
        jnb     Stch,sci026
        lcall   itir
        ljmp    sci999
sci026: mov     DPTR,#chi026
        lcall   cekchr
        jnb     Stch,sci027
        lcall   iba
        ljmp    sci999
sci027: mov     DPTR,#chi027
        lcall   cekchr
        jnb     Stch,sci028
        lcall   ite
        ljmp    sci999
sci028: mov     DPTR,#chi028
        lcall   cekchr
        jnb     Stch,sci029
        lcall   irai
        ljmp    sci999
sci029: mov     DPTR,#chi029
        lcall   cekchr
        jnb     Stch,sci030
        lcall   iber
        ljmp    sci999
sci030: mov     DPTR,#chi030
        lcall   cekchr
        jnb     Stch,sci031
        lcall   igan
        ljmp    sci999
sci031: mov     DPTR,#chi031
        lcall   cekchr
        jnb     Stch,sci032
        lcall   ida
        ljmp    sci999
sci032: mov     DPTR,#chi032
        lcall   cekchr
        jnb     Stch,sci033
        lcall   inyang
        ljmp    sci999
sci033: mov     DPTR,#chi033
        lcall   cekchr
        jnb     Stch,sci034
```

```
        lcall   iga
        ljmp    sci999
sci034: mov     DPTR,#chi034
        lcall   cekchr
        jnb     Stch,sci035
        lcall   ika
        ljmp    sci999
sci035: mov     DPTR,#chi035
        lcall   cekchr
        jnb     Stch,sci036
        lcall   ipa
        ljmp    sci999
sci036: mov     DPTR,#chi036
        lcall   cekchr
        jnb     Stch,sci037
        lcall   itan
        ljmp    sci999
sci037: mov     DPTR,#chi037
        lcall   cekchr
        jnb     Stch,sci038
        lcall   itor
        ljmp    sci999
sci038: mov     DPTR,#chi038
        lcall   cekchr
        jnb     Stch,sci039
        lcall   ivo
        ljmp    sci999
sci039: mov     DPTR,#chi039
        lcall   cekchr
        jnb     Stch,sci040
        lcall   ilu
        ljmp    sci999
sci040: mov     DPTR,#chi040
        lcall   cekchr
        jnb     Stch,sci041
        lcall   ime
        ljmp    sci999
sci041: mov     DPTR,#chi041
        lcall   cekchr
        jnb     Stch,sci042
        lcall   ine
        ljmp    sci999
sci042: mov     DPTR,#chi042
        lcall   cekchr
        jnb     Stch,sci043
        lcall   ikan
        ljmp    sci999
sci043: mov     DPTR,#chi043
        lcall   cekchr
        jnb     Stch,sci044
        lcall   iduk
        ljmp    sci999
sci044: mov     DPTR,#chi044
        lcall   cekchr
        jnb     Stch,sci045
        lcall   ialat
        ljmp    sci999
sci045: mov     DPTR,#chi045
        lcall   cekchr
        jnb     Stch,sci046
        lcall   ide
        ljmp    sci999
sci046: mov     DPTR,#chi046
        lcall   cekchr
        jnb     Stch,sci047
        lcall   itek
        ljmp    sci999
sci047: mov     DPTR,#chi047
        lcall   cekchr
        jnb     Stch,sci048
        lcall   idio
        ljmp    sci999
sci048: mov     DPTR,#chi048
        lcall   cekchr
        jnb     Stch,sci049
        lcall   ija
        ljmp    sci999
```

```
sci049: mov     DPTR,#chi049
        lcall   cekchr
        jnb     Stch,sci050
        lcall   irak
        ljmp    sci999
sci050: mov     DPTR,#chi050
        lcall   cekchr
        jnb     Stch,sci051
        lcall   idi
        ljmp    sci999
sci051: mov     DPTR,#chi051
        lcall   cekchr
        jnb     Stch,sci052
        lcall   ina
        ljmp    sci999
sci052: mov     DPTR,#chi052
        lcall   cekchr
        jnb     Stch,sci053
        lcall   imo
        ljmp    sci999
sci053: mov     DPTR,#chi053
        lcall   cekchr
        jnb     Stch,sci054
        lcall   igu
        ljmp    sci999
sci054: mov     DPTR,#chi054
        lcall   cekchr
        jnb     Stch,sci055
        lcall   ike
        ljmp    sci999
sci055: mov     DPTR,#chi055
        lcall   cekchr
        jnb     Stch,sci056
        lcall   imag
        ljmp    sci999
sci056: mov     DPTR,#chi056
        lcall   cekchr
        jnb     Stch,sci057
        lcall   inyiar
        ljmp    sci999
sci057: mov     DPTR,#chi057
        lcall   cekchr
        jnb     Stch,sci058
        lcall   ila
        ljmp    sci999
sci058: mov     DPTR,#chi058
        lcall   cekchr
        jnb     Stch,sci059
        lcall   ihan
        ljmp    sci999
sci059: mov     DPTR,#chi059
        lcall   cekchr
        jnb     Stch,sci060
        lcall   ieks
        ljmp    sci999
sci060: mov     DPTR,#chi060
        lcall   cekchr
        jnb     Stch,sci061
        lcall   ipan
        ljmp    sci999
sci061: mov     DPTR,#chi061
        lcall   cekchr
        jnb     Stch,sci062
        lcall   iwat
        ljmp    sci999
sci062: mov     DPTR,#chi062
        lcall   cekchr
        jnb     Stch,sci063
        lcall   ipi
        ljmp    sci999
sci063: mov     DPTR,#chi063
        lcall   cekchr
        jnb     Stch,sci064
        lcall   ijar
        ljmp    sci999
sci064: mov     DPTR,#chi064
        lcall   cekchr
```

```
        jnb     Stch,sci065
        lcall   iri
        ljmp    sci999
sci065: mov     DPTR,#chi065
        lcall   cekchr
        jnb     Stch,sci066
        lcall   ingan
        ljmp    sci999
sci066: mov     DPTR,#chi066
        lcall   cekchr
        jnb     Stch,sci067
        lcall   ijum
        ljmp    sci999
sci067: mov     DPTR,#chi067
        lcall   cekchr
        jnb     Stch,sci068
        lcall   ilah
        ljmp    sci999
sci068: mov     DPTR,#chi068
        lcall   cekchr
        jnb     Stch,sci069
        lcall   ige
        ljmp    sci999
sci069: mov     DPTR,#chi069
        lcall   cekchr
        jnb     Stch,sci070
        lcall   ita
        ljmp    sci999
sci070: mov     DPTR,#chi070
        lcall   cekchr
        jnb     Stch,sci071
        lcall   iran
        ljmp    sci999
sci071: mov     DPTR,#chi071
        lcall   cekchr
        jnb     Stch,sci072
        lcall   ikar
        ljmp    sci999
sci072: mov     DPTR,#chi072
        lcall   cekchr
        jnb     Stch,sci073
        lcall   iya
        ljmp    sci999
sci073: mov     DPTR,#chi073
        lcall   cekchr
        jnb     Stch,sci074
        lcall   ibe
        ljmp    sci999
sci074: mov     DPTR,#chi074
        lcall   cekchr
        jnb     Stch,sci075
        lcall   irat
        ljmp    sci999
sci075: mov     DPTR,#chi075
        lcall   cekchr
        jnb     Stch,sci076
        lcall   iim
        ljmp    sci999
sci076: mov     DPTR,#chi076
        lcall   cekchr
        jnb     Stch,sci077
        lcall   ipe
        ljmp    sci999
sci077: mov     DPTR,#chi077
        lcall   cekchr
        jnb     Stch,sci078
        lcall   idan
        ljmp    sci999
sci078: mov     DPTR,#chi078
        lcall   cekchr
        jnb     Stch,sci079
        lcall   iin
        ljmp    sci999
sci079: mov     DPTR,#chi079
        lcall   cekchr
        jnb     Stch,sci080
        lcall   ima
```

```
        ljmp    sci999
sci080: mov     DPTR,#chi080
        lcall   cekchr
        jnb     Stch,sci081
        lcall   isu
        ljmp    sci999
sci081: mov     DPTR,#chi081
        lcall   cekchr
        jnb     Stch,sci082
        lcall   izat
        ljmp    sci999
sci082: mov     DPTR,#chi082
        lcall   cekchr
        jnb     Stch,sci083
        lcall   icair
        ljmp    sci999
sci083: mov     DPTR,#chi083
        lcall   cekchr
        jnb     Stch,sci999
        lcall   inet
        ljmp    sci999
sci999: ret
;
scneng: clr     Stch                    ; reset status suku kata
sce000: mov     DPTR,#che000
        lcall   cekchr
        jnb     Stch,sce001
        lcall   eab
        ljmp    sce999
sce001: mov     DPTR,#che001
        lcall   cekchr
        jnb     Stch,sce002
        lcall   eac
        ljmp    sce999
sce002: mov     DPTR,#che002
        lcall   cekchr
        jnb     Stch,sce003
        lcall   eam
        ljmp    sce999
sce003: mov     DPTR,#che003
        lcall   cekchr
        jnb     Stch,sce004
        lcall   ean
        ljmp    sce999
sce004: mov     DPTR,#che004
        lcall   cekchr
        jnb     Stch,sce005
        lcall   ear
        ljmp    sce999
sce005: mov     DPTR,#che005
        lcall   cekchr
        jnb     Stch,sce006
        lcall   eso
        ljmp    sce999
sce006: mov     DPTR,#che006
        lcall   cekchr
        jnb     Stch,sce007
        lcall   ecu
        ljmp    sce999
sce007: mov     DPTR,#che007
        lcall   cekchr
        jnb     Stch,sce008
        lcall   eti
        ljmp    sce999
sce008: mov     DPTR,#che008
        lcall   cekchr
        jnb     Stch,sce009
        lcall   epere
        ljmp    sce999
sce009: mov     DPTR,#che009
        lcall   cekchr
        jnb     Stch,sce010
        lcall   eten
        ljmp    sce999
sce010: mov     DPTR,#che010
        lcall   cekchr
        jnb     Stch,sce011
```

```
        lcall   eres
        ljmp    sce999
sce011: mov     DPTR,#che011
        lcall   cekchr
        jnb     Stch,sce012
        lcall   elute
        ljmp    sce999
sce012: mov     DPTR,#che012
        lcall   cekchr
        jnb     Stch,sce013
        lcall   evate
        ljmp    sce999
sce013: mov     DPTR,#che013
        lcall   cekchr
        jnb     Stch,sce014
        lcall   eme
        ljmp    sce999
sce014: mov     DPTR,#che014
        lcall   cekchr
        jnb     Stch,sce015
        lcall   eter
        ljmp    sce999
sce015: mov     DPTR,#che015
        lcall   cekchr
        jnb     Stch,sce016
        lcall   ena
        ljmp    sce999
sce016: mov     DPTR,#che016
        lcall   cekchr
        jnb     Stch,sce017
        lcall   ede
        ljmp    sce999
sce017: mov     DPTR,#che017
        lcall   cekchr
        jnb     Stch,sce018
        lcall   etor
        ljmp    sce999
sce018: mov     DPTR,#che018
        lcall   cekchr
        jnb     Stch,sce019
        lcall   ebat
        ljmp    sce999
sce019: mov     DPTR,#che019
        lcall   cekchr
        jnb     Stch,sce020
        lcall   etery
        ljmp    sce999
sce020: mov     DPTR,#che020
        lcall   cekchr
        jnb     Stch,sce021
        lcall   efer
        ljmp    sce999
sce021: mov     DPTR,#che021
        lcall   cekchr
        jnb     Stch,sce022
        lcall   ebi
        ljmp    sce999
sce022: mov     DPTR,#che022
        lcall   cekchr
        jnb     Stch,sce023
        lcall   epo
        ljmp    sce999
sce023: mov     DPTR,#che023
        lcall   cekchr
        jnb     Stch,sce024
        lcall   elar
        ljmp    sce999
sce024: mov     DPTR,#che024
        lcall   cekchr
        jnb     Stch,sce025
        lcall   ebuf
        ljmp    sce999
sce025: mov     DPTR,#che025
        lcall   cekchr
        jnb     Stch,sce026
        lcall   eca
        ljmp    sce999
```

```
sce026: mov     DPTR,#che026
        lcall   cekchr
        jnb     Stch,sce027
        lcall   etan
        ljmp    sce999
sce027: mov     DPTR,#che027
        lcall   cekchr
        jnb     Stch,sce028
        lcall   etho
        ljmp    sce999
sce028: mov     DPTR,#che028
        lcall   cekchr
        jnb     Stch,sce029
        lcall   epa
        ljmp    sce999
sce029: mov     DPTR,#che029
        lcall   cekchr
        jnb     Stch,sce030
        lcall   ece
        ljmp    sce999
sce030: mov     DPTR,#che030
        lcall   cekchr
        jnb     Stch,sce031
        lcall   eci
        ljmp    sce999
sce031: mov     DPTR,#che031
        lcall   cekchr
        jnb     Stch,sce032
        lcall   ety
        ljmp    sce999
sce032: mov     DPTR,#che032
        lcall   cekchr
        jnb     Stch,sce033
        lcall   econ
        ljmp    sce999
sce033: mov     DPTR,#che033
        lcall   cekchr
        jnb     Stch,sce034
        lcall   educ
        ljmp    sce999
sce034: mov     DPTR,#che034
        lcall   cekchr
        jnb     Stch,sce035
        lcall   erent
        ljmp    sce999
sce035: mov     DPTR,#che035
        lcall   cekchr
        jnb     Stch,sce036
        lcall   edio
        ljmp    sce999
sce036: mov     DPTR,#che036
        lcall   cekchr
        jnb     Stch,sce037
        lcall   emo
        ljmp    sce999
sce037: mov     DPTR,#che037
        lcall   cekchr
        jnb     Stch,sce038
        lcall   etive
        ljmp    sce999
sce038: mov     DPTR,#che038
        lcall   cekchr
        jnb     Stch,sce039
        lcall   etro
        ljmp    sce999
sce039: mov     DPTR,#che039
        lcall   cekchr
        jnb     Stch,sce040
        lcall   etic
        ljmp    sce999
sce040: mov     DPTR,#che040
        lcall   cekchr
        jnb     Stch,sce041
        lcall   eer
        ljmp    sce999
sce041: mov     DPTR,#che041
        lcall   cekchr
```

```
        jnb     Stch,sce042
        lcall   epan
        ljmp    sce999
sce042: mov     DPTR,#che042
        lcall   cekchr
        jnb     Stch,sce043
        lcall   efil
        ljmp    sce999
sce043: mov     DPTR,#che043
        lcall   cekchr
        jnb     Stch,sce044
        lcall   equen
        ljmp    sce999
sce044: mov     DPTR,#che044
        lcall   cekchr
        jnb     Stch,sce045
        lcall   egra
        ljmp    sce999
sce045: mov     DPTR,#che045
        lcall   cekchr
        jnb     Stch,sce046
        lcall   epe
        ljmp    sce999
sce046: mov     DPTR,#che046
        lcall   cekchr
        jnb     Stch,sce047
        lcall   eput
        ljmp    sce999
sce047: mov     DPTR,#che047
        lcall   cekchr
        jnb     Stch,sce048
        lcall   enet
        ljmp    sce999
sce048: mov     DPTR,#che048
        lcall   cekchr
        jnb     Stch,sce049
        lcall   etec
        ljmp    sce999
sce049: mov     DPTR,#che049
        lcall   cekchr
        jnb     Stch,sce050
        lcall   edis
        ljmp    sce999
sce050: mov     DPTR,#che050
        lcall   cekchr
        jnb     Stch,sce051
        lcall   eef
        ljmp    sce999
sce051: mov     DPTR,#che051
        lcall   cekchr
        jnb     Stch,sce052
        lcall   eec
        ljmp    sce999
sce052: mov     DPTR,#che052
        lcall   cekchr
        jnb     Stch,sce053
        lcall   emag
        ljmp    sce999
sce053: mov     DPTR,#che053
        lcall   cekchr
        jnb     Stch,sce054
        lcall   eem
        ljmp    sce999
sce054: mov     DPTR,#che054
        lcall   cekchr
        jnb     Stch,sce055
        lcall   eror
        ljmp    sce999
sce055: mov     DPTR,#che055
        lcall   cekchr
        jnb     Stch,sce056
        lcall   ese
        ljmp    sce999
sce056: mov     DPTR,#che056
        lcall   cekchr
        jnb     Stch,sce057
        lcall   ement
```

```
        ljmp    sce999
sce057: mov     DPTR,#che057
        lcall   cekchr
        jnb     Stch,sce058
        lcall   ecy
        ljmp    sce999
sce058: mov     DPTR,#che058
        lcall   cekchr
        jnb     Stch,sce059
        lcall   evi
        ljmp    sce999
sce059: mov     DPTR,#che059
        lcall   cekchr
        jnb     Stch,sce060
        lcall   edan
        ljmp    sce999
sce060: mov     DPTR,#che060
        lcall   cekchr
        jnb     Stch,sce061
        lcall   eli
        ljmp    sce999
sce061: mov     DPTR,#che061
        lcall   cekchr
        jnb     Stch,sce062
        lcall   eano
        ljmp    sce999
sce062: mov     DPTR,#che062
        lcall   cekchr
        jnb     Stch,sce063
        lcall   ecur
        ljmp    sce999
sce063: mov     DPTR,#che063
        lcall   cekchr
        jnb     Stch,sce064
        lcall   edy
        ljmp    sce999
sce064: mov     DPTR,#che064
        lcall   cekchr
        jnb     Stch,sce065
        lcall   efec
        ljmp    sce999
sce065: mov     DPTR,#che065
        lcall   cekchr
        jnb     Stch,sce066
        lcall   eel
        ljmp    sce999
sce066: mov     DPTR,#che066
        lcall   cekchr
        jnb     Stch,sce067
        lcall   ene
        ljmp    sce999
sce067: mov     DPTR,#che067
        lcall   cekchr
        jnb     Stch,sce068
        lcall   eit
        ljmp    sce999
sce068: mov     DPTR,#che068
        lcall   cekchr
        jnb     Stch,sce069
        lcall   eex
        ljmp    sce999
sce069: mov     DPTR,#che069
        lcall   cekchr
        jnb     Stch,sce070
        lcall   efila
        ljmp    sce999
sce070: mov     DPTR,#che070
        lcall   cekchr
        jnb     Stch,sce071
        lcall   efre
        ljmp    sce999
sce071: mov     DPTR,#che071
        lcall   cekchr
        jnb     Stch,sce072
        lcall   efuel
        ljmp    sce999
sce072: mov     DPTR,#che072
```

```
        lcall   cekchr
        jnb     Stch,sce073
        lcall   eim
        ljmp    sce999
sce073: mov     DPTR,#che073
        lcall   cekchr
        jnb     Stch,sce074
        lcall   ein
        ljmp    sce999
sce074: mov     DPTR,#che074
        lcall   cekchr
        jnb     Stch,sce999
        lcall   equid
        ljmp    sce999
sce999: ret
;
cekchr: clr     A
        clr     C
        movc    A,@A+DPTR
        mov     B,R1
        subb    A,B
        cjne    A,#0,ckchr0
        inc     DPTR
        movc    A,@A+DPTR
        mov     B,R2
        subb    A,B
        cjne    A,#0,ckchr0
        inc     DPTR
        movc    A,@A+DPTR
        mov     B,R3
        subb    A,B
        cjne    A,#0,ckchr0
        inc     DPTR
        movc    A,@A+DPTR
        mov     B,R4
        subb    A,B
        cjne    A,#0,ckchr0
        inc     DPTR
        movc    A,@A+DPTR
        mov     B,R5
        subb    A,B
        cjne    A,#0,ckchr0
        setb    Stch
        ljmp    ckchr1
ckchr0: clr     Stch
ckchr1: ret
;
imut:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#000                 ; | address 000
        lcall   bcris0                  ; |
        ret                             ;/
;
ilak:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#005                 ; | address 005
        lcall   bcris0                  ; |
        ret                             ;/
;
ipem:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#010                 ; | address 010
        lcall   bcris0                  ; |
        ret                             ;/
;
ibang:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#015                 ; | address 015
        lcall   bcris0                  ; |
        ret                             ;/
;
ikit:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#020                 ; | address 020
        lcall   bcris0                  ; |
        ret                             ;/
;
```

```
ilis:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#025                 ; | address 025
        lcall   bcris0                  ; |
        ret                             ;/
;
itrik:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#030                 ; | address 030
        lcall   bcris0                  ; |
        ret                             ;/
;
imeng:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#035                 ; | address 035
        lcall   bcris0                  ; |
        ret                             ;/
;
iak:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#040                 ; | address 040
        lcall   bcris0                  ; |
        ret                             ;/
;
itif:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#045                 ; | address 045
        lcall   bcris0                  ; |
        ret                             ;/
;
ikan:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#050                 ; | address 050
        lcall   bcris0                  ; |
        ret                             ;/
;
isa:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#055                 ; | address 055
        lcall   bcris0                  ; |
        ret                             ;/
;
ituan:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#060                 ; | address 060
        lcall   bcris0                  ; |
        ret                             ;/
;
iarus:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#065                 ; | address 065
        lcall   bcris0                  ; |
        ret                             ;/
;
ipeng:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#070                 ; | address 070
        lcall   bcris0                  ; |
        ret                             ;/
;
iukur:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#075                 ; | address 075
        lcall   bcris0                  ; |
        ret                             ;/
;
ian:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#080                 ; | address 080
        lcall   bcris0                  ; |
        ret                             ;/
;
itena:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#085                 ; | address 085
        lcall   bcris0                  ; |
        ret                             ;/
```

```
;
iku:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#090                 ; | address 090
        lcall   bcris0                  ; |
        ret                             ;/
;
itub:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#095                 ; | address 095
        lcall   bcris0                  ; |
        ret                             ;/
;
ipo:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#100                 ; | address 100
        lcall   bcris0                  ; |
        ret                             ;/
;
isi:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#105                 ; | address 105
        lcall   bcris0                  ; |
        ret                             ;/
;
ipe:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#110                 ; | address 110
        lcall   bcris0                  ; |
        ret                             ;/
;
inang:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#115                 ; | address 115
        lcall   bcris0                  ; |
        ret                             ;/
;
ikal:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#120                 ; | address 120
        lcall   bcris0                  ; |
        ret                             ;/
;
itir:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#125                 ; | address 125
        lcall   bcris0                  ; |
        ret                             ;/
;
iba:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#130                 ; | address 130
        lcall   bcris0                  ; |
        ret                             ;/
;
ite:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#135                 ; | address 135
        lcall   bcris0                  ; |
        ret                             ;/
;
irai:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#140                 ; | address 140
        lcall   bcris0                  ; |
        ret                             ;/
;
iber:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#145                 ; | address 145
        lcall   bcris0                  ; |
        ret                             ;/
;
igan:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#150                 ; | address 150
        lcall   bcris0                  ; |
```

```
        ret                             ;/
;
ida:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#155                 ; | address 155
        lcall   bcris0                  ; |
        ret                             ;/
;
inyang: clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#160                 ; | address 160
        lcall   bcris0                  ; |
        ret                             ;/
;
iga:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#165                 ; | address 165
        lcall   bcris0                  ; |
        ret                             ;/
;
ika:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#170                 ; | address 170
        lcall   bcris0                  ; |
        ret                             ;/
;
ipa:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#175                 ; | address 175
        lcall   bcris0                  ; |
        ret                             ;/
;
itan:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#180                 ; | address 180
        lcall   bcris0                  ; |
        ret                             ;/
;
itor:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#185                 ; | address 185
        lcall   bcris0                  ; |
        ret                             ;/
;
ivo:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#190                 ; | address 190
        lcall   bcris0                  ; |
        ret                             ;/
;
ilu:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#195                 ; | address 195
        lcall   bcris0                  ; |
        ret                             ;/
;
ime:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#200                 ; | address 200
        lcall   bcris0                  ; |
        ret                             ;/
;
ine:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#205                 ; | address 205
        lcall   bcris0                  ; |
        ret                             ;/
;
;ikan:   clr     ISLO                   ;\
;        clr     ISHI                   ; |
;        mov     P0,#210                ; | address 210
;        lcall   bcris0                 ; |
;        ret                            ;/
;
iduk:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#215                 ; | address 215
```

```
        lcall   bcris0                  ; |
        ret                             ;/
;
ialat:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#220                 ; | address 220
        lcall   bcris0                  ; |
        ret                             ;/
;
ide:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#225                 ; | address 225
        lcall   bcris0                  ; |
        ret                             ;/
;
itek:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#230                 ; | address 230
        lcall   bcris0                  ; |
        ret                             ;/
;
idio:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#235                 ; | address 235
        lcall   bcris0                  ; |
        ret                             ;/
;
ija:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#240                 ; | address 240
        lcall   bcris0                  ; |
        ret                             ;/
;
irak:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#245                 ; | address 245
        lcall   bcris0                  ; |
        ret                             ;/
;
idi:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#250                 ; | address 250
        lcall   bcris0                  ; |
        ret                             ;/
;
ina:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#255                 ; | address 255
        lcall   bcris0                  ; |
        ret                             ;/
;
imo:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#004                 ; | address 260
        lcall   bcris0                  ; |
        ret                             ;/
;
igu:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#009                 ; | address 265
        lcall   bcris0                  ; |
        ret                             ;/
;
ike:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#014                 ; | address 270
        lcall   bcris0                  ; |
        ret                             ;/
;
imag:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#019                 ; | address 275
        lcall   bcris0                  ; |
        ret                             ;/
;
inyiar: setb    ISLO                    ;\
        clr     ISHI                    ; |
```

```
        mov     P0,#024                 ; | address 280
        lcall   bcris0                  ; |
        ret                             ;/
;
ila:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#029                 ; | address 285
        lcall   bcris0                  ; |
        ret                             ;/
;
ihan:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#034                 ; | address 290
        lcall   bcris0                  ; |
        ret                             ;/
;
ieks:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#039                 ; | address 295
        lcall   bcris0                  ; |
        ret                             ;/
;
ipan:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#044                 ; | address 300
        lcall   bcris0                  ; |
        ret                             ;/
;
iwat:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#049                 ; | address 305
        lcall   bcris0                  ; |
        ret                             ;/
;
ipi:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#054                 ; | address 310
        lcall   bcris0                  ; |
        ret                             ;/
;
ijar:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#059                 ; | address 315
        lcall   bcris0                  ; |
        ret                             ;/
;
iri:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#064                 ; | address 320
        lcall   bcris0                  ; |
        ret                             ;/
;
ingan:  setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#069                 ; | address 325
        lcall   bcris0                  ; |
        ret                             ;/
;
ijum:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#074                 ; | address 330
        lcall   bcris0                  ; |
        ret                             ;/
;
ilah:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#079                 ; | address 335
        lcall   bcris0                  ; |
        ret                             ;/
;
ige:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#084                 ; | address 340
        lcall   bcris0                  ; |
        ret                             ;/
;
ita:    setb    ISLO                    ;\
```

```
        clr     ISHI                    ; |
        mov     P0,#089                 ; | address 345
        lcall   bcris0                  ; |
        ret                             ;/
;
iran:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#094                 ; | address 350
        lcall   bcris0                  ; |
        ret                             ;/
;
ikar:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#099                 ; | address 355
        lcall   bcris0                  ; |
        ret                             ;/
;
iya:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#104                 ; | address 360
        lcall   bcris0                  ; |
        ret                             ;/
;
ibe:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#109                 ; | address 365
        lcall   bcris0                  ; |
        ret                             ;/
;
irat:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#114                 ; | address 370
        lcall   bcris0                  ; |
        ret                             ;/
;
iim:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#119                 ; | address 375
        lcall   bcris0                  ; |
        ret                             ;/
;
;ipe:    setb    ISLO                   ;\
;        clr     ISHI                   ; |
;        mov     P0,#124                ; | address 380
;        lcall   bcris0                 ; |
;        ret                            ;/
;
idan:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#129                 ; | address 385
        lcall   bcris0                  ; |
        ret                             ;/
;
iin:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#134                 ; | address 390
        lcall   bcris0                  ; |
        ret                             ;/
;
ima:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#139                 ; | address 395
        lcall   bcris0                  ; |
        ret                             ;/
;
isu:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#144                 ; | address 400
        lcall   bcris0                  ; |
        ret                             ;/
;
izat:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#149                 ; | address 405
        lcall   bcris0                  ; |
        ret                             ;/
;
```

```
icair:  setb    ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#154             ; | address 410
        lcall   bcris0              ; |
        ret                         ;/
;
inet:   setb    ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#159             ; | address 415
        lcall   bcris0              ; |
        ret                         ;/
;
eab:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#000             ; | address 000
        lcall   bcris1              ; |
        ret                         ;/
;
eac:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#005             ; | address 005
        lcall   bcris1              ; |
        ret                         ;/
;
eam:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#010             ; | address 010
        lcall   bcris1              ; |
        ret                         ;/
;
ean:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#015             ; | address 015
        lcall   bcris1              ; |
        ret                         ;/
;
ear:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#020             ; | address 020
        lcall   bcris1              ; |
        ret                         ;/
;
eso:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#025             ; | address 025
        lcall   bcris1              ; |
        ret                         ;/
;
ecu:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#030             ; | address 030
        lcall   bcris1              ; |
        ret                         ;/
;
eti:    clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#035             ; | address 035
        lcall   bcris1              ; |
        ret                         ;/
;
epere:  clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#040             ; | address 040
        lcall   bcris1              ; |
        ret                         ;/
;
eten:   clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#045             ; | address 045
        lcall   bcris1              ; |
        ret                         ;/
;
eres:   clr     ISLO                ;\
        clr     ISHI                ; |
        mov     P0,#050             ; | address 050
        lcall   bcris1              ; |
        ret                         ;/
```

```
;
elute:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#055                 ; | address 055
        lcall   bcris1                  ; |
        ret                             ;/
;
evate:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#060                 ; | address 060
        lcall   bcris1                  ; |
        ret                             ;/
;
eme:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#065                 ; | address 065
        lcall   bcris1                  ; |
        ret                             ;/
;
eter:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#070                 ; | address 070
        lcall   bcris1                  ; |
        ret                             ;/
;
ena:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#075                 ; | address 075
        lcall   bcris1                  ; |
        ret                             ;/
;
ede:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#080                 ; | address 080
        lcall   bcris1                  ; |
        ret                             ;/
;
etor:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#085                 ; | address 085
        lcall   bcris1                  ; |
        ret                             ;/
;
ebat:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#090                 ; | address 090
        lcall   bcris1                  ; |
        ret                             ;/
;
etery:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#095                 ; | address 095
        lcall   bcris1                  ; |
        ret                             ;/
;
efer:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#100                 ; | address 100
        lcall   bcris1                  ; |
        ret                             ;/
;
ebi:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#105                 ; | address 105
        lcall   bcris1                  ; |
        ret                             ;/
;
epo:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#110                 ; | address 110
        lcall   bcris1                  ; |
        ret                             ;/
;
elar:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#115                 ; | address 115
        lcall   bcris1                  ; |
```

```
        ret                             ;/
;
ebuf:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#120                 ; | address 120
        lcall   bcris1                  ; |
        ret                             ;/
;
eca:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#125                 ; | address 125
        lcall   bcris1                  ; |
        ret                             ;/
;
etan:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#130                 ; | address 130
        lcall   bcris1                  ; |
        ret                             ;/
;
etho:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#135                 ; | address 135
        lcall   bcris1                  ; |
        ret                             ;/
;
epa:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#140                 ; | address 140
        lcall   bcris1                  ; |
        ret                             ;/
;
ece:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#145                 ; | address 145
        lcall   bcris1                  ; |
        ret                             ;/
;
eci:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#150                 ; | address 150
        lcall   bcris1                  ; |
        ret                             ;/
;
ety:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#155                 ; | address 155
        lcall   bcris1                  ; |
        ret                             ;/
;
econ:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#160                 ; | address 160
        lcall   bcris1                  ; |
        ret                             ;/
;
educ:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#165                 ; | address 165
        lcall   bcris1                  ; |
        ret                             ;/
;
erent:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#170                 ; | address 170
        lcall   bcris1                  ; |
        ret                             ;/
;
edio:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#175                 ; | address 175
        lcall   bcris1                  ; |
        ret                             ;/
;
emo:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#180                 ; | address 180
```

```
        lcall   bcris1                  ; |
        ret                             ;/
;
etive:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#185                 ; | address 185
        lcall   bcris1                  ; |
        ret                             ;/
;
etro:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#190                 ; | address 190
        lcall   bcris1                  ; |
        ret                             ;/
;
etic:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#195                 ; | address 195
        lcall   bcris1                  ; |
        ret                             ;/
;
eer:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#200                 ; | address 200
        lcall   bcris1                  ; |
        ret                             ;/
;
epan:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#205                 ; | address 205
        lcall   bcris1                  ; |
        ret                             ;/
;
efil:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#210                 ; | address 210
        lcall   bcris1                  ; |
        ret                             ;/
;
equen:  clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#215                 ; | address 215
        lcall   bcris1                  ; |
        ret                             ;/
;
egra:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#220                 ; | address 220
        lcall   bcris1                  ; |
        ret                             ;/
;
epe:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#225                 ; | address 225
        lcall   bcris1                  ; |
        ret                             ;/
;
eput:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#230                 ; | address 230
        lcall   bcris1                  ; |
        ret                             ;/
;
enet:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#235                 ; | address 235
        lcall   bcris1                  ; |
        ret                             ;/
;
etec:   clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#240                 ; | address 240
        lcall   bcris1                  ; |
        ret                             ;/
;
edis:   clr     ISLO                    ;\
        clr     ISHI                    ; |
```

```
        mov     P0,#245                 ; | address 245
        lcall   bcris1                  ; |
        ret                             ;/
;
;ksg:    clr     ISLO                   ;\
;        clr     ISHI                   ; |
;        mov     P0,#250                ; | address 250
;        lcall   bcris1                 ; |
;        ret                            ;/
;
eef:    clr     ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#255                 ; | address 255
        lcall   bcris1                  ; |
        ret                             ;/
;
eec:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#004                 ; | address 260
        lcall   bcris1                  ; |
        ret                             ;/
;
emag:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#009                 ; | address 265
        lcall   bcris1                  ; |
        ret                             ;/
;
eem:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#014                 ; | address 270
        lcall   bcris1                  ; |
        ret                             ;/
;
eror:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#019                 ; | address 275
        lcall   bcris1                  ; |
        ret                             ;/
;
ese:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#024                 ; | address 280
        lcall   bcris1                  ; |
        ret                             ;/
;
ement:  setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#029                 ; | address 285
        lcall   bcris1                  ; |
        ret                             ;/
;
ecy:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#034                 ; | address 290
        lcall   bcris1                  ; |
        ret                             ;/
;
evi:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#039                 ; | address 295
        lcall   bcris1                  ; |
        ret                             ;/
;
edan:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#044                 ; | address 300
        lcall   bcris1                  ; |
        ret                             ;/
;
eli:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#049                 ; | address 305
        lcall   bcris1                  ; |
        ret                             ;/
;
eano:   setb    ISLO                    ;\
```

```
        clr     ISHI                    ; |
        mov     P0,#054                 ; | address 310
        lcall   bcris1                  ; |
        ret                             ;/
;
ecur:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#059                 ; | address 315
        lcall   bcris1                  ; |
        ret                             ;/
;
edy:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#064                 ; | address 320
        lcall   bcris1                  ; |
        ret                             ;/
;
efec:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#069                 ; | address 325
        lcall   bcris1                  ; |
        ret                             ;/
;
eel:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#074                 ; | address 330
        lcall   bcris1                  ; |
        ret                             ;/
;
ene:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#079                 ; | address 335
        lcall   bcris1                  ; |
        ret                             ;/
;
eit:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#084                 ; | address 340
        lcall   bcris1                  ; |
        ret                             ;/
;
eex:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#089                 ; | address 345
        lcall   bcris1                  ; |
        ret                             ;/
;
efila:  setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#094                 ; | address 350
        lcall   bcris1                  ; |
        ret                             ;/
;
efre:   setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#099                 ; | address 355
        lcall   bcris1                  ; |
        ret                             ;/
;
efuel:  setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#104                 ; | address 360
        lcall   bcris1                  ; |
        ret                             ;/
;
eim:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#109                 ; | address 365
        lcall   bcris1                  ; |
        ret                             ;/
;
ein:    setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#114                 ; | address 370
        lcall   bcris1                  ; |
        ret                             ;/
;
```

```
equid:  setb    ISLO                    ;\
        clr     ISHI                    ; |
        mov     P0,#119                 ; | address 375
        lcall   bcris1                  ; |
        ret                             ;/
;
bcris0: clr     CIS0
        clr     ISTR                    ;\
        lcall   dlybcr                  ; | select isd 0
        setb    ISTR                    ; | play, delay, stop isd
        setb    CIS0                    ; | deselect isd 0
        lcall   jeda                    ; |
        ret                             ;/
;
bcris1: clr     CIS1
        clr     ISTR                    ;\
        lcall   dlybcr                  ; | select isd 1
        setb    ISTR                    ; | play, delay, stop isd
        setb    CIS1                    ; | deselect isd 1
        lcall   jeda                    ; |
        ret                             ;/
;
line1:  mov     P0,#80h
        lcall   w_ins
        ret
;
line2:  mov     P0,#0C0h
        lcall   w_ins
        ret
;
tulis:  clr     A
        movc    A,@A+DPTR
        mov     P0,A
        lcall   w_chr
        inc     DPTR
        djnz    Hurf,tulis
        ret
;
wr_chr: movc    A,@A+DPTR
        mov     P0,A
        lcall   w_chr
        ret
;
w_ins:  clr     Enbl
        clr     Rest
        setb    Enbl
        clr     Enbl
        lcall   jdchr
        ret
;
w_chr:  clr     Enbl
        setb    Rest
        setb    Enbl
        clr     Enbl
        lcall   jdchr
        ret
;
lcdclr: mov     P0,#01h                 ; Display Clear
        lcall   w_ins
        lcall   jdchr
        lcall   jdchr
        ret
;
lcd_in: lcall   delayp
        mov     P0,#01h                 ; Display Clear
        lcall   w_ins
        mov     P0,#38h                 ; Function Set
        lcall   w_ins
        mov     P0,#0Dh                 ; Display On, Cursor, Blink
        lcall   w_ins
        mov     P0,#06h                 ; Entry Mode
        lcall   w_ins
        mov     P0,#02h                 ; Cursor Home
        lcall   w_ins
        ret
;
wt_wr:  djnz    Dly0,wt_wr
```

```
        djnz    Dly1,wt_wr
        ret
;
scnkbd: clr     A
        mov     R0,#255
        jb      Kclk,scnkb5
        mov     Ckb0,#3
scnkb0: mov     Ckb1,#8
        jb      Kclk,$
        jnb     Kclk,$
        clr     A
scnkb1: jb      Kclk,$
        mov     C,Kdta
        RRC     A
        jnb     Kclk,$
        djnz    Ckb1,scnkb1
        jb      Kclk,$
        jnb     Kclk,$
        jb      Kclk,$
        jnb     Kclk,$
        djnz    Ckb0,scnkb0
scnkb2: cjne    A,#224,scnkb3
        mov     Ckb0,#1
        ljmp    scnkb0
scnkb3: cjne    A,#240,scnkb4
        mov     Ckb0,#1
        ljmp    scnkb0
scnkb4: mov     R0,A
        mov     DPTR,#kbdmap
        movc    A,@A+DPTR
        mov     P0,A
scnkb5: ret
;
jdchr:  djnz    Dly0,jdchr
        ret
;
jeda:   lcall   scnkbd
        cjne    R0,#05,jda0             ; tekan F1
        ljmp    menu00
jda0:   cjne    R0,#06,jda1             ; tekan F2
        ljmp    menu10
jda1:   cjne    R0,#04,jda2             ; tekan F3
        ljmp    menu20
jda2:   djnz    Dly0,jeda
        ret
;
delays: lcall   jeda
        djnz    Dly1,delays
        ret
;
delayl: mov     Dly2,#3
dlyl:   lcall   delays
        djnz    Dly2,dlyl
        ret
;
delayp: mov     Dly2,#5
dlyp:   djnz    Dly0,$
        djnz    Dly1,dlyp
        djnz    Dly2,dlyp
        ret
;
dlybcr: mov     Dly2,#6
dlybc:  djnz    Dly0,dlybc
        djnz    Dly1,dlybc
        djnz    Dly2,dlybc
        ret
;
nama:   DB      ' Andi  Prasetyo '
        DB      ' NIM: 021.70.55 '
judul:  DB      ' Digital  Kamus '
        DB      '     Teknik     '
elckms: DB      ' Electric Kamus '
mskdta: DB      ' Masukkan Data  '
bcadta: DB      '   Baca Data    '
scanig: DB      '  Scaning ...   '
saving: DB      '  Saving ....   '
deletg: DB      '  Deleting ..   '
```

```
plwait: DB      '  Please Wait   '
nodata: DB      ' Data Not Found '
angka:  DB      '0123456789      '
chi000: DB      'mut  '
chi001: DB      'lak  '
chi002: DB      'pem  '
chi003: DB      'bang '
chi004: DB      'kit  '
chi005: DB      'lis  '
chi006: DB      'trik '
chi007: DB      'meng '
chi008: DB      'ak   '
chi009: DB      'tif  '
chi010: DB      'kan  '
chi011: DB      'sa   '
chi012: DB      'tuan '
chi013: DB      'arus '
chi014: DB      'peng '
chi015: DB      'ukur '
chi016: DB      'an   '
chi017: DB      'tena '
chi018: DB      'ku   '
chi019: DB      'tub  '
chi020: DB      'po   '
chi021: DB      'si   '
chi022: DB      'pe   '
chi023: DB      'nang '
chi024: DB      'kal  '
chi025: DB      'tir  '
chi026: DB      'ba   '
chi027: DB      'te   '
chi028: DB      'rai  '
chi029: DB      'ber  '
chi030: DB      'gan  '
chi031: DB      'da   '
chi032: DB      'nyang'
chi033: DB      'ga   '
chi034: DB      'ka   '
chi035: DB      'pa   '
chi036: DB      'tan  '
chi037: DB      'tor  '
chi038: DB      'vo   '
chi039: DB      'lu   '
chi040: DB      'me   '
chi041: DB      'ne   '
chi042: DB      'kan  '
chi043: DB      'duk  '
chi044: DB      'alat '
chi045: DB      'de   '
chi046: DB      'tek  '
chi047: DB      'dio  '
chi048: DB      'ja   '
chi049: DB      'rak  '
chi050: DB      'di   '
chi051: DB      'na   '
chi052: DB      'mo   '
chi053: DB      'gu   '
chi054: DB      'ke   '
chi055: DB      'mag  '
chi056: DB      'nyiar'
chi057: DB      'la   '
chi058: DB      'han  '
chi059: DB      'eks  '
chi060: DB      'pan  '
chi061: DB      'wat  '
chi062: DB      'pi   '
chi063: DB      'jar  '
chi064: DB      'ri   '
chi065: DB      'ngan '
chi066: DB      'jum  '
chi067: DB      'lah  '
chi068: DB      'ge   '
chi069: DB      'ta   '
chi070: DB      'ran  '
chi071: DB      'kar  '
chi072: DB      'ya   '
chi073: DB      'be   '
```

```
chi074: DB      'rat  '
chi075: DB      'im   '
chi076: DB      'pe   '
chi077: DB      'dan  '
chi078: DB      'in   '
chi079: DB      'ma   '
chi080: DB      'su   '
chi081: DB      'zat  '
chi082: DB      'cair '
chi083: DB      'net  '
;
che000: DB      'ab   '
che001: DB      'ac   '
che002: DB      'am   '
che003: DB      'an   '
che004: DB      'ar   '
che005: DB      'so   '
che006: DB      'cu   '
che007: DB      'ti   '
che008: DB      'pere '
che009: DB      'ten  '
che010: DB      'res  '
che011: DB      'lute '
che012: DB      'vate '
che013: DB      'me   '
che014: DB      'ter  '
che015: DB      'na   '
che016: DB      'de   '
che017: DB      'tor  '
che018: DB      'bat  '
che019: DB      'tery '
che020: DB      'fer  '
che021: DB      'bi   '
che022: DB      'po   '
che023: DB      'lar  '
che024: DB      'buf  '
che025: DB      'ca   '
che026: DB      'tan  '
che027: DB      'tho  '
che028: DB      'pa   '
che029: DB      'ce   '
che030: DB      'ci   '
che031: DB      'ty   '
che032: DB      'con  '
che033: DB      'duc  '
che034: DB      'rent '
che035: DB      'dio  '
che036: DB      'mo   '
che037: DB      'tive '
che038: DB      'tro  '
che039: DB      'tic  '
che040: DB      'er   '
che041: DB      'pan  '
che042: DB      'fil  '
che043: DB      'quen '
che044: DB      'gra  '
che045: DB      'pe   '
che046: DB      'put  '
che047: DB      'net  '
che048: DB      'tec  '
che049: DB      'dis  '
che050: DB      '     '
che051: DB      'ef   '
che052: DB      'ec   '
che053: DB      'mag  '
che054: DB      'em   '
che055: DB      'ror  '
che056: DB      'se   '
che057: DB      'ment '
che058: DB      'cy   '
che059: DB      'vi   '
che060: DB      'dan  '
che061: DB      'li   '
che062: DB      'ano  '
che063: DB      'cur  '
che064: DB      'dy   '
che065: DB      'fec  '
```

```
che066: DB      'el    '
che067: DB      'ne    '
che068: DB      'it    '
che069: DB      'ex    '
che070: DB      'fila  '
che071: DB      'fre   '
che072: DB      'fuel  '
che073: DB      'im    '
che074: DB      'in    '
che075: DB      'quid  '
;
kbdmap: DB      '          '            ; 0-9
        DB      '          '            ; 10-19
        DB      ' q1   zsaw'            ; 20-29
        DB      '2  cxde43 '            ; 30-39
        DB      '  vftr5  n'            ; 40-49
        DB      'bhgy6   mj'            ; 50-59
        DB      'u78  ,kio0'            ; 60-69
        DB      '9  ./l p- '            ; 70-79
        DB      '    [=    '            ; 80-89
        DB      ' ]        '            ; 90-99
        DB      '          '            ; 100-109
        DB      '          '            ; 110-119
        DB      '          '            ; 120-129
        DB      '          '            ; 130-139
        DB      '          '            ; 140-149
        DB      '          '            ; 150-159
        DB      '          '            ; 160-169
        DB      '          '            ; 170-179
        DB      '          '            ; 180-189
        DB      '          '            ; 190-199
        DB      '          '            ; 200-209
        DB      '          '            ; 210-219
        DB      '          '            ; 220-229
        DB      '          '            ; 230-239
        DB      '          '            ; 240-249
        DB      '          '            ; 250-255
;
        end
```

AT 89S8252

atures

- mpatible with MCS-51™ Products
- Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- to 6V Operating Range
- lly Static Operation: 0 Hz to 24 MHz
- ree-level Program Memory Lock
- 5 x 8-bit Internal RAM
- Programmable I/O Lines
- ree 16-bit Timer/Counters
- ne Interrupt Sources
- ogrammable UART Serial Channel
- I Serial Interface
- w-power Idle and Power-down Modes
- errupt Recovery From Power-down
- ogrammable Watchdog Timer
- al Data Pointer
- wer-off Flag

scription

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

AT89S8252 provides the following standard features: 8K bytes of downloadable h, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watch- timer, two data pointers, three 16-bit timer/counters, a six-vector two-level rupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In tion, the AT89S8252 is designed with static logic for operation down to zero fre- ncy and supports two software selectable power saving modes. The Idle Mode s the CPU while allowing the RAM, timer/counters, serial port, and interrupt sys- to continue functioning. The Power-down mode saves the RAM contents but zes the oscillator, disabling all other chip functions until the next interrupt or hard- reset.

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

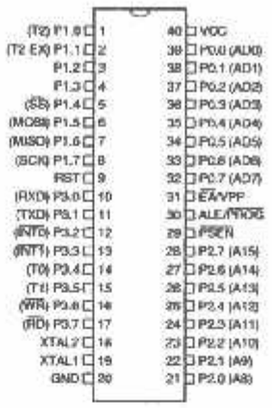
8-bit Microcontroller with 8K Bytes Flash

AT89S8252

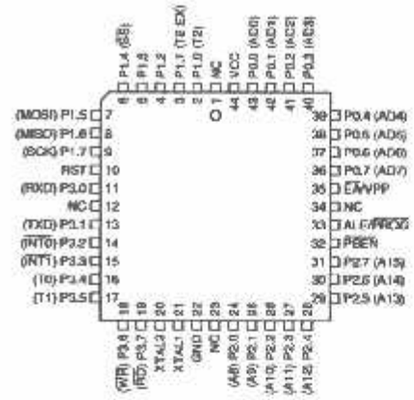
Rev. 0401E–02/00

Configurations

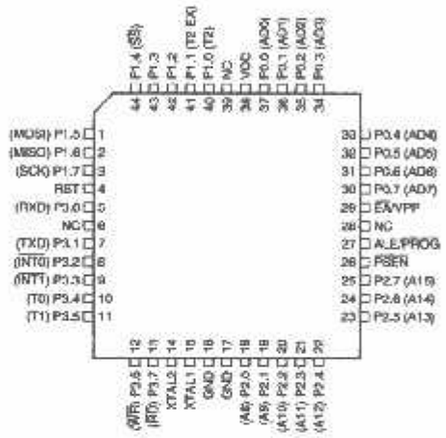
PDIP



PLCC



PQFP/TQFP



Description

ply voltage.

nd.

0

0 is an 8-bit open drain bbi-didirectional I/O port. As an ut port, each pin can sink eight TTL inputs. When 1s written to port 0 pins, the pins can be used as high-edance inputs.

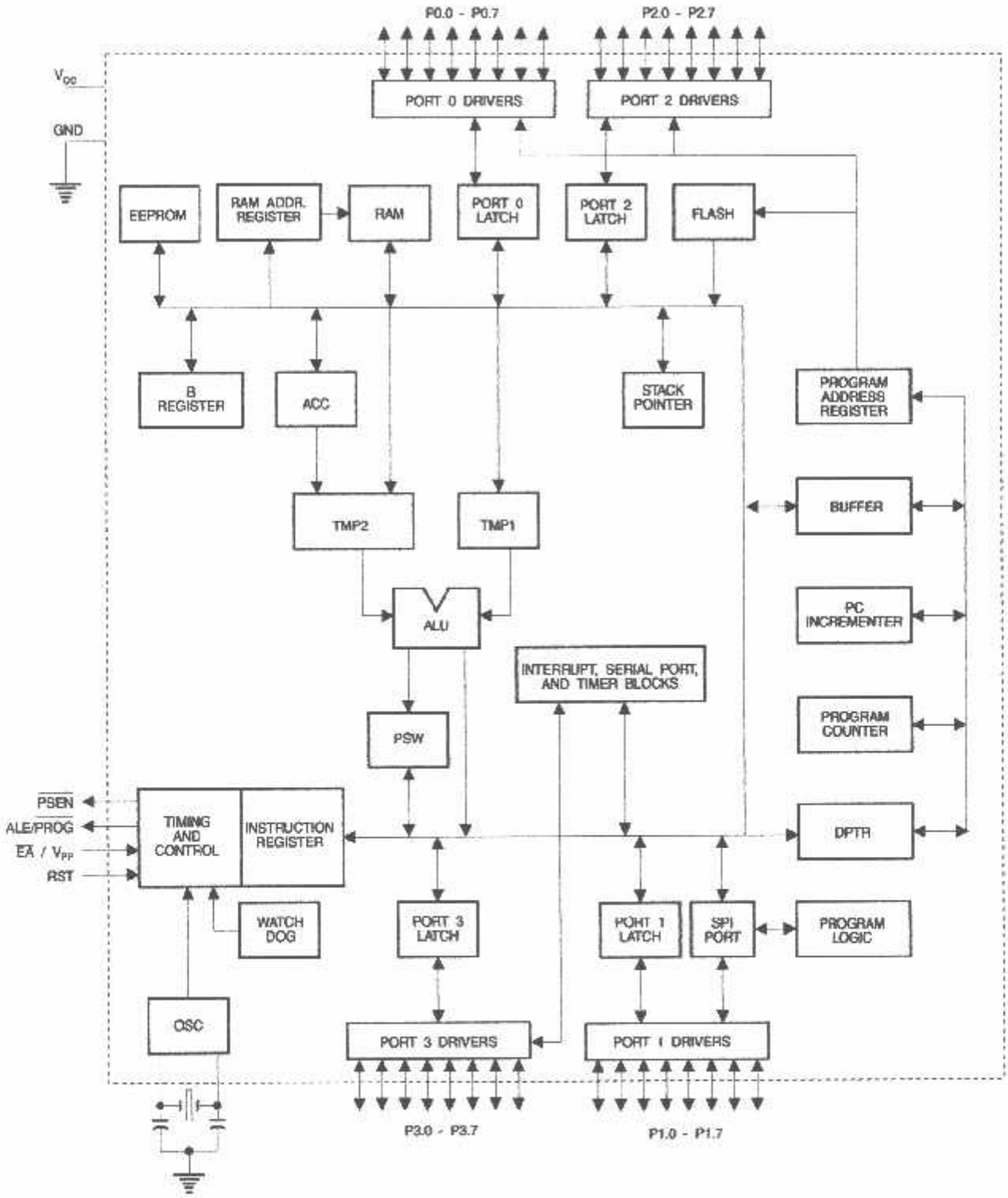
0 can also be configured to be the multiplexed low-ar address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

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 bi-directional 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.

ɔck Diagram


P0.0 - P0.7
P2.0 - P2.7
V_{CC}
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
EEPROM
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
BUFFER
TMP2
TMP1
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
PSEN
ALE/PROG
EA / V_{PP}
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
WATCH DOG
PORT 3 LATCH
PORT 1 LATCH
SPI PORT
PROGRAM LOGIC
OSC
PORT 3 DRIVERS
PORT 1 DRIVERS
P3.0 - P3.7
P1.0 - P1.7

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

ı Description

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

ı Pin	Alternate Functions
0	T2 (external count input to Timer/Counter 2), clock-out
1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
4	$\overline{SS}$ (Slave port select input)
5	MOSI (Master data output, slave data input pin for SPI channel)
6	MISO (Master data input, slave data output pin for SPI channel)
7	SCK (Master clock output, slave clock input pin for SPI channel)

: 1 also receives the low-order address bytes during
h programming and verification.

t 2

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

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

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

t 3

t 3 is an 8 bit bi-directional I/O port with internal pullups.
Port 3 output buffers can sink/source four TTL inputs.
ın 1s are written to Port 3 pins, they are pulled high by
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.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external pro-

n memory locations starting at 0000H up to FFFFH. :, however, that if lock bit 1 is programmed, $\overline{EA}$ will be nally latched on reset.

should be strapped to V_{CC} for internal program execu- s. This pin also receives the 12-volt programming ole voltage (V_{PP}) during Flash programming when 12- programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

e 1. AT89S8252 SFR Map and Reset Values

8H									0FFH
0H	B 00000000								0F7H
8H									0EFH
0H	ACC 00000000								0E7H
8H									0DFH
0H	PSW 00000000					SPCR 000001XX			0D7H
8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0H									0C7H
8H	IP XX000000								0BFH
0H	P3 11111111								0B7H
8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0H	P2 11111111								0A7H
8H	SCON 00000000	SBUF XXXXXXXX							9FH
0H	P1 11111111						WMCON 00000010		97H
8H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
0H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H

ecial Function Registers

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

that not all of the addresses are occupied, and unoc- ed addresses may not be implemented on the chip. d accesses to these addresses will in general return om data, and write accesses will have an indeterminate t.

r 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 new 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.

e 2. T2CON—Timer/Counter 2 Control Register

CON Address = 0C8H — Reset Value = 0000 0000B

Addressable

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
7	6	5	4	3	2	1	0

nbol	Function
	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.
F2	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).
LK	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.
LK	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.
EN2	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.
2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
$\overline{2}$	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).
/$\overline{RL2}$	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.

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

e 3. WMCON—Watchdog and Memory Control Register

ICON Address = 96H — Reset Value = 0000 0010B

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

nbol	Function
2 1)	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.
MWE	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.
MEN	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.
S	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1
TRST Y/$\overline{BSY}$	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.
TEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

Registers Control and status bits for the Serial Periph- Interface are contained in registers SPCR (shown in le 4) and SPSR (shown in Table 5). The SPI data bits contained in the SPDR register. Writing the SPI data ster during serial data transfer sets the Write Collision WCOL, in the SPSR register. The SPDR is double buff- l for writing and the values in SPDR are not changed by et.

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

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 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.

e 4. SPCR—SPI Control Register

CR Address = D5H — Reset Value = 0000 01XXB

t	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	7	6	5	4	3	2	1	0

mbol	Function
IE	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.
E	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.
RD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
TR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
OL	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.
HA	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.
R0 R1	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

e 5. SPSR – SPI Status Register

SR Address = AAH — Reset Value = 00XX XXXXB

it	SPIF	WCOL	–	–	–	–	–	–
	7	6	5	4	3	2	1	0

mbol	Function
IF	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.
OL	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.

le 6. SPDR – SPI Data Register

DR Address = 86H — Reset Value = unchanged

it	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	7	6	5	4	3	2	1	0

ta Memory – EEPROM and RAM

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

ɔn an instruction accesses an internal location above ress 7FH, the address mode used in the instruction cifies whether the CPU accesses the upper 128 bytes AM or the SFR space. Instructions that use direct essing access SFR space.

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

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

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

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

e that stack operations are examples of indirect essing, so the upper 128 bytes of data RAM are avail- as stack space.

on-chip EEPROM data memory is selected by setting EEMEN bit in the WMCON register at SFR address tion 96H. The EEPROM address range is from 000H to H. The MOVX instructions are used to access the ²ROM. To access off-chip data memory with the MOVX uctions, the EEMEN bit needs to be set to "0".

EEMWE bit in the WMCON register needs to be set to ɔefore any byte location in the EEPROM can be written. r software should reset EEMWE bit to "0" if no further ²ROM write is required. EEPROM write cycles in the al programming mode are self-timed and typically take ms. The progress of EEPROM write can be monitored eading the RDY/$\overline{BSY}$ bit (read-only) in SFR WMCON. Y/$\overline{BSY}$ = 0 means programming is still in progress and Y/$\overline{BSY}$ = 1 means EEPROM write cycle is completed another write cycle can be initiated.

ddition, during EEPROM programming, an attempted d from the EEPROM will fetch the byte being written the MSB complemented. Once the write cycle is com- ed, true data are valid at all bit locations.

ɔgrammable Watchdog Timer

programmable Watchdog Timer (WDT) operates from ndependent oscillator. The prescaler bits, PS0, PS1 PS2 in SFR WMCON are used to set the period of the tchdog Timer from 16 ms to 2048 ms. The available ɜr periods are shown in the following table and the actual timer periods (at V_{CC} = 5V) are within ±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 WDTRST 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			
PS2	PS1	PS0	Period (nominal)
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/$\overline{T2}$ 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

transition was detected. Since two machine cycles (24 llator periods) are required to recognize a 1-to-0 transi-, the maximum count rate is 1/24 of the oscillator uency. To ensure that a given level is sampled at least before it changes, the level should be held for at least full machine cycle.

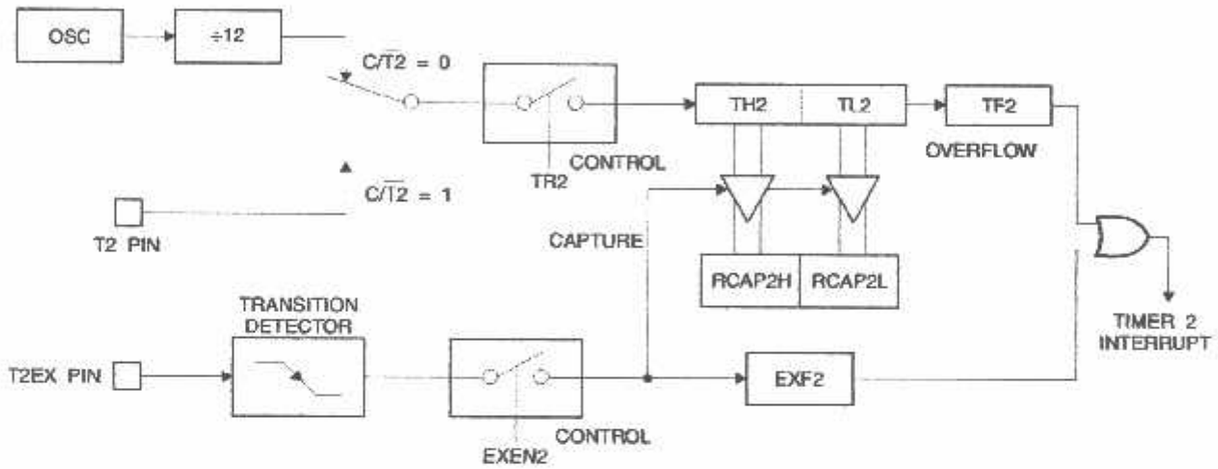
e 8. Timer 2 Operating Modes

CLK + 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.

ıre 1. Timer 2 in Capture Mode



:o-reload (Up or Down Counter)

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

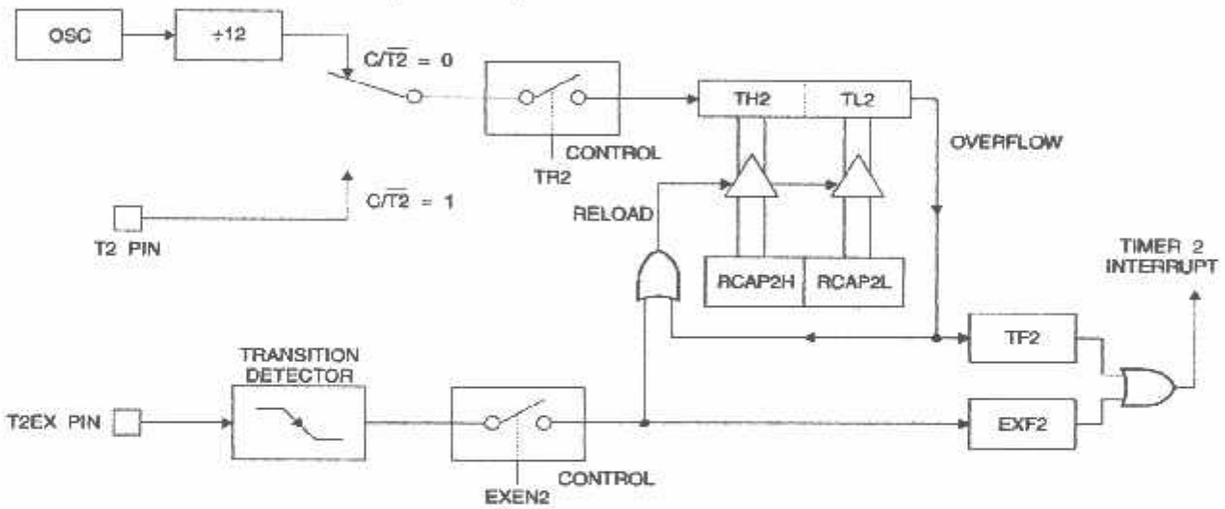
ire 2 shows Timer 2 automatically counting up when EN = 0. In this mode, two options are selected by bit EN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to FFH and then sets the TF2 bit upon overflow. The flow also causes the timer registers to be reloaded with 16 bit value in RCAP2H and RCAP2L. The values in AP2H and RCAP2L are preset by software. If EXEN2 = 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.

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



le 9. T2MOD – Timer 2 Mode Control Register

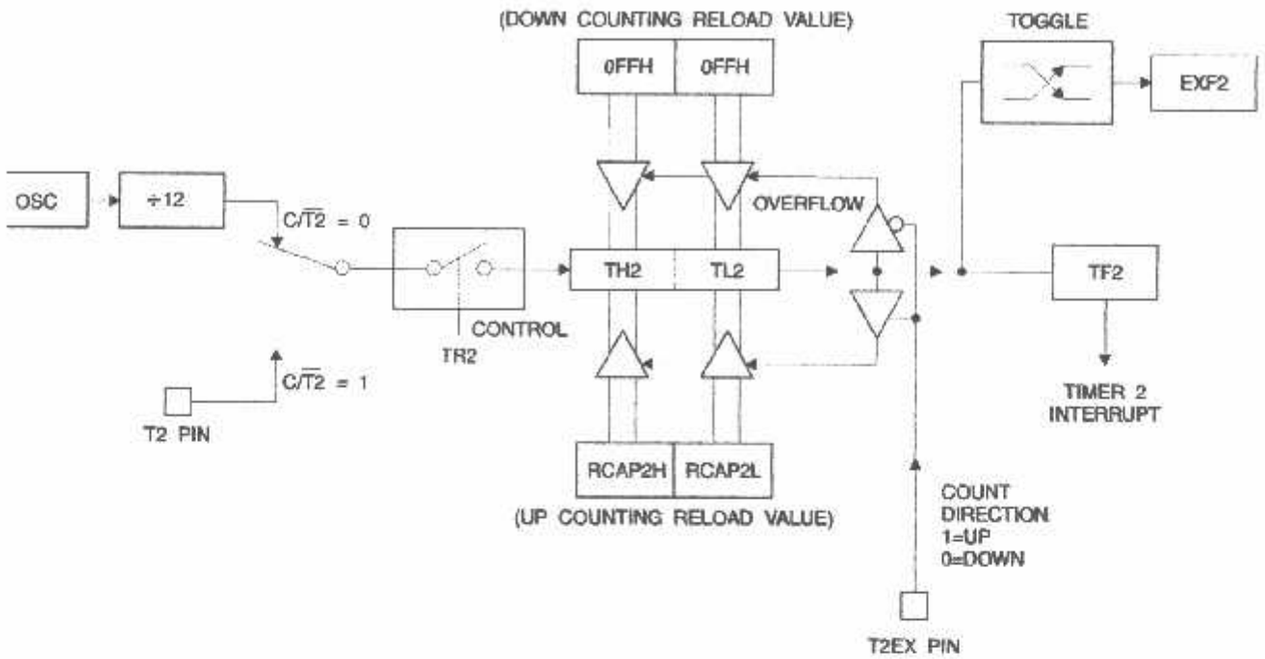
MOD Address = 0C9H — Reset Value = XXXX XX00B

t Bit Addressable

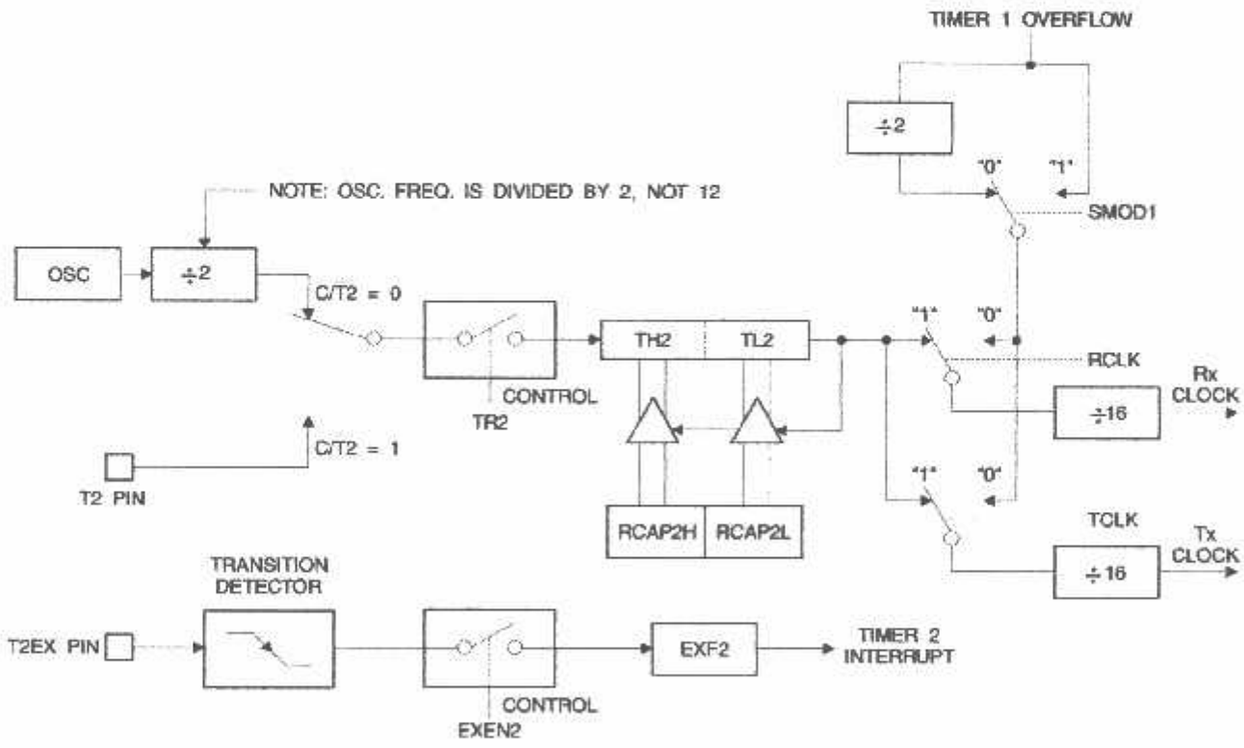
	–	–	–	–	–	–	T2OE	DCEN
lit	7	6	5	4	3	2	1	0

mbol	Function
	Not implemented, reserved for future use.
OE	Timer 2 Output Enable bit.
EN	When set, this bit allows Timer 2 to be configured as an up/down counter.

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



ıre 4. Timer 2 In Baud Rate Generator Mode



ud Rate Generator

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

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

baud rates in Modes 1 and 3 are determined by Timer verflow rate according to the following equation.

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

Timer can be configured for either timer or counter ration. In most applications, it is configured for timer ration ($C/\overline{T2} = 0$). The timer operation is different for er 2 when it is used as a baud rate generator. Normally, i timer, it increments every machine cycle (at 1/12 the llator frequency). As a baud rate generator, however, it ements every state time (at 1/2 the oscillator fre- ncy). The baud rate formula is given below.

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

re (RCAP2H, RCAP2L) is the content of RCAP2H and AP2L taken as a 16 bit unsigned integer.

er 2 as a baud rate generator is shown in Figure 4. This re is valid only if RCLK or TCLK = 1 in T2CON. Note a rollover in TH2 does not set TF2 and will not gener- an interrupt. Note too, that if EXEN2 is set, a 1-to-0 sition in T2EX will set EXF2 but will not cause a reload (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.

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, 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 to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

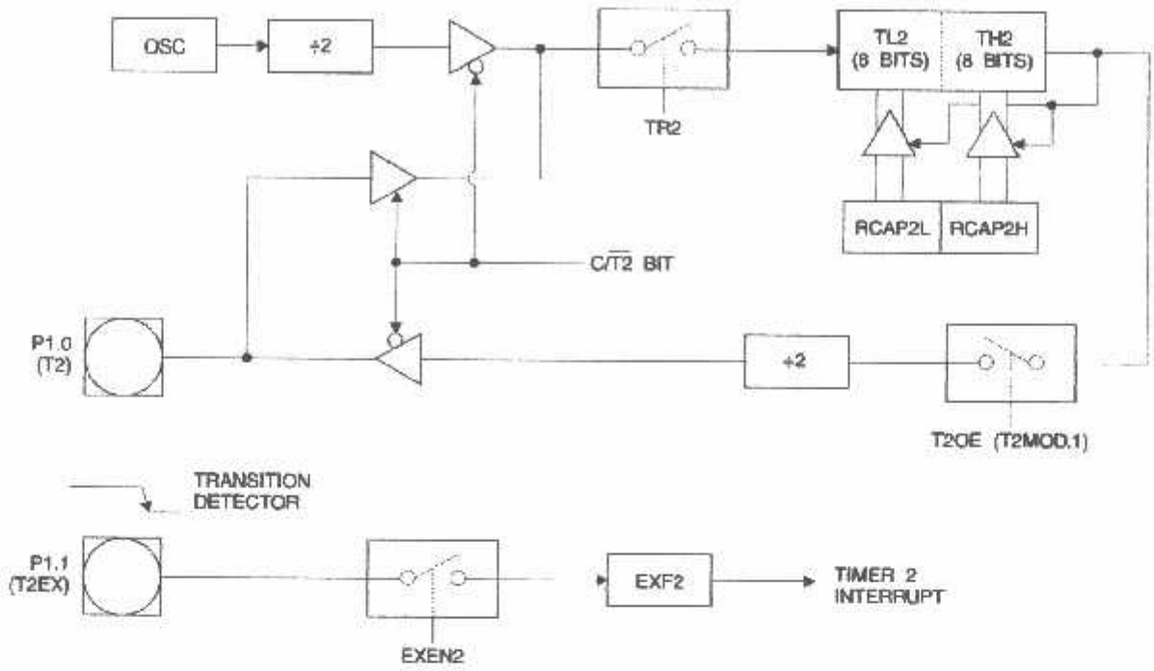
To configure the Timer/Counter 2 as a clock generator, bit $C/\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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.

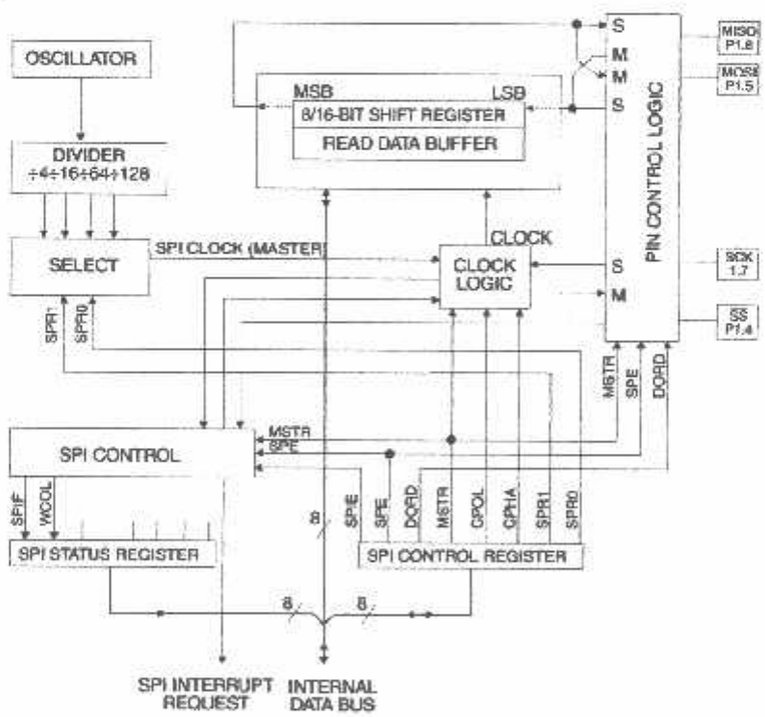
$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

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.

ıre 5. Timer 2 in Clock-out Mode



ıre 6. SPI Block Diagram



RT

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

rial Peripheral Interface

serial peripheral interface (SPI) allows high-speed syn-ɔnous data transfer between the AT89S8252 and pheral devices or between several AT89S8252 ices. The AT89S8252 SPI features include the wing:

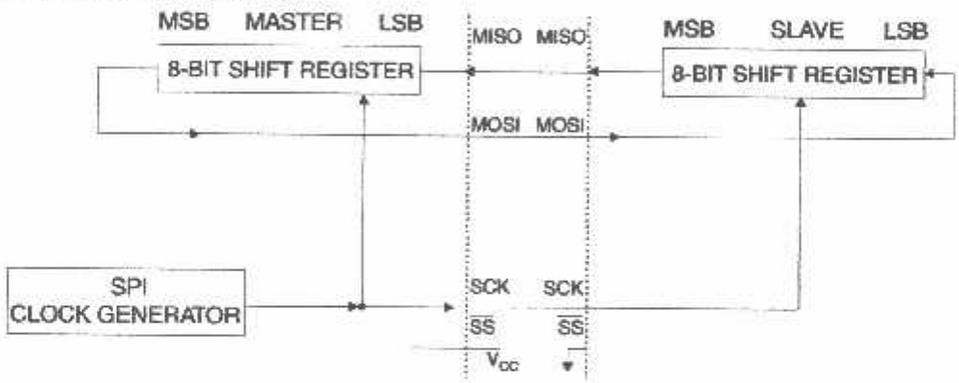
- ull-Duplex, 3-Wire Synchronous Data Transfer
- aster or Slave Operation
- 5 MHz Bit Frequency (max.)
- SB First or MSB First Data Transfer
- ɔur Programmable Bit Rates
- nd of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

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.

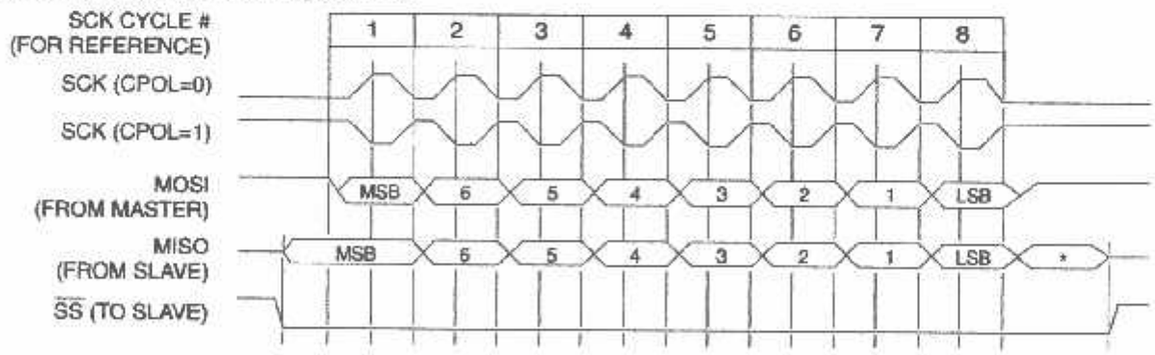
The Slave Select input, $\overline{SS}$/P1.4, is set low to select an individual SPI device as a slave. When $\overline{SS}$/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

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 Figure 8 and Figure 9.

ıre 7. SPI Master-slave Interconnection

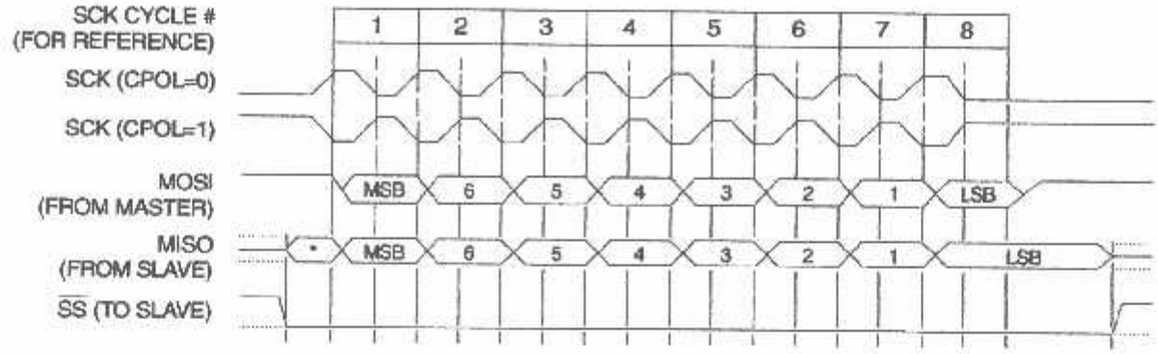


ıre 8. SPI transfer Format with CPHA = 0



t defined but normally MSB of character just received

ıre 9. SPI Transfer Format with CPHA = 1



: defined but normally LSB of previously transmitted character

errupts

AT89S8252 has a total of six interrupt vectors: two rnal interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts ıers 0, 1, and 2), and the serial port interrupt. These rupts are all shown in Figure 10.

h of these interrupt sources can be individually enabled isabled by setting or clearing a bit in Special Function ister IE. IE also contains a global disable bit, EA, which bles all interrupts at once.

ɜ that Table 10 shows that bit position IE.6 is unimple- ıted. In the AT89C51, bit position IE.5 is also ıplemented. User software should not write 1s to these ɔositions, since they may be used in future AT89 ducts.

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

Timer 0 and Timer 1 flags, TF0 and TF1, are set at ²2 of the cycle in which the timers overflow. The values then polled by the circuitry in the next cycle. However, Timer 2 flag, TF2, is set at S2P2 and is polled in the ıe cycle in which the timer overflows.

Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)

EA	–	ET2	ES	ET1	EX1	ET0	EX0

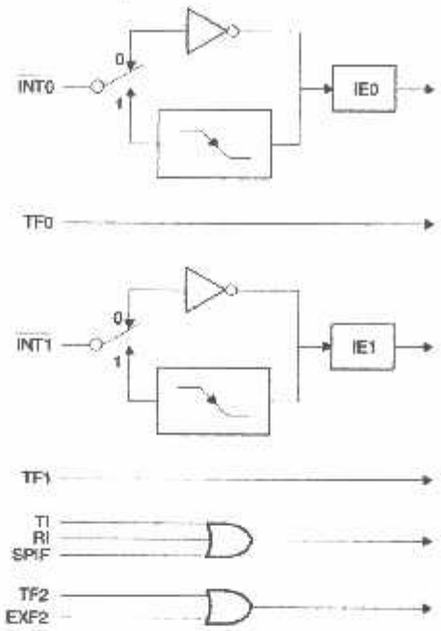
Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

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.

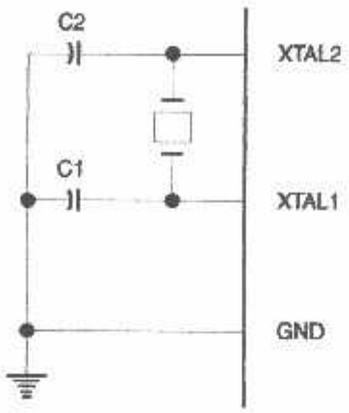
ire 10. Interrupt Sources



cillator Characteristics

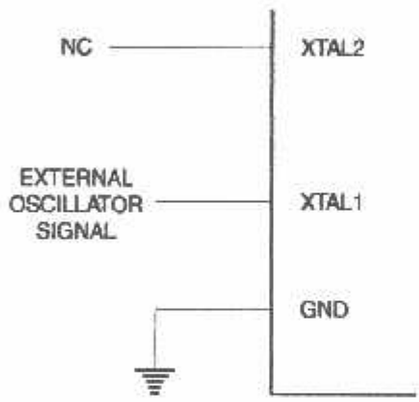
.L1 and XTAL2 are the input and output, respectively, n inverting amplifier that can be configured for use as on-chip oscillator, as shown in Figure 11. Either a quartz stal or ceramic resonator may be used. To drive the ce from an external clock source, XTAL2 should be left onnected while XTAL1 is driven, as shown in Figure 12. re are no requirements on the duty cycle of the external k signal, since the input to the internal clocking circuitry rough a divide-by-two flip-flop, but minimum and maxi- n voltage high and low time specifications must be erved.

Figure 11. Oscillator Connections



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

Figure 12. External Clock Drive Configuration



e Mode

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

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

tus of External Pins During Idle and Power-down Modes

de	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
	Internal	1	1	Data	Data	Data	Data
	External	1	1	Float	Data	Address	Data
ver-down	Internal	0	0	Data	Data	Data	Data
ver-down	External	0	0	Float	Data	Data	Data

wer-down Mode

ie power-down mode, the oscillator is stopped and the
uction that invokes power-down is the last instruction
cuted. The on-chip RAM and Special Function Regis-
retain their values until the power-down mode is
inated. Exit from power-down can be initiated either by
rdware reset or by an enabled external interrupt. Reset
efines the SFRs but does not change the on-chip RAM.
reset should not be activated before V_{CC} is restored to
normal operating level and must be held active long
ugh to allow the oscillator to restart and stabilize.

exit power-down via an interrupt, the external interrupt
st be enabled as level sensitive before entering power-
wn. The interrupt service routine starts at 16 ms (nomi-
after the enabled interrupt pin is activated.

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.

ck Bit Protection Modes[1][2]

	Program Lock Bits			
	LB1	LB2	LB3	Protection Type
	U	U	U	No internal memory lock feature.
2	P	U	U	MOVC 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.

es: 1. U = Unprogrammed
2. P = Programmed

ɔgramming the Flash and EEPROM

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

AT89S8252 is normally shipped with the on-chip Flash le and EEPROM Data memory arrays in the erased ə (I.e. contents = FFH) and ready to be programmed. ; device supports a High-voltage (12V) Parallel pro- mming mode and a Low-voltage (5V) Serial ɡramming mode. The serial programming mode pro- s a convenient way to download the AT89S8252 inside user's system. The parallel programming mode is com- ble with conventional third party Flash or EPROM ɡrammers.

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

Code and Data memory arrays on the AT89S8252 are ɡrammed byte-by-byte in either programming mode. An ›-erase cycle is provided with the self-timed program- g operation in the serial programming mode. There is ıeed to perform the Chip Erase operation to reprogram memory location in the serial programming mode ıss any of the lock bits have been programmed.

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

allel Programming Algorithm: To program and verify AT89S8252 in the parallel programming mode, the fol- ing sequence is recommended:

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.

Set $\overline{PSEN}$ pin to "L"

ALE pin to "H"

$\overline{EA}$ pin to "H" and all other pins to "H".

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.

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.
8. 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.
9. 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.

$\overline{Data}$ 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 P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{DATA}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{BSY}$ 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.

Program Verify: If 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.

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

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

ial Programming Fuse: A programmable fuse is avail- ə to disable Serial Programming if the user needs ximum system security. The Serial Programming Fuse only be programmed or erased in the Parallel Program- g Mode.

AT89S8252 is shipped with the Serial Programming le enabled.

ıding the Signature Bytes: The signature bytes are ı by the same procedure as a normal verification of ıtions 030H and 031H, except that P3.6 and P3.7 must ıulled to a logic low. The values returned are as follows:

(030H) = 1EH indicates manufactured by Atmel
(031H) = 72H indicates 89S8252

ɔgramming Interface

ry code byte in the Flash and EEPROM arrays can be ten, and the entire array can be erased, by using the ropriate combination of control signals. The write oper- ɔn cycle is self-timed and once initiated, will ɔmatically time itself to completion.

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

rial Downloading

h the Code and Data memory arrays can be pro- mmed using the serial SPI bus while RST is pulled to . The serial interface consists of pins SCK, MOSI (input) I MISO (output). After RST is set high, the Programming ıble instruction needs to be executed first before pro- m/erase operations can be executed.

auto-erase cycle is built into the self-timed programming ıration (in the serial mode ONLY) and there is no need rst execute the Chip Erase instruction unless any of the ‹ bits have been programmed. The Chip Erase opera- turns the content of every memory location in both the le and Data arrays into FFH.

› Code and Data memory arrays have separate address ces:

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:

1. Power-up sequence:
   Apply power between VCC 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.
2. 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.
3. 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.
4. 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.
5. 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:

ɜtruction Set

ıtruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
ɔgramming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
ip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
ad 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.
ite 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.
ad 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.
ite Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
ite Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

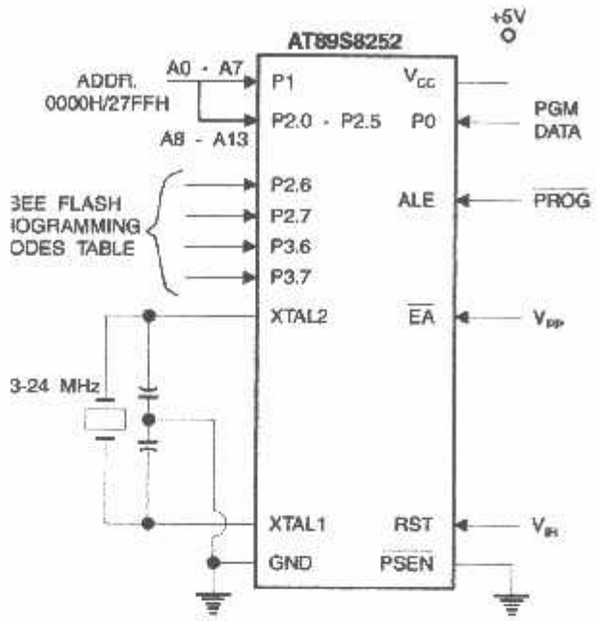
ɜ: 1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.

ash and EEPROM Parallel Programming Modes

de	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
rial Prog. Modes	H	h(1)	h(1)	x						
ip Erase	H	L	(2)	12V	H	L	L	L	X	X
ite (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
ad (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
ite 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
ad 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
ad Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
ad Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
rial Prog. Enable	H	L	(2)	12V	L	H	L	H	P0.0 = 0	X
rial Prog. Disable	H	L	(2)	12V	L	H	L	H	P0.0 = 1	X
ad Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

es:
1. "h" = weakly pulled "High" internally.
2. Chip Erase and Serial Programming Fuse require a 10 ms 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/BSY.
4. "X" = don't care

ure 13. Programming the Flash/EEPROM Memory



ure 14. Verifying the Flash/EEPROM Memory

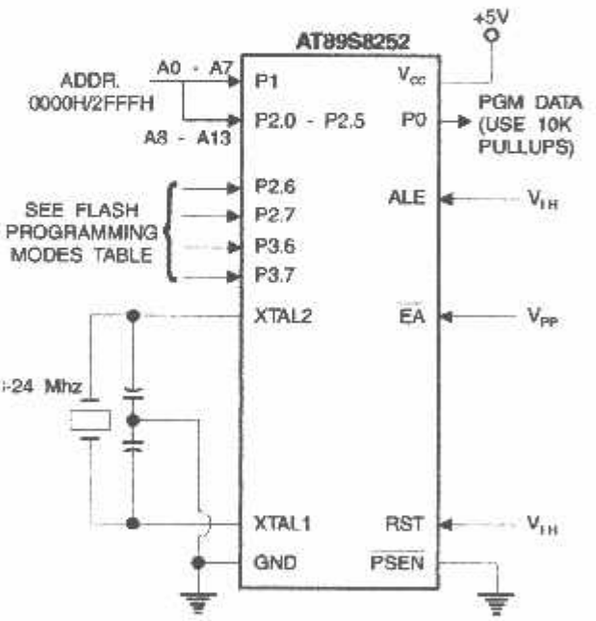
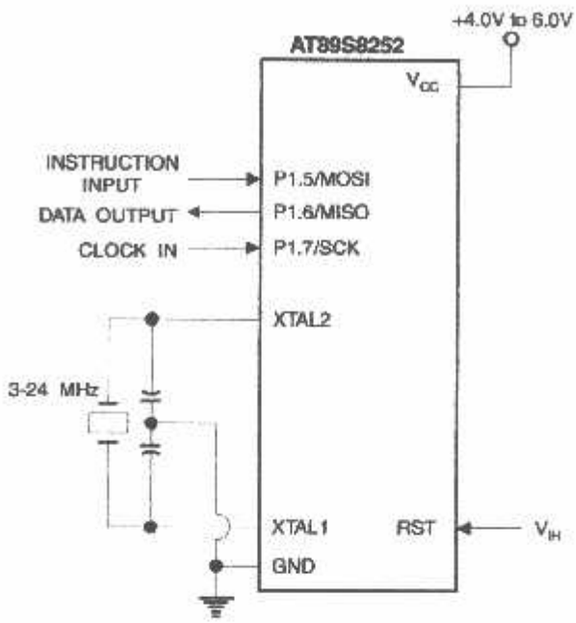


Figure 15. Flash/EEPROM Serial Downloading

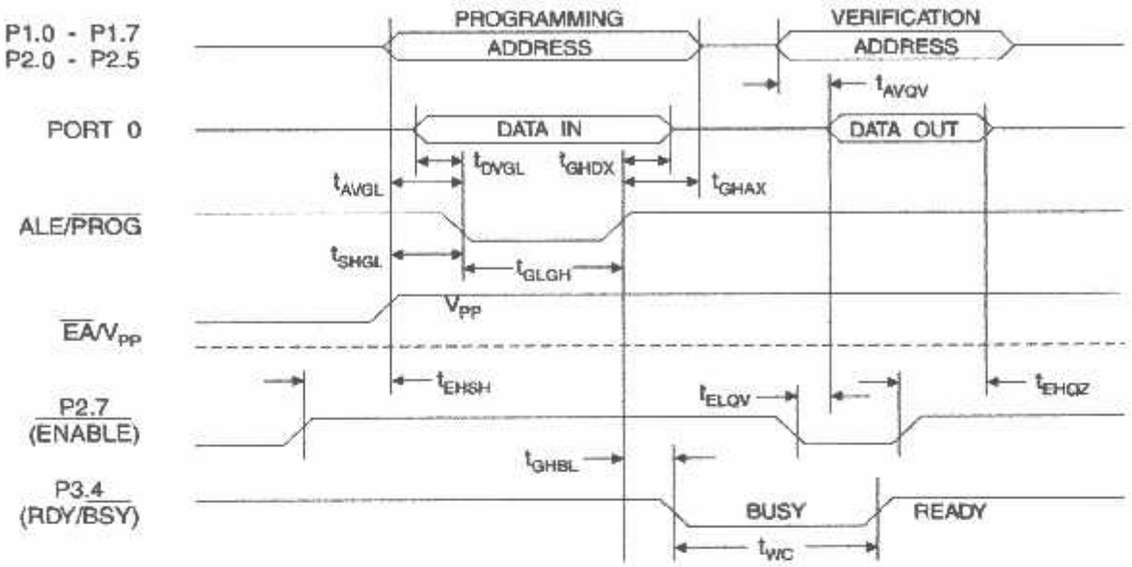


ısh Programming and Verification Characteristics – Parallel Mode

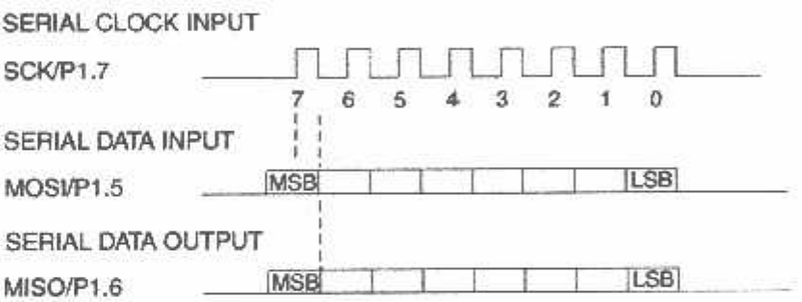
: 0°C to 70°C, V_{CC} = 5.0V ± 10%

mbol	Parameter	Min	Max	Units
₀	Programming Enable Voltage	11.5	12.5	V
	Programming Enable Current		1.0	mA
CLCL	Oscillator Frequency	3	24	MHz
GL	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
IAX	Address Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
GL	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
IDX	Data Hold after PROG	$48t_{CLCL}$		
SH	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
GL	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		µs
GH	$\overline{\text{PROG}}$ Width	1	110	µs
DV	Address to Data Valid		$48t_{CLCL}$	
DV	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
QZ	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
BL	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	µs
	Byte Write Cycle Time		2.0	ms

ısh/EEPROM Programming and Verification Waveforms – Parallel Mode



rial Downloading Waveforms



ısolute Maximum Ratings*

erating Temperature	-55°C to +125°C
rage Temperature	-65°C to +150°C
ltage on Any Pin h Respect to Ground	-1.0V to +7.0V
ximum Operating Voltage	6.6V
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.

Characteristics

values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0V ± 20%, unless otherwise noted.

mbol	Parameter	Condition	Min	Max	Units
	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
1	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
I	Input Hifh-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
IH	Input Hifh-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
L	Output Low-voltage [1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.5	V
L1	Output Low-voltage [1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.5	V
H	Output Hifh-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
IH1	Output Hifh-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RST	Reset Pull-down Resistor		50	300	KΩ
O	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode [2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

es: 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

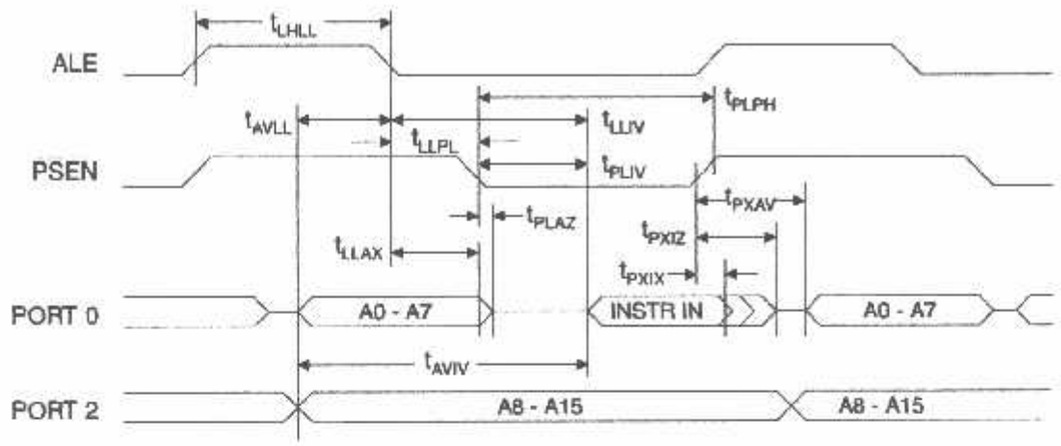
: Characteristics

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

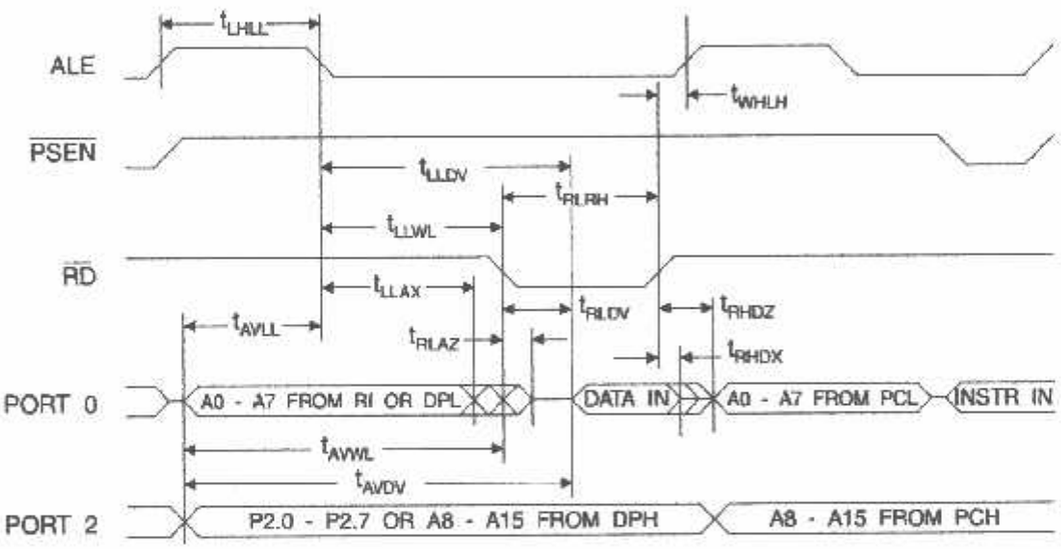
ternal Program and Data Memory Characteristics

mbol	Parameter	Variable Oscillator		Units
		Min	Max	
CLCL	Oscillator Frequency	0	24	MHz
LL	ALE Pulse Width	$2t_{CLCL}$ - 40		ns
LL	Address Valid to ALE Low	t_{CLCL} - 13		ns
AX	Address Hold after ALE Low	t_{CLCL} - 20		ns
IV	ALE Low to Valid Instruction In		$4t_{CLCL}$ - 65	ns
PL	ALE Low to $\overline{PSEN}$ Low	t_{CLCL} - 13		ns
PH	$\overline{PSEN}$ Pulse Width	$3t_{CLCL}$ - 20		ns
PV	$\overline{PSEN}$ Low to Valid Instruction In		$3t_{CLCL}$ - 45	ns
PIX	Input Instruction Hold after $\overline{PSEN}$	0		ns
PIZ	Input Instruction Float after $\overline{PSEN}$		t_{CLCL} - 10	ns
PAV	$\overline{PSEN}$ to Address Valid	t_{CLCL} - 8		ns
AIV	Address to Valid Instruction In		$5t_{CLCL}$ - 55	ns
PAZ	$\overline{PSEN}$ Low to Address Float		10	ns
RLRH	$\overline{RD}$ Pulse Width	$6t_{CLCL}$ - 100		ns
WLWH	$\overline{WR}$ Pulse Width	$6t_{CLCL}$ - 100		ns
RLDV	$\overline{RD}$ Low to Valid Data In		$5t_{CLCL}$ - 90	ns
RHDX	Data Hold after $\overline{RD}$	0		ns
RHDZ	Data Float after $\overline{RD}$		$2t_{CLCL}$ - 28	ns
LLDV	ALE Low to Valid Data In		$8t_{CLCL}$ - 150	ns
AVDV	Address to Valid Data In		$9t_{CLCL}$ - 165	ns
LLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	$3t_{CLCL}$ - 50	$3t_{CLCL}$ + 50	ns
AVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	$4t_{CLCL}$ - 75		ns
QVWX	Data Valid to $\overline{WR}$ Transition	t_{CLCL} - 20		ns
QVWH	Data Valid to $\overline{WR}$ High	$7t_{CLCL}$ - 120		ns
WHQX	Data Hold after $\overline{WR}$	t_{CLCL} - 20		ns
RLAZ	$\overline{RD}$ Low to Address Float		0	ns
WHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{CLCL} - 20	t_{CLCL} + 25	ns

ternal Program Memory Read Cycle



ternal Data Memory Read Cycle

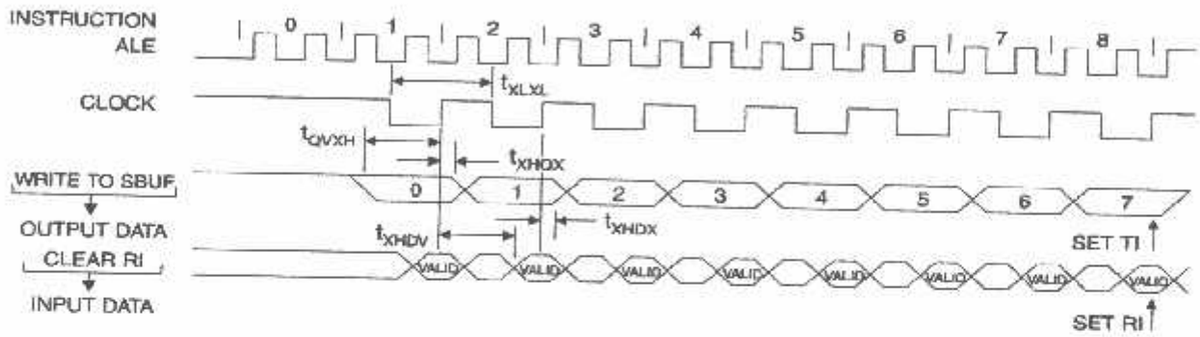


rial Port Timing: Shift Register Mode Test Conditions

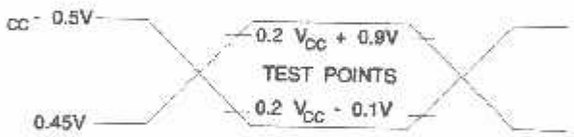
values in this table are valid for V_{CC} = 4.0V to 6V and Load Capacitance = 80 pF.

mbol	Parameter	Variable Oscillator		Units
		Min	Max	
L	Serial Port Clock Cycle Time	$12t_{CLCL}$		µs
XH	Output Data Setup to Clock Rising Edge	$10t_{CLCL}$ - 133		ns
QX	Output Data Hold after Clock Rising Edge	$2t_{CLCL}$ - 117		ns
DX	Input Data Hold after Clock Rising Edge	0		ns
DV	Clock Rising Edge to Input Data Valid		$10t_{CLCL}$ - 133	ns

ift Register Mode Timing Waveforms



: Testing Input/Output Waveforms[1]



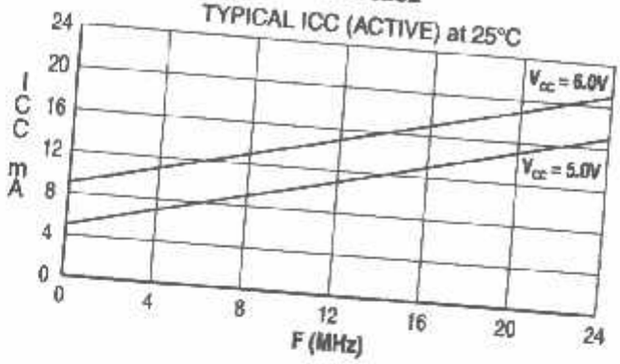
es: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

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.

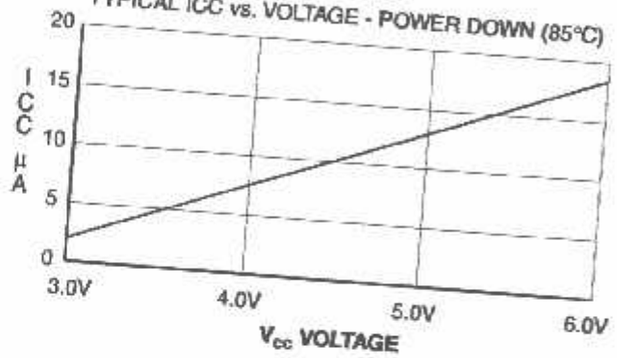
AT89S8252

TYPICAL ICC (ACTIVE) at 25°C



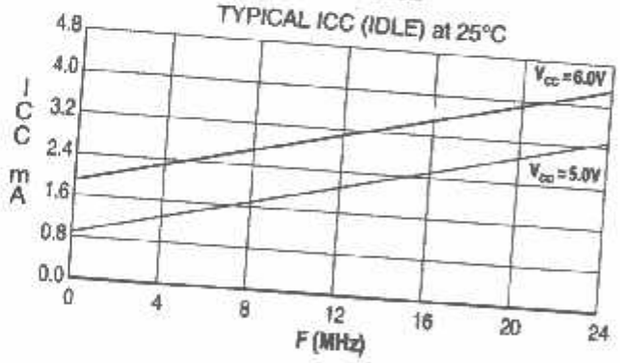
AT89S8252

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



AT89S8252

TYPICAL ICC (IDLE) at 25°C



Notes: 1. XTAL1 tied to GND for Icc (power-down)
2. Lock bits programmed

dering Information

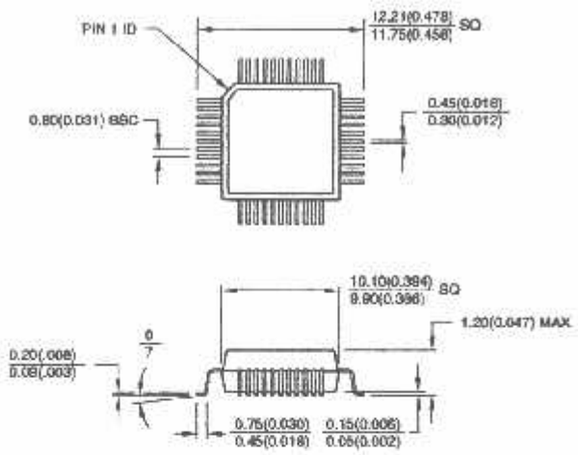
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 6.0V	AT89S8252-24AC AT89S8252-24JC AT89S8252-24PC AT89S8252-24QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
	4.0V to 6.0V	AT89S8252-24AI AT89S8252-24JI AT89S8252-24PI AT89S8252-24QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
33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

= Preliminary Information

	Package Type
4A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
4J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
0P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
4Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

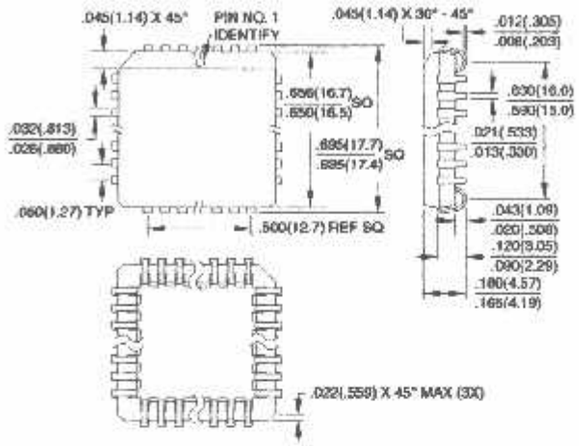
ckaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

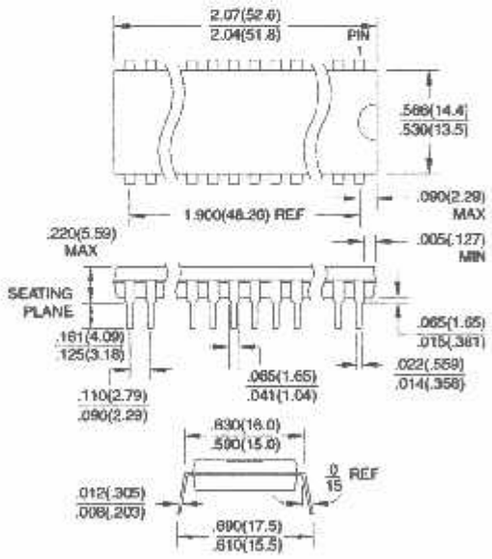


Controlling dimension: millimeters

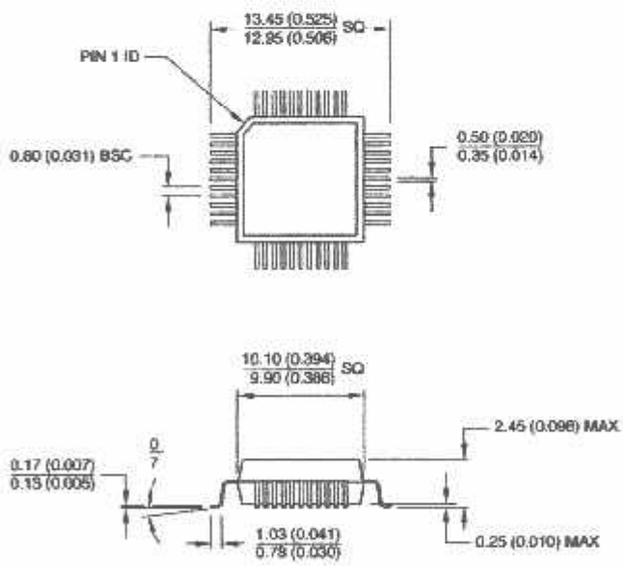
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

tmel Headquarters

)rporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

irope
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

;ia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

pan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

0401E–02/00/xM

ISD 25120

Winbond Electronics Corp.

ISD2560/75/90/120

SINGLE-CHIP, MULTIPLE-MESSAGES,

VOICE RECORD/PLAYBACK DEVICE

60-, 75-, 90-, AND 120-SECOND DURATION

Publication Release Date: May 2003
Revision 1.0

Winbond Electronics Corp.

1. GENERAL DESCRIPTION

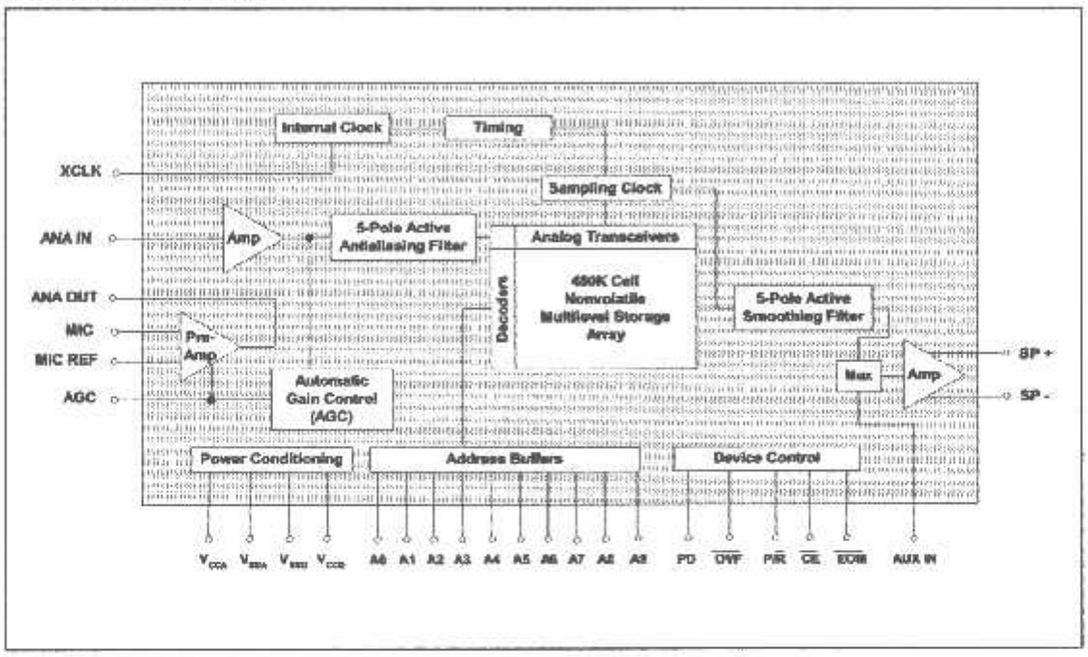
Winbond's ISD2500 ChipCorder® Series provide high-quality, single-chip, Record/Playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multi-level storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved. Recordings are stored into on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through Winbond'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.

2. FEATURES

- Easy-to-use single-chip, voice record/playback solution
- High-quality, natural voice/audio reproduction
- Single-chip with duration of 60, 75, 90, or 120 seconds.
- Manual switch or microcontroller compatible
- Playback can be edge- or level-activated
- Directly cascadable for longer durations
- Automatic power-down (push-button mode)
  - Standby current 1 µA (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, PDIP, SOIC and TSOP packaging
- Temperature = die (0°C to +50°C) and package (0°C to +70°C)

Winbond Electronics Corp.

3. BLOCK DIAGRAM

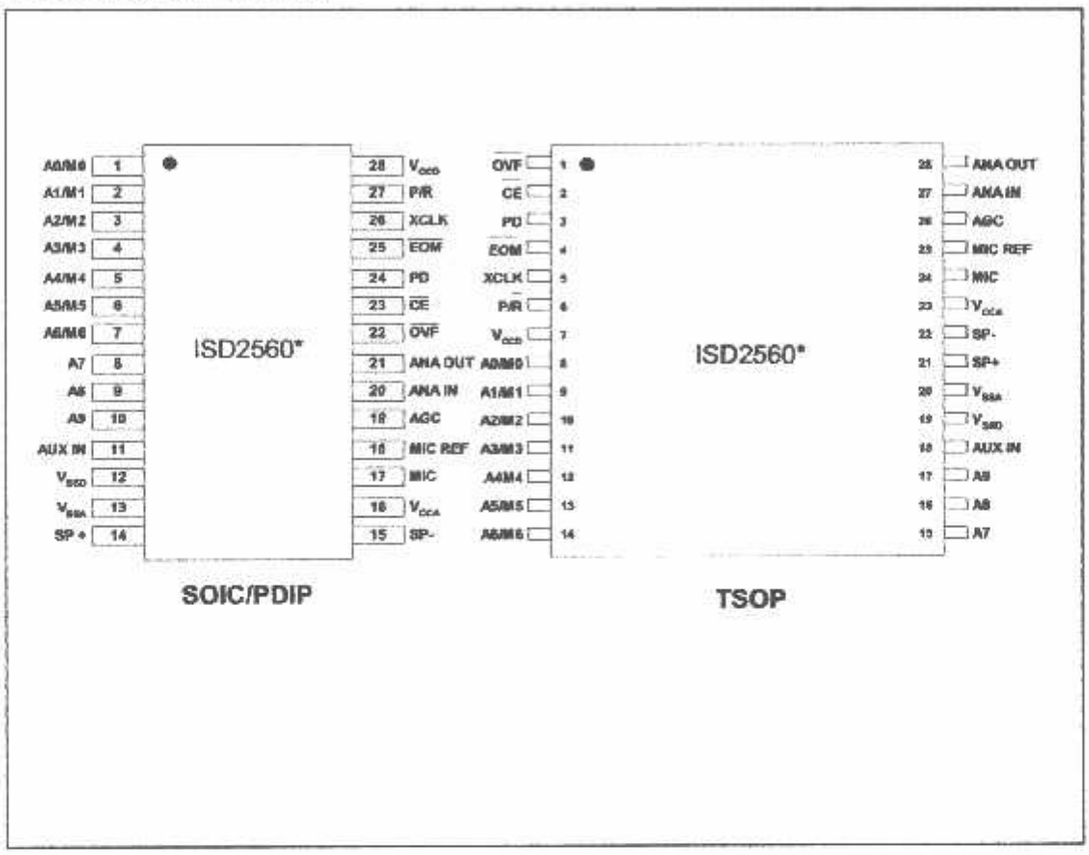


Publication Release Date: May 2003
Revision 1.0

Winbond Electronics Corp.

4. TABLE OF CONTENTS

5. PIN CONFIGURATION



* Same pinouts for ISD2575 / 2590 / 25120 products

6. PIN DESCRIPTION

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
Ax/Mx	1-10/ 1-7	8-17/ 8-14	**Address/Mode Inputs:** The Address/Mode Inputs have two functions depending on the level of the two Most Significant Bits (MSB) of the address pins (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 any internal address information during the operation. 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 on page 12. 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.
AUX IN	11	18	**Auxiliary Input:** 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.
V_{SSA}, V_{SSD}	13, 12	20, 19	**Ground:** 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.
SP+/SP-	14/15	21/22	**Speaker Outputs:** All devices in the ISD2500 series include an on-chip differential speaker driver, capable of driving 50 mW into 16 Ω from AUX IN (12.2mW from memory). [1] 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. [2] A single-end output may be used (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. However, the use of single-end output results in a 1 to 4 reduction in its output power.

[1] Connection of speaker outputs in parallel may cause damage to the device.
[2] Never ground or drive an unused speaker output.

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
V_{CCA}, V_{CCD}	16, 28	23, 7	**Supply Voltage:** 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.
MIC	17	24	**Microphone:** The microphone pin transfers input signal to the on-chip preamplifier. A built-in Automatic Gain Control (AGC) circuit controls the gain of this preamplifier from –15 to 24dB. An external microphone should be AC coupled to this pin via a series capacitor. The capacitor value, together with the internal 10 KΩ resistance on this pin, determines the low-frequency cutoff for the ISD2500 series passband. See Winbond's Application Information for additional information on low-frequency cutoff calculation.
MIC REF	18	25	**Microphone Reference:** 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.
AGC	19	26	**Automatic Gain Control:** 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 KΩ internal resistance and an external capacitor (C2 on the schematic of Figure 5 in section 11) 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 KΩ and 4.7 µF give satisfactory results in most cases.
ANA IN	20	27	**Analog Input:** The analog input transfers analog 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 KΩ 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.
ANA OUT	21	28	**Analog Output:** 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.

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
$\overline{OVF}$	22	1	**Overflow**: This signal pulses LOW at the end of memory array, 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.
$\overline{CE}$	23	2	**Chip Enable**: The $\overline{CE}$ input pin is taken LOW to enable all playback and record operations. The address pins and playback/record pin (P/$\overline{R}$) are latched by the falling edge of $\overline{CE}$. $\overline{CE}$ has additional functionality in the M6 (Push-Button) Operational Mode as described in the Operational Mode section.
PD	24	3	**Power Down**: When neither record nor playback operation, the PD pin should be pulled HIGH to place the part in standby 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 memory array. The PD pin has additional functionality in the M6 (Push-Button) Operation Mode as described in the Operational Mode section.
$\overline{EOM}$	25	4	**End-Of-Message**: 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 3.5V. 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 signal, indicating the device is currently recording or playing. This signal can conveniently drive an LED for visual indicator of a record or playback operation in process.

<table>
<tr><th rowspan="2">PIN NAME</th><th colspan="2">PIN NO.</th><th rowspan="2">FUNCTION</th></tr>
<tr><th>SOIC/ PDIP</th><th>TSOP</th></tr>
<tr><td>XCLK</td><td>26</td><td>5</td><td>External Clock: The external clock input has an internal pull-down device. The device is 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. If greater precision is required, the device can be clocked through the XCLK pin as follows:
<table>
<tr><th>Part Number</th><th>Sample Rate</th><th>Required Clock</th></tr>
<tr><td>ISD2560</td><td>8.0 kHz</td><td>1024 kHz</td></tr>
<tr><td>ISD2575</td><td>6.4 kHz</td><td>819.2 kHz</td></tr>
<tr><td>ISD2590</td><td>5.3 kHz</td><td>682.7 kHz</td></tr>
<tr><td>ISD25120</td><td>4.0 kHz</td><td>512 kHz</td></tr>
</table>
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.</td></tr>
<tr><td>P/$\overline{R}$</td><td>27</td><td>6</td><td>Playback/Record: The P/$\overline{R}$ input pin 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 pins 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, then End-Of-Message ($\overline{EOM}$) 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 $\overline{EOM}$ marker is encountered. The device can continue to pass an $\overline{EOM}$ marker if $\overline{CE}$ is held LOW in address mode, or in an Operational Mode. (See Operational Modes section)</td></tr>
</table>

7. FUNCTIONAL DESCRIPTION

7.1. DETAILED DESCRIPTION

Speech/Sound Quality

The Winbond's 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 the sound quality. Please refer to the ISD2560/75/90/120 Product Summary table below to compare the duration, sampling frequency and filter pass band.

The speech samples are stored directly into the on-chip nonvolatile memory without any digitization and compression associated like 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 various system requirements, the ISD2560/75/90/120 products offer single-chip solutions at 60, 75, 90, and 120 seconds. Parts may also be cascaded together for longer durations.

TABLE 1: 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

* 3db roll-off point

EEPROM Storage

One of the benefits of Winbond'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 a gang programmer.

7.2. OPERATIONAL MODES

The ISD2500 series is designed with several built-in Operational Modes that provide maximum functionality with minimum external components. These modes are described in details as below. The Operational Modes are accessed via the address pins and mapped beyond the normal message address range. When the two Most Significant Bits (MSB), A8 and A9, are HIGH, 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 of its memory. 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. This Operational Mode remains in effect until the next LOW-going $\overline{CE}$ signal, at which point the current mode(s) are sampled and executed.

TABLE 2: OPERATIONAL MODES

Mode [1]	Function	Typical Use	Jointly Compatible [2]
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete $\overline{EOM}$ markers	Position $\overline{EOM}$ 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/playback 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] Besides mode pin needed to be "1", A8 and A9 pin are also required to be "1" in order to enter into the related operational mode.

[2] Indicates additional Operational Modes which can be used simultaneously with the given mode.

7.2.1. Operational Modes Description

The Operational Modes can be used in conjunction with a microcontroller, or they can be hardwired 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 $\overline{CE}$ LOW pulse causes the internal address pointer to skip to the next message. This mode is used for playback only, and is typically used with the M4 Operational Mode.

M1 – Delete $\overline{EOM}$ Markers

The M1 Operational Mode allows sequentially recorded messages to be combined into a single message with only one $\overline{EOM}$ 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 ISD2500 device and will loop from beginning to end without $\overline{OVF}$ going LOW.

M4 – Consecutive Addressing

During normal operation, the address pointer will reset when a message is played through an $\overline{EOM}$ marker. The M4 Operational Mode inhibits the address pointer reset on $\overline{EOM}$, allowing messages to be played back consecutively.

M5 - $\overline{CE}$ -Level Activated

The default mode for ISD2500 devices is for $\overline{CE}$ to be edge-activated on playback and level-activated on record. The M5 Operational Mode causes the $\overline{CE}$ pin to be interpreted as level-activated as opposed to edge-activated during playback. This is especially useful for terminating playback operations using the $\overline{CE}$ signal. In this mode, $\overline{CE}$ LOW begins a playback cycle, at the beginning of the device memory. The playback cycle continues as long as $\overline{CE}$ is held LOW. When $\overline{CE}$ goes HIGH, playback will immediately end. A new $\overline{CE}$ LOW will restart the message from the beginning unless M4 is also HIGH.

M6 – Push-Button Mode

The ISD2500 series 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 $\overline{CE}$ goes HIGH.

When this operational mode is implemented, three of the pins on the device have alternate functionality as described in the table below.

TABLE 3: ALTERNATE FUNCTIONALITY IN PINS

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{CE}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse-activated)
$\overline{EOM}$	Active-HIGH Run Indicator

$\overline{CE}$ (START/PAUSE)

In Push-Button Operational Mode, $\overline{CE}$ 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 record cycle according to the level on the P/$\overline{R}$ pin. A subsequent pulse on the $\overline{CE}$ pin, before an $\overline{EOM}$ is reached in playback or an overflow condition occurs, will pause the current operation, and the address counter is not reset. Another $\overline{CE}$ pulse will cause the device to continue the operation from the place where it is paused.

PD (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.

$\overline{EOM}$ (RUN)

In Push-Button Operational Mode, $\overline{EOM}$ 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 P/$\overline{R}$ pin is taken LOW.
3. The $\overline{CE}$ pin is pulsed LOW. Recording starts, $\overline{EOM}$ goes HIGH to indicate an operation in progress.
4. When the $\overline{CE}$ pin is pulsed LOW. Recording pauses, $\overline{EOM}$ goes back LOW. The internal address pointers are not cleared, but the $\overline{EOM}$ marker is stored in memory to indicate as the message end. The P/$\overline{R}$ pin may be taken HIGH at this time. Any subsequent $\overline{CE}$ would start a playback at address 0.
5. The $\overline{CE}$ pin is pulsed LOW. Recording starts at the next address after the previous set $\overline{EOM}$ marker. $\overline{EOM}$ goes back HIGH.[3]
6. When the recording sequences are finished, the final $\overline{CE}$ pulse LOW will end the last record cycle, leaving a set $\overline{EOM}$ marker at the message end. Recording may also be terminated by a HIGH level on PD, which will leave a set $\overline{EOM}$ marker.

Playback in Push-Button Mode

1. The PD pin should be LOW.
2. The P/$\overline{R}$ pin is taken HIGH.
3. The $\overline{CE}$ pin is pulsed LOW. Playback starts, $\overline{EOM}$ goes HIGH to indicate an operation in progress.
4. If the $\overline{CE}$ pin is pulsed LOW or an $\overline{EOM}$ marker is encountered during an operation, the part will pause. The internal address pointers are not cleared, and $\overline{EOM}$ goes back LOW. The P/$\overline{R}$ 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. $\overline{CE}$ is again pulsed LOW. Playback starts where it left off, with $\overline{EOM}$ 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 $\overline{CE}$ 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.

[3] If the M1 Operational Mode pin is also HIGH, the just previously written $\overline{EOM}$ bit is erased, and recording starts at that address.

Good Audio Design Practices

Winbond 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 Application Information or below links for details.

Good Audio Design Practices

http://www.winbond-usa.com/products/isd_products/chipcorder/applicationinfo/apin11.pdf

Single-Chip Board Layout Diagrams

http://www.winbond-usa.com/products/isd_products/chipcorder/applicationinfo/apin12.pdf

8. TIMING DIAGRAMS

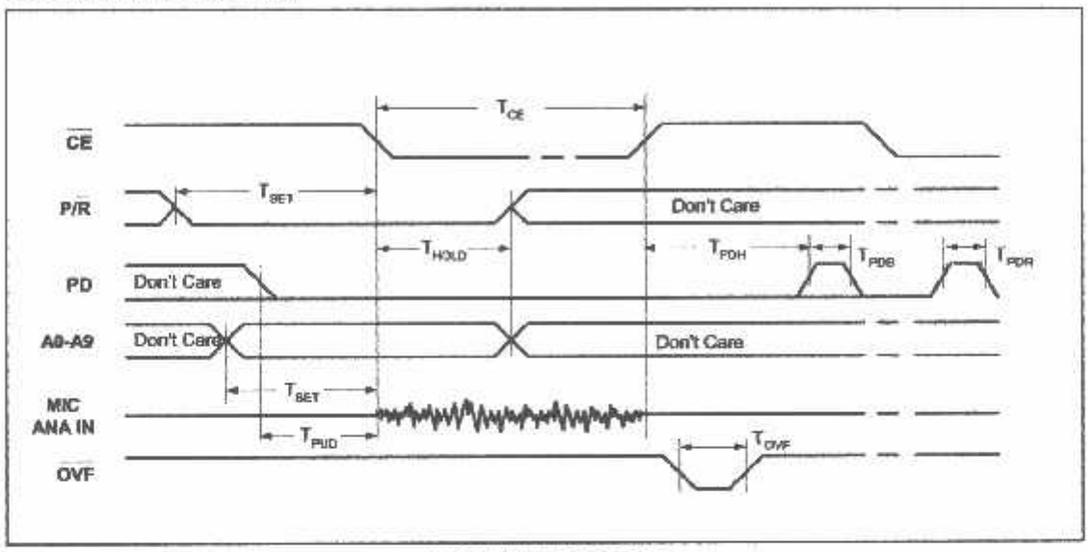


FIGURE 1: RECORD

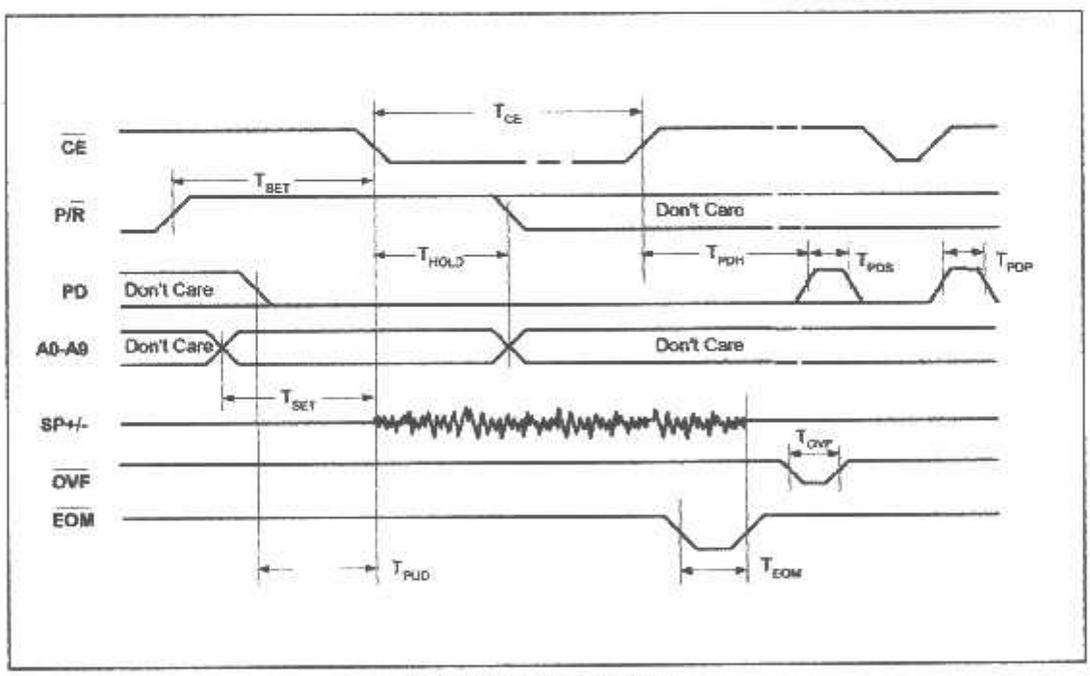


FIGURE 2: PLAYBACK

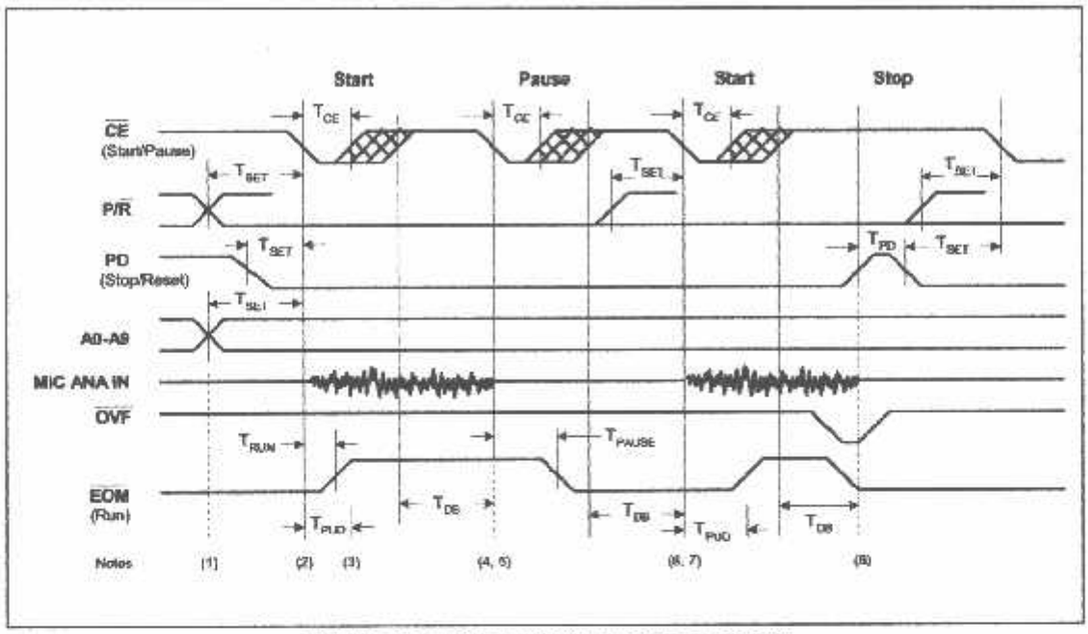


FIGURE 3: PUSH-BUTTON MODE RECORD

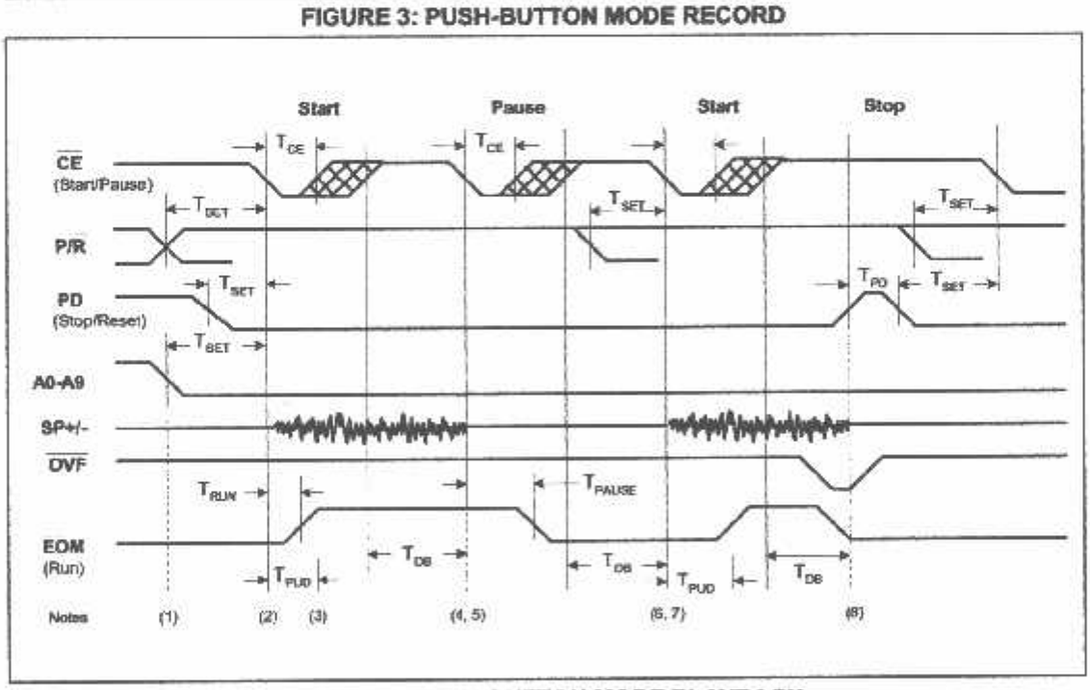


FIGURE 4: PUSH-BUTTON MODE PLAYBACK

Notes for Push-Button modes:

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.

9. ABSOLUTE MAXIMUM RATINGS

TABLE 4: ABSOLUTE MAXIMUM RATINGS (DIE)

CONDITION	VALUE
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pad	(V_{SS} –0.3V) to (V_{CC} +0.3V)
Voltage applied to any pad (Input current limited to ±20mA)	(V_{SS} –1.0V) to (V_{CC} +1.0V)
V_{CC} – V_{SS}	-0.3V to +7.0V

TABLE 5: ABSOLUTE MAXIMUM RATINGS (PACKAGED PARTS)

CONDITION	VALUE
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pin	(V_{SS} –0.3V) to (V_{CC} +0.3V)
Voltage applied to any pin (Input current limited to ±20 mA)	(V_{SS} –1.0V) to (V_{CC} +1.0V)
Lead temperature (Soldering – 10sec)	300°C
V_{CC} – V_{SS}	-0.3V to +7.0V

Note: Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability and performance. Functional operation is not implied at these conditions.

9.1 Operating Conditions

TABLE 6: OPERATING CONDITIONS (DIE)

CONDITION	VALUE
Commercial operating temperature range	0°C to +50°C
Supply voltage (V_{CC}) [1]	+4.5V to +6.5V
Ground voltage (V_{SS}) [2]	0V

TABLE 7: OPERATING CONDITIONS (PACKAGED PARTS)

CONDITION	VALUE
Commercial operating temperature range [3]	0°C to +70°C
Supply voltage (V_{CC}) [1]	+4.5V to +5.5V
Ground voltage (V_{SS}) [2]	0V

[1] $V_{CC} = V_{CCA} = V_{CCD}$

[2] $V_{SS} = V_{SSA} = V_{SSD}$

[3] Case Temperature

10. ELECTRICAL CHARACTERISTICS

10.1. PARAMETERS FOR PACKAGED PARTS

TABLE 8: DC PARAMETERS – Packaged Parts

PARAMETER	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
Input Low Voltage	V_{IL}			0.8	V	
Input High Voltage	V_{IH}	2.0			V	
Output Low Voltage	V_{OL}			0.4	V	I_{OL} = 4.0 mA
Output High Voltage	V_{OH}	V_{CC} - 0.4			V	I_{OH} = -10 µA
$\overline{OVF}$ Output High Voltage	V_{OH1}	2.4			V	I_{OH} = -1.6 mA
$\overline{EOM}$ Output High Voltage	V_{OH2}	V_{CC} – 1.0	V_{CC} - 0.8		V	I_{OH} = -3.2 mA
V_{CC} Current (Operating)	I_{CC}		25	30	mA	R_{EXT} = ∞ [3]
V_{CC} Current (Standby)	I_{SB}		1	10	µA	[3]
Input Leakage Current	I_{IL}			±1	µA	
Input Current HIGH w/Pull Down	I_{ILPD}			130	µA	Force V_{CC} [4]
Output Load Impedance	R_{EXT}	16			Ω	Speaker Load
Preamp Input Resistance	R_{MIC}	4	9	15	KΩ	MIC and MIC REF Pins
AUX IN Input Resistance	R_{AUX}	5	11	20	KΩ	
ANA IN Input Resistance	$R_{ANA\ IN}$	2.3	3	5	KΩ	
Preamp Gain 1	A_{PRE1}	21	24	26	dB	AGC = 0.0V
Preamp Gain 2	A_{PRE2}		-15	5	dB	AGC = 2.5V
AUX IN/SP+ Gain	A_{AUX}		0.98	1.0	V/V	
ANA IN to SP+/- Gain	A_{ARP}	21	23	26	dB	
AGC Output Resistance	R_{AGC}	2.5	5	9.5	KΩ	

Notes:

[1] Typical values @ T_A = 25° and V_{CC} = 5.0V.

[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

[3] V_{CCA} and V_{CCD} connected together.

[4] XCLK pin only.

Winbond Electronics Corp.

TABLE 9: AC PARAMETERS – Packaged Parts

CHARACTERISTIC	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
Sampling Frequency	F_S					
ISD2560			8.0		kHz	[7]
ISD2575			6.4		kHz	[7]
ISD2590			5.3		kHz	[7]
ISD25120			4.0		kHz	[7]
Filter Pass Band	F_{CF}					
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]
Record Duration	T_{REC}					
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]
Playback Duration	T_{PLAY}					
ISD2560		58.1	60.0	62.0	sec	Commercial Operation
ISD2575		72.6	75.0	77.5	sec	Commercial Operation
ISD2590		87.1	90.0	93.0	sec	Commercial Operation
ISD25120		116.1	120.0	123.9	sec	Commercial Operation
$\overline{CE}$ Pulse Width	T_{CE}		100		nsec	
Control/Address Setup Time	T_{SET}		300		nsec	
Control/Address Hold Time	T_{HOLD}		0		nsec	
Power-Up Delay	T_{PUD}					
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
PD Pulse Width (record)	T_{PDR}					
ISD2560			25.0		msec	
ISD2575			31.25		msec	
ISD2590			37.5		msec	
ISD25120			50.0		msec	

Winbond Electronics Corp.

TABLE 9: AC PARAMETERS – Packaged Parts (Cont'd)

CHARACTERISTIC	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
PD Pulse Width (Play)	T_{PDP}					
ISD2560			12.5		msec	
ISD2575			15.625		msec	
ISD2590			18.75		msec	
ISD25120			25.0		msec	
PD Pulse Width (Static)	T_{PDS}		100		nsec	[6]
Power Down Hold	T_{PDH}		0		nsec	
$\overline{EOM}$ Pulse Width	T_{EOM}					
ISD2560			12.5		msec	
ISD2575			15.625		msec	
ISD2590			18.75		msec	
ISD25120			25.0		msec	
Overflow Pulse Width	T_{OVF}		6.5		µsec	
Total Harmonic Distortion	THD		1	2	%	@ 1 kHz
Speaker Output Power	P_{OUT}		12.2	50	mW	R_{EXT} = 16 Ω[4]
Voltage Across Speaker Pins	V_{OUT}			2.5	V p-p	R_{EXT} = 600 Ω
MIC Input Voltage	V_{IN1}			20	mV	Peak-to-Peak[5]
ANA IN Input Voltage	V_{IN2}			50	mV	Peak-to-Peak
AUX Input Voltage	V_{IN3}			1.25	V	Peak-to-Peak; R_{EXT} = 16 Ω

Notes:

[1] Typical values @ T_A = 25°C and V_{CC} = 5.0V.

[2] All Min/Max limits are guaranteed by Winbond 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 K Ω 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. For greater stability, an external clock can be utilized (see Pin Descriptions)

[8] Filter specification applies to the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 6 dB drop by nature of passing through both filters.

10.1.1. Typical Parameter Variation with Voltage and Temperature (Packaged Parts)

Chart 1: Record Mode Operating Current (I_{CC})

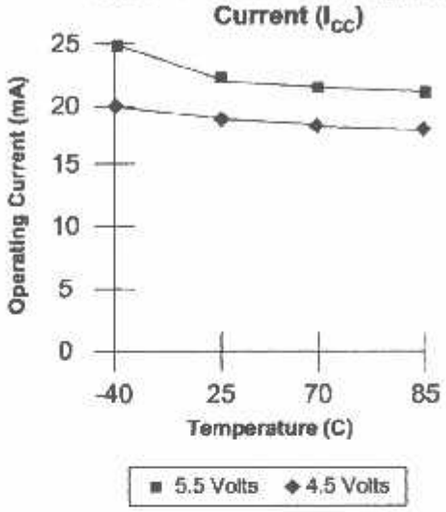


Chart 3: Standby Current (I_{SB})

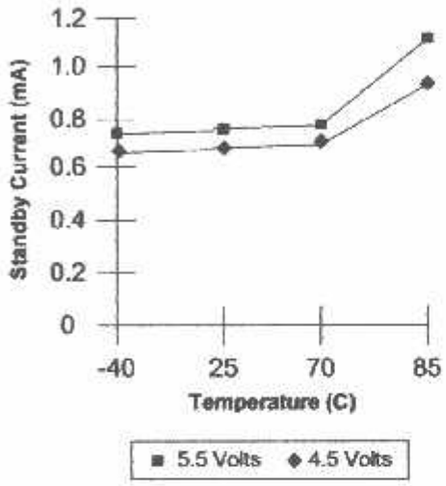


Chart 2: Total Harmonic Distortion

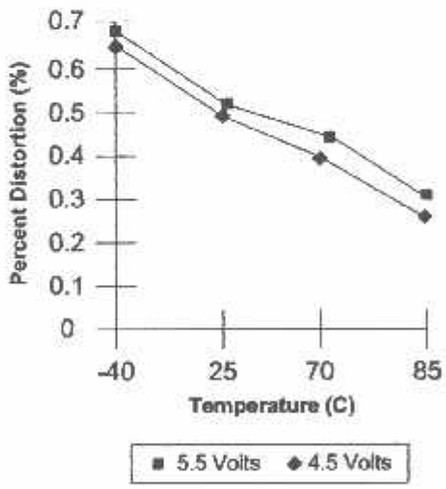
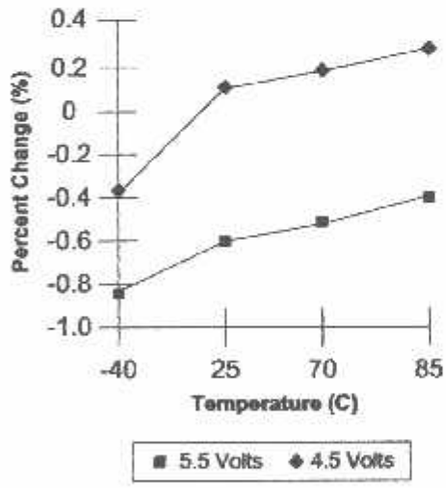


Chart 4: Oscillator Stability



10.2. PARAMETERS FOR DIE

TABLE 10: DC PARAMETERS – Die

PARAMETER	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
Input Low Voltage	V_{IL}			0.8	V	
Input High Voltage	V_{IH}	2.0			V	
Output Low Voltage	V_{OL}			0.4	V	I_{OL} = 4.0 mA
Output High Voltage	V_{OH}	V_{CC} - 0.4			V	I_{OH} = -10 µA
$\overline{OVF}$ Output High Voltage	V_{OH1}	2.4			V	I_{OH} = -1.6 mA
$\overline{EOM}$ Output High Voltage	V_{OH2}	V_{CC} – 1.0	V_{CC} - 0.8		V	I_{OH} = -3.2 mA
V_{CC} Current (Operating)	I_{CC}		25	30	mA	R_{EXT} = ∞ [3]
V_{CC} Current (Standby)	I_{SB}		1	10	µA	[2]
Input Leakage Current	I_{IL}			±1	µA	
Input Current HIGH w/Pull Down	I_{ILPD}			130	µA	Force V_{CC} [4]
Output Load Impedance	R_{EXT}	16			Ω	Speaker Load
Preamp IN Input Resistance	R_{MIC}	4	9	15	KΩ	MIC and MIC REF Pads
AUX IN Input Resistance	R_{AUX}	5	11	20	KΩ	
ANA IN Input Resistance	$R_{ANA\ IN}$	2.3	3	5	KΩ	
Preamp Gain 1	A_{PRE1}	21	24	26	dB	AGC = 0.0V
Preamp Gain 2	A_{PRE2}		-15	5	dB	AGC = 2.5V
AUX IN/SP+ Gain	A_{AUX}		0.98	1.0	V/V	
ANA IN to SP+/- Gain	A_{ARP}	21	23	26	dB	
AGC Output Resistance	R_{AGC}	2.5	5	9.5	KΩ	

Notes:

[1] Typical values @ T_A = 25°C and V_{CC} = 5.0V.

[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

[3] V_{CCA} and V_{CCD} connected together.

[4] XCLK pad only.

Winbond Electronics Corp.

TABLE 11: AC PARAMETERS – Die

CHARACTERISTIC	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
Sampling Frequency	F_S					
ISD2560			8.0		kHz	[7]
ISD2575			6.4		kHz	[7]
ISD2590			5.3		kHz	[7]
ISD25120			4.0		kHz	[7]
Filter Pass Band	F_{CF}					
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]
Record Duration	T_{REC}					
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]
Playback Duration	T_{PLAY}					
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]
$\overline{CE}$ Pulse Width	T_{CE}		100		nsec	
Control/Address Setup Time	T_{SET}		300		nsec	
Control/Address Hold Time	T_{HOLD}		0		nsec	
Power-Up Delay	T_{PUD}					
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
PD Pulse Width (Record)	T_{POR}					
ISD2560			25.0		msec	
ISD2575			31.25		msec	
ISD2590			37.5		msec	
ISD25120			50.0		msec	

TABLE 11: AC PARAMETERS – Die (Cont'd)

CHARACTERISTIC	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
PD Pulse Width (Play)	T_{PDP}					
ISD2560			12.5		msec	
ISD2575			15.625		msec	
ISD2590			18.75		msec	
ISD25120			25.0		msec	
PD Pulse Width (Static)	T_{PDS}		100		nsec	[6]
Power Down Hold	T_{PDH}		0		nsec	
$\overline{EOM}$ Pulse Width	T_{EOM}					
ISD2560			12.5		msec	
ISD2575			15.625		msec	
ISD2590			18.75		msec	
ISD25120			25.0		msec	
Overflow Pulse Width	T_{OVF}		6.5		µsec	
Total Harmonic Distortion	THD		1	3	%	@ 1 kHz
Speaker Output Power	P_{OUT}		12.2	50	mW	R_{EXT} = 16 Ω[4]
Voltage Across Speaker Pins	V_{OUT}			2.5	V p-p	R_{EXT} = 600 Ω
MIC Input Voltage	V_{IN1}			20	mV	Peak-to-Peak[5]
ANA IN Input Voltage	V_{IN2}			50	mV	Peak-to-Peak
AUX Input Voltage	V_{IN3}			1.25	V	Peak-to-Peak; R_{EXT} = 16 Ω

Notes:

[1] Typical values @ T_A = 25°C and V_{CC} = 5.0V.

[2] All Min/Max limits are guaranteed by Winbond 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 K Ω 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. For greater stability, an external clock can be utilized (see Pin Descriptions)

[8] Filter specification applies to the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 6 dB drop by nature of passing through both filters.

10.2.1. Typical Parameter Variation with Voltage and Temperature (Die)

Chart 5: Record Mode Operating Current (I_{CC})

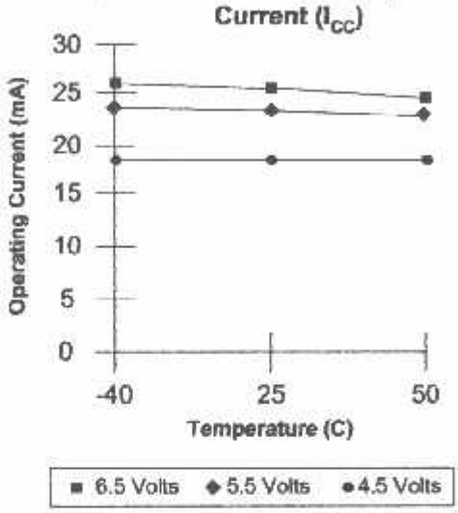


Chart 7: Standby Current (I_{SB})

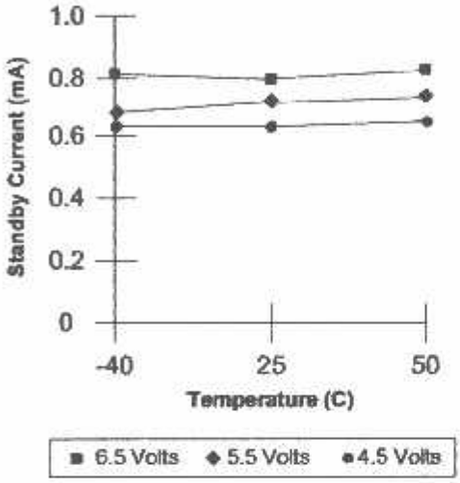


Chart 6: Total Harmonic Distortion

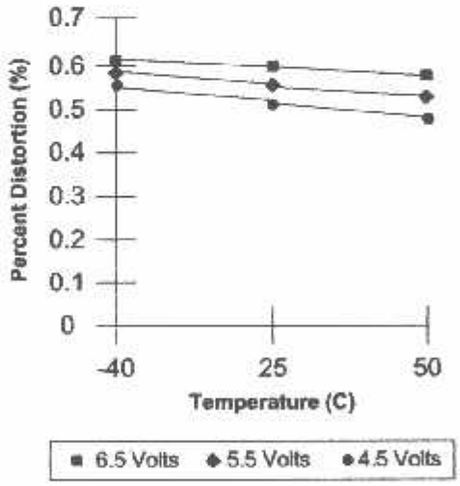
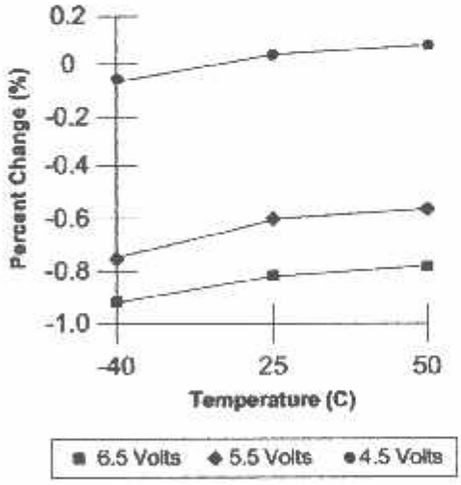


Chart 8: Oscillator Stability



Winbond Electronics Corp.

10.3. PARAMETERS FOR PUSH-BUTTON MODE

TABLE 12: PARAMETERS FOR PUSH-BUTTON MODE

PARAMETER	SYMBOL	MIN[2]	TYP[1]	MAX[2]	UNITS	CONDITIONS
$\overline{CE}$ Pulse Width (Start/Pause)	T_{CE}		300		nsec	
Control/Address Setup Time	T_{SET}		300		nsec	
Power-Up Delay	T_{PUD}					
ISD2560			25.0		msec	
ISD2575			31.25		msec	
ISD2590			37.25		msec	
ISD25120			50.0		msec	
PD Pulse Width (Stop/Restart)	T_{PD}		300		nsec	
$\overline{CE}$ to $\overline{EOM}$ HIGH	T_{RUN}	25		400	nsec	
$\overline{CE}$ to $\overline{EOM}$ LOW	T_{PAUSE}	50		400	nsec	
$\overline{CE}$ HIGH Debounce	T_{DB}					
ISD2560		70		105	msec	
ISD2575		85		135	msec	
ISD2590		105		160	msec	
ISD25120		135		215	msec	

Notes:

[1] Typical values @ $T_A = 25°C$ and $V_{CC} = 5.0V$.

[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

11. TYPICAL APPLICATION CIRCUIT

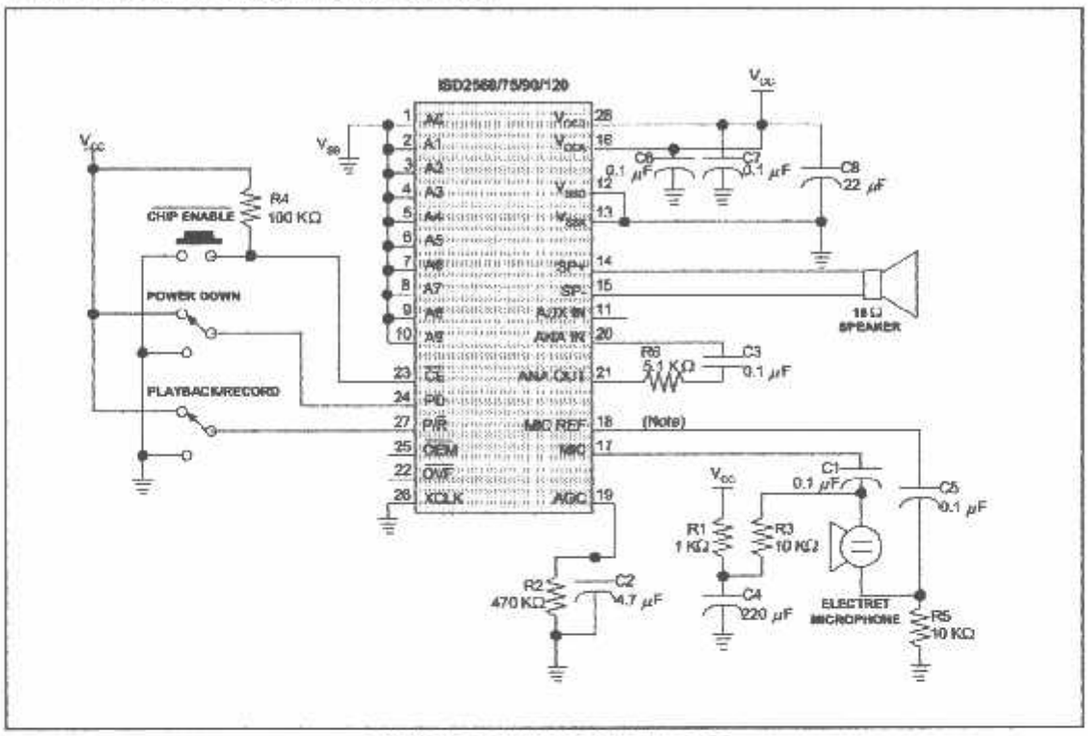


FIGURE 5: DESIGN SCHEMATIC

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 below.

TABLE 13: APPLICATION EXAMPLE – BASIC DEVICE CONTROL

Control Step	Function	Action
1	Power up chip and select Record/Playback Mode	1. PD = LOW, 2. P/$\overline{R}$ = As desired
2	Set message address for record/playback	Set addresses A0-A9
3A	Begin playback	P/$\overline{R}$ = HIGH, $\overline{CE}$ = Pulse LOW
3B	Begin record	P/$\overline{R}$ = LOW, $\overline{CE}$ = LOW
4A	End playback	Automatic
4B	End record	PD or $\overline{CE}$ = HIGH

TABLE 14: 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 command 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

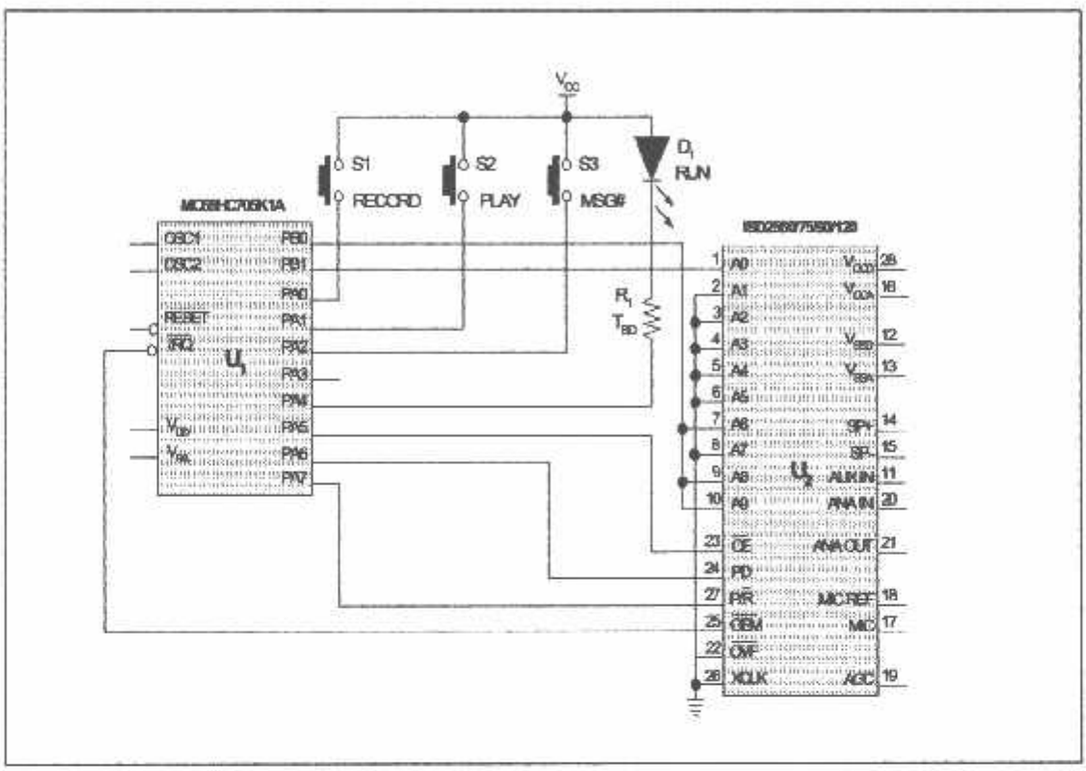


FIGURE 6: ISD2560/75/90/120 APPLICATION EXAMPLE – MICROCONTROLLER/ISD2500 INTERFACE

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 function desired in a particular application.

Note: Winbond does not recommend connecting address lines directly to a microprocessor bus. Address lines should be externally latched.

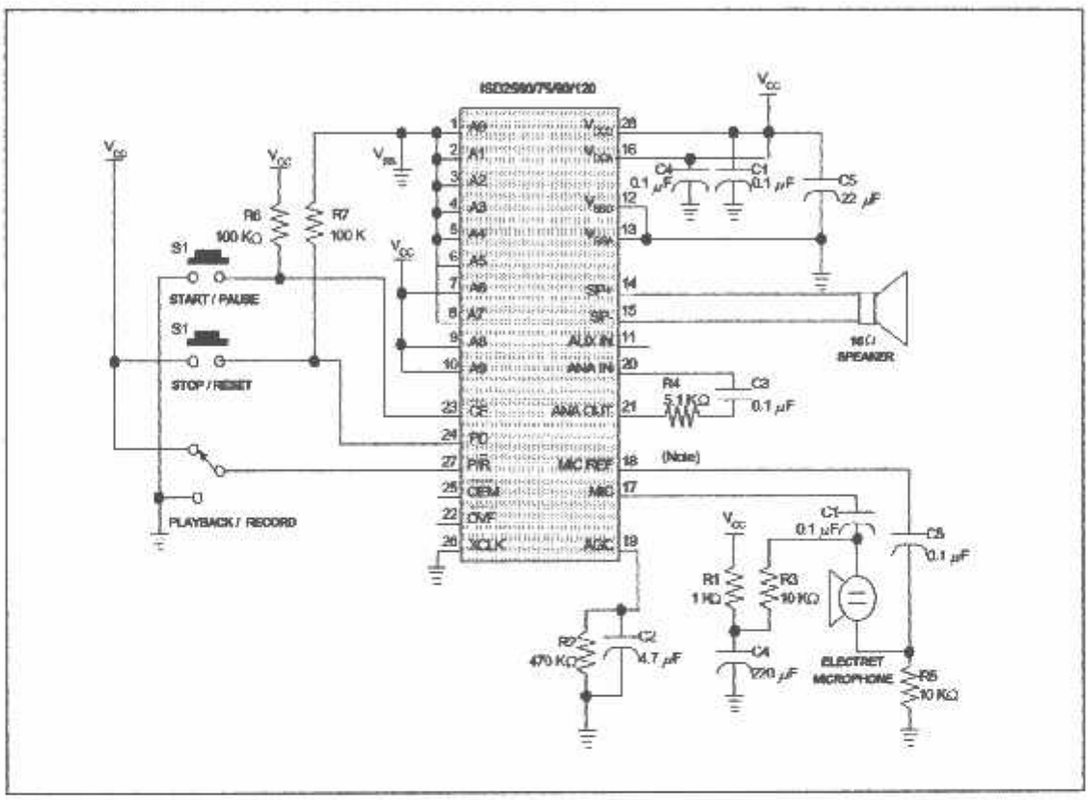


FIGURE 7: ISD2560/75/90/120 APPLICATION EXAMPLE – PUSH-BUTTON

Note: Please refer to page 13 for more details.

TABLE 15: APPLICATION EXAMPLE – PUSH-BUTTON CONTROL

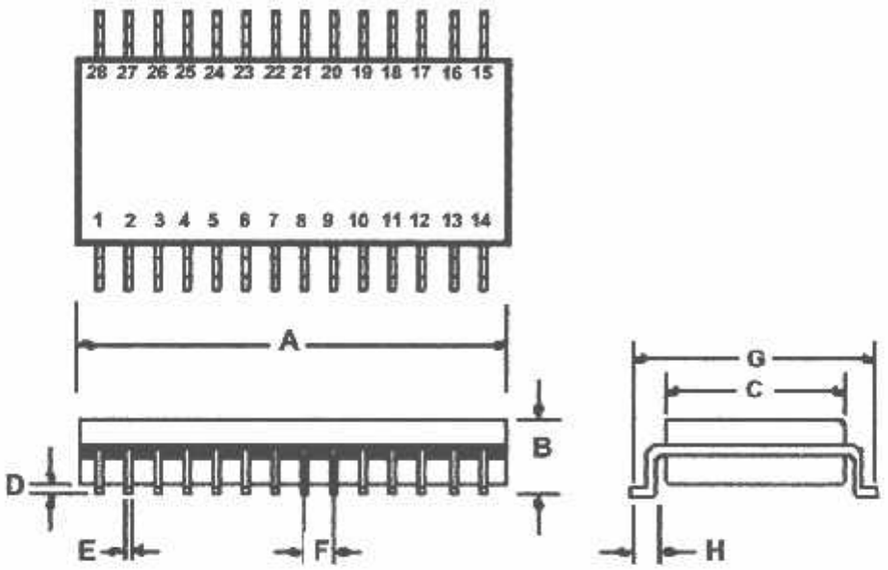
Control Step	Function	Action
1	Select Record/Playback Mode	$\overline{P/R}$ = As desired
2A	Begin playback	$\overline{P/R}$ = HIGH, $\overline{CE}$ = Pulse LOW
2B	Begin record	$\overline{P/R}$ = LOW, $\overline{CE}$ = Pulse LOW
3	Pause record or playback	$\overline{CE}$ = Pulsed LOW
4A	End playback	Automatic at $\overline{EOM}$ marker or PD = Pulsed HIGH
4B	End record	PD = Pulsed HIGH

TABLE 16: 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

12. PACKAGE DRAWING AND DIMENSIONS

12.1. 28-LEAD 300-MIL PLASTIC SMALL OUTLINE IC (SOIC)

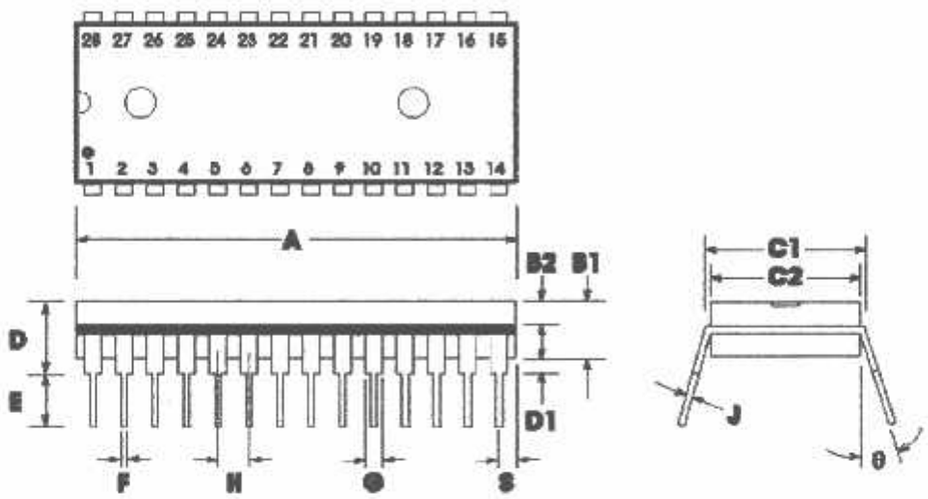


	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.701	0.706	0.711	17.81	17.93	18.06
B	0.097	0.101	0.104	2.46	2.56	2.64
C	0.292	0.296	0.299	7.42	7.52	7.59
D	0.005	0.009	0.0115	0.127	0.22	0.29
E	0.014	0.016	0.019	0.35	0.41	0.48
F		0.050			1.27	
G	0.400	0.406	0.410	10.16	10.31	10.41
H	0.024	0.032	0.040	0.61	0.81	1.02

Note: Lead coplanarity to be within 0.004 inches.

Winbond Electronics Corp.

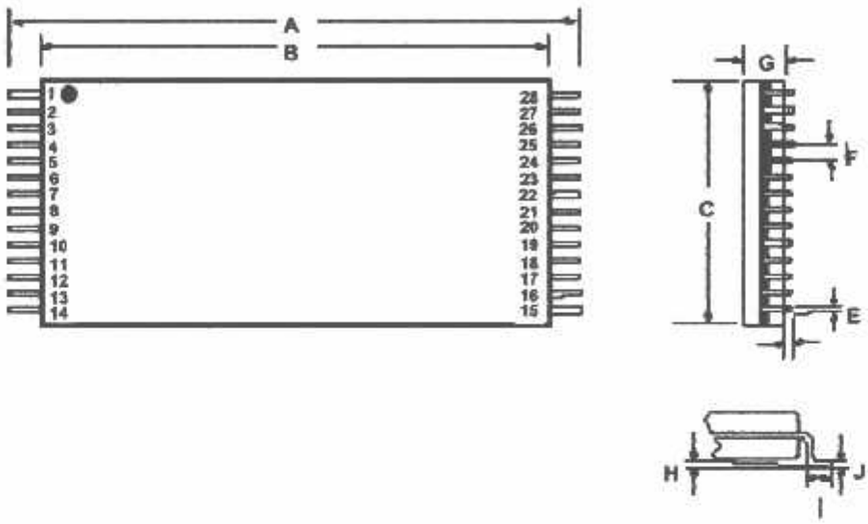
12.2. 28-LEAD 600-MIL PLASTIC DUAL INLINE PACKAGE (PDIP)



	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	1.62
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°

Winbond Electronics Corp.

12.3. 28-LEAD 8X13.4MM PLASTIC THIN SMALL OUTLINE PACKAGE (TSOP) TYPE 1



Plastic Thin Small Outline Package (TSOP) Type 1 Dimensions

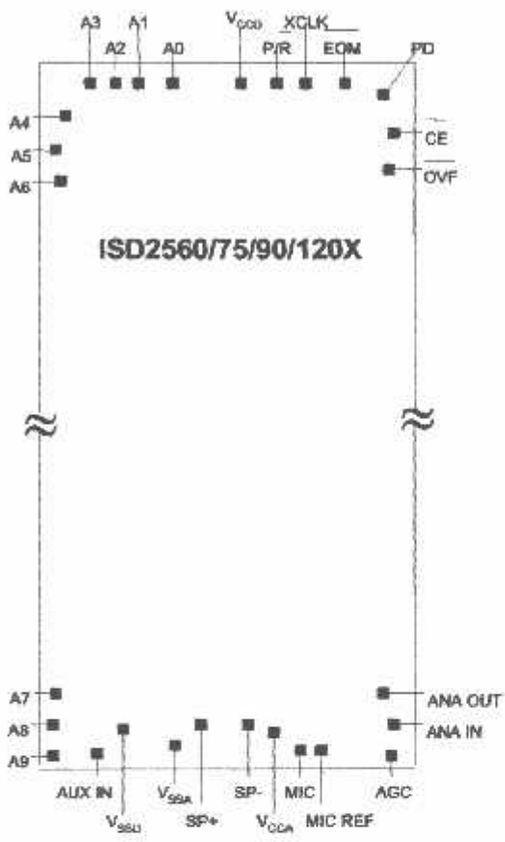
	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.520	0.528	0.535	13.20	13.40	13.60
B	0.461	0.465	0.469	11.70	11.80	11.90
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.007	0.009	0.011	0.17	0.22	0.27
F		0.0217			0.55	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0^0	3^0	6^0	0^0	3^0	6^0
I	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.*

12.4. ISD2560/75/95/120 Product Bonding Physical Layout (Die) [1]

ISD2560/75/95/120

- Die Dimensions
  X: 149.5 ± 1 mils
  Y: 262.0 ± 1 mils
- Die Thickness [2]
  11.8 ± .4 mils
- Pad Opening
  111 x 111 microns
  4.4 x 4.4 mils



Notes:

[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 Winbond factory for status and availability.

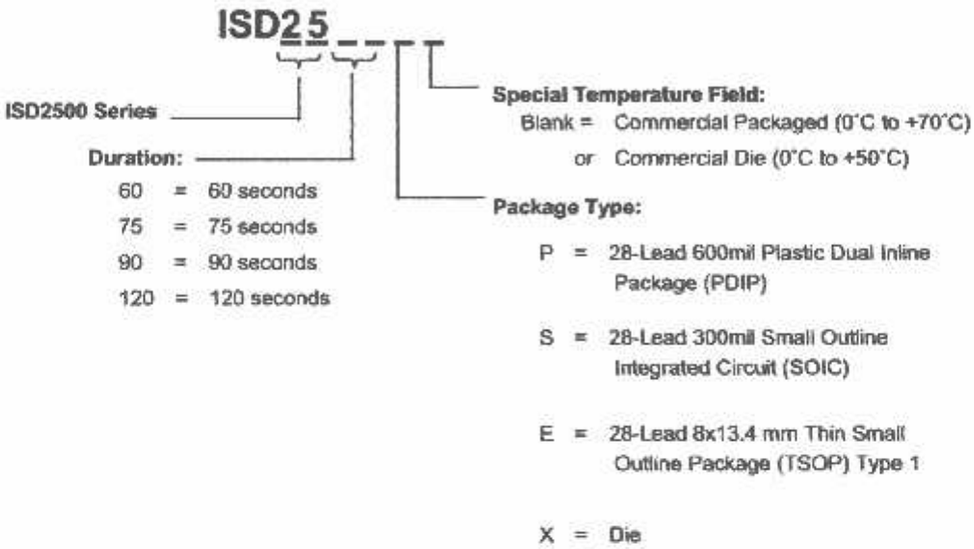
ISD2560/75/90/120 PRODUCT PAD DESIGNATIONS

(with respect to die center)

Pad	Pad Name	X Axis (µm)	Y Axis (µm)
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	-1410.2	-3115.7
V_{SSD}	V_{SS} Digital Power Supply	-1112.4	-3096.5
V_{SSA}	V_{SS} Analog Power Supply	-408.2	-3138.9
SP+	Speaker Output +	-46.65	-3068.4
SP-	Speaker Output -	386.1	-3068.4
V_{CCA}	V_{CC} Analog Power Supply	746.9	-3110.8
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
$\overline{OVF}$	Overflow Output	1674.6	2489.5
$\overline{CE}$	Chip Enable Input	1726.7	2824.4
PD	Power Down Input	1730.5	3094.0
$\overline{EOM}$	End of Message	1341.2	3122.1
XCLK	No Connect (optional)	986.5	3160.7
P/$\overline{R}$	Playback/Record	807.2	3163.4
V_{CCD}	V_{CC} Digital Power Supply	544.4	3159.6

13. ORDERING INFORMATION

Product Number Descriptor Key



When ordering ISD2560/75/90/120 products refer to the following part numbers which are supported in volume for this product series. Consult the local Winbond Sales Representative or Distributor for availability information.

Part Number	Part Number	Part Number	Part Number
ISD2560P	ISD2575P	ISD2590P	ISD25120P
ISD2560S	ISD2575S	ISD2590S	ISD25120S
ISD2560E	ISD2575E	ISD2590E	
ISD2560X	ISD2575X	ISD2590X	ISD25120X

For the latest product information, access Winbond's worldwide website at
http://www.winbond-usa.com

14. VERSION HISTORY

VERSION	DATE	PAGE	DESCRIPTION
0	Apr. 1998	All	Preliminary Specifications
1.0	May 2003	All	Re-format the document. Update TSOP pin configuration. Revise Overflow pad designation.

Winbond Electronics Corp.

The contents of this document are provided only as a guide for the applications of Winbond products. Winbond makes no representation or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to discontinue or make changes to specifications and product descriptions at any time without notice. No license, whether express or implied, to any intellectual property or other right of Winbond or others is granted by this publication. Except as set forth in Winbond's Standard Terms and Conditions of Sale, Winbond assumes no liability whatsoever and disclaims any express or implied warranty of merchantability, fitness for a particular purpose or infringement of any intellectual property.

Winbond products are not designed, intended, authorized or warranted for use as components in systems or equipments intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Winbond products are not intended for applications wherein failure of Winbond products could result or lead to a situation wherein personal injury, death or severe property or environmental injury could occur.

Application examples and alternative uses of any integrated circuit contained in this publication are for illustration only and Winbond makes no representation or warranty that such applications shall be suitable for the use specified.

ISD® and ChipCorder® are trademarks of Winbond Electronics Corporation.

The 100-year retention and 100K record cycle projections are based upon accelerated reliability tests, as published in the Winbond Reliability Report, and are neither warranted nor guaranteed by Winbond.

Information contained in this ISD® ChipCorder® data sheet supersedes all data for the ISD ChipCorder products published by ISD® prior to August, 1998.

This data sheet and any future addendum to this data sheet is(are) the complete and controlling ISD® ChipCorder® product specifications. In the event any inconsistencies exist between the information in this and other product documentation, or in the event that other product documentation contains information in addition to the information in this, the information contained herein supersedes and governs such other information in its entirety.

Winbond Electronics Corp.

Headquarters
No. 4, Creation Rd. III
Science-Based Industrial Park,
Hsinchu, Taiwan
TEL: 886-3-5770066
FAX: 886-3-5665577
http://www.winbond.com.tw/

Winbond Electronics Corporation America
2727 North First Street, San Jose,
CA 95134, U.S.A.
TEL: 1-408-9436666
FAX: 1-408-5441798
http://www.winbond-usa.com/

Winbond Electronics (Shanghai) Ltd.
27F, 299 Yan An W. Rd. Shanghai,
200336 China
TEL: 86-21-62365999
FAX: 86-21-62365998

Taipei Office
9F, No. 480, Pueguang Rd.
Neihu District,
Taipei, 114, Taiwan
TEL: 886 2 81777168
FAX: 886 2 87153579

Winbond Electronics Corporation Japan
7F Daini-ueno BLDG, 3-7-18
Shinyokohama Kohokuku,
Yokohama 222-0033
TEL: 81-45-4781881
FAX: 81-45-4781800

Winbond Electronics (H.K.) Ltd.
Unit 9, 15, 22/F, Millennium City,
No. 378 Kwun Tong Rd.,
Kowloon, Hong Kong
TEL: 852-27513100
FAX: 852-27552064

AT 28C64

64K (8K x 8) CMOS E²PROM

Features

- Fast Read Access Time - 120 ns
- Fast Byte Write - 200 µs or 1 ms
- Self-Timed Byte Write Cycle
  - Internal Address and Data Latches
  - Internal Control Timer
  - Automatic Clear Before Write
- Direct Microprocessor Control
  - READY/BUSY Open Drain Output
  - DATA Polling
- Low Power
  - 30 mA Active Current
  - 100 µA CMOS Standby Current
- High Reliability
  - Endurance: 10^4 or 10^5 Cycles
  - Data Retention: 10 Years
- 5V ± 10% Supply
- CMOS and TTL Compatible Inputs and Outputs
- JEDEC Approved Byte-Wide Pinout
- Commercial and Industrial Temperature Ranges

Description

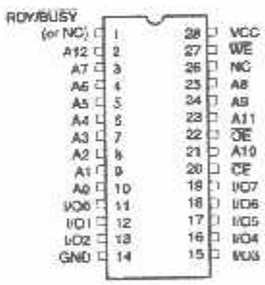
The AT28C64 is a low-power, high-performance 8,192 words by 8 bit nonvolatile Electrically Erasable and Programmable Read Only Memory with popular, easy to use features. The device is manufactured with Atmel's reliable nonvolatile technology.

(continued)

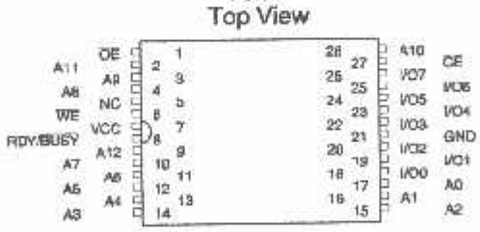
Pin Configurations

Pin Name	Function
A0 - A12	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
RDY/$\overline{BUSY}$	Ready/Busy Output
NC	No Connect
DC	Don't Connect

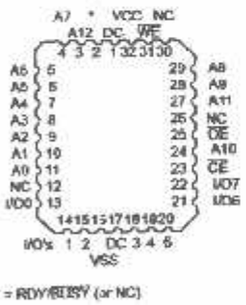
PDIP, SOIC
Top View



TSOP
Top View



LCC, PLCC
Top View



* = RDY/BUSY (or NC)

Note: PLCC package pins 1 and 17 are DON'T CONNECT.

0001G

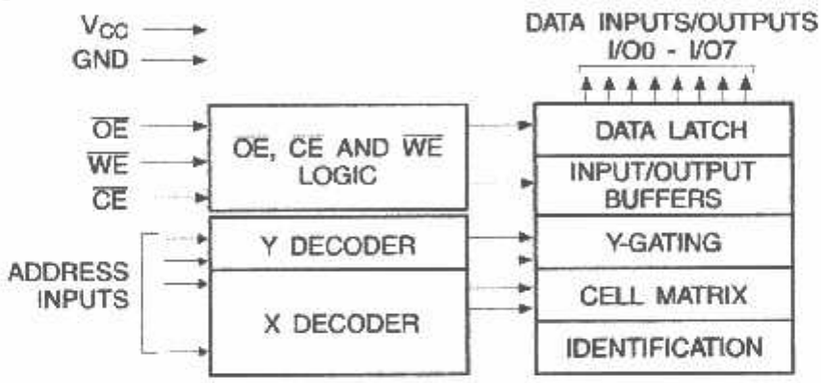
Description (Continued)

The AT28C64 is accessed like a Static RAM for the read or write cycles without the need for external components. During a byte write, the address and data are latched internally, freeing the microprocessor address and data bus for other operations. Following the initiation of a write cycle, the device will go to a busy state and automatically clear and write the latched data using an internal control timer. The device includes two methods for detecting the end of a write cycle, level detection of RDY/BUSY (unless pin 1 is N.C.) and DATA POLLING of I/O_7. Once the end of a write cycle has been detected, a new access for a read or write can begin.

The CMOS technology offers fast access times of 120 ns at low power dissipation. When the chip is deselected the standby current is less than 100 µA.

Atmel's 28C64 has additional features to ensure high quality and manufacturability. The device utilizes error correction internally for extended endurance and for improved data retention characteristics. An extra 32-bytes of E^2PROM are available for device identification or tracking.

Block Diagram



Absolute Maximum Ratings*

Temperature Under Bias	-55°C to +125°C
Storage Temperature	-65°C to +150°C
All Input Voltages (including NC Pins) with Respect to Ground	-0.6V to +6.25V
All Output Voltages with Respect to Ground	-0.6V to V_{CC} + 0.6V
Voltage on $\overline{OE}$ and A9 with Respect to Ground	-0.6V to +13.5V

*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.

Device Operation

READ: The AT28C64 is accessed like a Static RAM. When $\overline{CE}$ and $\overline{OE}$ are low and $\overline{WE}$ is high, the data stored at the memory location determined by the address pins is asserted on the outputs. The outputs are put in a high impedance state whenever $\overline{CE}$ or $\overline{OE}$ is high. This dual line control gives designers increased flexibility in preventing bus contention.

BYTE WRITE: Writing data into the AT28C64 is similar to writing into a Static RAM. A low pulse on the $\overline{WE}$ or $\overline{CE}$ input with $\overline{OE}$ high and $\overline{CE}$ or $\overline{WE}$ low (respectively) initiates a byte write. The address location is latched on the falling edge of $\overline{WE}$ (or $\overline{CE}$); the new data is latched on the rising edge. Internally, the device performs a self-clear before write. Once a byte write has been started, it will automatically time itself to completion. Once a programming operation has been initiated and for the duration of t_{WC}, a read operation will effectively be a polling operation.

FAST BYTE WRITE: The AT28C64E offers a byte write time of 200 µs maximum. This feature allows the entire device to be rewritten in 1.6 seconds.

READY/$\overline{BUSY}$: Pin 1 is an open drain READY/$\overline{BUSY}$ output that can be used to detect the end of a write cycle. RDY/$\overline{BUSY}$ is actively pulled low during the write cycle and is released at the completion of the write. The open drain connection allows for OR-tying of several devices to the same RDY/$\overline{BUSY}$ line. Pin 1 is not connected for the AT28C64X.

$\overline{DATA}$ POLLING: The AT28C64 provides $\overline{DATA}$ POLLING to signal the completion of a write cycle. During a write cycle, an attempted read of the data being written results in the complement of that data for I/O_7 (the other outputs are indeterminate). When the write cycle is finished, true data appears on all outputs.

WRITE PROTECTION: Inadvertent writes to the device are protected against in the following ways. (a) V_{CC} sense— if V_{CC} is below 3.8V (typical) the write function is inhibited. (b) V_{CC} power on delay— once V_{CC} has reached 3.8V the device will automatically time out 5 ms (typical) before allowing a byte write. (c) Write Inhibit— holding any one of $\overline{OE}$ low, $\overline{CE}$ high or $\overline{WE}$ high inhibits byte write cycles.

CHIP CLEAR: The contents of the entire memory of the AT28C64 may be set to the high state by the CHIP CLEAR operation. By setting $\overline{CE}$ low and $\overline{OE}$ to 12 volts, the chip is cleared when a 10 msec low pulse is applied to $\overline{WE}$.

DEVICE IDENTIFICATION: An extra 32-bytes of E^2PROM memory are available to the user for device identification. By raising A9 to 12 ± 0.5V and using address locations 1FE0H to 1FFFH the additional bytes may be written to or read from in the same manner as the regular memory array.

DC and AC Operating Range

		AT28C64-12	AT28C64-15	AT28C64-20	AT28C64-25
Operating Temperature (Case)	Com.	0°C - 70°C	0°C - 70°C	0°C - 70°C	0°C - 70°C
	Ind.	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C
V_{CC} Power Supply		5V ± 10%	5V ± 10%	5V ± 10%	5V ± 10%

Operating Modes

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I/O
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}
Write (2)	V_{IL}	V_{IH}	V_{IL}	D_{IN}
Standby/Write Inhibit	V_{IH}	X (1)	X	High Z
Write Inhibit	X	X	V_{IH}	
Write Inhibit	X	V_{IL}	X	
Output Disable	X	V_{IH}	X	High Z
Chip Erase	V_{IL}	V_H (3)	V_{IL}	High Z

Notes: 1. X can be V_{IL} or V_{IH}.
2. Refer to AC Programming Waveforms.
3. V_H = 12.0V ± 0.5V.

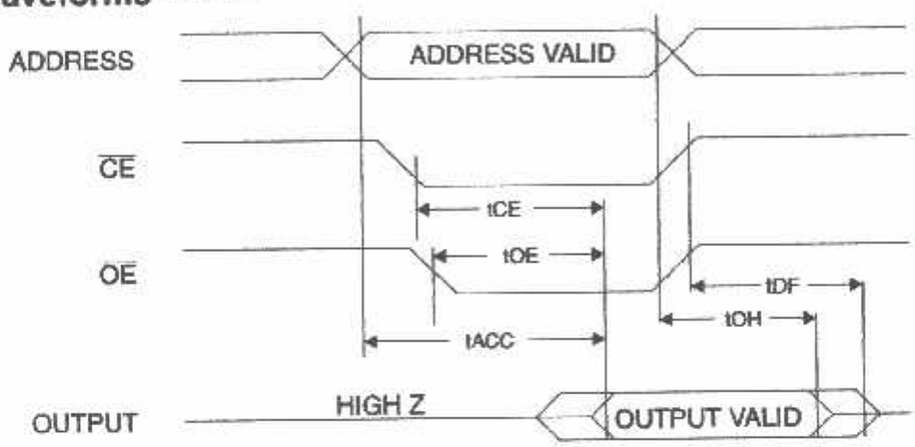
DC Characteristics

Symbol	Parameter	Condition		Min	Max	Units
I_{LI}	Input Load Current	V_{IN} = 0V to V_{CC} + 1V			10	µA
I_{LO}	Output Leakage Current	$V_{I/O}$ = 0V to V_{CC}			10	µA
I_{SB1}	V_{CC} Standby Current CMOS	$\overline{CE}$ = V_{CC} - 0.3V to V_{CC} + 1.0V			100	µA
I_{SB2}	V_{CC} Standby Current TTL	$\overline{CE}$ = 2.0V to V_{CC} + 1.0V	Com.		2	mA
			Ind.		3	mA
I_{CC}	V_{CC} Active Current AC	f = 5 MHz; I_{OUT} = 0 mA $\overline{CE}$ = V_{IL}	Com.		30	mA
			Ind.		45	mA
V_{IL}	Input Low Voltage				0.8	V
V_{IH}	Input High Voltage			2.0		V
V_{OL}	Output Low Voltage	I_{OL} = 2.1 mA = 4.0 mA for RDY/$\overline{BUSY}$			.45	V
V_{OH}	Output High Voltage	I_{OH} = -400 µA		2.4		V

AC Read Characteristics

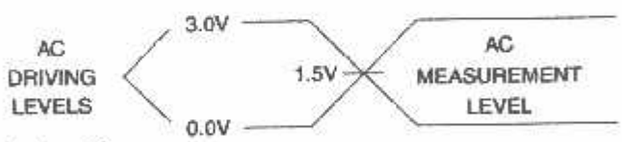
Symbol	Parameter	AT28C64-12		AT28C64-15		AT28C64-20		AT28C64-25		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay		120		150		200		250	ns
t_{CE} (1)	$\overline{CE}$ to Output Delay		120		150		200		250	ns
t_{OE} (2)	$\overline{OE}$ to Output Delay	10	60	10	70	10	80	10	100	ns
t_{DF} (3, 4)	$\overline{CE}$ or $\overline{OE}$ High to Output Float	0	45	0	50	0	55	0	60	ns
t_{OH}	Output Hold from $\overline{OE}$, $\overline{CE}$ or Address, whichever occurred first	0		0		0		0		ns

AC Read Waveforms (1, 2, 3, 4)



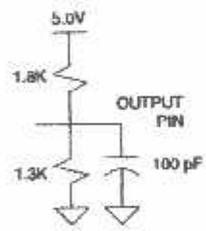
Notes:
1. $\overline{CE}$ may be delayed up to $t_{ACC} - t_{CE}$ after the address transition without impact on t_{ACC}.
2. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE} or by $t_{ACC} - t_{OE}$ after an address change without impact on t_{ACC}.
3. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first (C_L = 5 pF).
4. This parameter is characterized and is not 100% tested.

Input Test Waveforms and Measurement Level



t_R, t_F < 20 ns

Output Test Load



Pin Capacitance (f = 1 MHz, T = 25°C) (1)

	Typ	Max	Units	Conditions
C_{IN}	4	6	pF	V_{IN} = 0V
C_{OUT}	8	12	pF	V_{OUT} = 0V

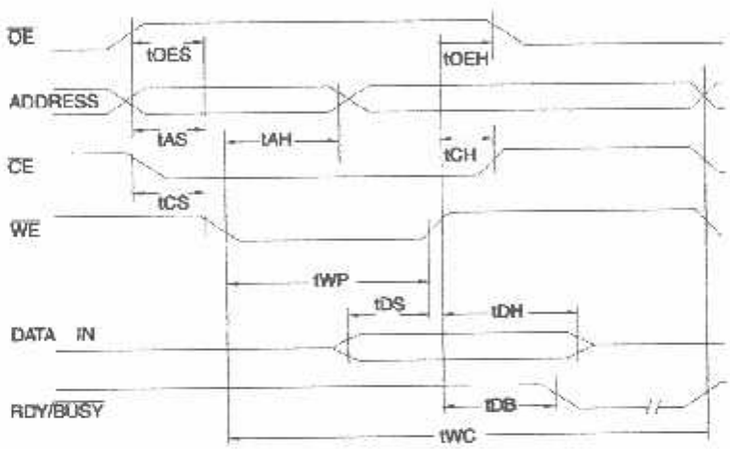
Note: 1. This parameter is characterized and is not 100% tested.

AC Write Characteristics

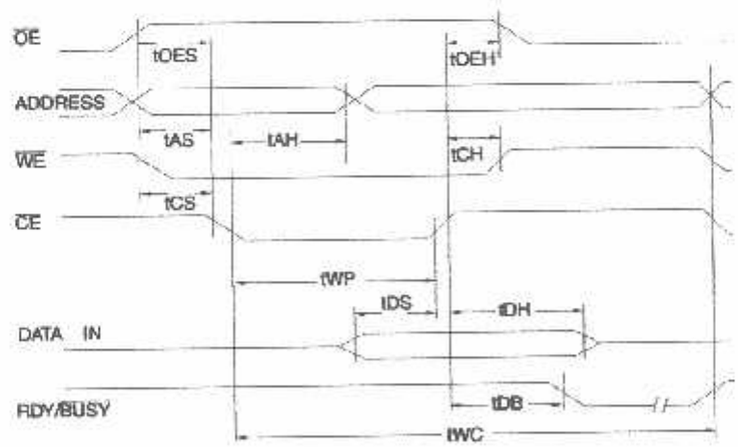
Symbol	Parameter		Min	Max	Units
t_{AS}, t_{OES}	Address, $\overline{OE}$ Set-up Time		10		ns
t_{AH}	Address Hold Time		50		ns
t_{WP}	Write Pulse Width ($\overline{WE}$ or $\overline{CE}$)		100	1000	ns
t_{DS}	Data Set-up Time		50		ns
t_{DH}, t_{OEH}	Data, $\overline{OE}$ Hold Time		10		ns
t_{CS}, t_{CH}	$\overline{CE}$ to $\overline{WE}$ and $\overline{WE}$ to $\overline{CE}$ Set-up and Hold Time		0		ns
t_{DB}	Time to Device Busy			50	ns
t_{WC}	Write Cycle Time	AT28C64		1.0	ms
		AT28C64E		200	µs

AC Write Waveforms

$\overline{WE}$ Controlled



$\overline{CE}$ Controlled

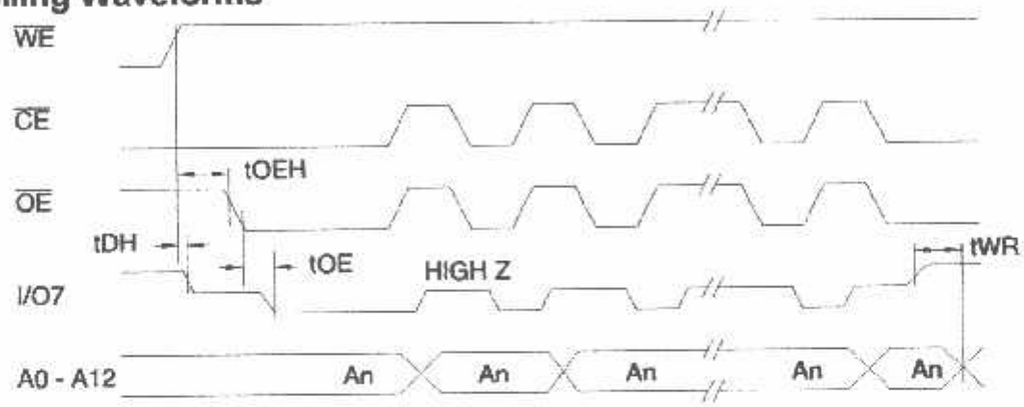


Data Polling Characteristics[1]

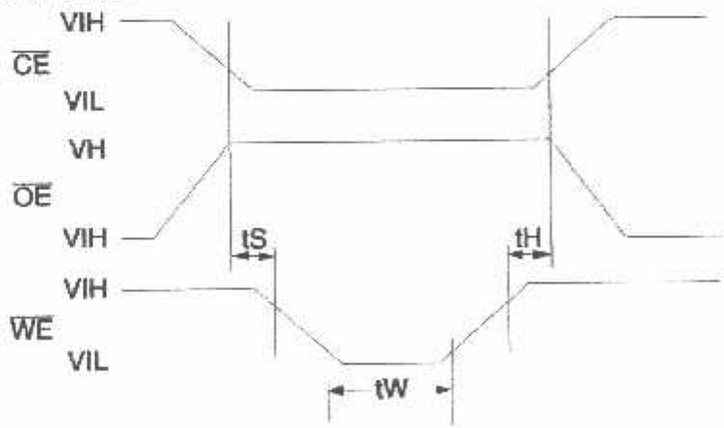
Symbol	Parameter	Min	Typ	Max	Units
t_{DH}	Data Hold Time	10			ns
t_{OEH}	$\overline{OE}$ Hold Time	10			ns
t_{OE}	$\overline{OE}$ to Output Delay[2]				ns
t_{WR}	Write Recovery Time	0			ns

Notes: 1. These parameters are characterized and not 100% tested.
2. See AC Read Characteristics.

Data Polling Waveforms

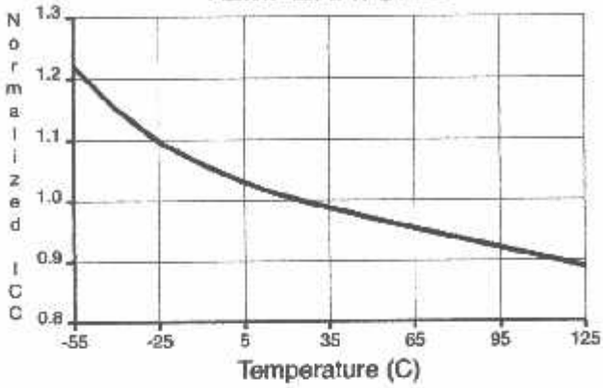


Chip Erase Waveforms

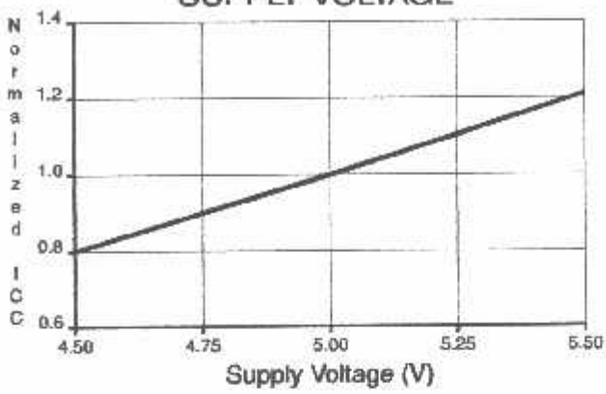


$t_S = t_H = 1\ \mu sec$ (min.)
$t_W = 10$ msec (min.)
$V_H = 12.0V \pm 0.5V$

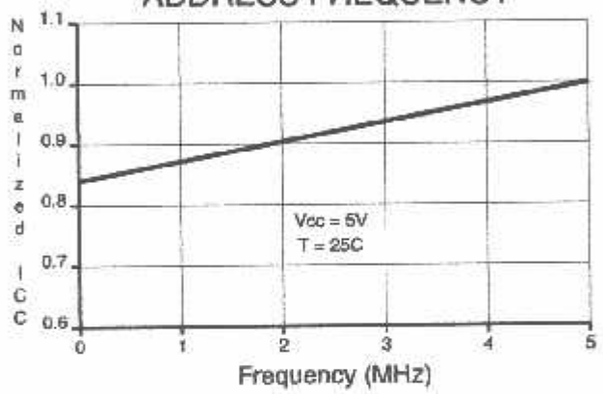
NORMALIZED SUPPLY CURRENT vs. TEMPERATURE



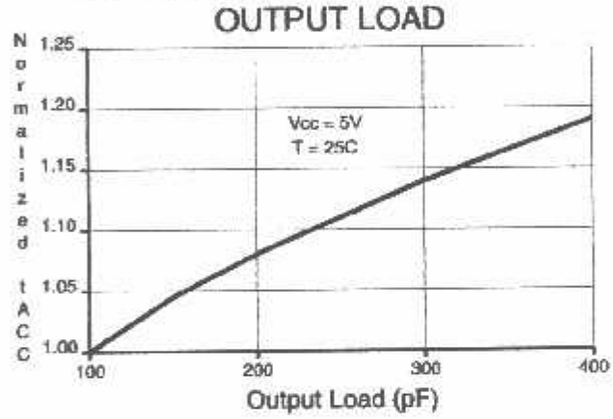
NORMALIZED SUPPLY CURRENT vs. SUPPLY VOLTAGE



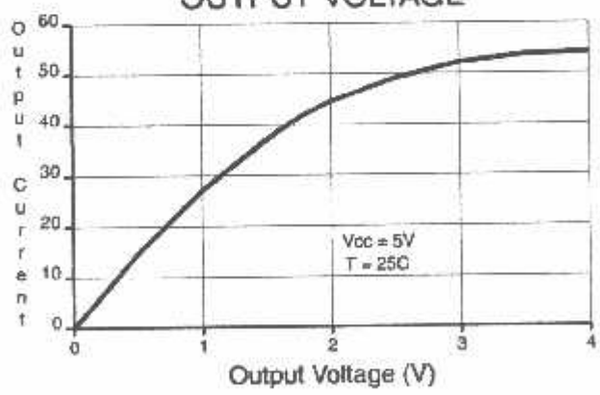
NORMALIZED SUPPLY CURRENT vs. ADDRESS FREQUENCY



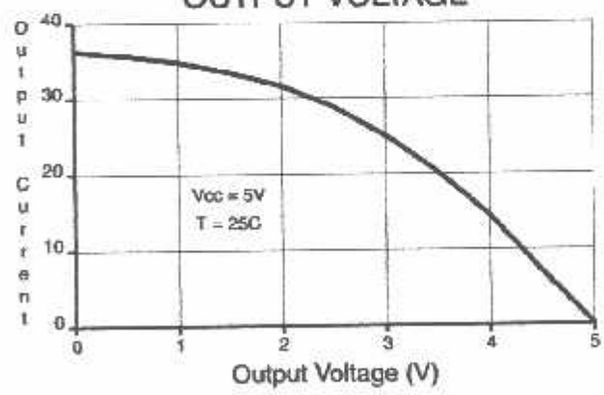
NORMALIZED ACCESS TIME vs. OUTPUT LOAD



OUTPUT SINK CURRENT vs. OUTPUT VOLTAGE



OUTPUT SOURCE CURRENT vs. OUTPUT VOLTAGE



Ordering Information [1]

t_{ACC} (ns)	I_{CC} (mA)		Ordering Code	Package	Operation Range
	Active	Standby			
120	30	0.1	AT28C64(E)-12JC AT28C64(E)-12PC AT28C64(E)-12SC AT28C64(E)-12TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64(E)-12JI AT28C64(E)-12PI AT28C64(E)-12SI AT28C64(E)-12TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)
150	30	0.1	AT28C64(E)-15JC AT28C64(E)-15PC AT28C64(E)-15SC AT28C64(E)-15TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64(E)-15JI AT28C64(E)-15PI AT28C64(E)-15SI AT28C64(E)-15TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)
200	30	0.1	AT28C64(E)-20JC AT28C64(E)-20PC AT28C64(E)-20SC AT28C64(E)-20TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64(E)-20JI AT28C64(E)-20PI AT28C64(E)-20SI AT28C64(E)-20TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)
250	30	0.1	AT28C64(E)-25JC AT28C64(E)-25PC AT28C64(E)-25SC AT28C64(E)-25TC AT28C64-W	32J 28P6 28S 28T DIE	Commercial (0°C to 70°C)
	45	0.1	AT28C64(E)-25JI AT28C64(E)-25PI AT28C64(E)-25SI AT28C64(E)-25TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)

Note: 1. See Valid Part Number table below.

Package Type	
32J	32 Lead, Plastic J-Leaded Chip Carrier (PLCC)
28P6	28 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
28S	28 Lead, 0.300" Wide, Plastic Gull Wing, Small Outline (SOIC)
28T	28 Lead, Plastic Thin Small Outline Package (TSOP)
W	Die
Options	
Blank	Standard Device: Endurance = 10K Write Cycles; Write Time = 1 ms
E	High Endurance Option: Endurance = 100K Write Cycles; Write Time = 200 µs

Ordering Information

t_{ACC} (ns)	I_{CC} (mA)		Ordering Code	Package	Operation Range
	Active	Standby			
150	30	0.1	AT28C64X-15JC AT28C64X-15PC AT28C64X-15SC AT28C64X-15TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64X-15JI AT28C64X-15PI AT28C64X-15SI AT28C64X-15TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)
200	30	0.1	AT28C64X-20JC AT28C64X-20PC AT28C64X-20SC AT28C64X-20TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64X-20JI AT28C64X-20PI AT28C64X-20SI AT28C64X-20TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)
250	30	0.1	AT28C64X-25JC AT28C64X-25PC AT28C64X-25SC AT28C64X-25TC	32J 28P6 28S 28T	Commercial (0°C to 70°C)
	45	0.1	AT28C64X-25JI AT28C64X-25PI AT28C64X-25SI AT28C64X-25TI	32J 28P6 28S 28T	Industrial (-40°C to 85°C)

Valid Part Numbers

The following table lists standard Atmel products that can be ordered.

Device Numbers	Speed	Package and Temperature Combinations
AT28C64 X	12	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64 X	15	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64 X	20	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64 X	25	JC, JI, PC, PI, SC, SI, TC, TI

Package Type	
32J	32 Lead, Plastic J-Leaded Chip Carrier (PLCC)
28P6	28 Lead, 0.600" Wide Plastic Dual Inline Package (PDIP)
28S	28 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
28T	28 Lead, Plastic Thin Small Outline Package (TSOP)

LM 386

August 2000

National Semiconductor

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.

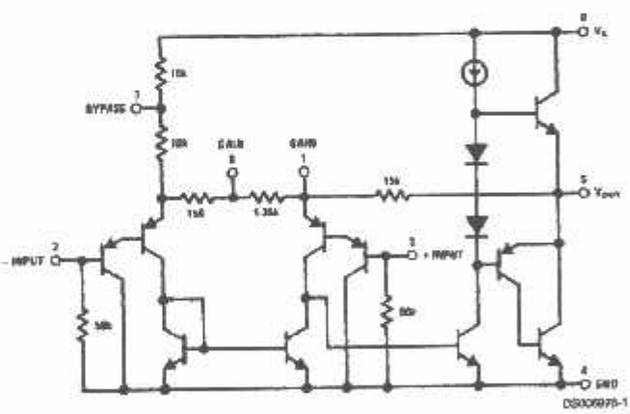
Features

- 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 = 8Ω, P_O = 125mW, f = 1kHz)
- Available in 8 pin MSOP package

Applications

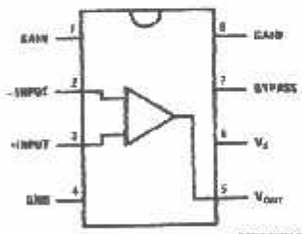
- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams



DS006976-1

Small Outline, Molded Mini Small Outline, and Dual-In-Line Packages



DS006976-2

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°C

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage (V_S)					
LM386N-1, -3, LM386M-1, LM386MM-1		4		12	V
LM386N-4		5		18	V
Quiescent Current (I_Q)	V_S = 6V, V_{IN} = 0		4	8	mA
Output Power (P_{OUT})					
LM386N-1, LM386M-1, LM386MM-1	V_S = 6V, R_L = 8Ω, THD = 10%	250	325		mW
LM386N-3	V_S = 9V, R_L = 8Ω, THD = 10%	500	700		mW
LM386N-4	V_S = 16V, R_L = 32Ω, THD = 10%	700	1000		mW
Voltage Gain (A_V)	V_S = 6V, f = 1 kHz		26		dB
	10 µF from Pin 1 to 8		46		dB
Bandwidth (BW)	V_S = 6V, Pins 1 and 8 Open		300		kHz
Total Harmonic Distortion (THD)	V_S = 6V, R_L = 8Ω, P_{OUT} = 125 mW		0.2		%
	f = 1 kHz, Pins 1 and 8 Open				
Power Supply Rejection Ratio (PSRR)	V_S = 6V, f = 1 kHz, C_{BYPASS} = 10 µF		50		dB
	Pins 1 and 8 Open, Referred to Output				
Input Resistance (R_{IN})			50		kΩ
Input Bias Current (I_{BIAS})	V_S = 6V, 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 kΩ resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35 kΩ 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 kΩ resistor). For 6 dB effective bass boost: R = 15 kΩ, the lowest value for good stable operation is R = 10 kΩ if pin 8 is open. If pins 1 and 8 are bypassed then R as low as 2 kΩ can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

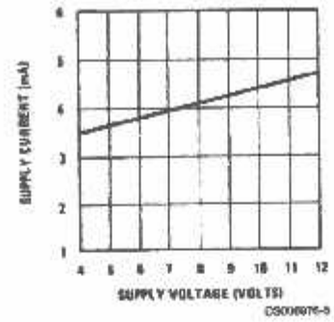
INPUT BIASING

The schematic shows that both inputs are biased to ground with a 50 kΩ 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 kΩ 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 kΩ, 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.

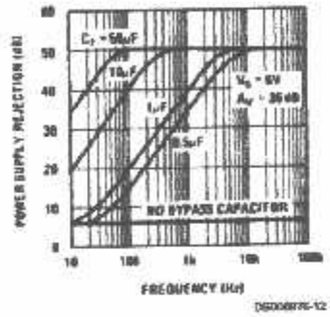
When using the LM386 with higher gains (bypassing the 1.35 kΩ resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1 µF capacitor or a short to ground depending on the dc source resistance on the driven input.

Typical Performance Characteristics

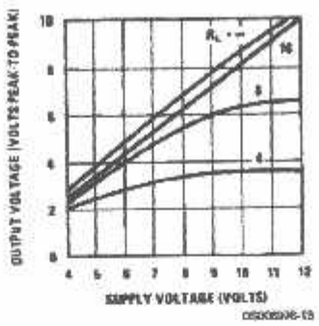
Quiescent Supply Current vs Supply Voltage



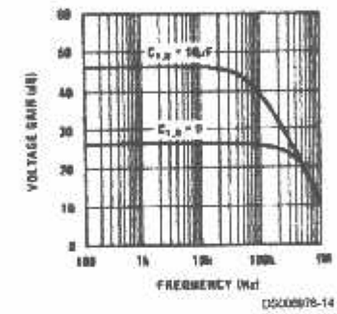
Power Supply Rejection Ratio (Referred to the Output) vs Frequency



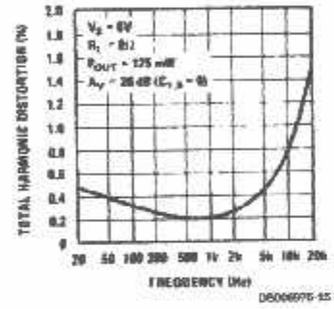
Peak-to-Peak Output Voltage Swing vs Supply Voltage



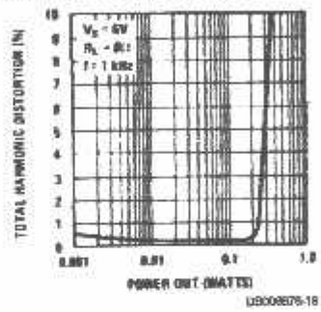
Voltage Gain vs Frequency



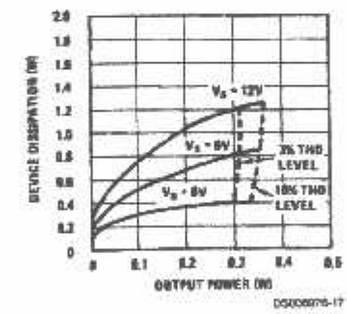
Distortion vs Frequency



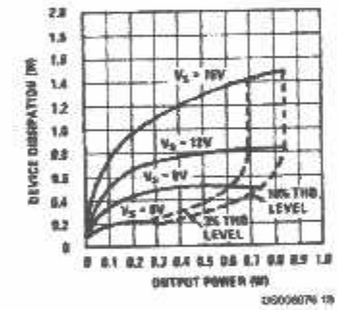
Distortion vs Output Power



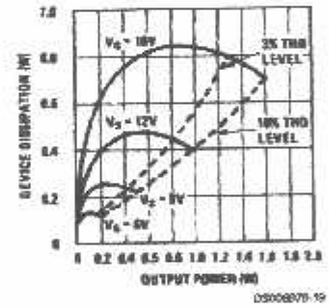
Device Dissipation vs Output Power—4Ω Load



Device Dissipation vs Output Power—8Ω Load

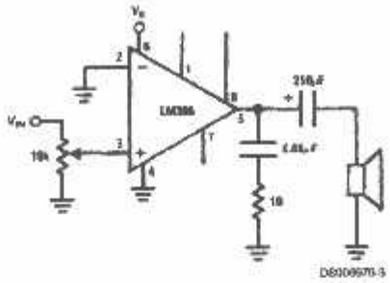


Device Dissipation vs Output Power—16Ω Load

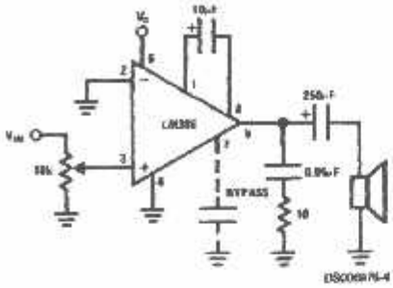


Typical Applications

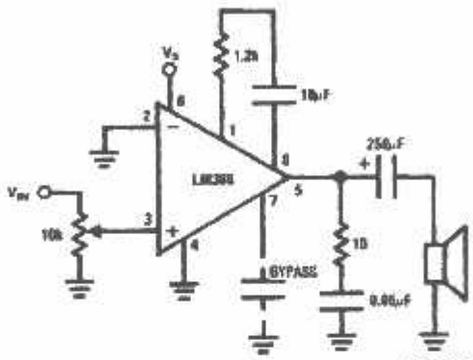
Amplifier with Gain = 20
Minimum Parts



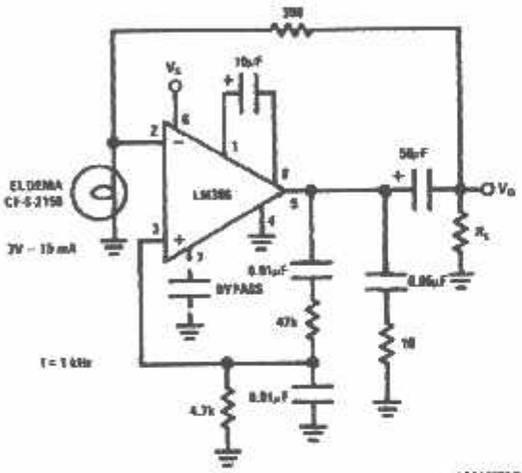
Amplifier with Gain = 200



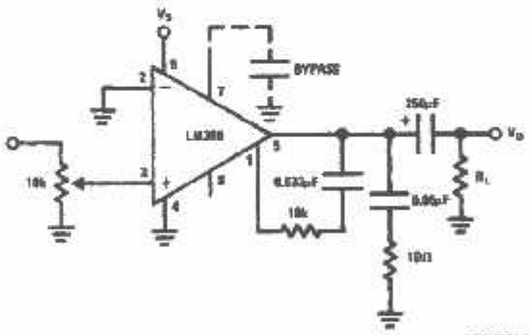
Amplifier with Gain = 50



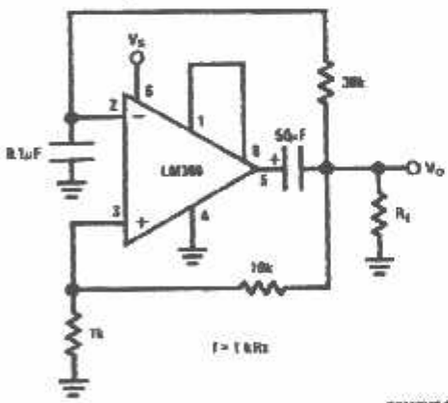
Low Distortion Power Wienbridge Oscillator



Amplifier with Bass Boost

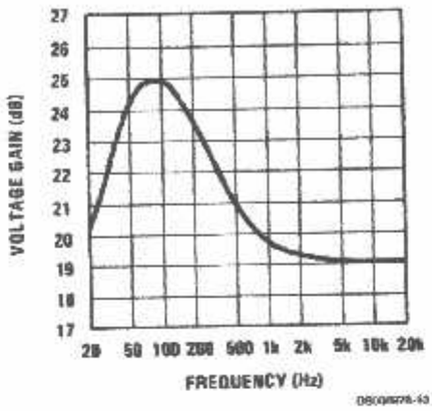


Square Wave Oscillator

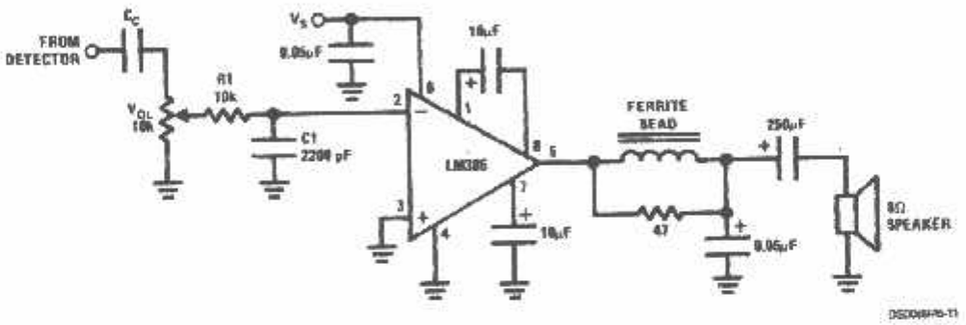


Typical Applications (Continued)

Frequency Response with Bass Boost



AM Radio Power Amplifier



Note 4: Twist Supply lead and supply ground very tightly.

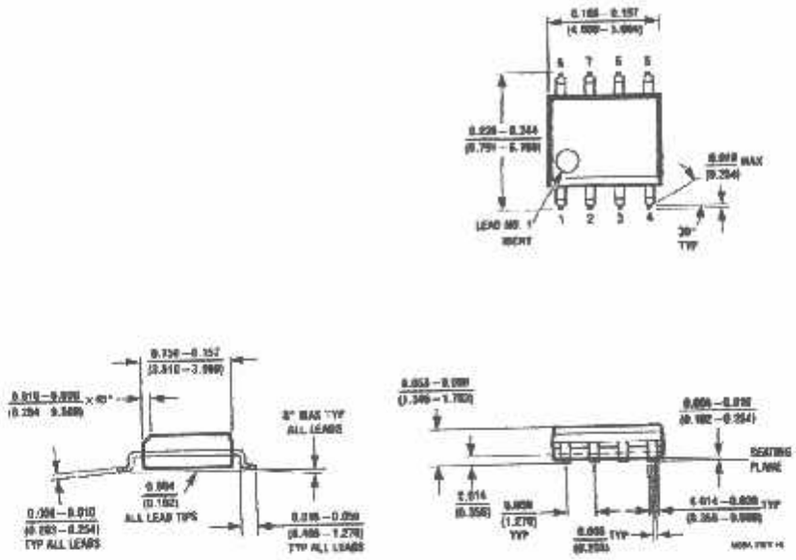
Note 5: Twist speaker lead and ground very tightly.

Note 6: Ferrite bead in Ferroxcube K5-001-001/3B with 3 turns of wire.

Note 7: R1C1 band limits input signals.

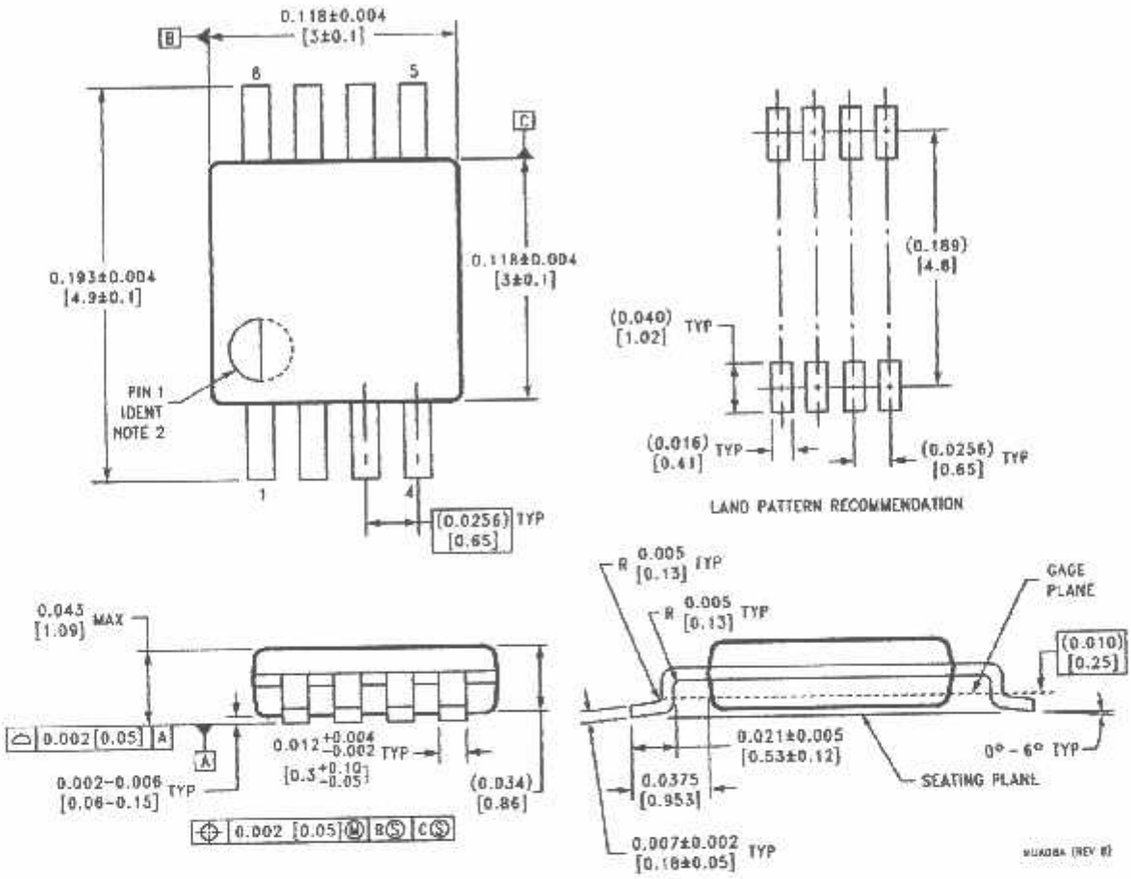
Note 8: All components must be spaced very closely to IC.

Physical Dimensions inches (millimeters) unless otherwise noted



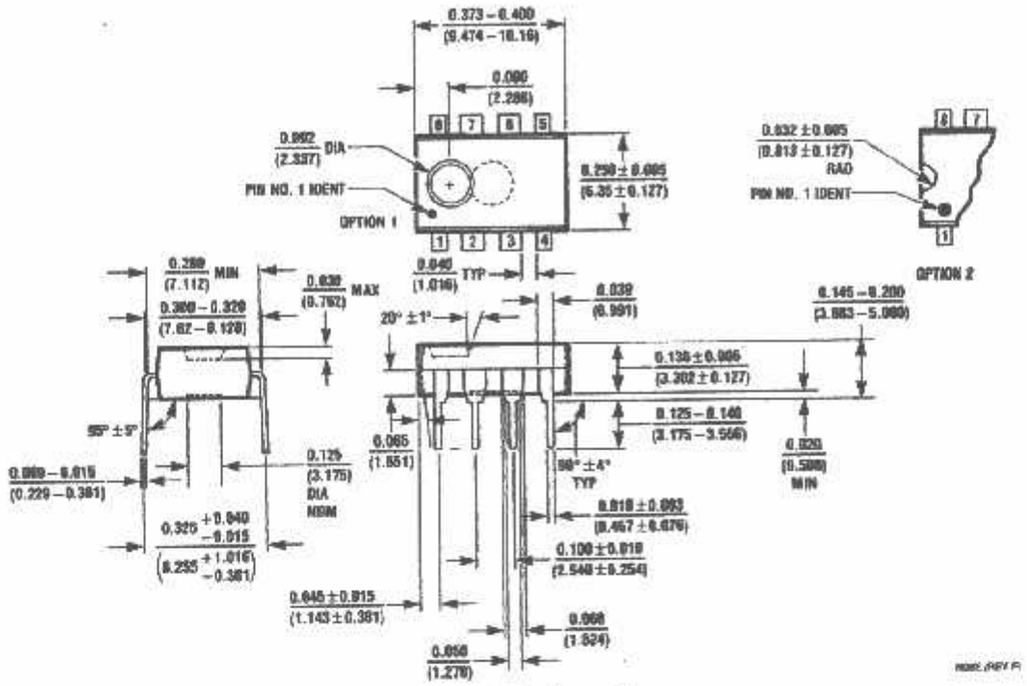
SO Package (M)
Order Number LM386M-1
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



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

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

74HCT573

HARRIS SEMICONDUCTOR

CD74HC373, CD74HCT373, CD54HC573, CD74HC573, CD74HCT573

High Speed CMOS Logic
Octal Transparent Latch, Three-State Output

November 1997

Features

- Common Latch Enable Control
- Common Three-State Output Enable Control
- Buffered Inputs
- Three-State Outputs
- Bus Line Driving Capacity
- Typical Propagation Delay = 12ns at V_{CC} = 5V, C_L = 15pF, T_A = 25°C (Data to Output for HC373)
- Fanout (Over Temperature Range)
  - Standard Outputs . . . . . 10 LSTTL Loads
  - Bus Driver Outputs . . . . . 15 LSTTL Loads
- Wide Operating Temperature Range . . . -55°C to 125°C
- Balanced Propagation Delay and Transition Times
- Significant Power Reduction Compared to LSTTL Logic ICs
- HC Types
  - 2V to 6V Operation
  - High Noise Immunity: N_{IL} = 30%, N_{IH} = 30% of V_{CC} at V_{CC} = 5V
- HCT Types
  - 4.5V to 5.5V Operation
  - Direct LSTTL Input Logic Compatibility, V_{IL} = 0.8V (Max), V_{IH} = 2V (Min)
  - CMOS Input Compatibility, $I_I \leq 1\mu A$ at V_{OL}, V_{OH}

Description

The Harris CD74HC373, CD74HCT373, CD54HC573, CD74HC573, and CD74HCT573 are high speed Octal Transparent Latches manufactured with silicon gate CMOS technology. They possess the low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LSTTL devices. The CD74HCT373 and CD74HCT573 are functionally as well as pin compatible with the standard 74LS373 and 74LS573.

The outputs are transparent to the inputs when the latch enable ($\overline{LE}$) is high. When the latch enable ($\overline{LE}$) goes low the data is latched. The output enable ($\overline{OE}$) controls the three-state outputs. When the output enable ($\overline{OE}$) is high the outputs are in the high impedance state. The latch operation is independent to the state of the output enable. The 373 and 573 are identical in function and differ only in their pinout arrangements.

Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
CD54HC573F	-55 to 125	20 Ld CERDIP	F20.3
CD74HC373E	-55 to 125	20 Ld PDIP	F20.3
CD74HCT373E	-55 to 125	20 Ld PDIP	E20.3
CD74HC573E	-55 to 125	20 Ld PDIP	E20.3
CD74HCT573E	-55 to 125	20 Ld PDIP	E20.3
CD74HC373M	-55 to 125	20 Ld SOIC	M20.3
CD74HCT373M	-55 to 125	20 Ld SOIC	M20.3
CD74HC573M	-55 to 125	20 Ld SOIC	M20.3
CD74HCT573M	-55 to 125	20 Ld SOIC	M20.3

NOTES:

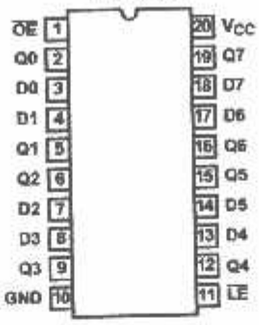
1. When ordering, use the entire part number. Add the suffix 96 to obtain the variant in the tape and reel.
2. Wafer or die for this part number are available which meets all electrical specifications. Please contact your local sales office or Harris customer service for ordering information.

CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

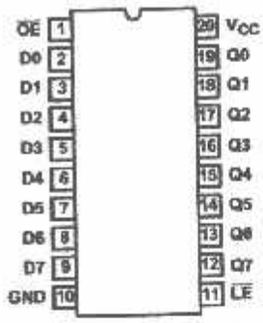
File Number 1679.1

Pinout

CD74HC373, CD74HCT373
(PDIP, SOIC)
TOP VIEW

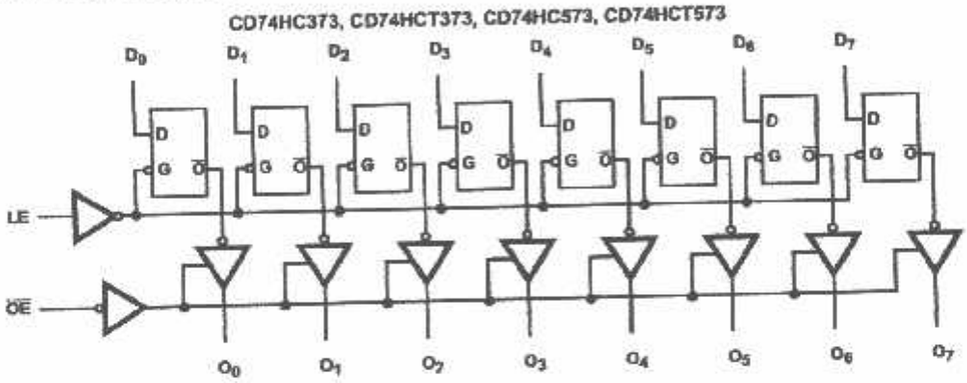


CD54HC573, CD74HC573, CD74HCT573
(PDIP, SOIC, CERDIP)
TOP VIEW

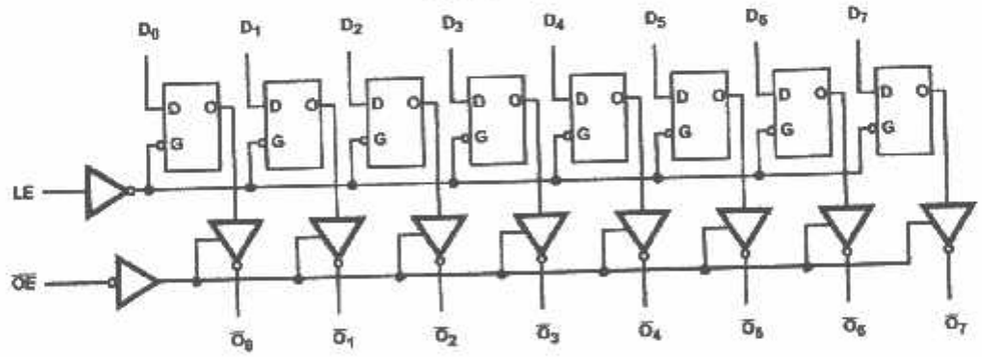


Functional Block Diagrams

CD74HC373, CD74HCT373, CD74HC573, CD74HCT573



CD74HCT573



TRUTH TABLE

OUTPUT ENABLE	LATCH ENABLE	DATA	OUTPUT
L	H	H	H
L	H	L	L
L	L	l	L
L	L	h	H
H	X	X	Z

NOTE: H = High Voltage Level, L = Low Voltage Level, X = Don't Care, Z = High Impedance State, l = Low voltage level one set-up time prior to the high to low latch enable transition, h = High voltage level one set-up time prior to the high to low latch enable transition.

Absolute Maximum Ratings

DC Supply Voltage, V_{CC} ..... -0.5V to 7V
DC Input Diode Current, I_{IK}
For V_I < -0.5V or V_I > V_{CC} + 0.5V ..... ±20mA
DC Output Diode Current, I_{OK}
For V_O < -0.5V or V_O > V_{CC} + 0.5V ..... ±20mA
DC Drain Current, per Output, I_O
For -0.5V < V_O < V_{CC} + 0.5V ..... ±35mA
DC Output Source or Sink Current per Output Pin, I_O
For V_O > -0.5V or V_O < V_{CC} + 0.5V ..... ±25mA
DC V_{CC} or Ground Current, I_{CC} ..... ±50mA

Operating Conditions

Temperature Range, T_A ..... -55°C to 125°C
Supply Voltage Range, V_{CC}
HC Types ..... 2V to 6V
HCT Types ..... 4.5V to 5.5V
DC Input or Output Voltage, V_I, V_O ..... 0V to V_{CC}
Input Rise and Fall Time
2V ..... 1000ns (Max)
4.5V ..... 500ns (Max)
6V ..... 400ns (Max)

Thermal Information

Thermal Resistance (Typical, Note 3)	θ_{JA} (°C/W)	θ_{JA} (°C/W)
PDIP Package	125	N/A
CERDIP Package	85	24
SOIC Package	120	N/A

Maximum Junction Temperature (Plastic Package) ..... 150°C
Maximum Storage Temperature Range ..... -65°C to 150°C
Maximum Lead Temperature (Soldering 10s) ..... 300°C
(SOIC - Lead Tips Only)

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

3. θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

DC Electrical Specifications

PARAMETER	SYMBOL	TEST CONDITIONS		V_{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
		V_I (V)	I_O (mA)		MIN	TYP	MAX	MIN	MAX	MIN	MAX	
HC TYPES												
High Level Input Voltage	V_{IH}	-	-	2	1.5	-	-	1.5	-	1.5	-	V
				4.5	3.15	-	-	3.15	-	3.15	-	V
				6	4.2	-	-	4.2	-	4.2	-	V
Low Level Input Voltage	V_{IL}	-	-	2	-	-	0.5	-	0.5	-	0.5	V
				4.5	-	-	1.35	-	1.35	-	1.35	V
				6	-	-	1.8	-	1.8	-	1.8	V
High Level Output Voltage CMOS Loads	V_{OH}	V_{IH} or V_{IL}	-0.02	2	1.9	-	-	1.9	-	1.9	-	V
			-0.02	4.5	4.4	-	-	4.4	-	4.4	-	V
			-0.02	6	5.9	-	-	5.9	-	5.9	-	V
High Level Output Voltage TTL Loads			-6	4.5	3.98	-	-	3.84	-	3.7	-	V
			-7.8	6	5.48	-	-	5.34	-	5.2	-	V
Low Level Output Voltage CMOS Loads	V_{OL}	V_{IH} or V_{IL}	0.02	2	-	-	0.1	-	0.1	-	0.1	V
			0.02	4.5	-	-	0.1	-	0.1	-	0.1	V
			0.02	6	-	-	0.1	-	0.1	-	0.1	V
Low Level Output Voltage TTL Loads			6	4.5	-	-	0.26	-	0.33	-	0.4	V
			7.8	6	-	-	0.26	-	0.33	-	0.4	V
Input Leakage Current	I_I	V_{CC} or GND	-	6	-	-	±0.1	-	±1	-	±1	µA
Quiescent Device Current	I_{CC}	V_{CC} or GND	0	6	-	-	8	-	80	-	160	µA

DC Electrical Specifications (Continued)

PARAMETER	SYMBOL	TEST CONDITIONS		V_{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
		V_I (V)	I_O (mA)		MIN	TYP	MAX	MIN	MAX	MIN	MAX	
Three-State Leakage Current	-	V_{IL} or V_{IH}	V_O = V_{CC} or GND	6	-	-	±0.5	-	±5	-	±10	µA
HCT TYPES												
High Level Input Voltage	V_{IH}	-	-	4.5 to 5.5	2	-	-	2	-	2	-	V
Low Level Input Voltage	V_{IL}	-	-	4.5 to 5.5	-	-	0.8	-	0.8	-	0.8	V
High Level Output Voltage CMOS Loads	V_{OH}	V_{IH} or V_{IL}	-0.02	4.5	4.4	-	-	4.4	-	4.4	-	V
High Level Output Voltage TTL Loads			-6	4.5	3.98	-	-	3.84	-	3.7	-	V
			-7.8	6	5.48	-	-	5.34	-	5.2	-	V
Low Level Output Voltage CMOS Loads	V_{OL}	V_{IH} or V_{IL}	0.02	4.5	-	-	0.1	-	0.1	-	0.1	V
Low Level Output Voltage TTL Loads			6	4.5	-	-	0.26	-	0.33	-	0.4	V
			7.8	6	-	-	0.26	-	0.33	-	0.4	V
Input Leakage Current	I_I	V_{CC} to GND	-	5.5	-	-	±0.1	-	±1	-	±1	µA
Quiescent Device Current	I_{CC}	V_{CC} or GND	0	5.5	-	-	8	-	80	-	160	µA
Three-State Leakage Current	-	V_{IL} or V_{IH}	V_O = V_{CC} or GND	6	-	-	±0.5	-	±5	-	±10	µA
Additional Quiescent Device Current Per Input Pin: 1 Unit Load (Note 4)	ΔI_{CC}	V_{CC} -2.1	-	4.5 to 5.5	-	100	360	-	450	-	490	µA

NOTE:

4. For dual-supply systems theoretical worst case (V_I = 2.4V, V_{CC} = 5.5V) specification is 1.8mA.

HCT Input Loading Table

INPUT	UNIT LOADS	
	HCT373	HCT573
$\overline{OE}$	1.5	1.25
Dn	0.4	0.3
$\overline{LE}$	0.6	0.65

NOTE: Unit Load is ΔI_{CC} limit specified in DC Electrical Specifications table, e.g., 360µA max at 25°C.

Prerequisite For Switching Specifications

PARAMETER	SYMBOL	TEST CONDITIONS	V_{CC} (V)	25°C			-40°C TO 85°C		-55°C TO 125°C		UNITS
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
HC TYPES											
$\overline{LE}$ Pulse Width	t_W	-	2	80	-	-	100	-	120	-	ns
			4.5	16	-	-	20	-	24	-	ns
			6	14	-	-	17	-	20	-	ns
Set-up Time Data to $\overline{LE}$	t_{SU}	-	2	50	-	-	85	-	75	-	ns
			4.5	10	-	-	13	-	15	-	ns
			6	9	-	-	11	-	13	-	ns
Hold Time, Data to $\overline{LE}$ (573)	t_H	-	2	40	-	-	50	-	60	-	ns
			4.5	8	-	-	10	-	12	-	ns
			6	7	-	-	9	-	10	-	ns
Hold Time, Data to $\overline{LE}$ (373)	t_H	-	2	5	-	-	5	-	5	-	ns
			4.5	5	-	-	5	-	5	-	ns
			6	5	-	-	5	-	5	-	ns
HCT TYPES											
$\overline{LE}$ Pulse Width	t_W	-	4.5	16	-	-	20	-	24	-	ns
Set-up Time Data to $\overline{LE}$	t_W	-	4.5	13	-	-	16	-	20	-	ns
Hold Time, Data to $\overline{LE}$	t_H	-	4.5	10	-	-	13	-	15	-	ns

Switching Specifications Input t_r, t_f = 6ns

PARAMETER	SYMBOL	TEST CONDITIONS	V_{CC} (V)	25°C		-40°C TO 85°C	-55°C TO 125°C	UNITS
				TYP	MAX	MAX	MAX	
HC TYPES								
Propagation Delay, Data to Qn (HC/HCT373)	t_{PLH}, t_{PHL}	C_L = 50pF	2	-	150	190	225	ns
			4.5	-	30	38	45	ns
			6	-	26	33	38	ns
		C_L = 15pF	5	12	-	-	-	ns
Propagation Delay, Data to Qn (HC/HCT573)	t_{PLH}, t_{PHL}	C_L = 50pF	2	-	175	220	265	ns
			4.5	-	35	44	53	ns
			6	-	30	37	45	ns
		C_L = 15pF	5	14	-	-	-	ns
Propagation Delay, $\overline{LE}$ to Qn	t_{PLH}, t_{PHL}	C_L = 50pF	2	-	175	220	265	ns
			4.5	-	35	44	53	ns
			6	-	30	37	45	ns
		C_L = 15pF	5	14	-	-	-	ns
Output Enabling Time	t_{PZL}, t_{PZH}	C_L = 50pF	2	-	150	190	225	ns
			4.5	-	30	38	45	ns
			6	-	26	33	38	ns
		C_L = 15pF	5	12	-	-	-	ns

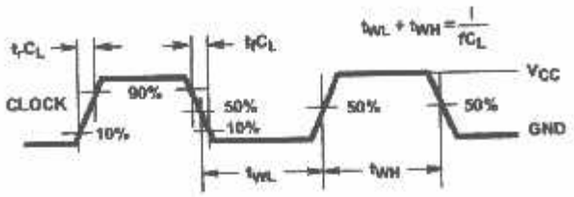
Switching Specifications Input t_r, t_f = 6ns **(Continued)**

PARAMETER	SYMBOL	TEST CONDITIONS	V_{CC} (V)	25°C		-40°C TO 85°C	-55°C TO 125°C	UNITS
				TYP	MAX	MAX	MAX	
Output Disabling Time	t_{PLZ}, t_{PHZ}	C_L = 50pF	2	-	150	190	225	ns
			4.5	-	30	38	45	ns
			6	-	26	33	38	ns
		C_L = 15pF	5	12	-	-	-	ns
Output Transition Time	t_{TLH}, t_{THL}	C_L = 50pF	2	-	60	75	90	ns
			4.5	-	12	15	18	ns
			6	-	10	13	15	ns
Input Capacitance	C_I	-	-	-	10	10	10	pF
Three-State Output Capacitance	C_O	-	-	-	20	20	20	pF
Power Dissipation Capacitance (Notes 5, 6)	C_{PD}	-	5	51	-	-	-	pF
HCT TYPES								
Propagation Delay, Data to Qn (HC/HCT373)	t_{PLH}, t_{PHL}	C_L = 50pF	4.5	-	32	40	48	ns
		C_L = 15pF	5	13	-	-	-	ns
Propagation Delay, Data to Qn (HC/HCT573)	t_{PLH}, t_{PHL}	C_L = 50pF	4.5	-	35	44	53	ns
		C_L = 15pF	5	17	-	-	-	ns
Propagation Delay, LE to Qn	t_{PLH}, t_{PHL}	C_L = 50pF	4.5	-	35	44	53	ns
		C_L = 15pF	5	14	-	-	-	ns
Output Enabling Time	t_{PZL}, t_{PZH}	C_L = 50pF	4.5	-	35	44	53	ns
		C_L = 15pF	5	14	-	-	-	ns
Output Disabling Time	t_{PLZ}, t_{PZH}	C_L = 50pF	4.5	-	35	44	53	ns
		C_L = 15pF	5	14	-	-	-	ns
Output Transition Time	t_{TLH}, t_{THL}	C_L = 50pF	4.5	-	12	15	18	ns
Input Capacitance	C_I	-	-	-	10	10	10	pF
Three-State Output Capacitance	C_O	-	-	-	20	20	20	pF
Power Dissipation Capacitance (Notes 5, 6)	C_{PD}	-	5	53	-	-	-	pF

NOTES:

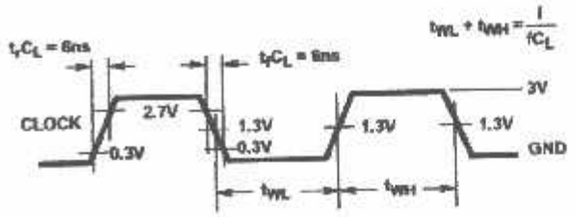
5. C_{PD} is used to determine the no-load dynamic power consumption, per latch.
6. P_D (total power per latch) = $V_{CC}^2 f_i (C_{PD} + C_L)$ where f_i = Input Frequency, C_L = Output Load Capacitance, V_{CC} = Supply Voltage.

Test Circuits and Waveforms



NOTE: Outputs should be switching from 10% V_{CC} to 90% V_{CC} in accordance with device truth table. For f_{MAX}, input duty cycle = 50%.

FIGURE 1. HC CLOCK PULSE RISE AND FALL TIMES AND PULSE WIDTH



NOTE: Outputs should be switching from 10% V_{CC} to 90% V_{CC} in accordance with device truth table. For f_{MAX}, input duty cycle = 50%.

FIGURE 2. HCT CLOCK PULSE RISE AND FALL TIMES AND PULSE WIDTH

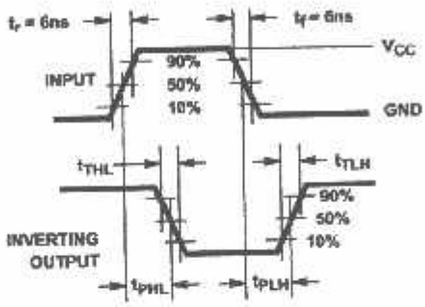


FIGURE 3. HC TRANSITION TIMES AND PROPAGATION DELAY TIMES, COMBINATION LOGIC

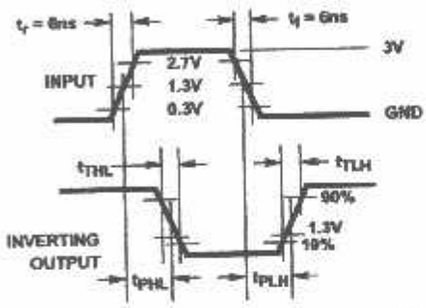


FIGURE 4. HCT TRANSITION TIMES AND PROPAGATION DELAY TIMES, COMBINATION LOGIC

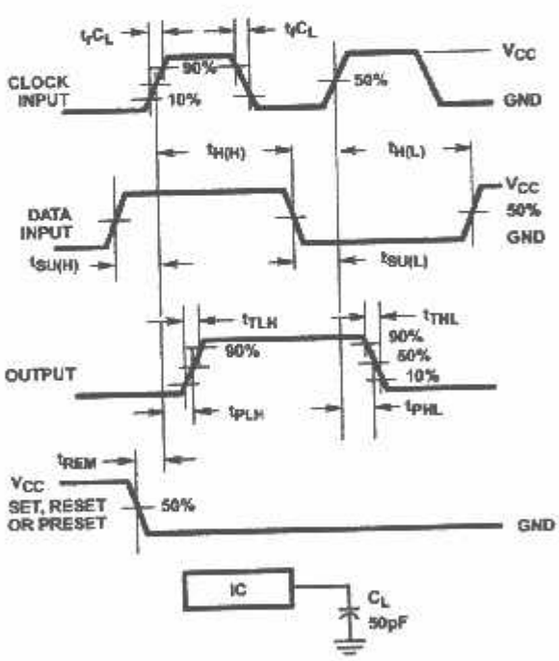


FIGURE 5. HC SETUP TIMES, HOLD TIMES, REMOVAL TIME, AND PROPAGATION DELAY TIMES FOR EDGE TRIGGERED SEQUENTIAL LOGIC CIRCUITS

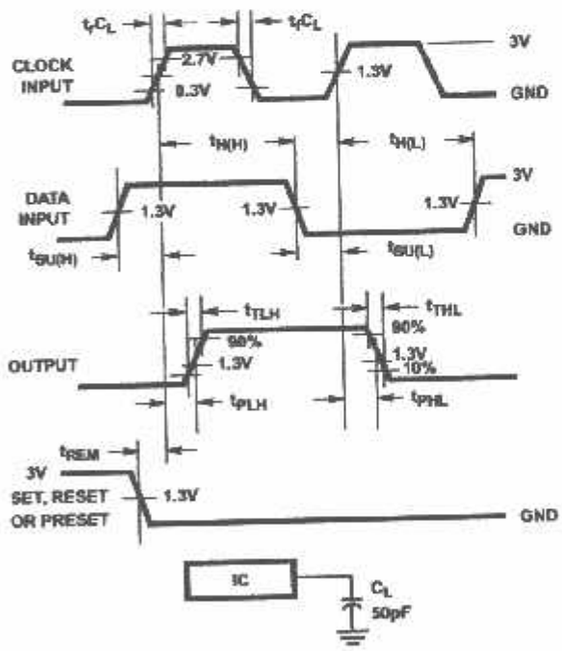


FIGURE 6. HCT SETUP TIMES, HOLD TIMES, REMOVAL TIME, AND PROPAGATION DELAY TIMES FOR EDGE TRIGGERED SEQUENTIAL LOGIC CIRCUITS

Test Circuits and Waveforms (Continued)

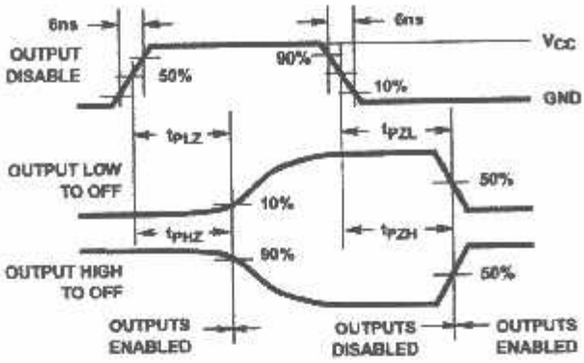


FIGURE 7. HC THREE-STATE PROPAGATION DELAY WAVEFORM

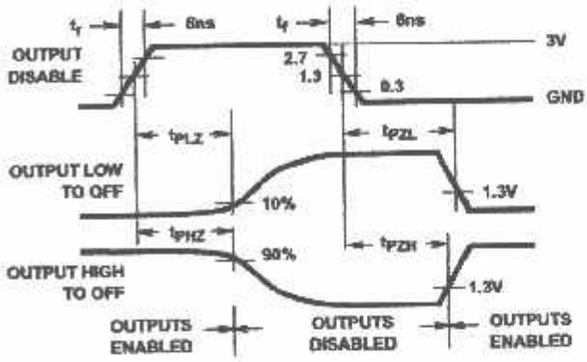
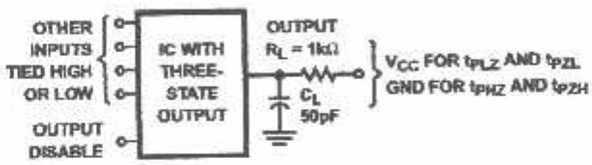


FIGURE 8. HCT THREE-STATE PROPAGATION DELAY WAVEFORM



NOTE: Open drain waveforms t_{PLZ} and t_{PZL} are the same as those for three-state shown on the left. The test circuit is Output R_L = 1kΩ to V_{CC}, C_L = 50pF.

FIGURE 9. HC AND HCT THREE-STATE PROPAGATION DELAY TEST CIRCUIT

Sales Office Headquarters

For general information regarding Harris Semiconductor and its products, call **1-800-4-HARRIS**

NORTH AMERICA
Harris Semiconductor
P. O. Box 883, Mail Stop 53-210
Melbourne, FL 32902
TEL: 1-800-442-7747
(407) 729-4984
FAX: (407) 729-5321

EUROPE
Harris Semiconductor
Mercure Center
100, Rue de la Fusee
1130 Brussels, Belgium
TEL: (32) 2.724.2111
FAX: (32) 2.724.22.05

ASIA
Harris Semiconductor PTE Ltd.
No. 1 Tannery Road
Cencon 1, #09-01
Singapore 1334
TEL: (65) 748-4200
FAX: (65) 748-0400

HARRIS SEMICONDUCTOR