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

PENCATAT DAFTAR HADIR DOSEN DAN PENAMPIL RUANG KULIAH DENGAN KOMPUTER DI KAMPUS ITN - MALANG

SKRIPSI

Disusun Oleh :

OKIS ARDHI PRATOMO LUKAS
99.17.035

APRIL 2005

LEMBAR PERSETUJUAN

PENCATAT DAFTAR HADIR DAN PENAMPIL RUANG KULIAH DENGAN KOMPUTER DI KAMPUS ITN - MALANG

SKRIPSI

Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat Guna Mencapai Gelar Sarjana Teknik

Disusun Oleh :

OKIS ARDHI PRATOMO LUKAS
NIM : 99.17.035

Menyetujui,

Dosen Pembimbing I	Dosen Pembimbing II
(Ir. Teguh Herbasuki, MT)	(I Komang Somawirata, ST)
NIP. 1038900209	NIP. 1030100361

Mengetahui,
Ketua Jurusan Teknik Elektro / Elektronika S-1

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

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

"Atau kamu berdoa juga, tetapi kamu tidak menerima apa – apa, karena kamu salah berdoa, sebab yang minta kamu itu hendak kamu habiskan untuk memuaskan hawa nafsumu"
(Yak 4:3)

Akan tetapi jikalau kamu menjalankan hukum utama yang tertulis dalam kitab suci "Kasihilah sesamamu manusia seperti dirimu sendiri", kamu berbuat baik
(Yak 2:8)

"Hidup adalah untuk berubah dan supaya sempurna harus sering berubah" (John Henry Newman)

Terimalah kenyataan bahwa tak seorang pun dapat sungguh memahami diri anda, begitu pula sebaliknya.

LEMBAR PERSEMBAHAN

Lembar ini Ku Persembahkan Sebagai Rasa Terima Kasih Kepada Teman – Teman Saudara dan Some One Special Yang Telah Mendukung Aku Selama Ini

Puji SyukurKepada Tuhan Yesus Kristus Dan Bunda Maria Perawan Sejati Atas Terkabulnya Doa - Doa Yang Selama Ini Aku Panjatkann kepada-Mu (khususnya Doa Novena 3x Salam Maria), Yang Udah Memberikan Kemudahan dan Kelancaran Dalam Menyelesaikan Skripsi ini..Amin
Terima Kasih Kepada Ayahanda (J.A Soekasno Bsc) dan Ibunda (F.Yuniati Srimulatsih, SPd) Tercinta Dengan Susah Payah Selalu Memberikan Dukungan Baik Dari Segi Materi Maupun Moral Dan Tak Lupa Atas Doa dan Restu Yang Selalu Mengantar Aku Dalam Menjalankan Skripsi ini Dengan Baik.
Terima Kasih Kepada Eyang Kakung (Alm. Soehadi PHP) dan Eyang Putri (Hj. Soefingah) Atas Doa, Nasehat dan Saran – Saran Yang Diberikan Kepadaku
Terima Kasih Buat Adik – Adikku Tersayang (Andreas, Bernardinus, dan Yustinus) , Sekarang Mas Lukas Jadi Insinyur Nih, mudah – mudahan kita selalu kompak dan selalu bersatu dalam satu ikatan Keluarga.
Anak-Anak Bendungan Sengguruh (sasana Park N Town) Matur Suwun Yooo atas Bantuannya, Smoga Rasa Kebersamaan Dan Kekeluargaan Selalu Terjaga.
Buat Sayangku Linda Herlistianti, Engkaulah Be Inspired And Spirit In My Heart, Smoga Tuhan Selalu Memberikan Jalan Terbaik Buat Kita Dalam Menempuh Hidup Yang Penuh Cobaan N Rintangan Nantinya
Gang Sempor 24 Choki, Dani , Wawan, Amin dan Karyo
Geng Poharin Ari, Ruslan, Dimas, Roni, dkk.. Sahabat-sahabatku Tita, Neni, Titik Unbraw, Diana UM, Eko (Kamto), Yudi (Makasih yoooo atas pinjem Printernya), Wisnu-Klaten, Dewi Prita,
Temen – temen seperjuangan ELKA : Natora, Wahid, Cahyo Aan, David, Lidya Dan Semuanya Yang Ga' Bisa Aku Sebutin Satu-Persatu.
Dan Aku Sadar Bahwa Ini Bukan Akhir Dari Perjuanganku, Masih Banyak Lagi Perjuangan Yang Lain Yang Penuh Tantangan Dan Rintangan.
Your Love Is Forever To Me☺

Success For You All
Love Will United Us Forever
And Ever

ABSTRAKSI

PENCATAT DAFTAR HADIR DOSEN DAN PENAMPIL RUANG KULIAH DENGAN KOMPUTER DI KAMPUS ITN MALANG

(Okis Ardhi Pratomo Lukas, NIM : 99.17.035, Teknik Elektronika S-1, 55 Halaman)
Pembimbing: Ir. Teguh Herbasuki, MT dan I Komang Somawirata, ST.
Insitut Teknologi Nasional Malang

Kata Kunci : *AT89S51*, EEPROM AT24C64, *Latch*, *Decoder*, LCD, Keypad

Teknologi dibuat dan diciptakan untuk membantu pekerjaan manusia atau untuk menjadi suatu pekerjaan menjadi mudah dan efisien. Penerapan teknlogi dimanfaatkan oleh semua bidang, salah satunya dalam judul Skripsi ini yaitu Pencatat Daftar Hadir Dosen dan Penampil Ruang Kuliah Dengan Komputer Di Kampus ITN Malang.

Untuk merancang dan membuat karya ini dapat dipakai beberapa macam teknologi salah satunya yaitu mikrokontroller, dalam perancangan alat digunakan keluarga ATMEL yaitu seri AT89S51 sebagai pengendalinya. Selain itu juga memakai EEPROM AT24C64 sebagai penyimpan nomor PIN, nama dosen dan NIP, keypad sebagai perintah untuk memasukkan nomor PIN dan LCD sebagai penampil kode PIN.

Prinsip kerja dari sistem ini adalah sebuah minimum sistem mikrokontroler AT89S51 beserta *chip card reader* (slot) yang dipasang setiap ruang kuliah untuk membaca *chip card*-nya setelah pengendali memerintahkan perintah " Masukkan PIN..." kemudian setelah PIN benar maka mikrokontroler bekerja. Hal itu pemasangan alat tersebut dipasang pada setiap ruang kuliah dan komputer sebagai penampil ruang kuliah yang diinterfaceckan dengan program Microsoft Visual Basic V6.0 melalui RS-232 ke RS-485. Apabila dosen itu menunda kuliah karena ada kesibukan di luar kampus maka si dosen harus memasukkan *chip card*-nya kembali dan dicatat pada perintah " Jam Masuk dan Jam keluar" pada program Visual Basic V6.0.

KATA PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Esa atas berkat bimbingan dan rahmat - Nya, sehingga penulis dapat menyelesaikan laporan Skripsi dengan judul:

" PENCATAT DAFTAR HADIR DOSEN DAN PENAMPIL RUANG KULIAH DENGAN KOMPUTER DI KAMPUS ITN – MALANG "

Pembuatan Skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Laporan Skripsi ini merupakan tanggung jawab tertulis atas ilmu pengetahuan yang didapat selama penyusun mengikuti kuliah.

Atas selesainya Skripsi ini, penulis mengucapkan terima kasih kepada :

- Bapak Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
- Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro Elektronika S - 1.
- Ibu Ir. Mimien Mustikawati, selaku Sekretaris Jurusan Teknik Elektro / Elektronika S - 1.
- Bapak Ir. Teguh Herbasuki, MT selaku Dosen Pembimbing I dan Bapak I Komang Somawirata, ST selaku dosen pembimbing II yang telah memberikan bimbingan, pengarahan, serta ilmu-ilmu yang sangat berharga sehingga Skripsi ini dapat terselesaikan

- Bapak dan Ibuku tercinta yang selalu mendoakan dan penuh kasih sayang memberi dorongan selama penulis menyelesaikan Skripsi ini.
- Adik - adikku tercinta yang telah memberi dukungan moril kepada penulis dalam menyelesaikan skripsi ini.
- Warga Bendungan sengguruh (Sasana PARK N TOWN) atas kekeluargaan dan kerjasamanya.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohan maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Maret 2005

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan pembangunan, teknologi elektronika mengalami berbagai kemajuan yang begitu cepat, menyebabkan manusia tidak akan lepas dari penggunaan berbagai macam peralatan elektronika yang ada, baik itu penggunaan peralatan yang menggunakan perangkat keras maupun perangkat lunak elektronika. Pada perkembangannya teknologi elektronika menuntut manusia untuk merekayasa suatu peralatan elektronika yang praktis.

Dari uraian diatas salah satu peralatan elektronika yang berkembang dewasa ini banyak yang menggunakan perangkat lunak yang akan menjalankan suatu peralatan elektronika. Dengan menggunakan peralatan yang diprogram oleh suatu perangkat lunak, maka akan diperoleh kemudahan dalam pembuatan peralatan elektronika yang akan dirancang. Sebagai wujud dari penggunakan perangkat elektronika adalah digunakan sebagai suatu sistem absensi secara elektronik.

Pada perancangan dan pembuatan sistem absensi elektronik ini akan diaplikasikan pada dosen di ITN Malang, sehingga diharapkan akan memudahkan sistem pencatatan dosen pada saat akan mengajar dan kapan selesai mengajar. Maka dengan alat ini akan dapat diketahui kehadiran dari dosen pada saat ada jadwal mengajar dan akan dicatat kehadirannya secara otomatis oleh komputer.

1.2. Rumusan Masalah

Untuk memecahkan permasalahan yang ada maka dapat diambil rumusan masalah, maka pada skripsi ini masalah dititik beratkan pada :

1. Bagaimana membuat unit masukan yang akan memberikan identifikasi pada seseorang melalui kartu.
2. Bagaimana merencanakan dan membuat perangkat keras dan perangkat lunak yang dapat mengendalikan sistem.
3. Bagaimana merencanakan dan membuat struktur database yang dapat digunakan sebagai sumber informasi pada komputer.

1.3. Tujuan Masalah

Tujuan pokok dari penulisan skripsi ini adalah merancang dan merealisasikan peralatan elektronika yang dapat digunakan sebagai sistem daftar hadir elektronik pada dosen di ITN Malang.

1.4. Batasan Masalah

Agar permasalahan yang dibahas tidak meluas maka perlu adanya pembatasan permasalahan. Adapun batasan masalah yang meliputi :

1. Pembahasan ditekankan pada teknik kontrol dari Mikrokontroler AT89S51, yang diaplikasikan pada sistem daftar hadir Elektronik pada dosen di ITN Malang.
2. Database hanya digunakan untuk menyimpan informasi tentang absensi dosen dan tidak membahas tentang gaji karyawan.

3. Tidak membahas jenis komputer yang digunakan.
4. Tidak membahas masalah catu daya.
5. Tidak membahas masalah model alat dan chip card yang digunakan.

1.5. Metodologi Perencanaan

Adapun langkah-langkah yang diambil untuk menyelesaikan perubahan teoritis pada perancangan dan pembuatan alat ini adalah sebagai berikut :

1. Studi literatur, yaitu mengumpulkan data dan bahan – bahan acuan yang dapat digunakan untuk perencanaan dan pembuatan alat.
2. Perencanaan dan pembuatan alat yang digunakan dengan cara pendekatan seara *hardware* dan *software*.
3. Perencanaan blok diagram.
4. Pembuatan alat secara keseluruhan, menyatukan rangkaian dan masing masing blok diagram untuk mendapatkan rangkaian secara lengkap.
5. Menguji peralatan sebagai unit pemroses.
6. Menyusun naskah skripsi.

1.6. Sistematika Penulisan

Penyusunan laporan skripsi ini secara garis besar terdiri dari lima bab, yang masing – masing bab berisi pembahasan tertentu dengan rincian sebagai berikut :

BAB I	: PENDAHULUAN Berisi latar belakang permasalahan, rumusan masalah, tujuan masalah, batasan masalah, metodologi penulisan, sistematika penulisan dan relevansi dari penyusunan tugas akhir.
BAB II	: LANDASAN TEORI Membahas tentang teori penunjang secara umum yang diperlukan dengan memakai referensi – referensi dari beberapa sumber dimana teori ini merupakan acuan dalam perencanaan dan pembuatan alat.
BAB III	: PERENCANAAN DAN PEMBUATAN ALAT Membahas tentang perencanaan dan pembuatan perangkat lunak dari sistem.
BAB IV	: PENGUJIAN ALAT Membahas pengujian peralatan secara keseluruhan dan analisa hasil pengujian.
BAB V	: PENUTUP Berisikan tentang kesimpulan dari pembahasan pada bab bab sebelumnya dan kemungkinan pengembangan pengembangan maupun aplikasi yang dapat dilakukan pada alat yang dirancang bangun.

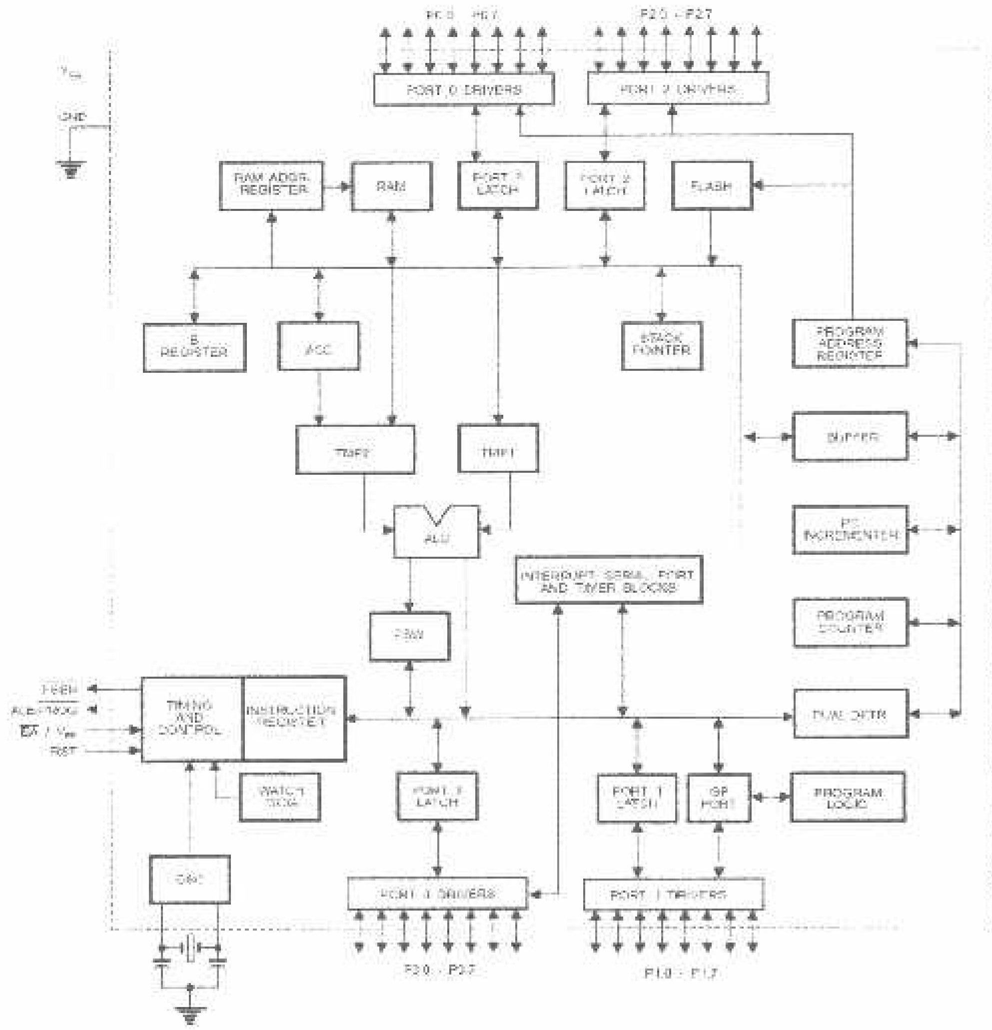
BAB II

LANDASAN TEORI

2.1. Mikrokontroler AT89S51

2.1.1. Arsitektur AT89S51

Mikrokontroler AT89S51 merupakan salah satu anggota keluarga MCS-51, yaitu suatu komponen produksi ATMEL yang berorientasi kontrol (*microcontroller*). Intel mengklarifikasikan dalam kelompok *embedded microcontroller*, yang artinya adalah mikrokontroler yang dapat diprogram ulang (*reprogrammable*). Di dalam *chip* mikrokontroler AT89S51 ini sudah tersedia berbagai macam peralatan pendukung mikroprosesor seperti RAM, *serial port*, *bus – bus* data dan lainnya yang membuat pemakai *chip* ini dapat menekan penambahan komponen pendukung. Spesifikasi perangkat keras dari mikrokontroler AT89S51 adalah sebagai berikut :



Gambar 2 – 1
Blok Diagram AT89S51
Sumber : *Datasheet ATMEL AT89S51*

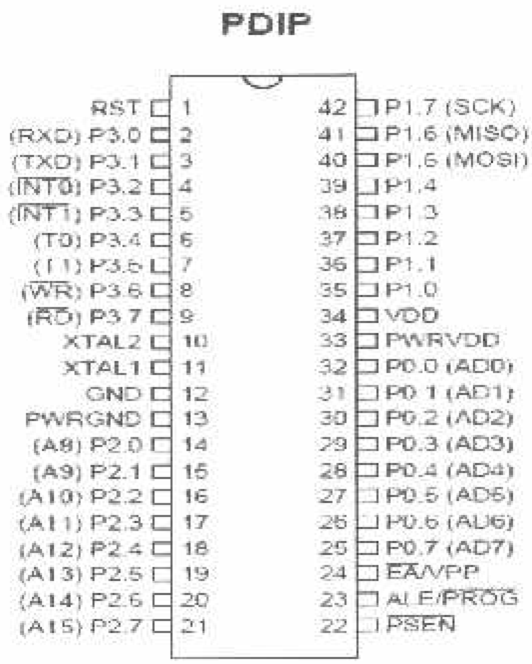
AT89S51 adalah mikrokontroler keluaran atmel dengan 4K byte Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89S51 merupakan memori dengan teknologi nonvolatile memori, artinya isi memori tersebut dapat diisi ulang ataupun dihapus berulang kali.

Memori ini biasa digunakan untuk menyimpan instruksi (Perintah) berstandar MCS – 51 code sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *Single Chip Operation* (Mode Operasi Keping Tunggal) yang tidak memerlukan *Eksternal Memori* (Memori luar) untuk menyimpan source code tersebut.

IC ATMEL AT89S51 menyediakan standart berikut:

- 4K Bytes memori yang dapat diprogram ulang
- 128 Bytes internal RAM
- 32 jalur I/O (Input dan Output) yang dapat diprogram
- Sepasang 16 bit Timer dan Counter
- Dual data Pointer (DPTR)
- Watchdog Timer
- ISP Port
- Mendukung serial Port secara penuh
- Waktu Pemrograman yang singkat

Sebagai tambahan AT89S51 dirancang menggunakan logika yang statis untuk mode pengoperasian yang menuju ke frekwensi dasar dan pendukungan terhadap dua Software, serta dapat memilih model *power saving*nya. Mode idle akan berhenti ketika CPU sedang menjalankan RAM, *Timer/Counter*, *Serial Port* dan *interrupt system* untuk terus melanjutkan fungsinya. Model power down akan menyimpan isi dari RAM tapi akan memberhentikan osilator dan akan menghentikan semua chip lain yang sedang berfungsi sampai terdapat adanya gangguan dari luar atau hardware di reset.



Gambar 2 – 2
Pin – Pin AT89S51
Sumber : *Datasheet ATMEL AT89S51*

2.1.2. Pin Deskripsi

VCC : Power Supply

GND : Ground

Port 0 : Port 0 berfungsi sebagai 8 bit I/O bertipe *open drain bi-directional.* Sebagai port keluaran masing – masing pin dapat menyerap arus sebesar 8 masukan TTL (sekitar 3,8 mA). Ketika diberikan logika '1' pada pin port 0 ini maka pin – pin port 0 ini akan dapat digunakan sebagai inputan berimpedansi tinggi. Port 0 juga dapat dikonfigurasikan pada sebagai bus alamat/data selama proses pengaksesan data memori dan program eksternal. Jika digunakan dalam mode ini port 0 memiliki internal Pull Up.

Port 0 juga menerima kode – kode data yang diberkan padanya selama proses pemrograman dan memberikan kode – kode selama proses verifikasi program yang telah tersimpan didalam memori. Dalam hal ini dibutuhkan eksternal Pull Up selama proses verifikasi program.

Port 1 : Port 1 berfungsi sebagai 8 bit I/O Bi-directional yang dilengkapi dengan internal Pull Up. Ketika diberikan logika '1' pin ini akan di Pull Up secara internal sehingga dapat digunakan sebagai input. Sebagai inputan jika pin – pin ini dihubungkan ke ground maka masing – masing pin ini dapat menghantarkan arus karena di Pull High secara internal. Port 1 juga menerima *Low Order Address Bytes* selama melakukan verifikasi program.

Pada port 1 di AT89S51 pin ini mempunyai alternatif seperti pada tabel berikut ini:

Tabel 2 – 1

Fungsi – Fungsi Alternative Port 1

Port Pin	Alternate Funtions
P1.5	MOSI (Master Output Slave Input)*
P1.6	MISO (Master Input Slave Output)*
P1.7	SCK (Serial Clock)*

Port 2 : Port 2 berfungsi sebagai 8 bit I/O Bi-directional yang dilengkapi dengan internal Pull Up Penyangga keluaran port 2 dapat

memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA).

Jika diberikan logika '1' pada pin – pin port 2, maka masing – masing pin akan di Pull High secara internal sehingga dapat digunakan sebagai inputan. Sebagai inputan jika pin – pin port 2 dihubungkan ke ground (di Pull Low), maka , masing – masing pin dapat menghantarkan arus karena di Pull High secara internal.

Port 2 akan memberikan byte alamat bagian tinggi (High Byte) selama pengambilan instruksi dari memori program eksternal dan selama pengaksesan memori data eksternal yang menggunakan perintah dengan alamat 16 bit (misalkan **MOVX@DPTR**). Dalam aplikasi ini , jika ingin mengirimkan '1', maka digunakan Pull Up internal yang sudah disediakan. Selama pengaksesan memori data eksternal yang menggunaka perintah 8 bit (misalkan **MOVX@RI**), port 2 akan mengirimkan isi dari SFR P2 (*Special Function Register Port 2*). Port 2 juga menerima alamat bagian tinggi (High Order Address) selama pemrograman dan verifikasi memori.

Port 3 : Port 3 sebagai 8 bit I/O Bi-directional yang dilengkapi dengan Pull Up Internal. Penyangga keluaran port 3 daqpat memberkan atau menyerap arus empat masukan TTL (sekitar 1,6 mA).

Jika diberikan logika '1' pada pin pin port 3, maka masing – masing pin akan di Pull High oleh Pull Up internal sehingga dapat digunakan sebagai inputan. Sebagai inputan, jika pin – pin port 3 dihubungkan ke ground, maka masing – masing kaki akan memberikan arus karena di Pull High secara internal.

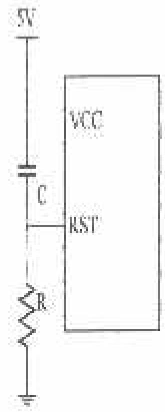
Seperti Port 1, port 3 juga mempunyai fungsi – fungsi alternatif yang diberikan oleh AT89S51 seperti pada tabel berikut ini:

Tabel 2 – 2

Fungsi – Fungsi Alternatif Port 3

Port Pin	Fungsi Alternatif
P3.0	RXD (Serial Input Port)
P3.1	TXD (Serial Output Port)
P3.2	$\overline{\text{INT0}}$ (Eksternal Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (Eksternal Interrupt 1)
P3.4	T0 (Timer 0 Eksternal Input)
P3.5	T1 (Timer 1 Eksternal Input)
P3.6	WR (Eksternal Data Memory Write Strobe)
P3.7	RD (Eksternal Data Memory Read Strobe)

Reset : Inputan Reset akan memberikan logika High '1' pada pin ini dengan jangka waktu yang ditentukan oleh lamanya pengosongan data muatan kapasitor. Jangka waktu minimal adalah 2 siklus mesin (24 periode frekwensi clock) ditambah waktu start On Osilator.



Gambar 2 - 3
Rangkaian Power On Reset
Sumber : *www.atmel.com*

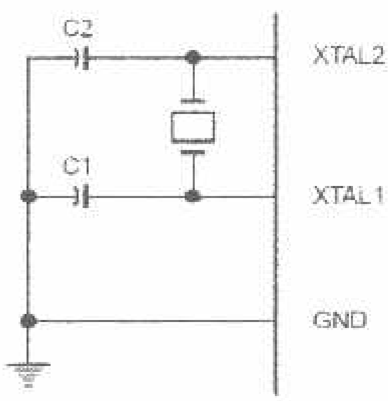
ALE/$\overline{\text{PROG}}$: Keluaran ALE (*Address Latch Enable*) menghasilkan pulsa – pulsa untuk menutup byte rendah (*Low Byte*) alamat selama mengakses memori eksternal. Pin ini juga berfungsi sebagai inputan pulsa program (*The Program Pulse Input*) atau $\overline{\text{PROG}}$ selama melakukan Flash Program. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekwensi kristal dan dapat digunakan sebagai pewaktuan (*Timing*) atau pendetakan (*Clocking*) rangkaian eksternal. Sebagai catatan ada sebuah pulsa yang dilewati selama pengaksesan memori data eksternal. Jika dikehendaki operasi ALE dapat di nonaktifkan dengan cara mengatur bit 0 dari SFR (*Special Function Register*) lokasi 8Eh. Jika diberi logika '1' ALE hanya akan aktif selama menemui instruksi **MOVX** atau **MOVC**. Selain itu, pin ini secara perlahan akan di Pull High. Mematikan bit ALE tidak akan ada efeknya jika mikrokontroller mengeksekusi program secara eksternal.

$\overline{\text{PSEN}}$: **$\overline{\text{PSEN}}$** (*Program Store Enable*) merupakan sinyal baca untuk memori program eksternal. Ketika mikrokontroller AT89S51 menjalankan kode dari program eksternal, **$\overline{\text{PSEN}}$** akan diaktifkan sebanyak 2 kali per siklusnya, kecuali dua aktivasi **$\overline{\text{PSEN}}$** dilompati (Diabaikan) saat mengakses memori data eksternal.

$\overline{\text{EA}}$/VPP : **$\overline{\text{EA}}$/VPP** (*External Access Enable*). **$\overline{\text{EA}}$** harus selalu dihubungkan ke Ground karena digunakan untuk mengakses eksternal memori dengan lokasi 0000H sampai FFFFH. Catatan sekalipun bit '1' sudah terkunci dan terprogram, maka EA akan terkunci pada reset. **$\overline{\text{EA}}$** juga harus dihubungkan ke **Vcc** untuk melakukan menjalankan program secara internal. Pada saat Flash Programming pin ini mendapatkan tegangan sebesar 12 Volt.

XTAL1 : Merupakan input ke penguat pembalik osilator dan ke rangkaian operasi Clock internal.

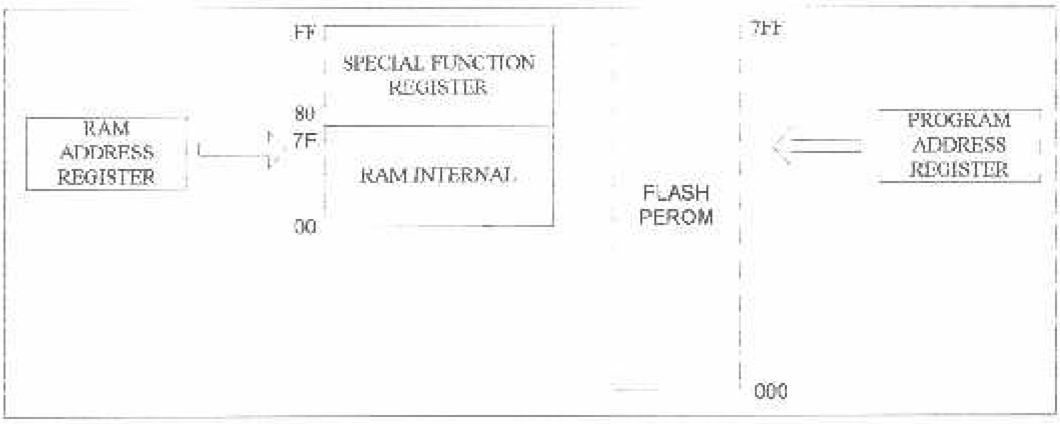
XTAL2 : Keluaran dari penguat pembalik osilator.



Gambar 2 – 4
Rangkaian Kristal
Sumber : *Datasheet AT89S51*

Mikrokontroler AT89S51 memiliki rangkaian osilator internal dengan mengacu pada frekwensi referensi pada pin XTAL1 dan XTAL2.

Struktur Memori



Gambar 2 – 5
Alamat RAM Internal dan Flash PEROM
Sumber : *Panduan Praktis Antarmuka & Pemrograman mC AT89c51, Gramedia*

AT89S51 mempunyai struktur atas memori yang terdiri atas :

- RAM Internal, memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara.
- *Special Function Register* (Register Fungsi Khusus), memori yang berisi register – register yang mempunyai fungsi – fungsi khusus yang disediakan oleh mikrokontroler tersebut, seperti timer, serial dan lain – lain.
- *Flash PEROM*, memori yang digunakan untuk menyimpan instruksi – instruksi MCS – 51.

AT89S51 mempunyai struktur memori yang terpisah antara RAM Internal dan Flash PEROM – nya. RAM Internal dialamati oleh RAM *Address Register* (Register Alamat RAM) sedangkan Flash PEROM yang menyimpan perintah – perintah MCS – 51 dialamati oleh *Program Address Register* (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, walaupun RAM Internal dan Flash PEROM, mempunyai alamat awal yang sama, yaitu 00, namun secara fisik kedua memori tersebut tidak saling berhubungan.

2.1.3. RAM Internal

RAM Internal terdiri atas:

⇒ Register BANKS

AT89S51 mempunyai delapan buah register yang terdiri atas R0 hingga R7. Kedelapan buah register ini selalu terletak pada alamat 00H hingga 07H pada setiap kali sistem direset. Namun, posisi R0 hingga R7 dapat dipindah ke BANK 1 (08 hingga 0FH), BANK 2 (10H hingga 17H), atau BANK 3 (18H hingga 1FH) dengan mengatur bit RS0 dan RS1.

⇒ Bit Addressable RAM

RAM pada alamat 20H hingga 2FH dapat diakses secara pengalamatan bit (*Bit Addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat diset, clear, AND dan OR. Sebagai contoh, pada saat terjadi instruksi Setb 67H, hal ini sama dengan menset bit MSB dari alamat 2C yaitu:

```
MOV   A,2CH          ; Pindahkan data dari alamat 2CH ke Acc A
Orl   A,#10000000B   ; Set MSB Acc A
MOV   2CH,A          ; Pindahkan data dari Acc A ke alamat 2CH
```

Dengan adanya sistem bit addressable RAM, proses yang seharusnya dijalankan dengan tiga cycle seperti listing diatas dapat digantikan dengan sebuah instruksi yang hanya membutuhkan satu instruksi saja.

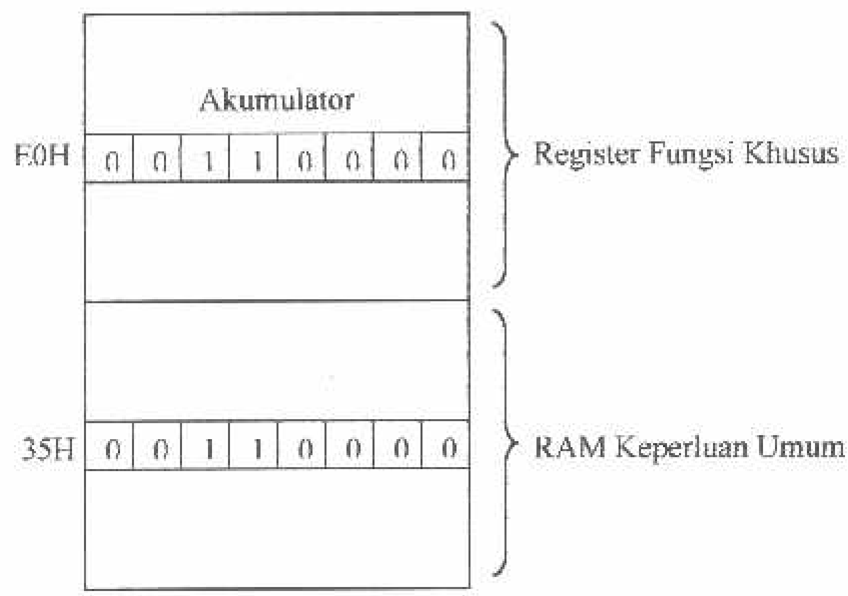
Dalam aplikasinya, lokasi yang dapat diakses dengan pengalamatan bit ini dapat juga digunakan untuk menandai suatu lokasi bit tertentu baik berupa Register Fungsi Khusus yang dapat dialamati secara bit (termasuk Register I/O) ataupun lokasi – lokasi tertentu yang dapat dialamati secara bit.

⇒ RAM Keperluan Umum

RAM keperluan umum dapat dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang menunjukan lokasi yang dialamati seperti pada contoh berikut:

MOV A,35H ; Baca data dari alamat 35H dan disimpan di Accumulator.

Alamat yang
ditunjuk langsung



Gambar 2 – 6
Pengalamatan Secara Langsung
Sumber : Panduan Praktis Antarmuka & Pemrograman mC AT89c51, Gramedia

Sedangkan pengalamatan secara tak langsung pada lokasi dari RAM Internal ini adalah akses dari data memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1. R0 dan R1 adalah dua buah register pada mikrokontroler berarsitertur MCS – 51 yang dapat digunakan sebagai pointer dari sebuah lokasi memori pada RAM Internal. Sebagai Contoh:

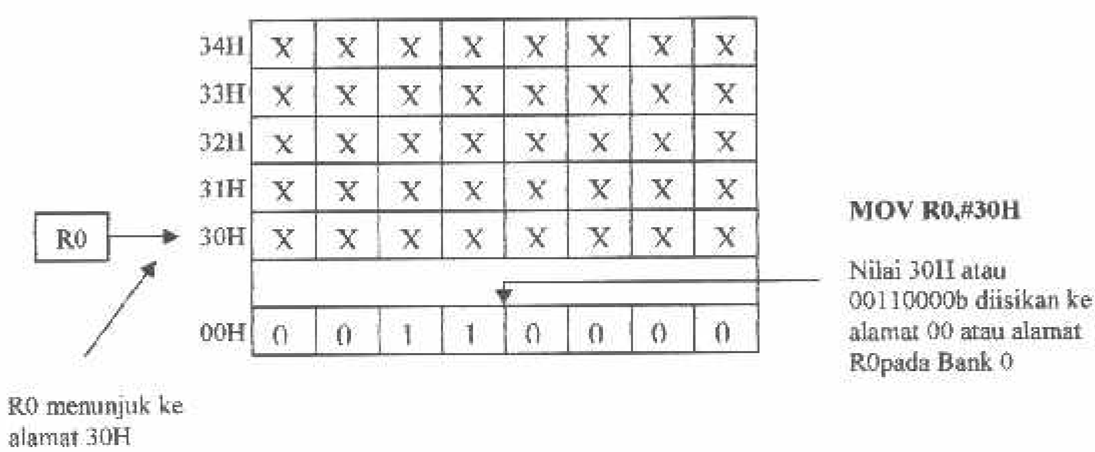
```
MOV  R0,#35H      ; R0 diisi dengan data 35H
MOV  A,@R0        ; Baca data di alamat yang dijuntuk oleh R0
```

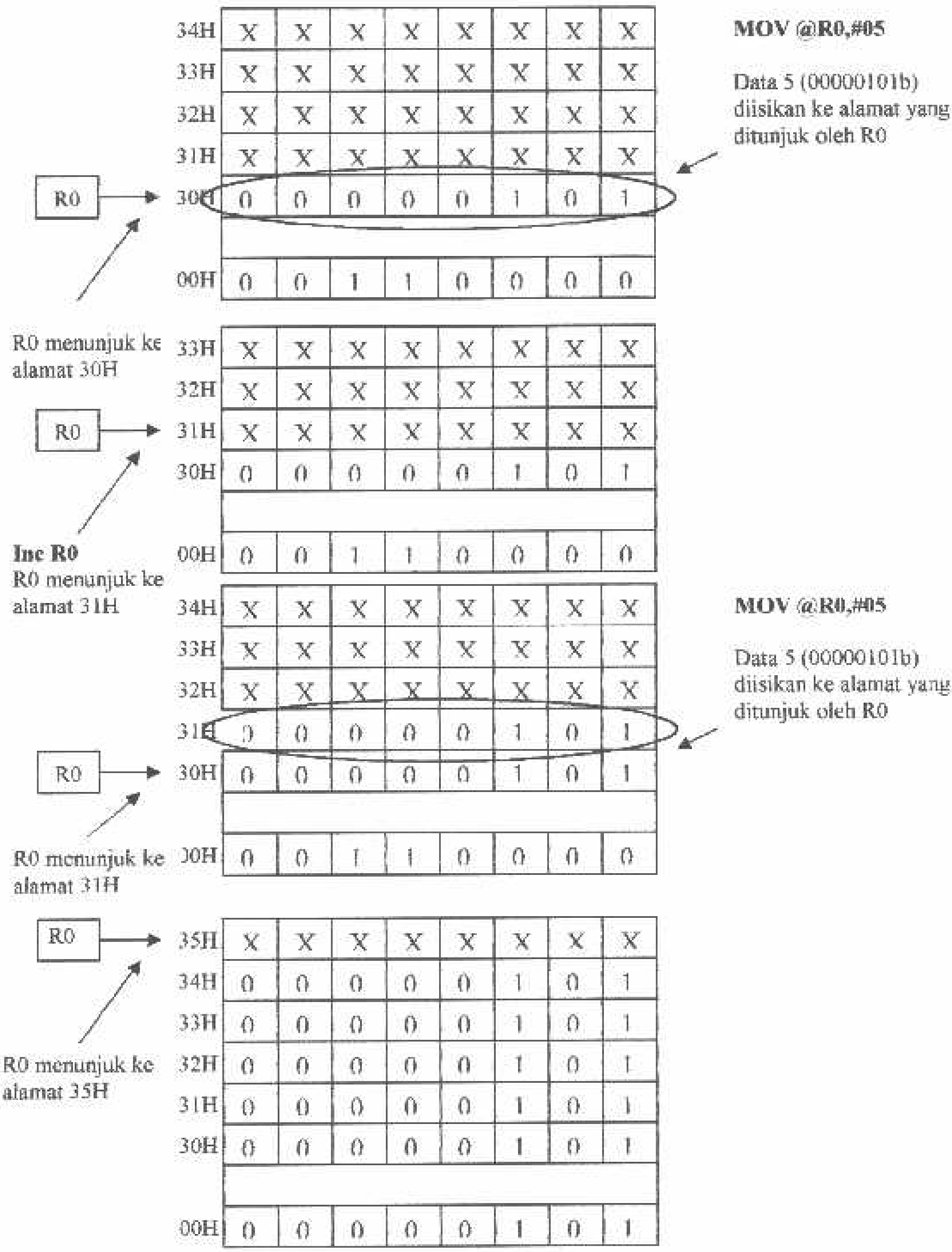
Pengalamatan secara tak langsung biasa digunakan untuk mengakses beberapa lokasi memori dengan letak yang beraturan seperti pada contoh berikut:

```
MOV     R0,#30H          ; R0 diisi dengan data 30H
Loop:
MOV     @R0,#05          ; Data 5 diisikan ke alamat yang ditunjuk
                         ; oleh R0

Inc     R0               ; R0 menunjuk ke alamat selanjutnya
Cjne    R0,#35H,Loop     ; Jika R0 belum mencapai 35H, lompat ke
                         ; label Loop
```

Pada gambar 2-7 yang merupakan step – step proses yang terjadi pada contoh program di atas, proses pemindahan data 5 ke alamat – alamat yang ditunjuk oleh R0 dilakukan berulang – ulang hingga R0, register yang berfungsi penyimpan nilai dari alamat yang diakses atau sebagai *pointer* (penunjuk) alamat yang diakses mencapai 35H. Pengalamatan – pengalamatan tersebut juga berlaku pada lokasi yang dapat dialamati secara bit maupun Register Bank.





Gambar 2 – 7
Step – Step Yang Terjadi Pada Pemindahan Data 5 ke Alamat 30H hingga 34H
Sumber : Panduan Praktis Antarmuka & Pemrograman mC AT89c51, Gramedia

2.1.4. Register Fungsi Khusus

AT89S51 mempunyai 21 ***Special Function Registers*** (Register Fungsi Khusus) yang terletak pada antara alamat 80H hingga FFH. Beberapa dari register – register ini juga bisa dialamati dengan pengalamatan bit sehingga dapat dioperasikan seperti yang ada pada RAM yang lokasinya dapat dialamati dengan pengalamatan bit.

⇒ Accumulator

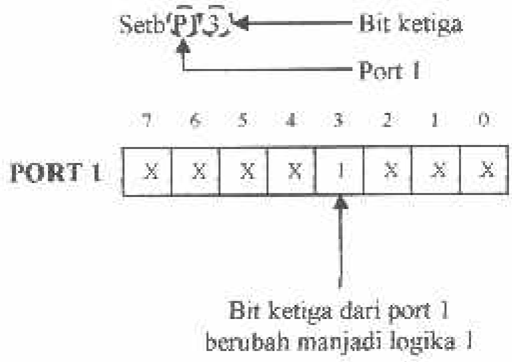
Register ini terletak pada alamat E0H. Hampir semua operasi aritmatik dan operasi logika selalu menggunakan register ini. Untuk proses pengambilan dan pengiriman data ke memori eksternal juga diperlukan register ini.

⇒ Port

AT89S51 mempunyai empat buah Port, yaitu Port 0, Port 1, Port 2 dan Port 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Namun, jika digunakan eksternal memoriataupun fungsi – fungsi special, seperti Eksternal Interrupt, Serial ataupun Eksternal Timer, Port 0, Port 2 dan Port 3 tidak dapat digunakan sebagai Port dengan fungsi umum. Untuk itu Port 1 yang dikhususkan untuk port dengan fungsi umum.

Semua Port ini dapat diakses dengan pengalamatan secara bit sehingga dapat dilakukan perubahan output pada tiap – tiap pin dari port ini tanpa mempengaruhi port - port yang lainnya.

Sebagai contoh, jika dilakukan instruksi Setb P1.3, maka bit ketiga dari port 1 akan berkondisi high (5V) tanpa mempengaruhi bit – bit yang lain pada port ini.



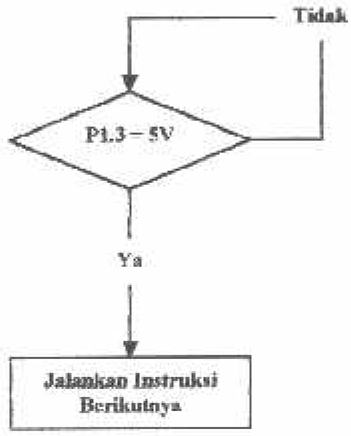
Gambar 2 – 8
Bit – Bit Port
Sumber : Panduan Praktis Antarmuka & Pemrograman mC AT89c51, Gramedia

Seperti yang tampak pada gambar 2.7, bit ketiga dari port 1 terletak pada alamat 93H oleh karena itu instruksi Setb P1.3 dapat diganti dengan instruksi Setb 93H.

Port ini digunakan untuk menunggu sinyal yang dikirim oleh komponen lain yang merupakan sinyal positif (5V) misalnya, dengan instruksi berikut ini:

```
Tunggu:
        Jnb     P1.3, tunggu
```

Selama kondisi pada port 1 pin ketiga masih low (0V), program akan terus melompat ke alamat yang ditunjukkan oleh label "tunggu" sehingga dapat diartikan bahwa program berhenti di alamat tersebut hingga terjadi sinyal positif (5V). Setelah sinyal positif (5V) muncul di bit ketiga dari port 1; program akan menuju ke alamat yang berikutnya.



Gambar 2 – 9
Diagram Alir Deteksi Bit Ketiga Port 1
Sumber : Panduan Praktis Antarmuka, Pemrograman mC AT89c51, Gramedia

⇒ PSW (*Program Status Word*)

Program status word atau PSW terletak pada alamat D0H yang terdiri atas beberapa bit sebagai berikut:

PSW	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
D0H	CY	AC	F0	RS1	RS0	OV	-	P

- Flag Carry

Flag Carry (terletak pada alamat D7H) mempunyai fungsi sebagai pendeteksi terjadinya kelebihan pada operasi penjumlahan atau terjadi pinjam (*borrow*) pada operasi pengurangan. Misalnya jika data pada accumulator adalah FFH dan dijumlahkan dengan bilangan satu atau lebih, akan terjadi kelebihan dan membuat carry menjadi Set, sedangkan jika data pada accumulator adalah 00H dan dikurangkan dengan bilanagan satu atau lebih, akan terjadi pemimjaman dan membuat carry juga menjadi set.

- Flag Auxilary Carry

  Flag Auxilary Carry akan selalu Set pada saat proses penjumlahan terjadi carry dari bit ketiga hingga bit keempat.

- Flag 0

  Flag 0 digunakan untuk tujuan umum bergantung pada kebutuhan pemakai.

- Bit Pemilih Register Bank

  Register Bank Select Bits (RS0 dan RS1) atau Bit Pemilih Register Bank digunakan untuk menentukan lokasi dari Register Bank (R0 hingga R7) pada memori. RS0 dan RS1 selalu bernilai nol setiap kali system direset sehingga lokasi dari R0 hingga R7 akan berada di alamat 00H hingga 07H.

- Flag Overflow

  Flag Overflow akan diset jika pada operasi aritmatik menghasilkan bilangan yang lebih besar dari pada 128 atau lebih kecil dari – 128.

  Contoh:

```
MOV  A,#80h; A = 128
Add   A,#10H        ; A = 128 + 16, A = 144 ATAU 90H
```

  Hasil dari operasi aritmatikdi atas adalah lebih besar dari 128, yaitu 144 atau 90H maka bit Overflow akan Set.

- Bit Pariti

Bit Pariti akan diset jika jumlah bit 1 dalam accumulator adalah ganjil dan akan clear jika jumlah bit 1 dalam accumulator genap. Jika data dalam accumulator adalah 10101110b atau AEH pariti akan diset. Data AEH mempunyai lima bit yang berkondisi 1 atau dapat disebut mempunyai bit 1 dalam jumlah yang ganjil. Bit pariti ini digunakan untuk proses yang berhubungan dengan serial port yaitu sebagai *Check sum*.

⇒ Register B

Register B digunakan bersama accumulator untuk proses aritmatik selain dapat juga difungsikan sebagai register biasa. Register ini juga bersifat *Bit Addressable*.

⇒ Stack Pointer

Stack Pointer merupakan sebuah register 8 bit yang terletak di alamat 81H. Isi dari Stack Pointer ini merupakan alamat dari data yang disimpan di stack. Stack Pointer dapat diedit atau dibiarkan saja mengikuti standart sesudah terjadi reset. Jika Stack Pointer diisi data 5FH, area untuk proses penyimpanan dan pengambilan data dari dan ke stack adalah sebesar 32 byte, yaitu antara 60H hingga 7FH karena AT89S51 mempunyai Internal RAM sebesar 128 byte.

⇒ Data Pointer Two Byte Register (DPTR)

Data Pointer Two Byte Register atau DPTR merupakan register 16 bit dan terletak pada alamat 82H untuk DPL (Data Pointer Low) dan 83H untuk DPH (Data Pointer High). DPTR biasa digunakan untuk mengakses source code ataupun data yang terletak di memori eksternal.

Contoh:

```
MOV  A, #01h
MOV  DPTR, #2000H
MOVX         @Dptr,A
```

Listing diatas berfungsi untuk menuliskan data 01H ke dalam alamat 2000H. Pertama, data 01H diisikan ke accumulator. Kemudian, DPTR yang berfungsi untuk menunjukan alamat penyimpanan data diisi dengan 2000H. terakhir, isi dari accumulator A disimpan ke lokasi memori yang ditunjukan oleh DPTR (*Indirect Addressing*).

⇒ Register Timer

AT89S51 mempunyai dua buah 16 bit Timer/Counter, yaitu Timer 0 dan Timer 1. Timer 0 terletak pada alamat 8AH untuk TL0 dan 8CH untuk TH0 dan Timer 1 terletak pada alamat 8BH untuk TL1 dan 8DH untuk TH1.

⇒ Register Port Serial

AT89S51 mempunyai sebuah *on chip serial port* (serial port dalam keping) yang dapat digunakan untuk berkomunikasi dengan peralatan lain yang menggunakan serial port juga seperti modem, shift register dan lain – lain.

Buffer (Penyangga) untuk proses pengiriman maupun pengambilan data terletak pada register SBUF, yaitu pada alamat 99H. Sedangkan untuk mengatur mode serial dapat dilakukan dengan mengubah isi dari SCON yang terletak pada alamat 98H.

⇒ Register Interupt

AT89S51 mempunyai lima buah interupsi dengan sua level prioritas interupsi. Interupsi akan selalu nonaktif setiap kali system di – reset. Register – register yang berhubungan dengan interrupt adalah *Interrupt Enable Register* (IE) atau Register Pengaktif Interupsi pada alamat A8H untuk mengatur keaktifan tiap – tiap interrupt dan Interrupt Priority Register (IP) atau Register Prioritas Interupsi pada alamat B8H.

⇒ Register Kontrol Power

Register ini terdiri atas SMOD yang digunakan untuk melipat dua baud rate dari port serial, dua buah bit untuk flag fungsi umum pada bit ketiga dan bit kedua, power Down (PD) bit dan idle (IDL) bit.

Pada mode idle hubungan antara CPU dan internal clock terputus, namun kondisi port tetap pada kondisi terakhir, ALE dan PSEN menjadi high, timermasih tetap bekerja. Mode idle berakhir pada saat terjadi interupsi, reset atau kondisi – kondisi lain yang mereset IDL bit.

Pada Mode Power Down oscillator dan semua fungsi berhenti, RAM tetap pada kondisi terakhir, begitupula dengan port dan ALE maupun PSEN akan berkondisi 0. Mode Power Down berakhir pada saat terjadi reset.

⇒ Time Counter Control (TCON)

TCON merupakan suatu register yang berisi bit – bit untuk memulai atau menghentikan pencacah/pewaktu.

⇒ Serial Control Buffer (SBUFF)

Register ini digunakan untuk menampung data masukan (SBUFF IN) ataupun keluaran (SBUFF OUT) dari serial port.

2.2. LCD

Pada sistem yang direncanakan akan digunakan LCD (Liquid Crystal Display) sebagai tampilan. LCD yang digunakan adalah jenis TM162ABC yang merupakan LCD dua baris dengan tiap barisnya terdiri dari 16 karakter.

LCD ini membutuhkan 3 sinyal kontrol, R/W (*read/write*) untuk menentukan apakah data akan dibaca atau ditulis, E (*Enable*) yang merupakan sinyal untuk meng-enable-kan dan RS (*Register Select*) untuk memilih register yang diakses. LCD TM162ABC memiliki 2 register yaitu register data dan register instruksi.

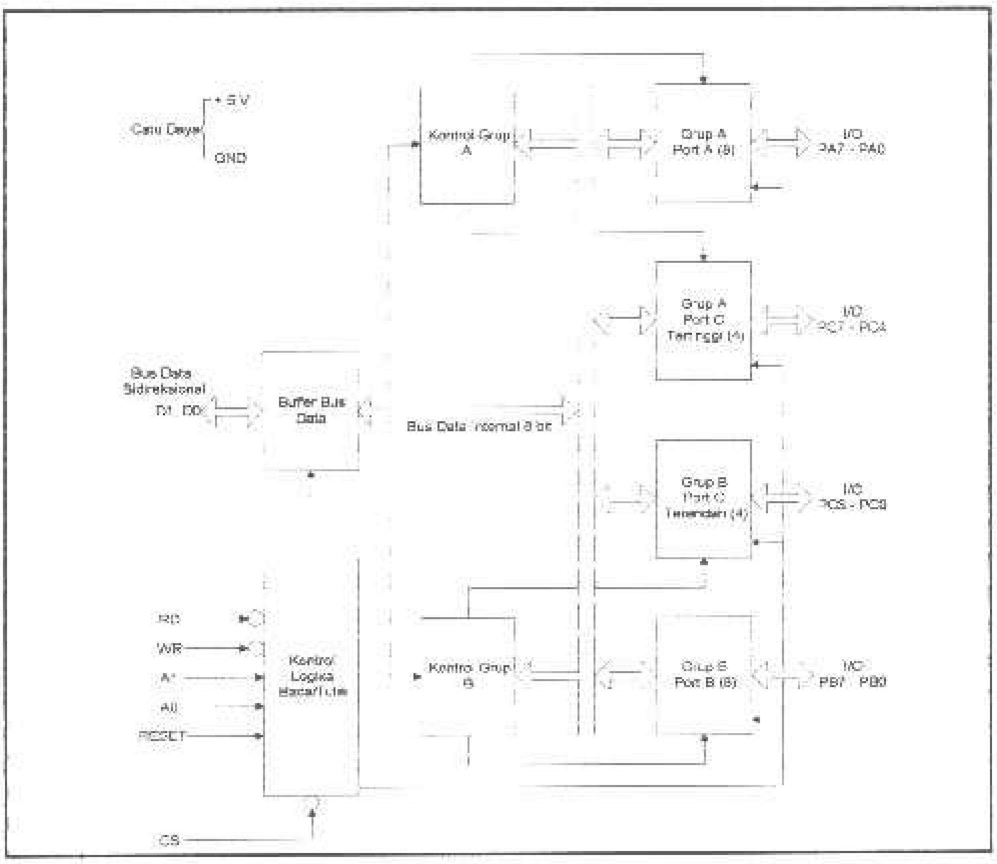
Dalam sistem ini, LCD menempati ruang alamat A000H-A001H. Pin R/W dihubungkan ke *ground* atau selalu berlogika 0 karena dalam perancangan. LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin A0 sistem mikrokontroler. Pengaktifan LCD ini selanjutnya tergantung pada pin E. Dimana pin E ini tergantung dari CS5 dari addres dekoder dan perintah write mikrokontroler. Berikut ini merupakan pin-pin LCD berserta konfigurasinya:

Tabel 2 – 3
Pin-Pin LCD Dan Konfigurasinya
Sumber : LCD Module User Manual

Nama pin	jumlah	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Trisate bidirectional lower data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
DB4-DB7	4	I/O	MPU	Tristatebidirectional upper fourdata bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
E	1	Input	MPU	Sinyal operasi dimulai: sinyal aktif baca/tulis
R/W	1	Input	MPU	Sinyalpilih data dan tulis (0:tulis,1:baca)
RS	1	-	Power supply	Sinyal pilih register 0: Instruction register (write) Busy flag dan addess counter (read) 1:Data register (write dan read)
VLC	1	-	Power supply	Penyetelan kontras pada tampilan LCD
VDD	1	-	Power supply	+ 5V
VSS	1	-	Power supply	Ground 0V

2.3. PPI 8255

PPI 8255 merupakan suatu *chip* yang dirancang untuk keperluan antarmuka dalam suatu sistem mikroprosesor. Chip ini memiliki 24 penyemat I/O yang masing-masing dapat diprogram dalam dua grup yaitu grup A dan grup B. Grup A terdiri dari *port* A ditambah dengan *port* C *upper* (C_4-C_7) sedangkan grup B terdiri dari *port* dan *port* C *lower* (C_0-C_3). Blok diagram PPI 8255 ditunjukkan dalam Gambar 2 – 10.

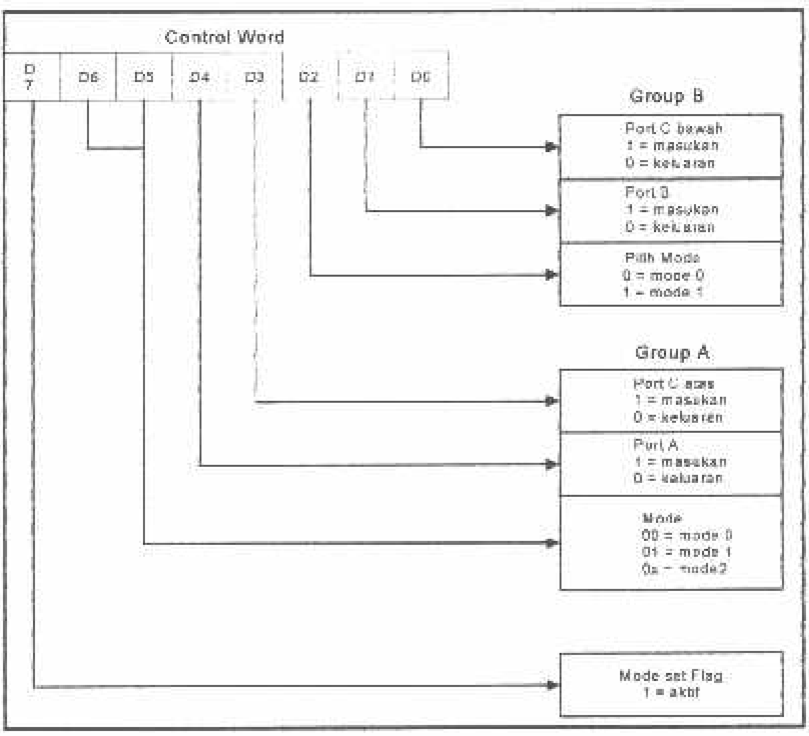


Gambar 2 – 10
Blok Diagram PPI 8255
Sumber : *Datasheet Harris Semiconductor*

Selain itu, pada *chip* ini terdapat tiga *port* yang dapat diprogram sebagai jalur keluaran atau masukan, yaitu :

- *Port* A, terdiri dari PA_0-PA_7
- *Port* B, terdiri dari PB_0-PB_7
- *Port* C, terdiri dari PC_0-PC_3 (*lower*) dan PC_4-PC_7 (*upper*)

Konfigurasi fungsi setiap *port* pada PPI ditentukan oleh *control word* yang diberikan ke *bus* data PPI pada awal program sebelum PPI digunakan. Proses ini biasa disebut inisialisasi PPI dan merupakan syarat mutlak yang harus diberikan kepada PPI. *Control word* ini mengandung berbagai informasi yang berhubungan dengan pemanfaatan PPI, seperti mode operasi, fungsi masing-masing *port* dan aktivitas PPI. *Control word* ini akan disimpan pada *control register*. Format masing-masing *bit* dari *control word* ditunjukkan dalam Gambar 2 11.



Gambar 2 – 11

***Control Word* PPI 8255**

Sumber : *Datasheet Harris Semiconductor*

Sebelum memakai peralatan PIO mikrokontroler harus menginisialisasinya dengan melaksanakan dua operasi dasar yaitu :

- Penentuan modus.
- Penunjukan arah saluran.

Untuk menentukan modus atau mode operasi PPI 8255 dilaksanakan dengan mengatur struktur *bit* pada *Control word* yang dikirimkan melalui program. *Control word* tersebut disimpan dalam *register* kontrol dengan alamat A_0=1 dan A_1=1. Bagian internal logic akan mengatur transfer data dan mengontrol informasi pada *bus* data internal. Mode kontrol dikirimkan ke dua buah *port* kontrol yaitu grup A dan grup B. Grup A mengatur penentuan mode dari *port* A dan 4 *bit* tertinggi dari *port* C. Grup B mengontrol *port* B dan 4 *bit* terendah dari *port* C. Pada Tabel 2 - 4 ditunjukkan operasi dasar PPI 8255.

Tabel 2 – 4
Operasi Dasar PPI 8255
Sumber : *Datasheet Harris Semiconductor*

A0	A1	Operasi I/O
0	0	*Bus* Data—*Port* A
0	1	*Bus* Data—*Port* B
1	0	*Bus* Data—*Port* B
1	1	*Bus* Data—Kontrol

PPI 8255 memiliki tiga mode operasi yang memiliki fungsi berbeda. Pemilihan mode ini dilakukan secara *software* dengan melakukan pengaturan *control word*.

a. Mode Operasi 0 (*Basic input/output*)

Konfigurasi ini digunakan untuk melakukan operasi masukan dan keluaran yang sederhana pada ketiga *port*. Pada mode ini tidak dibutuhkan proses jabat tangan (*handshaking*). Fungsi dasar dari mode 0 ini adalah :

- Dua 8-*bit port* dan dua 4-*bit port*.
- Masing-masing *port* dapat berfungsi sebagai masukan atau keluaran.
- Keluaran terkunci (*output latched*).
- Masukan tidak terkunci.

b. Mode Operasi 1 (*Strobed Input Output*)

Mode ini digunakan untuk mentransfer data pada *port* dengan menggunakan sinyal jabat tangan (*handshaking*). Dalam mode ini, port A dan port B memanfaatkan jalur-jalur pada *port* C untuk membangkitkan sinyal jabat tangan. Fungsi dasar dari mode 1 adalah :

- Dua grup (grup A dan grup B).
- Masing-masing grup terdiri dari 8-*bit* jalur data dan 4-*bit* jalur kontrol.
- Jalur data 8-*bit* dapat digunakan sebagai masukan atau keluaran dan keduanya terkunci.
- Jalur kontrol 4-*bit* digunakan untuk mengendalikan dan mengetahui status dari jalur data 8-*bit*.

c. Mode Operasi 2 (*Strobed Bidirectional Bus I/O*)

Dengan konfigurasi ini, sebuah jalur data 8-*bit* dapat digunakan untuk mengirim atau menerima data (*bidirectional*). Fungsi dasar mode operasi 2 ini adalah:

- Hanya digunakan pada grup A
- Terdapat sebuah 8-*bit*, *bidirectional bus* (*port* A) dan 5-*bit* jalur kontrol (*port* C).
- Masukan dan keluaran terkunci (*input and output latched*).
- 5-*bit* jalur kontrol pada port C digunakan untuk mengendalikan dan mengetahui status dari 8-*bit bidirectional bus* (*port* A).

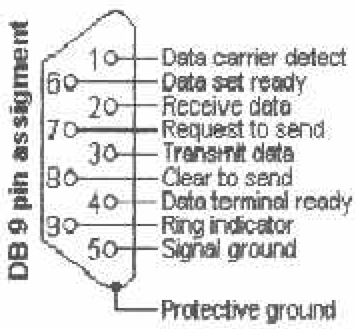
2.4. RS-232

2.4.1. Interface

Interface merupakan suatu piranti yang dapat menghubungkan satu peralatan ke peralatan lain sehingga peralatan tersebut dapat berkomunikasi. Sedangkan suatu cara/proses untuk menghubungkan dua sistem yang berbeda agar bisa bekerja sama disebut interfacing.

2.4.2. Konfigurasi pin

Jika suatu komputer dihubungkan dengan komputer lain atau suatu periferal lain, maka bisa dilakukan hubungan komunikasi data. Sebuah komputer dapat bertukar informasi melalui *interface port I/O* serial yang disebut *interface* RS 232. RS 232 merupakan interface antara peralatan terminal data dan peralatan komunikasi data dengan menggunakan data biner serial sebagai data yang ditransmisikan. Salah satu konektor yang biasa digunakan untuk komunikasi serial ini adalah konektor DB 9. Konektor DB 9 ditunjukkan pada Gambar 2 – 12.



Gambar 2 – 12
Konektor DB 9

Pada dasarnya semua pin-pin memiliki fungsi-fungsi tetapi untuk keperluan RS 232 ini hanya beberapa saja yang penting.

Fungsi yang ada untuk konektor DB 9 adalah sebagaimana tampak dalam Tabel 2 – 5 berikut :.

TABEL 2 – 5
KONFIGURASI PIN DB 9
Sumber : *B&B Electronics Mfg Co, USA, 2001, halaman 5*

EIA 232 (RS 232) FUNCTION	PIN
Data Carrier Detect (DCD)	1
Received Data (RxD)	2
Transmitted Data (TxD)	3
Data Terminal Ready (DTR)	4
Sinyal Ground (GND)	5
Data Set Ready (DSR)	6
Request To Send (RTS)	7
Clear To Send (CTS)	8
Ring Indikator (RI)	9

Jalur data (TxD dan RxD) untuk transport data, TxD adalah jalur output pada komputer, data dikirim dari pin ini. Sedangkan RxD adalah penerima untuk komputer, data yang datang akan diterima oleh pin ini. Pin ke empat adalah output (RTS) di mana sebuah sinyal akan diberikan pada alat yang dihubungkan dengan maksud meminta kiriman data. CTS adalah sinyal masukan yang menunggu sinyal dari alat yang terhubung. Ketika alat tersebut menerima sinyal RTS dan bisa menerima data maka ia akan mengirimkan sinyal balik yang merupakan CTS. DTR adalah sinyal keluaran yang memberi tanda bahwa ada alat yang terhubung dan akan mengirimkan data. DSR merupakan sinyal input yang mana jika alat yang terhubung manerima sinyal DTR ia akan memberi sinyal balik kemudian diterima sebagai sinyal DSR.

2.4.3. Protokol Komunikasi pada RS 232

Beberapa protokol dalam interface RS 232 adalah:

- Start Bit

  Merupakan sebuah bit dengan logic "0" dimana bit ini yang menandakan bahwa akan ada karakter atau data yang mengikutinya. Bit ini langsung diberikan oleh sinyal device tanpa harus mensetnya terlebih dahulu.

- Data Bit

  Merupakan bit yang mewakili dari karakter yang diikutinya *data bit* ini dapat diset sepanjang antara 5 sampai 8 bit.

- Pariti Bit

  Merupakan bit yang digunakan sebagai *error checking* pada *receiver* , apabila terjadi kesalahan maka *receiver* akan menset *error flag* (parity error) pada special register. *Parity bit* ini menghitung jumlah data yang berlogic '1' pada data bit. Perhitungan jumlah data bit tersebut tergantung dari jenis *parity* yang diset. Untuk parity *EVEN* maka jumlah data bit yang berlogic '1' ditambah dengan *parity bit* akan menghasilkan jumlah yang ganjil. Sedangkan untuk *parity MARK* merupakan *parity bit* selalu berlogic '1' begitu pula pada space, parity bit selalu berlogic ' 0' dan *parity NONE* disini *parity bit* yang diabaikan.

- Stop Bit

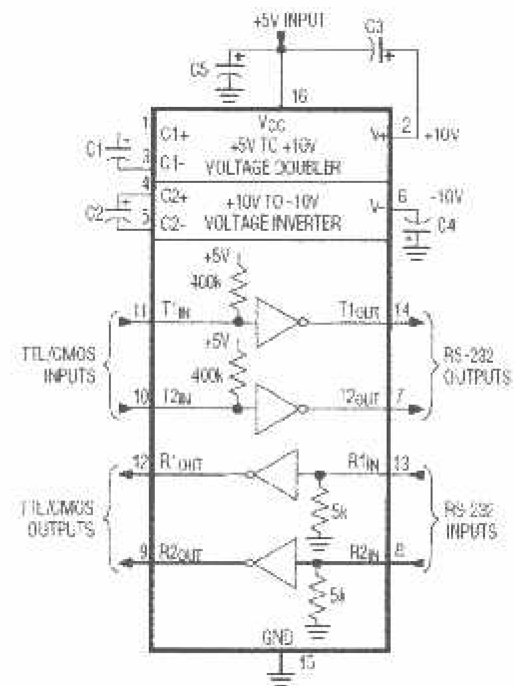
  Merupakan bit yang menandakan akhir dari suatu paket data (biasanya 1 byte data). Seperti pada start bit, bit ini langsung diberikan dari serial device. *Stop bit* ini dapat diset panjangnya menjadi satu bit, satu setengah dan dua bit.

- Baud Rate

  Sebenarnya *baut rate* berarti pergantian kondisi tiap detik *(State Change of the Line persecond)*, tetapi karena hanya ada 2 kondisi pada serial *(logic* 0 *dan 1)* maka dapat juga digunakan untuk menunjukkan kecepatan dari transmisi *(bits per second)*.

Saluran data pada port seri PC menggunakan standard RS232, dimana logic 0 (low) dinyatakan sebagai tegangan antara +3 Volt sampai +10 Volt dan logic 1 (high) dinyatakan sebagai tegangan antara -3 Volt sampai -10 Volt. Level tegangan ini tidak sesuai dengan level tegangan yang dipakai pada port seri AT89S51 yang menggunakan standard TTL (Transistor Transistor Logic), yaitu level tegangan baku dalam rangkaian – rangkaian digital.

Dalam standar TTL logic 0 (low) dinyatakan sebagai tegangan antara 0 Volt sampai 0.8 Volt, dan logic 1 (high) dinyatakan sebagai tegangan antara 3.5 Volt sampai 5 Volt. Karena perbedaan tegangan tersebut, agar port seri PC tidak merusak port seri AT89S51 antara keduanya dipasangkan IC MAX232 sebagai penyesuai tegangan. Rangkaian dasar dari MAX 232 dapat dilihat pada gambar 2 -13 berikut ini:

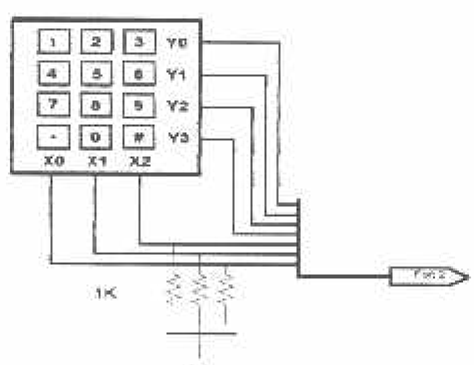


Gambar 2 – 13
Rangkaian Operasi MAX 232
Sumber : *Data Sheet Maxim Integrated Products*

2.5. Papan Tombol (*Keypad*)

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port* 1 mikrokontroler AT89S51.

Papan tombol tersebut mempunyai matrik 4 baris dan 4 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port* C PPI 8255 yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan *ground* (berlogika 0) dan *port* C (PC4-PC7) yang difungsikan sebagai *input* mikrokontroler. Sedangkan deretan baris dihubungkan ke *port* C (PC0-PC3) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri. Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. *Port* ini difungsikan sebagai *output* dari mikrokontroler. Dengan demikian kalau tombol tidak ditekan maka masukan *port* C (PC4-PC7) di pin yang terhubung tombol tersebut berlogika 0 dan bila tombol ditekan akan berlogika 1.

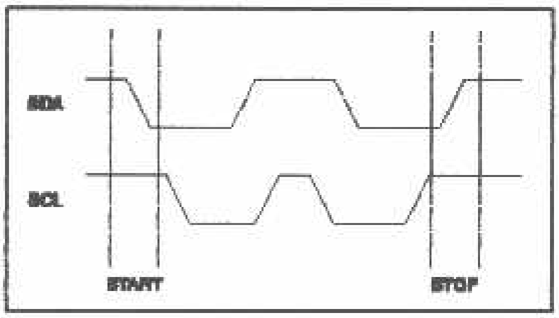


Gambar 2 – 14
Skematik Rangkaian *Keypad*
Sumber : Data Sheet Maxim Integrated Products

2.6. EEPROM AT24C64

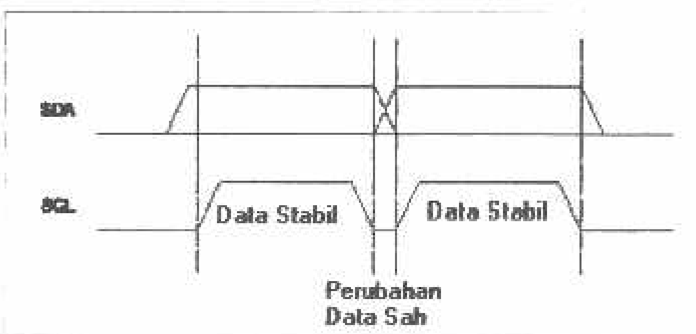
EEPROM AT24C64 memerlukan pengalamatan 8 bit yang mengawali kondisi awal yang membuat EEPROM dapat dibaca maupun ditulis. Pada pengalamatan ini diperlukan urutan akses yaitu 1010 sebagai data awalan. Tiga bit data berikutnya data pengalamatan EEPROM, namun untuk tipe AT24C64 tidak memerlukan pengalamatan device, biti-bit ini harus diberik logika nol. Bit ke delapan dari pengalamatan device menandakan operasi baca atau operasi tulis yang akan dilakukan. Untuk operasi pembacaan bit ini harus berlogika tinggi dan untuk operasi penulisan bit ini harus berlogika rendah.

Pin *Serial Data* (SDA) dan pin *Serial Clock* (SCL) memegang peranan di dalam mengendalikan kerja dari memori *serial* AT24C64. Kedua pin tersebut memerlukan tahanan *pull-up* sebesar 10KΩ *Serial data* digunakan untuk memtransfer data dua arah secara serial. *Serial clock* digunakan untuk meng-clock data. *Clock* sisi positif digunakan untuk memasukkan setiap data ke dalam EEPROM dan clock sisi negatif digunakan untuk mengeluarkan data dari EEPROM. Perubahan data pada SDA selama kondisi pin SCL dalam keadaan *high* menunjukkan adanya kondisi mulai dan kondisi berhenti dari memori. Diagram waktu untuk kondisi mulai kerja dan kondisi berhenti dari memori dapat dilihat pada Gambar 2-15.



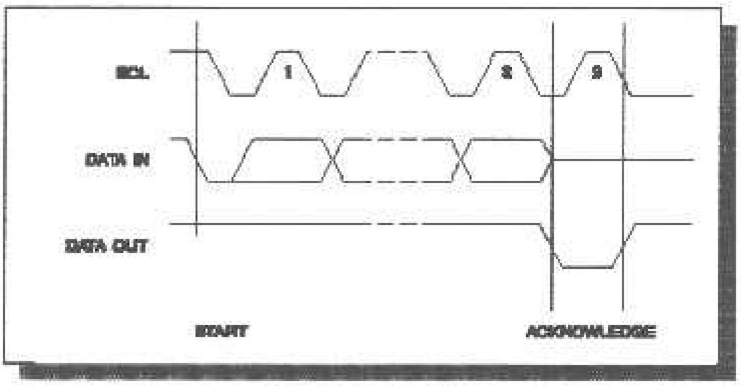
Gambar 2 – 15
Diagram waktu untuk kondisi start dan stop pada memori.
Sumber: *Datasheet AT24C64*

Selain untuk kondisi mulai dari kondisi berhenti, perubahan data pada pin SDA hanya akan sah apabila pin SCL dan keadaan kondisi *low*. Diagram waktu yang menunjukkan kapan perubahan data pada SDA akan sah dapat dilihat pada Gambar 2-16.



Gambar 2 – 16
Diagram waktu yang menunjukkan kapan perubahan data sah
Sumber: *Datasheet AT24C64*

Seperti yang telah disebutkan didepan bahwa EEPROM AT24C64 memiliki kapasitas sebesar 8192 word dengan masing-masing word sebesar 8 bit. Selain itu EEPROM yang dipakai adalah jenis serial EEPROM. Oleh sebab itu, setiap kali pengiriman data sebesar 8 bit dari mikrokontroller maka data tersebut akan dikirim tiap bit-nya secara serial. Setiap selesai menerima masing-masing word EEPROM secara otomatis akan mengirimkan *zero acknowledge* pada clock ke sembilan. Timing diagram yang menunjukkan *zero acknowledge* dapat dilihat pada Gambar 2-17.



Gambar 2 – 17
Diagram waktu yang menunjukkan *zero acknowledge*
Sumber: *Datasheet AT24C64*

2.7. Visual Basic 6.0

Visual Basic 6.0 merupakan bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. pengguna dapat membuat program dengan aplikasi GUI *(Graphical User Interface)* atau program yang memungkinkan pemakai komputer berkomunikasi dengan komputer tersebut dengan menggunakan modus grafik atau gambar.

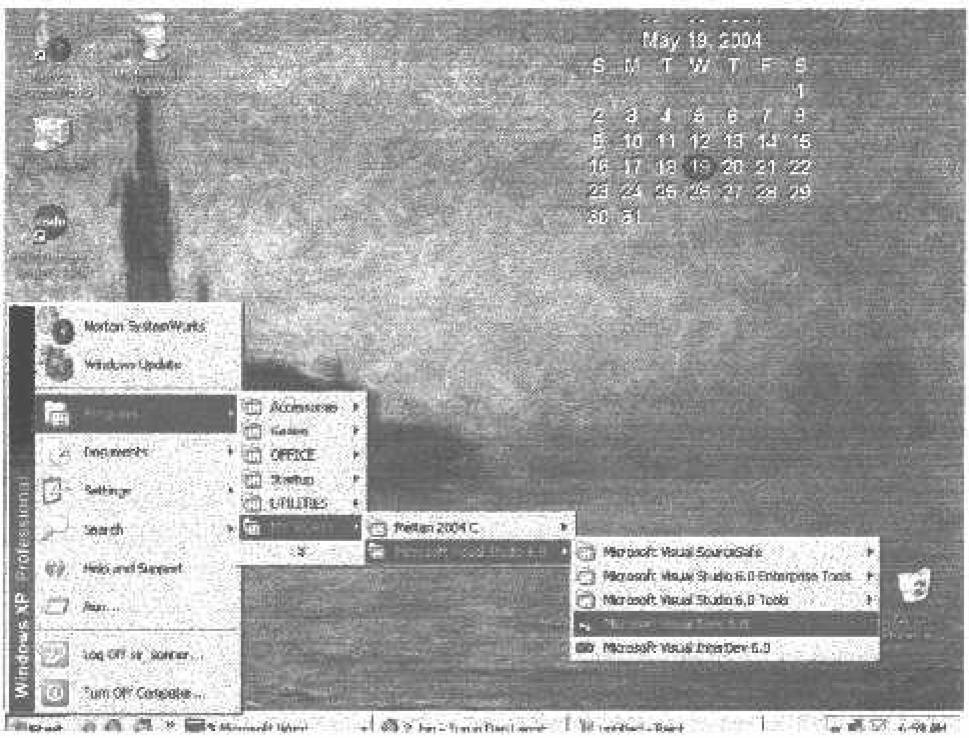
Visual Basic 6.0 berawal dari bahasa pemrograman BASIC *(Beginners All-purpose Symbolic Instruction Code)*. Karena bahasa BASIC cukup mudah dipelajari dan populer maka hampir setiap programmer menguasai bahasa ini.

Tahun 1980-an sistem operasi DOS cukup populer di kalangan pemakai PC karena di dalamnya disertakan bahasa BASIC yang dikenal dengan QBASIC (QuickBasic). Sistem tersebut sekarang sudah jarang digunakan. Di era Windows, Microsoft menciptakan Visual Basic yang terus mengalami penyempurnaan hingga Visual Basic 6.0 ini.

2.7.1. Memulai Visual Basic

Pada bagian ini akan dijelaskan bagaimana cara menjalankan Visual Basic 6.0 pada sistem operasi Windows. Cara pertama yang dapat dilakukan untuk memulai Microsoft Visual Basic 6.0 adalah:

1. Klik tombol Start pada Taskbar, kemudian pilih Program dari tampilan menu utama
2. Dari tampilan menu yang ada, pilih Visual Basic 6.0. perhatikan Gambar 2 - 18 yang menunjukkan proses cara menjalankan Visual Basic 6.0 dengan menggunakan Windows.



Gambar 2 – 18
Tampilan Menu Start

2.7.2. Tampilan Awal Visual Basic

Secara otomatis, pada saat pertama kali menjalankan Visual Basic, akan tampil kotak dialog New Project. Pada kotak dialog tersebut terdapat tiga pilihan tabulasi dengan keterangan sebagai berikut:

Tabel 2 - 6
Pilihan Tabulasi

Tabulasi	Keterangan
New	Pilihan ini digunakan untuk membuat project baru dengan berbagai macam pilihan.
Existing	Pilihan ini digunakan untuk membuka project yang pernah dibuat sebelumnya dengan menentukan folder sekaligus nama file.
Recent	Pilihan ini digunakan untuk membuka project yang telah dibuat dan terakhir kali dibuka.

2.8. Komponen Visual Basic 6.0

2.8.1. Title Bar

Title bar merupakan batang judul dari program Visual Basic 6.0 yang terletak pada bagian paling atas dari jendela program yang berfungsi untuk menampilkan judul atau namajendela. Selain itu title bar juga berfungsi untuk:

a. Memindah posisi jendela dengan menggunakan proses drag and drop pada posisi title bar tersebut.

b. Mengatur ukuran jendela dari ukuran Maximize ke ukuran Restore ataupun sebaliknya dengan melakukan klik ganda pada posisi title bar tersebut.

2.8.2. Control Menu

Control Menu merupakan sebuah elemen yang terletak pada bagian sudut kiri atas dari jendela Visual Basic. Dalam sistem operasi Windows, elemen ini tampil dalam bentuk ikon program. Ketika meng-klik Control Menu, akan tampil daftar menu perintah yang digunakan untuk mengubah ukuran jendela, memindah letak jendela dan juga dapat untuk keluar dari program Microsoft Visual Basic 6.0. Untuk lebih jelasnya, dapat dilihat fungsi dari masing-masing perintah pada tabel di bawah ini:

Tabel 2 - 7
Fungsi Perintah /Tombol

Tombol	Fungsi
Restore	Mengubah ukuran jendela Visual Basic ke ukuran relatif
Minimize	Mengubah ukuran jendela Visual Basic ke ukuran minimal.
Maximize	Mengubah ukuran jendela Visual Basic ke ukuran maksimal.
Move	Memindah posisi jendela Visual Basic.
Size	Mengatur ukuran jendela Visual Basic.
Close	Keluar dari program Visual Basic.

2.8.3. Menu Bar

Menu bar merupakan batang menu yang terletak di bawah title bar yang berfungsi untuk menampilkan pilihan menu atau perintah untuk mengoperasikan program Visual Basic. Saat pertama kali jendela program Visual Basic terbuka, ada tiga belas menu utama, yaitu: File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Windows dan Help.

Tampilan pilihan menu dalam program Visual Basic memiliki beberapa variasi yang masing-masing mempunyai pengertian yang berbeda dengan penjelasan sebagai berikut:

a. Menu dengan tanda segitiga yang terletak pada bagian kanan, di mana apabila menu tersebut dipilih maka akan menampilkan submenu pilihan berikutnya.
b. Menu yang secara langsung akan menjalankan suatu perintah yang dipilih.
c. Menu dengan tanda titik tiga pada bagian kanan, apabila dipilih maka akan menampilkan kotak dialog perintah.
d. Menu yang tampil redup yang mengindikasikan bahwa menu tersebut tidak dapat untuk dipilih.
e. Menu dengan gambar ikon di sebelah kirinya yang menandakan bahwa selain dapat dengan menggunakan menu tersebut untuk menjalankan suatu perintah, Anda dapat menekan tombol yang ada di toolbar.
f. Menu dengan Shortcut key di sebelah kanannya yang menandakan bahwa selain dengan menggunakan menu tersebut untuk menjalankan suatu perintah, Anda dapat menekan tombol kombinasi pada keyboard. Misalnya kombinasi antara tombol Shift + F7 untuk mewakili perintah menu View - Object.

2.8.4. Toolbar

Toolbar merupakan sebuah batang yang berisi kumpulan tombol yang terletak di bagian bawah menu bar yang dapat digunakan untuk menjalankan suatu perintah. Pada kondisi default program Visual Basic hanya menampilkan toolbar Standard.

Untuk lebih jelasnya tentang fungsi masing-masing tombol pada toolbar Standard, perhatikan penjelasan berikut:

Tabel 2 - 8
Fungsi tombol pada toolbar Standard

Nama Tombol	Fungsi
Add Project	Menambahkan project baru,Dengan pilihan: • Standard EXE • ActiveX EXE • ActiveX DLL • ActiveX Control
Add Form	Menambahkan item, dengan pilthan: • Form • User Control • MDI Form • Property Page • Module • Add File • Class Module • User Document
Menu Editor	Menampilkan kotak dialog Menu Editor.
Open Project	Membuka project yang sudah pernah dibuat ebelumnya.
Save Project Group	Menyimpan project.
Cut	Memotong kontrol yang ada di jendela form atau teks yang ada di jendela code.
Copy	Menyalin kontrol yang ada di jendela form atau teks yang ada di Jendela code.
Paste	Menempelkan kontrol atau teks yang sudah dipotong dengan perintah Cut atau disalin dengan perintah Copy.
Find	Mencari teks pada jendela code.
Undo	Membatalkan suatu perintah an dialankan sebelumn a..
Redo	Mengulangi suatu perintah yang pernah dibatalkan
Start	Menjalankan program.
Break	Menghentikan program yang sedang jalan untuk Sementara.
End	Menghentikan program yang sedang dijalankan.
Project Explorer	Menampilkan jendela ProjectExplorer.
Properties Window	Menampilkan jendela Properties.
	Menampilkan jendela Form
Window	Layout.
Object	Menampilkan jendela Object
Browser	Browser.
Toolbox	Menampilkan jendela Toolbox.

2.9. Properties Windows

Properties Window merupakan sebuah jendela digunakan untuk menampung nama properti dari yang terpilih. Pengaturan properti pada program Basic merupakan hal yang sangat penting membedakan objek yang satu dengan yang lainnya.

Pada jendela properti ditampilkan jenis dan nama objek yang bisa dipilih urut berdasarkan abjad pada tab Alphabetic atau berdasarkan kategori pada tab Categorized. Untuk menampilkan jendela Properties, dapat menggunakan prosedur berikut:

1. Klik tombol Properties Window pada toolbar Standard.
2. Pilih perintah View - Properties Window.
3. Shorcut key F4.

2.1.8.1. Form Layout Window

Form Layout Window merupakan sebuah jendela yang digunakan untuk mengatur posisi dari form pada form saat program dijalankan. Pada saat mengarahkan pointer mouse ke bagian form, maka pinter mouse akan berubah menjadi anak panah empat arah (pointer pengatur posisi). Untuk memindah posisi form pada layar monitor dapat dilakukan dengan proses drag and drop.

Untuk menampilkan jendela Form Layout, dapat menggunakan prosedur berikut:

1. Klik tombol Form Layout Window pada toolbar Standard.
2. Pilih perintah View - Form Layout Window.

2.9.2. Immediate Windows

Immediate Window merupakan sebuah jendela yang digunakan untuk mencoba beberapa perintah dengan mengetikkan baris program dan dapat secara langsung melihat hasilnya. Hal tersebut biasa dilakukan dan sangat membantu proses pengujian suatu perintah sebelum dipasang di dalam program.

Untuk menampilkan jendela Immediate, menggunakan prosedur berikut:

1. Pilih perintah View - Immediate Window.
2. Shortcut key Ctrl+G.
3. Klik tombol Immediate Window pada toolbar Debug.

2.9.2.1 Form Windows

Form Window merupakan jendela desain dari sebuah program aplikasi. Kita dapat mendesain sebuah program aplikasi dengan menempatkan kontrol-kontrol yang ada di bagian toolbox pada area form.

2.9.2.2. Code Windows

Code Window merupakan sebuah jendela yang digunakan untuk menuliskan kode program dari kontrol yang dipasang pada jendela form dengan cara memilih terlebih dahulu kontrol tersebut pada kotak objek.

2.9.2.3. Event

Event merupakan suatu kejadian yang akan diterima oleh suatu objek. Event yang diterima oleh objek berfungsi untuk menjalankan kode program yang ada didalam objek tersebut.

2.10. Module

Module hampir sama fungsinya dengan form, tetapi module tidak berisi objek dan bentuk standar. dan module berisi kode program atau prosedur yang dapat digunakan oleh program aplikasi.

2.10.1. Mengolah Toolbox

Pada kondisi default, pada bagian toolbox terdapat satu tabulasi yaitu tabulasi General. Saat menambahkan item kontrol baru secara otomatis akan ditampilkan pada tab yang aktif (General). Kita dapat mengelompokkan item-item yang akan ditambahkan ke bagian toolbox dengan menggunakan tabulasi buatan sendiri. Diharapkan semua item yang ditambahkan ke toolbox dapat terorganisasi sesuai dengan aplikasi yang dibuat.

1. Menambah Tabulasi

Untuk menambahkan tabulasi ke bagian toolbox, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian toolbox.
2. Pilih perintah Add Tab sehingga akan tampil sebuah kotak dialog.
3. Ketikkan nama tabulasi pada kotak New lab Name dan klik tombol OK.

2. Menghapus Tabulasi

Untuk menghapus tabulasi dari bagian toolbox. Dapat menggunakan prosedur berikut:

1. Klik kanan bagian tabulasi yang ingin dihapus.
2. Pilih perintab Delete Tab.

3. Mengganti Nama Tabulasi

Untuk mengganti nama tabulasi, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian tabulasi yang ingin diganti namanya.
2. Pilih perintah Rename Tab.
3. Ketikkan nama tabulasi yang baru dan klik tombol OK.

4. Memindah Posisi Tabulasi

Untuk meminndah posisi tabulasi, dapat menggunakan prosedur berikut:

1. Klik kanan pada bagian tabulasi yang ingin Anda pindah posisinya.
2. Pilih perintah Move Up untuk memindah posisi ke atas atau Move Down untuk memindah posisi ke bawah.

5. Menambahkan Item Kontrol

Untuk menambahkan item kontrol, dapat menggunakan prosedur berikut:

1. Aktifkan tabulasi untuk menambahkan item pada bagian kontrol tersebut dengan cara klik.
2. Berikan salah satu perintah berikut:
   a. Pilih perintah Project - Components.
   b. Klik kanan pada bagian tabulasi toolbox dan pilih perintah Components.
   c. Shortcut key Ctrl + y
3. Pada tabulasi Controls, aktifkan kotak periksa dari beberapa pilihan kontrol item yang ingin ditambahkan dan klik tombol Close.

BAB III

PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Peralatan ini membutuhkan perangkat keras dan perangkat lunak. Perangkat keras sendiri terdiri dari pembaca *chip card* yaitu *chip card reader*, mikrokontroler dan *Personal Computer* (PC). Untuk menyimpan database. Mikrokontroler AT89S51 untuk pengendalian masukan dan keluaran yang dimanfaatkan pada LCD yang fungsinya sebagai penampil tulisan yang sudah diprogram, begitu juga pada keypad sebagai perintah untuk memasukan PIN serta sebagai penerima data dari *chip card reader* untuk dikirim ke PC. Sedangkan program aplikasi pada PC digunakan Microsoft Visual Basic V6.0 untuk menerima dan menyimpan data dari *chip card reader* di PC, sedangkan untuk mengaktifkan mikrokontroler menggunakan bahasa *Assembler*.

Perancangan dan pembuatan alat ini membutuhkan sebuah kartu (*chip card*), pembaca kartu (*chip card reader)* dan Mikrokontroler AT89S51. Kartu disini berfungsi sebagai identitas dosen sebagai pengajar di kampus ITN – Malang , dimana berisi sebuah PIN yang disimpan dalam IC AT24C64.

3.1. Perencanaan Perangkat Keras dan Perangkat Lunak

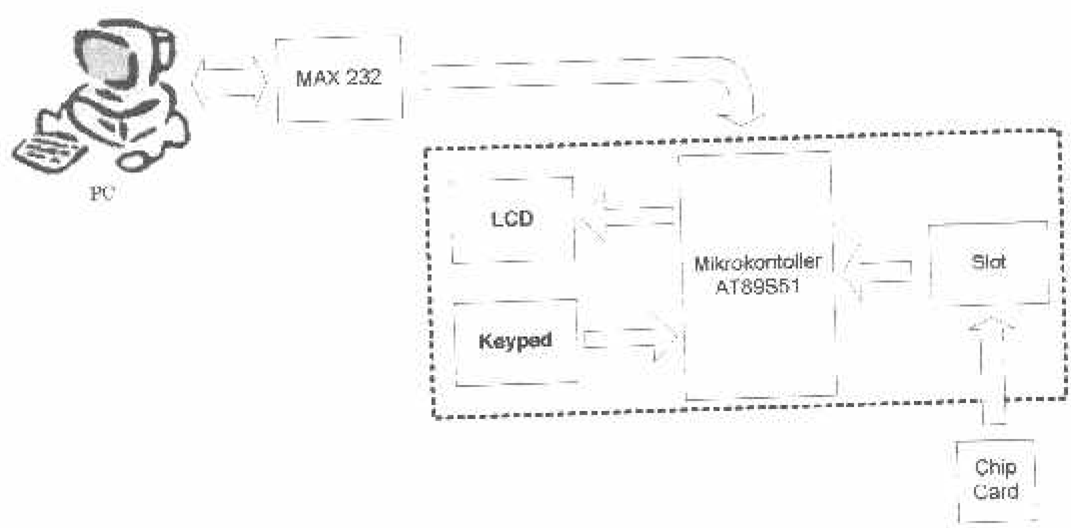
Perencanaan *hardware* dan *software* ini meliputi perencanaan minimum sistem berbasis mikrokontroler AT89S51 dan *flowchart* program.

3.2.1. Perencanaan Perangkat Keras

Minimum sistem ini terdiri atas mikrokontroler jenis AT89S51 dan komponen pendukung lainnya agar sistem dapat bekerja dengan optimal, seperti *latch* tipe 74LS573, *decoder* tipe 74LS138 dan PPI 8255.

Latch berfungsi untuk memisahkan bit – bit data dan bit – bit alamat yang termultipleks. Sedangkan *decoder* untuk membagi peta alamat komponen yang digunakan.

Berikut adalah diagram blok dari perangkat keras :



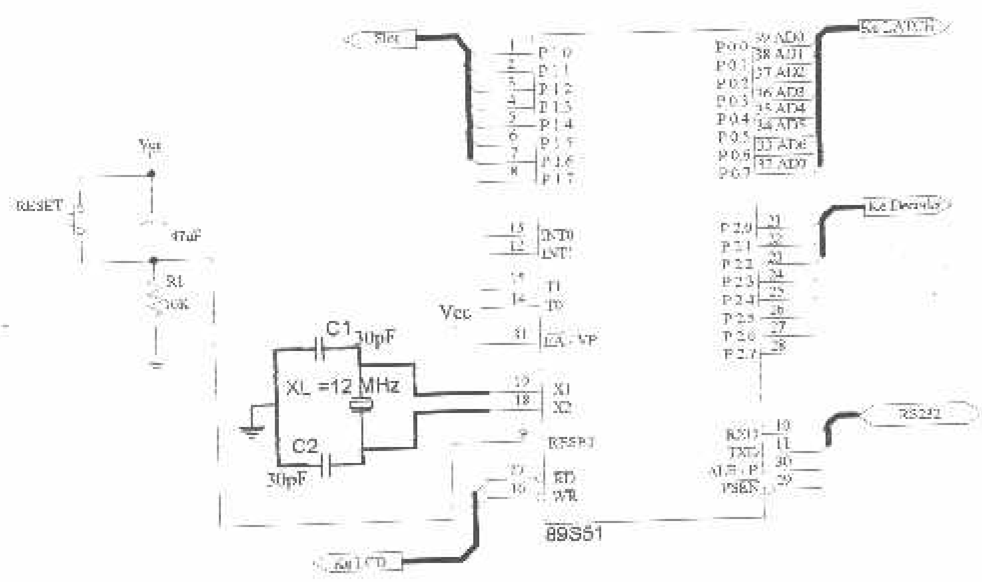
Gambar 3 – 1
Blok Diagram Sistem
Sumber : *Perancangan*

- PC digunakan sebagai media penyimpanan data.
- MAX 232 untuk mengubah level tegangan TTL ke level tegangan komputer.
- Mikrokontroler AT89S51 digunakan sebagai pengendali masukan dan keluaran.

- LCD digunakan untuk menampilkan data yang sudah dibaca dan diterima oleh PC.
- *Chip Card* sebagai kartu identitas dosen yang berisi nomor PIN

3.2.1.1. Mikrokontroler AT89S51

Mikrokontroler AT 89S51 harus didukung oleh beberapa rangkaian lain agar dapat melakukan prosesnya, yaitu berupa rangkaian *clock* dan *reset*. Selain itu juga harus ditentukan dalam penggunaan port – portnya dan sinyal – sinyal yang digunakan mendukung proses yang dilakukan. Rangkaian moikrokontroler AT89S51 yang akan diantarmukakan ke keypad, modul kartu dosen yang berisi EEPROM 24C64, modul LCD dan modul MAX232. Pada masing – masing blok memiliki rangkaian yang sama. Berikut adalah gambar 3 – 2 rangkaian mikrokontroler adalah sebagai berikut :



Gambar 3 – 2
Rangkaian Mikrokontroler AT89S51
Sumber : *Perancangan*

Dalam sistem mikrokontroler ini direncanakan penggunaan port yang tersedia sebagai berikut :

1. Port 1.0 – 1.7 sebagai jalur slot untuk *chip reader*.
2. Port 0.0 – 0.7 sebagai jalur untuk *latch* 74LS573.
3. Port 2.5 – 2-7 sebagai jalur data untuk *decoder* 74LS138.
4. Port 3.6 – 3.7 sebagai jalur data untuk pembaca dan menulis modul LCD.
5. Port 1.0 –1.1 sebagai jalur data dan clock dari kartu dosen (EEPROM AT 24C64).
6. Port 3.0 (RXD) dan Port 3.1 (TXD) sebagai jalur komunikasi serial dalam jaringan RS-232 dan sinyal – sinyal kontrol untuk RS-232.

Hal – hal itu adalah sebagai berikut :

- **Clock**

  Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber clock (pewaktuan) yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini akan menggunakan osilator internal yang sudah tersedia di dalam chip AT89S51. Untuk menentukan frekuensi osilatornya, cukup dengan cara menghunungkan kristal pada pin XTAL1 dan XTAL2 serta dua buah kapasitor ke ground. Dengan menggunakan crystal diatas maka dapat dihitung waktu yang diperlukan untuk satu siklus mesin.

  f = 12 Mhz

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

$$T = \frac{1}{12Mhz} = \frac{1}{12}\ \mu s$$

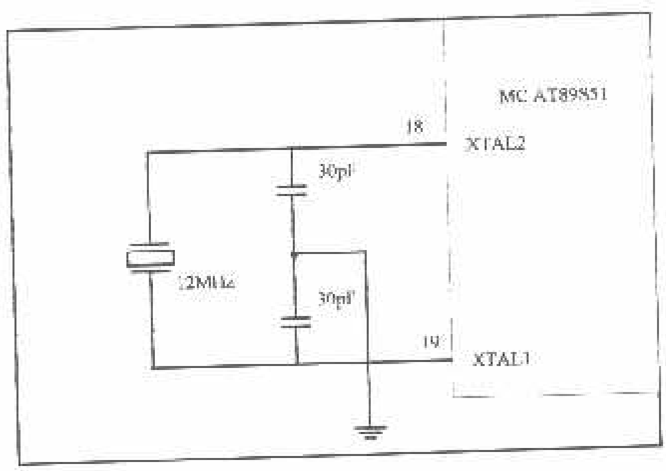
Maka untuk satu siklus mesin dari mikrokontroler besarnya adalah :

Tme = 12 x T

$$Tme = 12 \times \frac{1}{12}\ \mu s$$

Tme = 1 μs

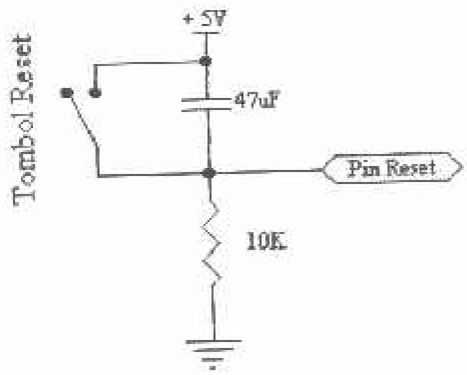
Besar *Crystal* disesuaikan dengan kecepatan yang diharapkan untuk transfer data melalui pin *serial interface* AT89S51 tersebut. Sistem ini dirancang untuk memiliki kemampuan *baud rate* sebesar 9600bps, sehinggan dipilih kristal dengan nilai 12 MHz sesuai dengan spesifikasi dalam Gambar 3 - 3. memperlihatkan rangkaian rangkaian pewaktu yang digunakan.



Gambar 3 – 3
Rangkaian Pewaktu
Sumber : *www.atmel.com*

- **Rangkaian Reset**

Reset pada mikrokontroler merupakan masukkan aktif High '1' Pulsa transisi dari rendah '0' ke tinggi '1' akan mereset mikrokontroler menuju alamat 0000H. Pin reset dihubungkan dengan rangkaian power on reset seperti pada gambar 3 – 4.



Gambar 3 – 4
Rangkaian Power On Reset
Sumber : *www.atmel.com*

Rangkaian reset bertujuan agar mikrokontroler dapat menjalankan proses dari awal. Rangkaian reset untuk mikrokontroler dirancang agar mempunyai kemampuan power on reset, yaitu reset yang terjadi pada saat sistem dinyalakan untuk pertama kalinya. Reset juga dapat dilakukan secara manual dengan menekan tombol reset yang berupa switch push button.

Rangkaian Reset terbentuk oleh komponen R dan C yang sudah baku (ditetapkan oleh perusahaan pembuat IC AT 89S51). Nilai R yang dipakai adalah 10 kΩ dan C 47 μF.

Sedangkan untuk mencari frekuensi dari reset tersebut dengan menggunakan rumus sebagai berikut :

$$fo = \frac{1}{1{,}1R.C}$$

Sehingga dengan komponen resistor dengan nilai 10 Kohm serta kapasitor dengan nilai 47 uF akan dihasilkan frekuensi.

$$fo = \frac{1}{1{,}1.RC}$$

$$= \frac{1}{1{,}1.10^3.47^{-6}}$$

$$fo = 1{,}93Hz$$

Maka Periode Clock $T = \frac{1}{fo}$

$$T = \frac{1}{1{,}93}$$

$= 0{,}52$ detik

- **Pembagian Port**

Mikrokontroler AT89S51 mempunyai 4 buah port yaitu port 0 (P0), port 1 (P1), port 2 (P2) dan port 3 (P3). Mikrokontroler memiliki EPROM internal, maka port 0 digunakan untuk saluran data dan alamat bagian rendah yang dimultipleks (AD0 – AD7)dan alamat bagian tinggi dikeluarkan pada port 2 (A8 – A15). Pada port 3, yang berfungsi sebagai bit kontrol RD dan WR adalah menggunakan rangkaian PPI 8255, sedangkan untuk bit kontrol Tx dan Rx terletak pada komunikasi serial pada sistem mikrokontroler Dalam perancangan port 1 digunakan sebagai masukan dari *keypad*.

- **Sinyal kontrol Memory Eksternal**

Untuk mengakses memori eksternal yang digunakan pada perancangan alat ini, maka harus digunakan sinyal kontrol RD, WR, ALE, PSEN dan EA. Sinyal – sinyal tersebut digunakan untuk mengontrol proses pemindahan data mikrokontroler. Untuk mengakses memori program diperlukan sinyal PSEN yang digunakan untuk mengeksekusi program yang tersimpan pada EPROM. Selain itu harus diperhatikan juga sinyal EA dan ALE. EA harus dihubungkan dengan Vcc (logika 0) yang berarti bahwa memori eksternal diakses mulai alamat 0000H sampai FFFFH, atau dengan kata lain pengaksesan seluruh alamat dilakukan di luar chip AT89S51. sinyal ALE berfungsi untuk memisahkan bit – bit data dan bit – bit alamat rendah yang dimultipleks pada port 0.

3.2.1.2. Decoder dan Peta Alamat

Proses aliran data yang diperlukan oleh mikrokontroler tergantung dari pengalamatan yang dipilih mikrokontroler tersebut. Untuk memilih komponen mana yang diakses maka diperlukan suatu rangkaian decoder. Decoder ini akan dihubungkan dengan alamat tinggi yang dikeluarkan mikrokontroler.

Peta alamat pada sistem ini direncanakan dibagi perdelapan kilo byte, yang berarti bahwa alamat mikrokontroler yang tersedia sebesar 64 kilo byte dibagi menjadi 8 blok. Masing – masing blok dipilih dengan menggunakan 3 bit alamat tinggi yang dikeluarkan melalui port 2. pada Tabel 3 - 1. memperlihatkan pemetaan alamat pada sistem ini.

Tabel 3 – 1
Pemetaan Alamat
Sumber : www.faichildsemi.com

Chip Select	**Alamat**	**Fungsi**
CS0	0000-1FFF	-
CS1	2000-3FFF	-
CS2	4000-5FFF	-
CS3	6000-7FFF	-
CS4	8000-9FFF	-
CS5	A000-BFFF	LCD
CS6	C000-DFFF	-
CS7	E000-FFFF	PPI 8255

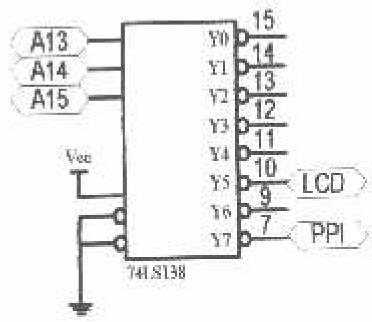
Blok-blok yang masih kosong dapat dimanfaatkan untuk penambahan memori atau untuk keperluan lainnya. Pemilihan alamat-alamat ini disesuaikan dengan keperluan dan perencanaan yang dapat dilakukan untuk pengembangan berikutnya.

Dekoder yang dipergunakan adalah dekoder 74LS138, suatu dekoder 3 ke 8. Tabel fungsinya seperti terlihat dalam Tabel 3 – 2.

Tabel 3 – 2
Tabel dekoder 74LS138
Sumber : www.fairchildsemi.com

INPUT					OUTPUT							
ENABLE		SELECT										
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	H	H	H	H	H	L	H	H	H	H	H
H	L	H	H	H	H	H	H	L	H	H	H	H
H	L	H	H	H	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Dekoder ini dikondisikan selalu membaca data masukannya setiap saat, sehingga pin G1 diberi logika tinggi dan pin G2A, G2B dihubungkan dengan pentanahan. Saluran alamat yang digunakan untuk memilih alamat diambil dari mikrokontroler pada bit A13, A14 dan A15, dihubungkan dengan pin A, B dan C pada dekoder. Rangkaian dekoder ditunjukkan dalam Gambar 3 – 5.



Gambar 3 – 5
Rangkaian Decoder 74LS138
Sumber : www.fairchildsemi.com

3.2.1.3. Latch

Latch digunakan untuk memisahkan antara bit-bit data dengan bit-bit alamat rendah yang dikeluarkan pada port 0 oleh mikrokontroler AT89S51. Tipe yang akan dipakai adalah tipe 74LS573. Latch ini merupakan latch berdelapan tipe D transparan yang berarti bahwa saat enable dan C dalam keadaan logika tinggi maka keluaran Q akan mengikuti masukan data D. Bila enable diberi logika rendah maka keluaran akan dilatch pada level data yang telah diset sebelumnya (Q0). Tabel fungsi 74LS573 dapat dilihat dalam Tabel 3 – 3.

Tabel 3 – 3
Tabel fungsi Latch 74LS573
Sumber : www.fairchildsemi.com

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q0
H	X	X	Z

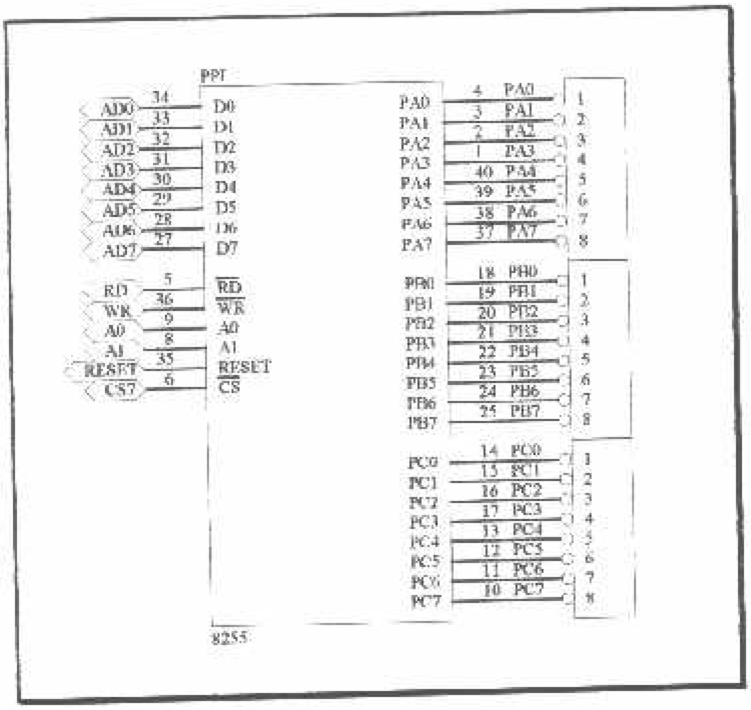
Pin enable C pada latch ini dihubungkan dengan sinyal ALE yang dikeluarkan oleh mikrokontroler. Sedangkan saluran masukan latch dihubungkan dengan port 0 yang mengeluarkan bit-bit data dan alamat rendah.

Sinyal ALE berlogika tinggi maka bit-bit yang dikeluarkan melalui port 0 adalah bit-bit alamat, dan 74LS573 akan meneruskan bit-bit ini ke keluarannya yang terhubung dengan saluran alamat komponen yang dituju. Apabila sinyal ALE dalam keadaan logika rendah maka 74LS573 menahan bit-bit tersebut sehingga bit-bit ini akan dibaca sebagai data oleh komponen yang terhubung.

3.2.1.4. PPI 8255

Mikrokontroler AT89S51 memiliki 4 buah port, yaitu port 0, port 1, port 2 dan port 3. Tetapi karena tidak mempunyai EPROM internal, maka untuk mengakses EPROM eksternal, port 0 dan port 2 difungsikan menjadi bus alamat dan bus data. Sementara itu beberapa bit dari port 3 juga memiliki fungsi lain yaitu sinyal RD, WR, TxD, RxD, INT0, INT1, T0 dan T1. Jadi port bebas yang tersisa adalah port 1. Untuk menambah jumlah port digunakanlah PPI 8255.

Dalam perencanaan, PPI 8255 ditempatkan pada alamat E000H-E003H. Pin A0 dan A1 PPI yang dipergunakan untuk memilih register yang diakses dihubungkan ke A0 dan A1 sistem mikrokontroler. Sementara pin CS IC 8255 dihubungkan ke CS7 (keluaran dari *address decoder* 74LS138). Rangkaian PPI 8255 seperti terlihat dalam Gambar 3 - 6.



Gambar 3 – 6
PIN PPI 8255
Sumber : Datasheet Harris Semiconductor

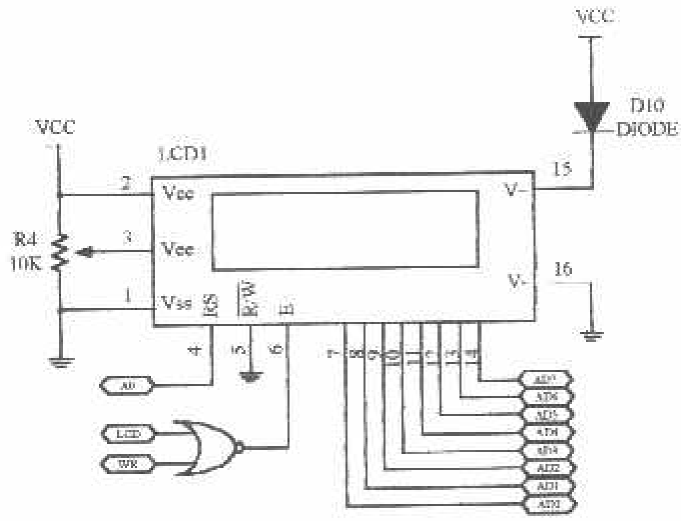
3.2.1.5. LCD

Pada sistem yang direncanakan akan digunakan LCD (Liquid Crystal Display) sebagai tampilan. LCD yang digunakan adalah jenis TM162ABC yang merupakan LCD dua baris dengan tiap barisnya terdiri dari 16 karakter.

LCD ini membutuhkan 3 sinyal kontrol, R/W (*read/write*) untuk menentukan apakah data akan dibaca atau ditulis, E (*Enable*) yang merupakan sinyal untuk meng-enable-kan dan RS (*Register Select*) untuk memilih register yang diakses. LCD TM162ABC memiliki 2 register yaitu register data dan register instruksi.

Dalam sistem ini, LCD menempati ruang alamat A000H-A001H. Pin R/W dihubungkan ke *ground* atau selalu berlogika 0 karena dalam perancangan.

LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin A0 sistem mikrokontroler. Pengaktifan LCD ini selanjutnya tergantung pada pin E. Dimana pin E ini tergantung dari CS5 dari addres dekoder dan perintah write mikrokontroler. Rangkaian LCD seperti terlihat dalam Gambar 3 - 7.

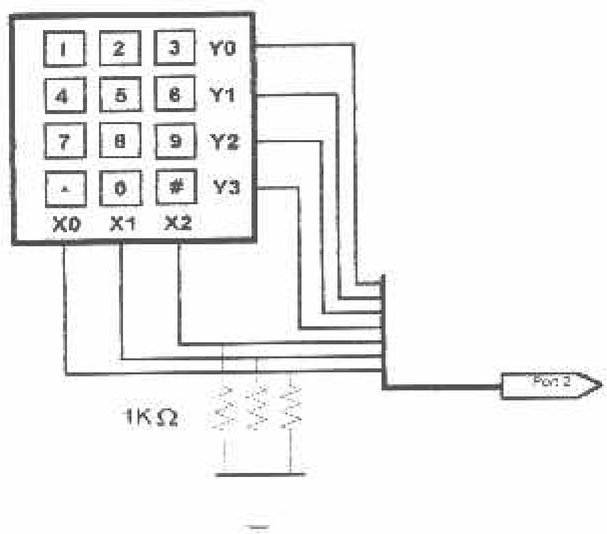


Gambar 3 – 7
Rangkaian LCD LM162ABC
Sumber : LCD Module

3.2.1.6. Papan Tombol (*Keypad*)

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port* C PPI 8255.

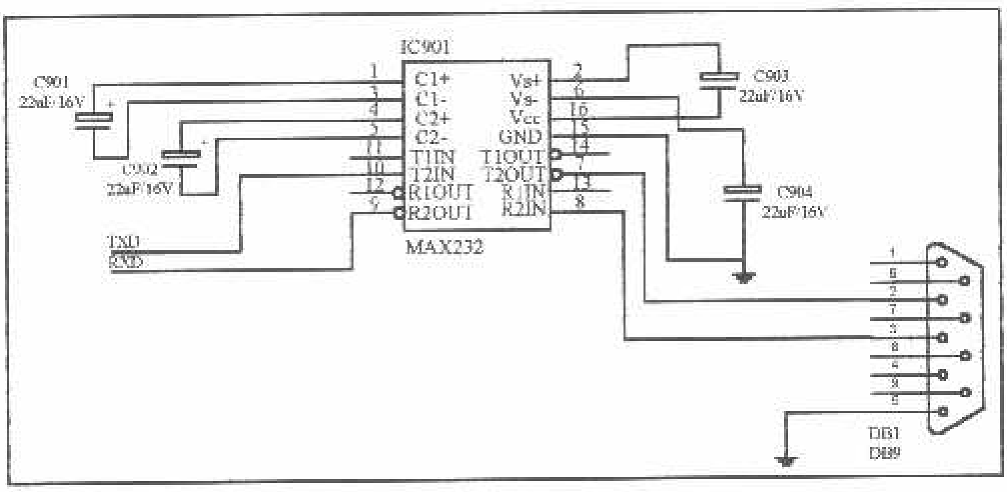
Papan tombol tersebut mempunyai matrik 4 baris dan 4 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port* C PPI 8255 yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan *ground* (berlogika 0) dan *port* C (PC4-PC7) yang difungsikan sebagai *input* mikrokontroler. Sedangkan deretan baris dihubungkan ke *port* C (PC0-PC3) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri. Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. *Port* ini difungsikan sebagai *output* dari mikrokontroler. Dengan demikian kalau tombol tidak ditekan maka masukan *port* C (PC4-PC7) di pin yang terhubung tombol tersebut berlogika 0 dan bila tombol ditekan akan berlogika 1. Rangkaian papan tombol tersebut dapat dilihat dalam Gambar 3 - 7



Gambar 3 – 8
Rangkaian Papan Tombol (*keypad*)
Sumber : Perancangan

3.2.1.7. Interface Unit RS232

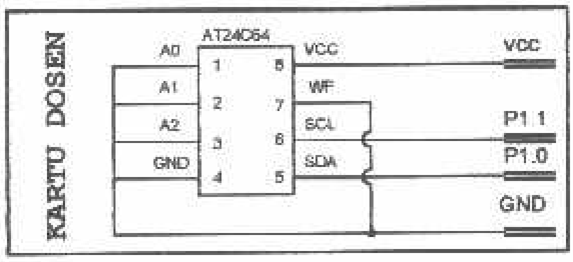
Data yang keluar masuk port serial komputer menggunakan standar komunikasi serial RS232. Pada standart komunikasi serial sesuai dengan spesifikasi standart RS 232-C, level tegangan yang digunakan adalah level tegangan yang berkisar antara –3 volt hingga 15 volt untuk kondisi logika '1' atau yang disebut dengan keadaan *mark* dan antara +3 volt hingga +15 volt untuk kondisi logika '0' atau disebut dengan keadaan *space* atau dengan kata lain standart RS 232-C menggunakan logika negatif/terbalik. Untuk dapat digunakan pada RS485, dalam membentuk jaringan komunikasi data antara komputer sentral dengan PPDS, level tegangan RS232 dari komputer harus diubah menjadi level tegangan TTL karena masukan DI (*driver input*) RS485 menggunakan level tegangan TTL. Begitu pula jika RS485 akan menyampaikan data ke komputer, maka level tegangan data tersebut harus diubah dulu sesuai standar RS232. Nilai kapasitor yang digunakan telah ditentukan sebesar 1 uF sesuai dengan *datasheet* MAX232. Rangkaian untuk *Interface Unit RS232* ditunjukkan pada Gambar 3 - 9.



Gambar 3 – 9
Rangkaian RS232
Sumber : www.maxim-ic.com

3.2.1.8. Perancangan Kartu

Dalam perencanaan dan pembuatan alat ini, digunakan EEPROM AT24C64 sebagai data penyimpan data dosen dalam chip card. Data yang disimpan di memori EEPROM nantinya bisa diganti nomor PIN secara berulang – ulang yang nantinya dapat diketahui oleh si pembawa kartu tersebut. Apabila kartu tersebut kita masukan pada slot sebagai chip reader maka secara otomatis oleh PROM di MCU. Gambar 3 - 8 dibawah ini adalah rangkaian dari kartu dosen



Gambar 3 – 10
Rangkaian Kartu Dosen
Sumber : Perancangan

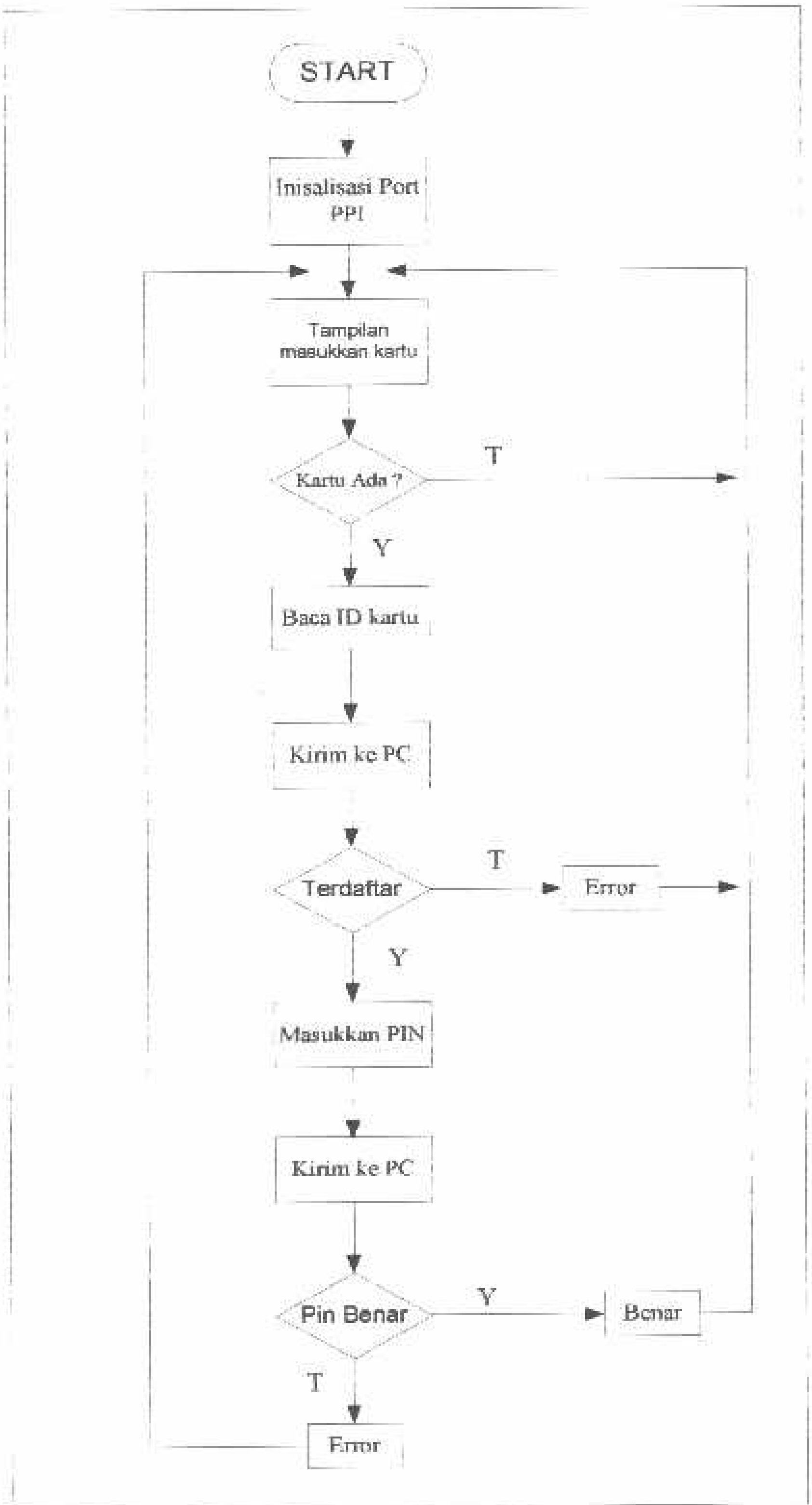
3.2.2. Perancangan Software

Perancangan software disini mengunakan bahasa pemrograman menggunakan bahasa C untuk memprogram mikrokontroler dan Microsoft Visual Basic versi 6.0 untuk menampilkan data dan menyimpan data pada komputer.

3.2.2.1. Software Mikrokontroller

untuk pemakaian mikrokontroler di dalam suatu sistem, perlu direncanakan perangkat lunak mikrokontroler yang dapat mengatur sistem tersebut. Perangkat lunak disini adalah susunan perintah – perintah (program) di dalam memori yang harus dilaksanakan oleh mikrokontroler.

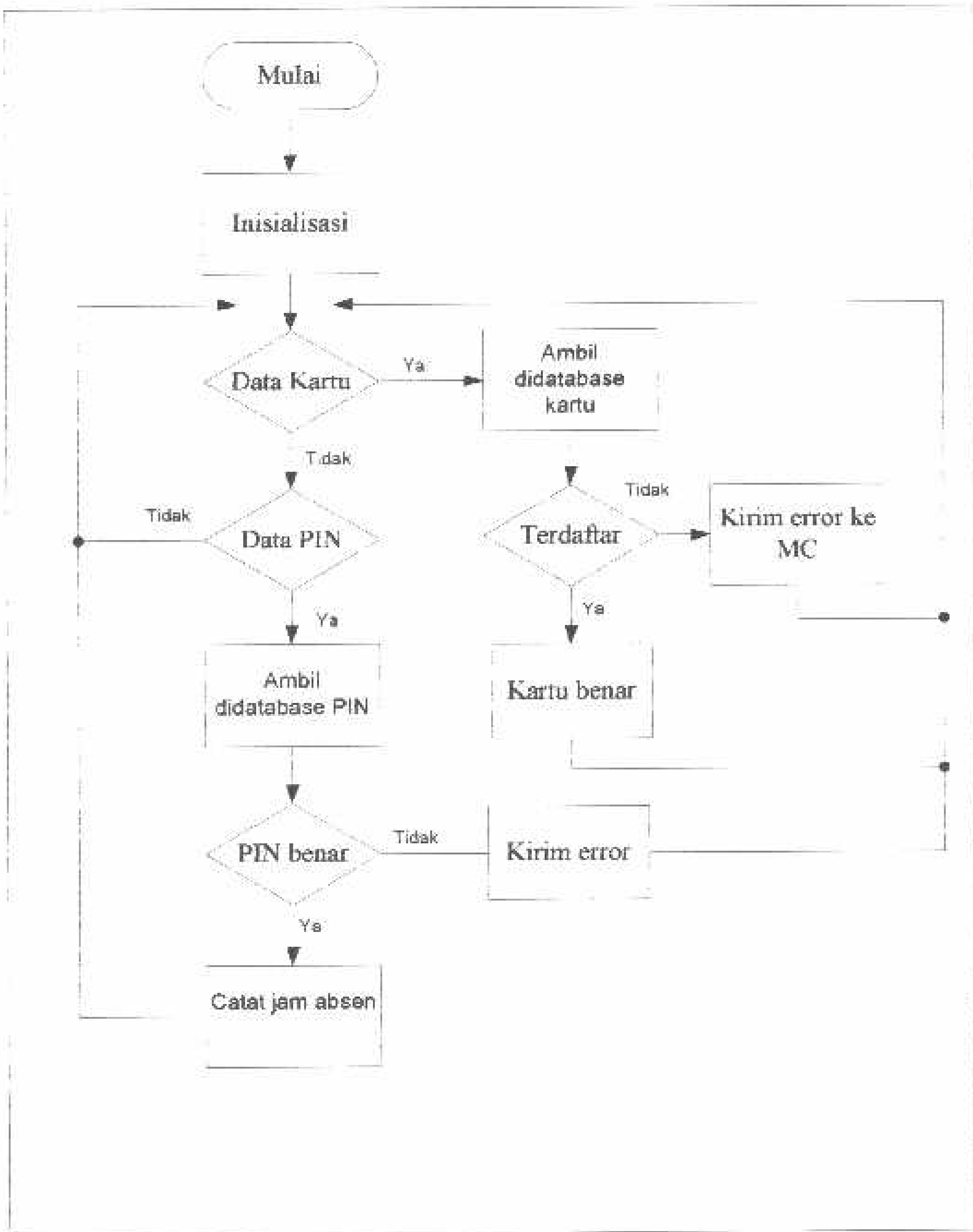
Perencanaan perangkat lunak (*software*) didsarkan perencanaan perangkat keras yang telah dibuat sebelumnya, untuk mendapatkan sistem kerja yang diharapkan. Software dari alat tersebut terdapat dibagian lampiran dan diagram alirnya adalah sebagai berikut



Gambar 3 – 11
Flowchart Mikrokontroler
Sumber : Perancangan

3.2.2.2. Software Tampilan Komputer

Perancangan software tampilan dan database digunakan Microsoft Visual Basic V6.0.



Gambar 3 – 12
Flowchart Software Microsoft Visual Basic V6.0
Sumber : Perancangan

BAB IV

PENGUJIAN ALAT

4.1. Umum

Pada bab ini membahas tentang pengujian alat yang telah dibuat. Secara umum, pengujian ini bertujuan untuk mengetahui apakah alat dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pengujian dilakukan pada masing – masing blok diagram sesuai pada gambar 3 – 1 dan juga pengujian alat secara keseluruhan.

4.2. Pengujian Rangkaian Sistem Mikrokontroller

4.2.1. Tujuan

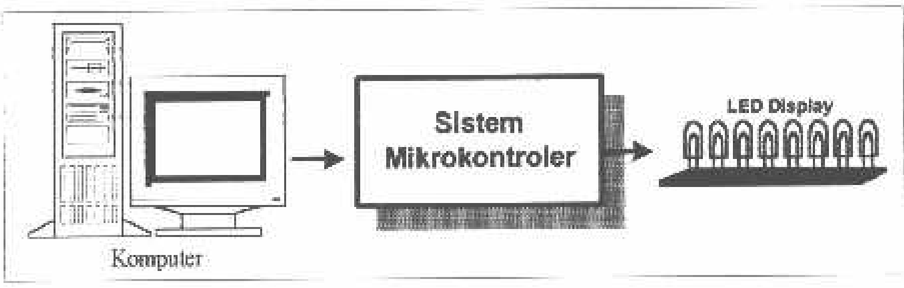
Untuk mengetahui apakah mikrokontroller dapat melaksanakan program yang tersimpan dalam *flash memory* dan RAM dengan benar.

4.2.2. Peralatan yang digunakan

1. *Personal Computer* (PC).
2. Sistem mikrokontroler
3. Led Display.

4.2.3. Prosedur Pengujian

1. Merangkai peralatan yang digunakan sesuai Gambar 4 - 1. Sistem mikrokontroler terdiri atas mikrokontroler AT89S51.



Gambar 4 – 1
Diagram Blok Pengujian Mikrokontroler

2. Membuat program *assembler* yang digunakan dalam pengujian mikrokontroler ii meruapak program sederhana yang meletakkan 0FH dan F0H pada ACC secara bergantian kemudian memindahkannya pada port 1 AT89S51. Program yang dibuat adalah sebagai berikut :

```
ORG    0000H
              a. JMP        START
    ii. START :  MOV        A,#0FH
              a. MOV        P1, A
              b. CALL       TUNDA
              c. MOV        A, #F0H
              d. MOV        P1, A
              e. JMP        START
    iii. TUNDA  :     MOV         R3, #0FFH
    iv. TUNDA1 :      MOV         R2, #0FFH
              a. DJNZ       R2, $
```

b. MOV R1, #0FH

c. DJNZ R1, $

d. DJNZ R3, TUNDA1

e. REF

f END

3. Memasang catu daya rangkaian sebesar 5 volt
4. Mendownload program diatas
5. Mengamati keluaran pada LED Display.

4.2.4. Hasil Pengujian Mikrokontroler

Hasil pengujian ditunjukkan dalam tabel 4 – 1.

Tabel 4 – 1
Diagram blok Pengujian Mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Satu	1	1	1	1	0	0	0	0
Dua	0	0	0	0	1	1	1	1

Dari hasil pengujian dalam tabel 4 – 1 dapat dilihat bahwa *port* 1 memberikan logika $0F_H$ dan $F0_H$ secara bergantian sesuai dengan isi program.

4.3. Pengujian LCD

4.3.1. Tujuan

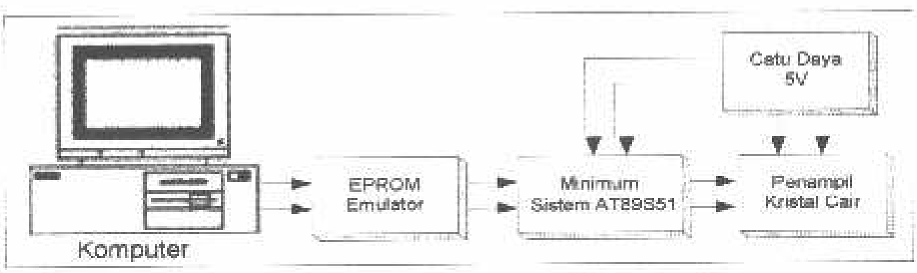
Untuk mengetahui kemampuan rangkaian tampilan yang sudah dibuat apakah dapat mendukung sistem yang direncanakan untuk menampilkan data pada *LCD*.

4.3.2. Peralatan yang digunakan :

1. *Personal Computer*.
2. Minimum sistem mikrokontroler AT89S51.
3. Catu daya 5V DC.

4.3.3. Prosedur Percobaan

1. Merangkai peralatan seperti dalam Gambar 4 - 2.



Gambar 4 – 2
Diagram blok Pengujian LCD

2. Membuat perangkat lunak pengujian rangkaian LCD. Program ini berisi inisialisasi mikrokontroler, PPI dan LCD.
3. Mengaktifkan catu daya
4. Mengoperasikan program dengan bantuan EPROM emulator. Hasil keluaran akan ditunjukkan pada layar penampil kristal cair.

4.3.4. Hasil Pengujian

Dari hasil pengujian didapatkan bahwa rangkaian LCD dapat menampilkan karakter-karakter, sesuai dengan data yang dikirimkan oleh EPROM *Emulator*. Tampilan penampil kristal cair terdiri atas 2 baris yang masing-masing mempunyai 16 karakter.

Gambar 4 – 3
Hasil Pengujian LCD

4.4. Pengujian Papan Tombol (*Keypad*)

4.4.1. Tujuan

Untuk mengetahui konfigurasi logika keluaran dari unit papan tombol saat tombol ditekan.

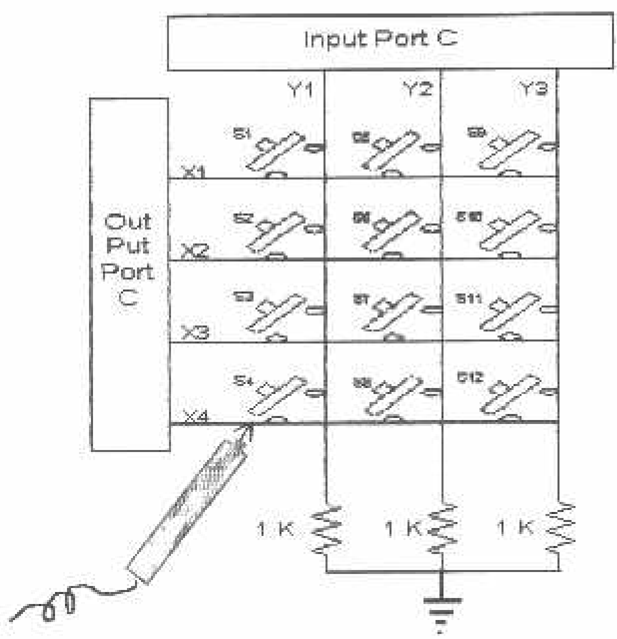
4.4.2. Peralatan Yang Digunakan

1. Catu daya 5V DC.
2. Unit papan tombol.
3. Logic probe.

4.4.3. Prosedur Pengujian

1. Menyiapkan catu daya dengan keluaran 5 V DC, rangkaian papan tombol seperti dalam bab 3 dan *logic probe*.

2. Menyusun rangkaian pengujian papan tombol seperti terlihat dalam Gambar 4-4 serta memastikan bahwa hubungan antar pin pada masing-masing saklar telah benar.
3. Menjalankan program pengujian keypad, amati keluaran pin Port C bagian output. Langkah ini dilakukan dalam keadaan tidak ada penekanan tombol sama sekali.
4. Menekan sembarang tombol kemudian amati keluaran pin Port C bagian output.



Gambar 4 – 4
Diagram Blok Papan Tombol (*keypad*)

Dalam pengujian ini keluaran yang diamati adalah proses *scanning* yang terjadi pada lajur baris dan kolom. Lajur baris merupakan bagian output sedangkan lajur kolom merupakan bagian input. Untuk mengetahui kebenaran rangkaian keypad yang telah dibuat maka keluaran dari rangkaian keypad ini akan ditampilkan ke port 1 MCU AT89S51.

4.4.4. Hasil Pengujian

Hasil pengujian ditunjukkan dalam Tabel 4-2.

Tabel 4 – 2
Hasil Pengujian Papan Tombol (*keypad*)

Input Tombol yang ditekan		Output Port 1			
		D	C	B	A
Nomor tombol	Definisi tombol				
-	-	X	X	X	X
(S1)	1	0	0	0	1
(S2)	4	0	1	0	0
(S3)	7	0	1	1	1
(S4)	#	1	0	1	0
(S5)	2	0	0	1	0
(S6)	5	0	1	0	1
(S7)	8	1	0	0	0
(S8)	0	0	0	0	0
(S9)	3	0	0	1	1
(S10)	6	0	1	1	0
(S11)	9	1	0	0	1
(S12)	*	1	0	1	1

Dari hasil pengujian, didapatkan data seperti dalam Tabel 4-2, maka dapat diketahui bahwa saat tombol ditekan maka keluaran port 1 mikrokontroler AT89S51 akan berlogika sesuai dengan tombol yang ditekan. Hasil pengujian dalam Tabel 4-2 terlihat bahwa rangkaian papan tombol yang telah direalisasikan sesuai dengan unjuk kerja perencanaan.

4.5. Pengujian PPI 8255

4.5.1. Tujuan

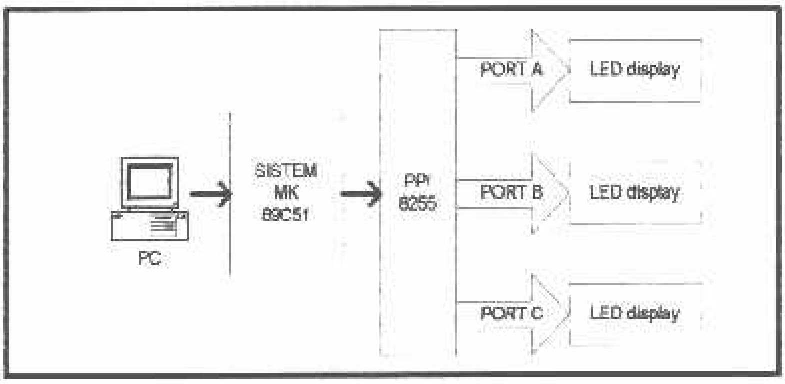
Untuk mengetahui unjuk kerja dari PPI 8255.

4.5.2. Peralatan Yang Dibutuhkan

1. Komputer.
2. Sistem mikrokontroller AT89C51.
3. PPI 8255.
4. LED

4.5.3. Prosedur Pengujian

1. Untuk menguji PPI, sistem mikrokontroller yang sudah diuji dihubungkan dengan PPI dan keluaran dari masing-masing port PPI dihubungkan dengan LED *display*.
2. Merangkai rangkaian seperti pada iagram blok untuk pengujian PPI diperlihatkan dalam Gambar 4-5.
3. Bila PPI dapat bekerja dengan baik, maka LED *display* akan menyala menunjukkan kode biner dari masukan.



Gambar 4-5
Diagram Blok Pengujian PPI 8255

4.5.4. Hasil Pengujian

Hasil pengujian dari PPI ditunjukkan dalam Tabel 4-3. Dalam tabel tersebut, nilai 1 menunjukkan bahwa LED dalam keadaan menyala, sedangkan nilai 0 menunjukkan LED dalam keadaan mati.

Tabel 4-3
Hasil Pengujian PPI 8255

No.	Masukan	Keluaran pada LED display							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	FFh	1	1	1	1	1	1	1	1
2	F0h	1	1	1	1	0	0	0	0
3	84h	1	0	0	0	0	1	0	0
4	0Eh	0	0	0	0	1	1	1	0

4.6. Pengujian Transfer Data Serial

4.6.1. Tujuan

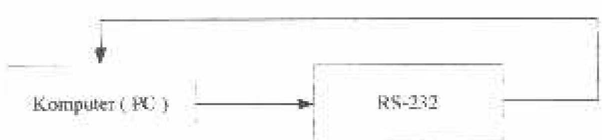
Untuk mengetahui apakah data yang dikirim mikrokontroler ke komputer dapat diterima dengan benar.

4.6.2. Peralatan yang digunakan

1. Komputer
2. Sistem mikrokontroler dengan antarmuka RS232 dan RS485.
3. EPROM *Emulator* EL-TECH Model EE-02.

4.6.3. Prosedur pengujian

1. Menyusun rangkaian seperti dalam Gambar 4 - 3
2. Membuat program transfer data pada sistem mikrokontroller seperti yang ditunjukkan dalam Gambar 4 – 4. Dengan program tersebut mikrokontroller hanya akan mengirim data 'MENCOBA', 'KIRIM', dan 'SERIAL' ke komputer.
3. *Download* program ke EPROM Emulator dan eksekusi program.
4. Mencatat hasil yang tampak dalam layar komputer.



Gambar 4-6
Rangkaian Penguji Transfer Data

```
;Program Komunikasi Serial

;*** SET BAUD RATE Dari TBL1 ***
B1200   EQU     232
PCON    EQU     87H
        ORG     0000H
        JMP     MULAI

        ORG     0023H
SERIAL: JBC     TI,OUT232
        PUSH    ACC
        MOV     A,SBUF
        MOV     SBUF,A
        POP     ACC
OUT232: CLR     TI
        CLR     RI
        RETI

DELAY:  PUSH    05H
        PUSH    06H
        PUSH    07H
        MOV     07H,#03H
DEL1:   MOV     06H,#0FFH
DEL2:   MOV     05,#0FFH
DEL3:   DJNZ    R5,DEL3
        DJNZ    R6,DEL2
        DJNZ    R7,DEL1
        POP     07H
        POP     06H
        POP     05H
        RET
WAIT9600:PUSH   05H
        PUSH    06H
        MOV     06H,#0F0H
WAIT156: MOV    05,#020H
WAIT256: DJNZ   R5,WAIT256
        DJNZ    R6,WAIT156
        POP     06H
        POP     05H
        RET

INIT232: MOV    TH1,#B1200
        MOV     TMOD,#22H
        ANL     PCON,#7FH
        MOV     SCON,#50H
        SETB    PS
        SETB    ES
        SETB    TR1
        SETB    EA
        CALL    DELAY
        RET

SEND232: PUSH   ACC
LOOP:   CLR     A
        MOVC    A,@A+DPTR
        CJNE    A,#'$',SENDNOW
        JMP     OUTSEND
SENDNOW: MOV    SBUF,A
        CALL    WAIT9600
        INC     DPTR
        JMP     LOOP
OUTSEND: POP    ACC
        RET

MULAI:  CALL    INIT232
        MOV     DPTR,#KOMEN1
        CALL    SEND232
        MOV     DPTR,#KOMEN2
        CALL    SEND232
        MOV     DPTR,#KOMEN3
        CALL    SEND232
HAB:    AJMP    HAB

KOMEN1: DB 'MENCOBA',13,'$'
KOMEN2: DB 'KIRIM',13,'$'
KOMEN3: DB 'SERIAL',13,'$'
        END
```

Gambar 4 – 7
Program Pengujian Transfer Data Mikrokontroler dengan menggunakan RS232

4.6.4. Hasil Pengujian

Hasil pengujian transfer data serial ini ditunjukkan Tabel 4 – 2.

Tabel 4 – 8
Hasil Pengujian Transfer Data Serial

No.	Data yg dikirim Mikrokontroler	No.	Data yg diterima Komputer
1.	MENCOBA	1.	MENCOBA
2.	KIRIM	2.	KIRIM
3.	SERIAL	3.	SERIAL

4.7. Pengujian Perangkat Lunak

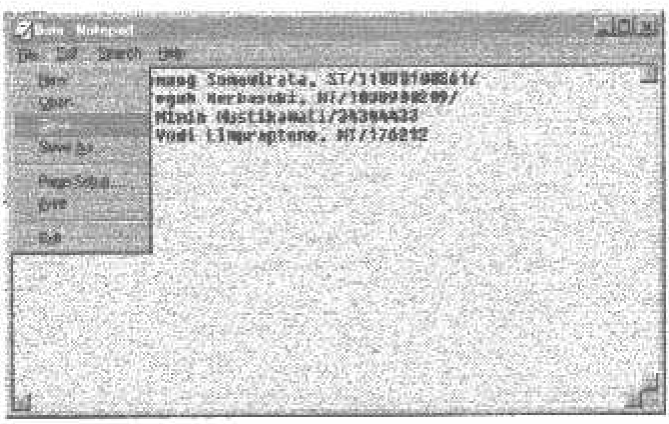
4.7.1. Pengujian Penyimpan Data

Pengujian pada penyimpanan data adalah dengan menambah data, edit data dan menghapus data dosen dengan menggunakan aplikasi *notepad*

Gambar 4 – 9
Tampilan Data Utama

4.7.1.1. Tambah Data

Penambahan data dosen dapat dilakukan setelah urutan sebelumnya dengan menggunakan aplikasi notepad kemudian "save". Pada tampilan menu utama klik "update data" maka akan tersimpan dalam database



Gambar 4 – 10
Tampilan Untuk Menambah Data

4.7.1.2. Edit Data

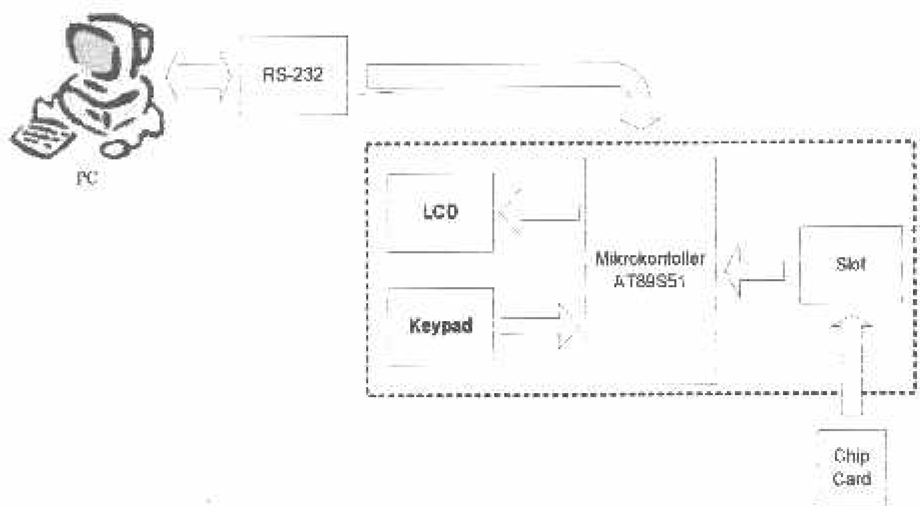
Edit data digunakan untuk mengganti data berupa Nama dan NIP. Sedangkan PIN dapat diganti pada saat kartu dimasukkan pada *Card ChipReader* dengan inputan dari komputer.

4.7.1.3. Hapus Data

Cara menghapus data dilakukan tidak seperti lainnya karena disini telah menggunakan aplikasi *notepad* yang dapat berjalan pada program Visual Basic Versi 6.0 yaitu tinggal menghapus kemudian disimpan. Kemudian klik update data.

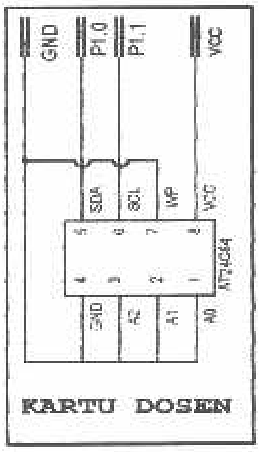
4.8. Pengujian Sistem Keseluruhan

Pengujian sistem kesuruhan dilakukan dengan membentuk suatu hubungan antara chip card reader, sistem rangkaian dan *Personal Computer (PC)*.



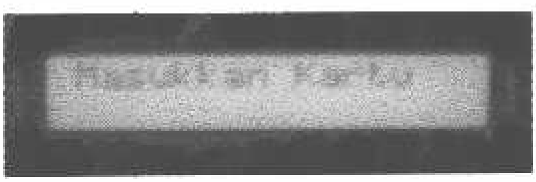
Gambar 4 – 11
Susunan Pengujian Keseluruhan

Chip card reader akan membaca kartu dosen yang sudah dilengkapi dengan IC AT24C64 sebagai penyimpan data PIN, Nama Dosen, NIP, berikut adalah kartu yang digunakan:

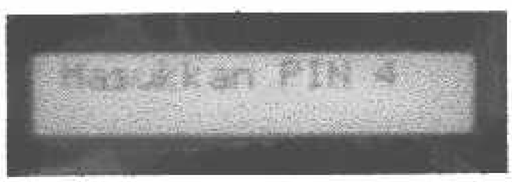


Gambar 4 – 12
Kartu Dosen

Setelah kartu dibaca oleh chip card reader, kemudian identitas kartu yang sudah berupa data diterima oleh mikrokontroller untuk ditampilkan di LCD terlebih dahulu.



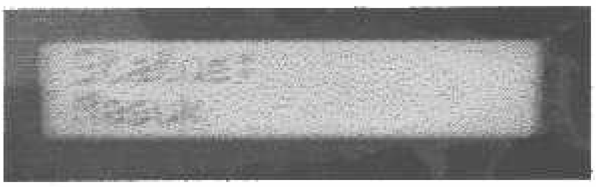
(a) Sebelum ada data masuk



(b) Sesudah Data Diterima Dari Chip Card Reader

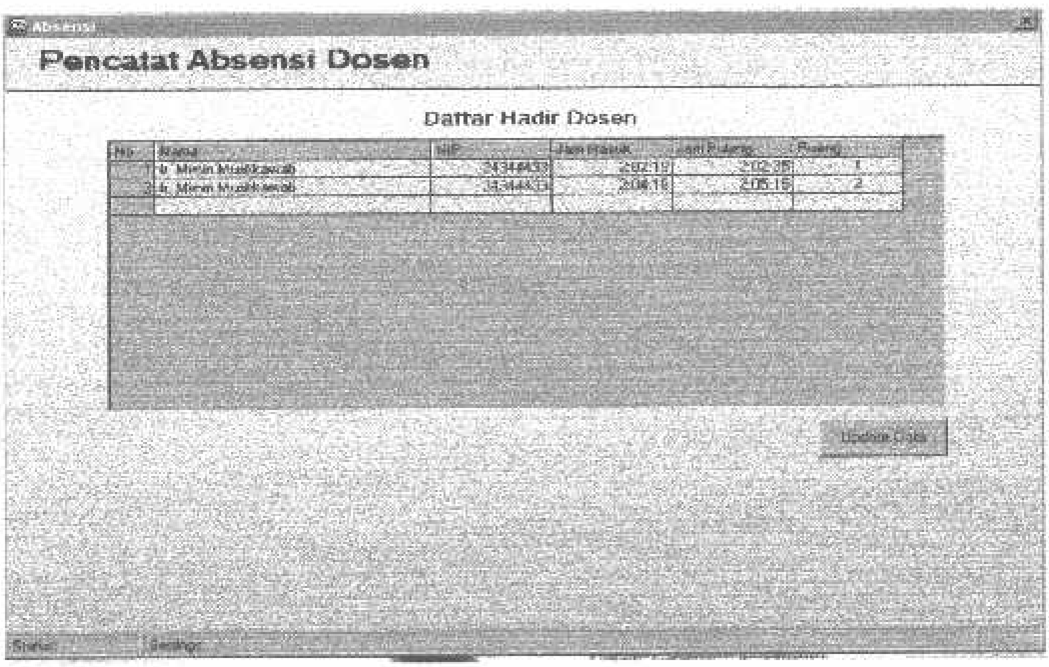
Gambar 4 – 13
Tampilan Pada LCD

Berikutnya data akan dikirim ke komputer untuk dicari databasenya, bila data ditemukan maka akan mencatat waktu masuk atau waktu keluar dari pemilih identitas, program juga akan mengirim PIN untuk ditampilkan di LCD. Sedangkan jika data yang diterima komputer tidak ditemukan di database maka tidak ada proses penyimpanan data berupa waktu masuk atau waktu keluar, program juga akan mengirim data ke mikrokontroller yang berupa informasi bahwa data tidak ditemukan di database. Gambar berikut adalah tampilan pada LCD ketika menerima data dari komputer:



Gambar 4 – 14
Mikrokontroler Menerima Data dari Komputer

Tampilan utama perangkat lunak di komputer pada saat melakukan proses kerja yang direncanakan ditunjukan pada gambar berikut:



Gambar 4 – 15
Tampilan Utama

Dari pengujian pengujian-pengujian yang dilakukan diatas, dapat disimpulkan bahwa keseluruhan sistem sudah dapat bekerja sesuai yang direncanakan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan dan pembuatan alat ini maka dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Sistem mikrokontroler sebagai pemroses perangkat sensor serta menjalin komunikasi dengan PC melalui port serial, dengan kecepatan komunikasi data sebesar 9600 bps dan frekuensi mikrokontroler sebesar 12 MHz.
2. Modul reader chip dapat dihubungkan satu dengan yang lainnya dengan 2 kawat saluran membentuk saluran satu reader anatara dengan menggunakan RS485.
3. LCD dapat menampilkan instruksi sesuai dengan database di komputer
4. IC AT24C64 dapat menyimpan data identitas pengaman pada kartu.
5. Metode komnukasi data serial menggunakan komnukasi satu arah (Duplex).

5.2. Saran - saran

Bagi kawan – kawan mahasiswa yang ingin merancang dan mengembangkan alat ini maka:

1. Sebaiknya untuk meningkatkan ketelitian dan kecepatan dalam pembacaan kartu digunakan sensor yang menggunakan laser.

2. Sebaiknya dikembangkan untuk komunikasi jarak jauh, tidak lagi menggunakan 2 kawat tapi menggunakan gelombang radio band VHF dan UHF.
3. Sebaiknya dikembangkan menggunakan foto sebagai identitas, untuk mengontrol pintu, lampu ruang perkuliahan, tempat parkir.
4. Pemilihan jenis IC dan aplikasi yang lain dalam pembuatan kartu sehingga didapat hasil yang maksimal.

DAFTAR PUSTAKA

1. Sinnema, Williem and Mc. Govern, Tom, 1980. *Digital, Analog and Data Comunication.* Second Edition. Prentice Hall International Editions Inc.
2. Proakis, John G. 1983. *Digital Communication.* Mc. Graw – Hill International Book Company, Singapore.
3. Budiharto, Widodo 2004. *Interfacing Komputer dan Mikrokontroler.* Elex Media Komputindo.
4. Nalwan, Paulus Andi 2003. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89S51.* Elex Media Komputindo.
5. Atmel,1997. *AT89S51 Flash Microcontroller.* USA, http://www.atmel.com.
6. Dallas, Semiconductor. 1999. *Application Note & Fundamental of RS-232 Serial Communication.* USA : http:// www.dallsemi.com.
7. Maxim, 1996. *Low Power, Slew-Rate-Limited RS-485 / RS-422 Transceiver.* USA: http:// www.maxim-ic.com.

LAMPIRAN

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Jl. Bendungan Sigura – gura No.2
MALANG

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa	: Okis Ardhi Pratomo Lukas
NIM	: 99.17.035
Jurusan	: Teknik Elektro
Konsentrasi	: Teknik Elektronika S - 1
Judul Skripsi	: PENCATAT DAFTAR HADIR DOSEN DAN PENAMPIL RUANG KULIAH DENGAN KOMPUTER DI KAMPUS ITN – MALANG

Dipertahankan dihadapan Team Penguji Skripsi Jenjang Strata Satu (S – 1) pada :

Hari : Rabu
Tanggal : 30 Maret 2005
Nilai : 76,25 (B)

Panitia Majelis Penguji :

Ketua	Sekretaris
(Ir. Mochtar Asroni, MSME)	(Ir. Yudi Limpraptono, MT)
NIP.Y. 1018100036	NIP. Y. 1039500274

Anggota Penguji

(Ir. Widodo Puji M, MT)	(Ir. Yudi Limpraptono, MT)
Penguji Pertama	Penguji Kedua

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Jl. Bendungan Sigura-gura No. 2
Malang

LEMBAR BIMBINGAN SKRIPSI

1. Nama : Okis Ardhi Pratomo Lukas
2. NIM : 99.17.035
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Elektronika
5. Judul Skripsi :

   " Pencatat Daftar Hadir Dosen dan Penampil Ruang Kuliah Yang Dikontrol Dengan Komputer Di Kampus ITN-Malang "
6. Tanggal Bimbingan Skripsi : 01 April 2004
7. Selesai Menulis Skripsi : 26 Maret 2005
8. Pembimbing : Ir. Teguh Herbasuki, MT
9. Telah Dievaluasi : 80 (Delapan Puluh)

Mengetahui	Diperiksa dan Disetujui
Ketua Jurusan Elektro S - 1	Dosen Pembimbing
Ir. F. Yudi Limpraptono, MT	Ir. Teguh Herbasuki, MT
NIP.Y. 1039500274	NIP. 1038900209

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Jl. Bendungan Sigura-gura No. 2
Malang

LEMBAR BIMBINGAN SKRIPSI

1. Nama : Okis Ardhi Pratomo Lukas
2. NIM : 99.17.035
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Elektronika
5. Judul Skripsi :

   " Pencatat Daftar Hadir Dosen dan Penampil Ruang Kuliah Yang Dikontrol Dengan Komputer Di Kampus ITN-Malang "
6. Tanggal Bimbingan Skripsi : 01 April 2004
7. Selesai Menulis Skripsi : 26 Maret 2005
8. Pembimbing : I Komang Somawirata, ST
9. Telah Dievaluasi : 80 (Delapan Puluh)

Mengetahui	Diperiksa dan Disetujui
Ketua Jurusan Elektro S - 1	Dosen Pembimbing
Ir. F. Yudi Limpraptono, MT	I Komang Somawirata, ST
NIP.Y. 1039500274	NIP. 1030100361

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

FORMULIR BIMBINGAN SKRIPSI

Nama : Okis Ardhi Pratomo L.
Nim : 9917035
Masa Bimbingan : 1-Apr-2004 s\d 2-Oct-2004
Judul Skripsi : Pencatat daftar hadir dosen dan penampil informasi ruang kuliah yang dikontrol dengan komputer di kampus ITN-Malang

No	Tanggal	Uraian	Paraf Pembimbing
1.	28/06 04	Konsultasi bab I	[paraf]
2.	09/07 04	Konsultasi bab IV	[paraf]
3.	09/08 04	Konsultasi bab II & III	[paraf]
4.	09/09 04	Konsultasi bab V	[paraf]
5.	18/09 04	Acc seminar	[paraf]
6.	21/09 04	Mencoba alat	[paraf]
7.	18/03 05	Revisi alat	[paraf]
8.	27/03 05	Acc ujian	[paraf]
9.			
10.			

Malang, 27-MARET-2005
Dosen Pembimbing,

[tanda tangan]

Ir. Teguh Herbasuki, MT

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

FORMULIR BIMBINGAN SKRIPSI

Nama : Okis Ardhi Pratomo L.
Nim : 9917035
Masa Bimbingan : 1-Apr-2004 s\d 2-Oct-2004
Judul Skripsi : Pencatat daftar hadir dosen dan penampil informasi ruang kuliah yang dikontrol dengan komputer di kampu ITN-Malang

No	Tanggal	Uraian	Paraf Pembimbing
1.	28/06 '04	Konsultasi BAB I	
2.	09/07 '04	Konsultasi BAB IV	
3.	09/08 '04	Konsultasi BaB II & III	
4.	09/09 '04	Konsultasi BAB V	
5.	18/09 '04	ACC SEMINAR	
6.	21/09 '04	Mencoba Alat.	
7.	18/03 '05	Revisi Alat.	
8.	27/03 '05	ACC Ujian.	
9.			
10.			

Malang, 27 MARET 2005
Dosen Pembimbing

I Komang Somawirata,ST

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : OKIS Ardhi P
N I M : 99 17 035
Perbaikan meliputi :

Malang, 200

( )

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

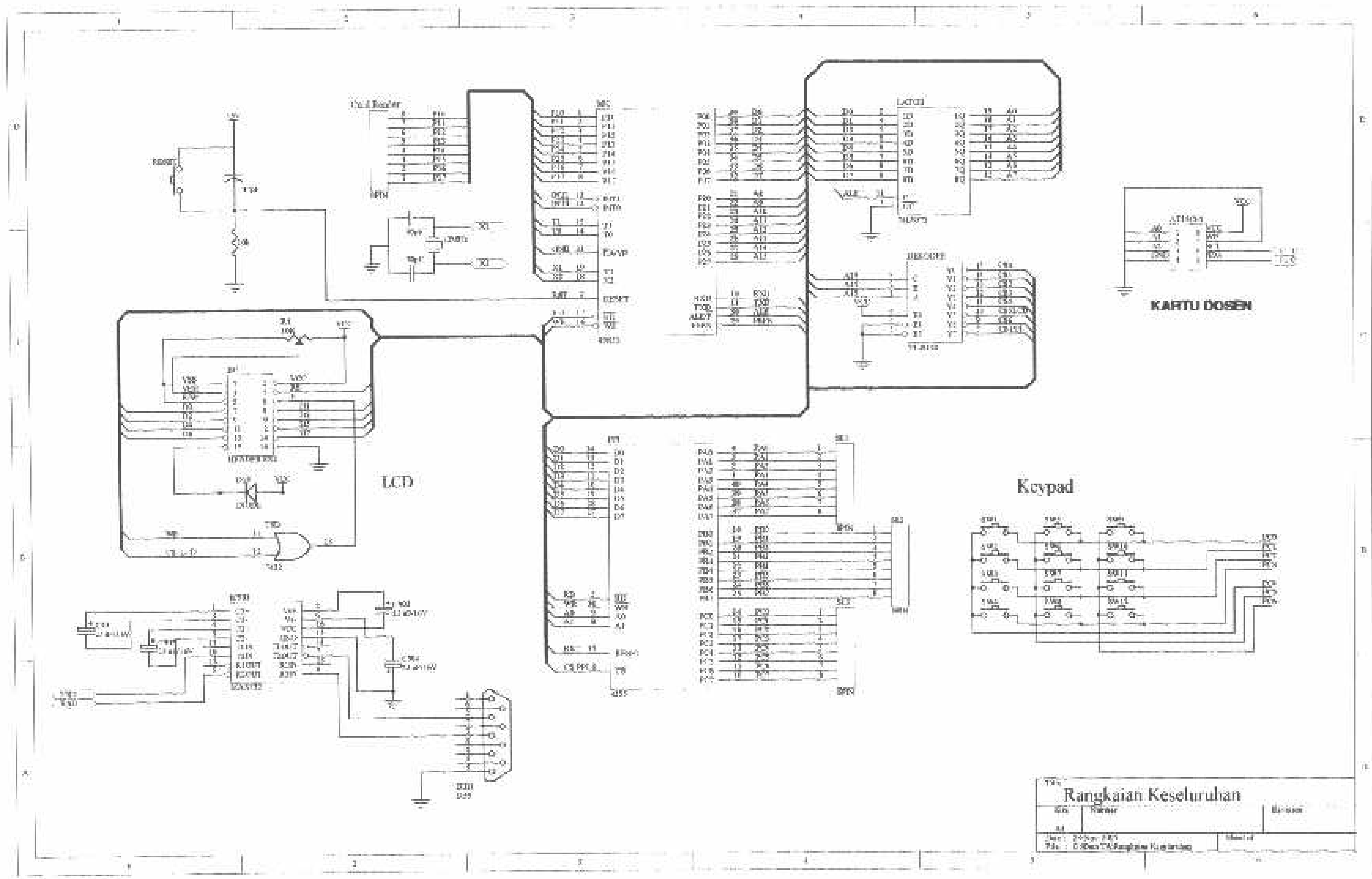
Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA :
N I M :
Perbaikan meliputi :

Malang, 200

( )

I


LCD
Keypad
KARTU DOSEN
Rangkaian Keseluruhan

```
DataKartu       Data    30h
DtPin0          Data    31h
DtPin1          Data    32h
DtPin2          Data    33h
DtPin3          Data    34h
DtPin4          Data    35h
DtPin5          Data    36h
DataBox         Data    37h

Awal:
        Mov     SP,#40h
        Mov     R0,#1
        Call
Keypad_Initialization
        Call
LCD_Initialization
        Call
Serial_Initialization
        Mov     DataBox,#'&'
        Mov     A,DataBox
        Call
Serial_Transmit

Start:
        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov
DPTR,#txtMasukKartu
        Call    LCD_String
        Clr     RI

CekKartu1:
        Jnb
P1.0,ProsesKartu1
        Jnb
P1.1,ProsesKartu2
        Jnb
P1.2,ProsesKartu3
        Jnb
P1.3,ProsesKartu4
        Jmp     CekKartu1
ProsesKartu1:
        Mov     DataKartu,#'1'
        Jmp     ProsesKartu
ProsesKartu2:
        Mov     DataKartu,#'2'
        Jmp     ProsesKartu
ProsesKartu3:
        Mov     DataKartu,#'3'
        Jmp     ProsesKartu
ProsesKartu4:
        Mov     DataKartu,#'4'
        Jmp     ProsesKartu
ProsesKartu:

        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov
DPTR,#txtMasukPin
        Call    LCD_String
        Mov     A,#' '
        Call    LCD_Data
        Mov     A,DataKartu
        Call    LCD_Data

        Call    LCD_Line2

        Mov     R0,#1
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin0,A

        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf1
        Jmp     Awal

asdfasdf1:
        Mov     R0,#1
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin1,A
        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf2
        Jmp     Awal

asdfasdf2:
        Mov     R0,#1
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin2,A
        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf3
        Jmp     Awal

asdfasdf3:
        Mov     R0,#1
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin3,A
        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf4
        Jmp     Awal
asdfasdf4:
        Mov     R0,#1
```

```
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin4,A
        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf5
        Jmp     Awal
asdfasdf5:
        Mov     R0,#1
        Mov     R1,#'*'
        Call    Keypad_Get
        Mov     DtPin5,A
        Mov     A,P1
        Orl     A,#11110000b
        Cjne
A,#11111111b,asdfasdf6
        Jmp     Awal
asdfasdf6:

        Call
Serial_Initialization
        Clr     EA
        Clr     TI
        Mov     A,#'~'
        Call
Serial_Transmit
        Mov     A,DtPin0
        Call
Serial_Transmit
        Mov     A,DtPin1
        Call
Serial_Transmit
        Mov     A,DtPin2
        Call
Serial_Transmit
        Mov     A,DtPin3
        Call
Serial_Transmit
        Mov     A,DtPin4
        Call
Serial_Transmit
        Mov     A,DtPin5
        Call
Serial_Transmit
        Mov     A,DataBox
        Call
Serial_Transmit
        Mov     A,DataKartu
        Call
Serial_Transmit
        Mov     A,#'@'
        Call
Serial_Transmit
;-----------
GetKonfirmasi:
        Call    Serial_Receive
        Cjne
A,#'E',NextKonfirmasi
        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov     DPTR,#txtSalah
        Call    LCD_String
        Call    Delay_Fix_1s
        Call    Delay_Fix_1s
        Call    QuitCard
        Call    Delay_Fix_1s
        Jmp     Start
NextKonfirmasi:
        Cjne
A,#'M',StatusKeluar
        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov
DPTR,#txtStatus
        Call    LCD_String
        Call    LCD_Line2
        Mov     DPTR,#txtMasuk
        Call    LCD_String
        Call    Delay_Fix_1s
        Call    Delay_Fix_1s
        Call    Delay_Fix_1s
        Call    QuitCard
        Call    Delay_Fix_1s
        Jmp     Start
StatusKeluar:
        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov
DPTR,#txtStatus
        Call    LCD_String
        Call    LCD_Line2
        Mov
DPTR,#txtKeluar
        Call    LCD_String
        Call    Delay_Fix_1s
        Call    Delay_Fix_1s
        Call    Delay_Fix_1s
        Call    QuitCard
        Call    Delay_Fix_1s
        Jmp     Start

Subroutine QuitCard:
        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov
DPTR,#txtSilakanQuit
        Call    LCD_String
        Mov     A,DataKartu
        Cjne
A,#'1',QuidCard1
        Jnb     P1.0,$
```

```
        Ret
QuidCard1:
        Cjne
A,#'2',QuidCard2
        Jnb      P1.1,$
        Ret
QuidCard2:
        Cjne
A,#'3',QuidCard3
        Jnb      P1.2,$
        Ret
QuidCard3:
        Jnb      P1.3,$
        Ret
EndSub

;*****************************
; subrutin clear lcd1
;*****************************
Subroutine LCD_Clrscr:
        Push     ACC
        Mov      A,#01h
        Call     LCD_Command
        Pop      ACC
EndSub
;*****************************
; subrutin menempatkan kursor
;*****************************
Subroutine
LCD_Cursor_Position:
        Push     ACC
        Mov      A,R0
        Anl      A,#0f0h
        Cjne
A,#10h,lcd_cursor_position1
        Mov      A,R0
        Anl      A,#0fh
        Orl      A,#80h
        Call     LCD_Command
        Jmp
lcd_cursor_position_end
lcd_cursor_position1:
        Cjne
A,#20h,lcd_cursor_position_end
        Mov      A,R0
        Anl      A,#0fh
        Orl      A,#0c0h
        Call     LCD_Command
lcd_cursor_position_end:
        Pop      ACC
EndSub
;*****************************
************
; kirim data ke lcd1
;*****************************
************
Subroutine LCD_Data:
        Push     DPL
        Push     DPH
        Push     ACC
        Mov      DPTR,#Lcd1
        Movx     @DPTR,A
        Call     LCD_Delay
        Lcall    LCD_Delay
        Lcall    LCD_Delay
        Lcall    LCD_Delay
        Pop      ACC
        Pop      DPH
        Pop      DPL
EndSub
;*****************************
***************
; kirim instruksi ke lcd1
;*****************************
***************
lcd0     Equ      0a000h
;lcd control operation
lcd1     Equ      lcd0+1
;lcd data operation
Subroutine LCD_Command:
        Push     DPL
        Push     DPH
        Mov      DPTR,#lcd0
        Movx     @DPTR,A
        Lcall    LCD_Delay
        Lcall    LCD_Delay
        Pop      DPH
        Pop      DPL
        Ret
EndSub
Subroutine LCD_Delay:
        Push     01h
        Push     02h
        Push     03h
        Mov      3,#5
tundalqqqa:
        Mov      1,#30
        Djnz     1,$
        Mov      2,#40
        Djnz     2,$
        Mov      2,#1
        Djnz     2,$
        Djnz     3,tundalqqqa
        Pop      03h
        Pop      02h
        Pop      01h
EndSub
Subroutine LCD_Delete_Line1:
        Push     ACC
        Push     DPH
        Push     DPL
        Call     LCD_Line1
```

```
        Mov
DPTR,#txt_blank
        Call    LCD_String
        Call    LCD_Line1
        Pop     DPL
        Pop     DPH
        Pop     ACC
EndSub
Subroutine LCD_Delete_Line2:
        Push    ACC
        Push    DPH
        Push    DPL
        Call    LCD_Line2
        Mov
DPTR,#txt_blank
        Call    LCD_String
        Call    LCD_Line2
        Pop     DPL
        Pop     DPH
        Pop     ACC
        Ret
txt_blank:      Db      '
',0
EndSub
Subroutine LCD_Initialization:
        Mov     A,#38h
;function set 8 bit
        Lcall   LCD_Command
        Mov     A,#0ch
;display on, cursor off, blink
off
        Lcall   LCD_Command
        Mov     A,#06h
;increment, no display shift
        Lcall   LCD_Command
        Mov     A,#01h
;clear display
        Lcall   LCD_Command
EndSub
Subroutine LCD_Line1:
        Push    ACC
        Mov     A,#80h
        Lcall   LCD_Command
        Pop     ACC
EndSub
Subroutine LCD_Line2:
        Push    ACC
        Mov     A,#0c0h
        Lcall   LCD_Command
        Pop     ACC
EndSub
Subroutine LCD_Off:
        Push    ACC
        Mov     A,#00001000b
        Lcall   LCD_Command
        Pop     ACC
EndSub
Subroutine LCD_On:
        Push    ACC
        Mov     A,#00001100b
        Lcall   LCD_Command
        Pop     ACC
EndSub
;-----
Subroutine LCD_String:
        Push    ACC
        Push    DPL
        Push    DPH
getcar1:
        Clr     A               ;
mengambil data dari eprom
        Movc    A,@A+DPTR
        Cjne    A,#0,tammpil1
; tes apakah data habis?
        Ljmp    mettu1
tammpil1:               ;
keluarkan data ke lcd

        Call    LCD_Data
        Inc     DPTR            ;
naikkan dptr
        Ljmp    getcar1
mettu1:
        Pop     DPH
        Pop     DPL
        Pop     ACC
EndSub
;*****************************
***
; subrutin tampil text pada
lcd1
; dari data di ram
;*****************************
***
Subroutine LCD_String_Ram:
        Push    ACC
        Push    DPL
        Push    DPH
        Push    00h
getcar1z:
        Movx    A,@DPTR
        Cjne    A,#0,tammpil1z
; tes apakah data habis?
        Ljmp    mettu1z
tammpil1z:              ;
keluarkan data ke lcd
        Call    LCD_Data
        Inc     DPTR            ;
naikkan dptr
        Ljmp    getcar1z

mettu1z:
```

```
         Pop      00h
         Pop      DPH
         Pop      DPL
         Pop      ACC
EndSub
Subroutine LCD_Blink_Off:
         Push     ACC
         Mov      A,#00001100b
;
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Blink_On:
         Push     ACC
         Mov      A,#00001101b
;
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Cursor_Off:
         Push     ACC
         Mov      A,#00001100b
; cursor on
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Cursor_On:
         Push     ACC
         Mov      A,#00001110b
; cursor on
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Cursor_Left:
         Push     ACC
         Mov      A,#00010000b
; cursor left
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Cursor_Right:
         Push     ACC
         Mov      A,#00010100b
; cursor left
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Display_Left:
         Push     ACC
         Mov      A,#00011000b
; display left
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Display_Right:
         Push     ACC
         Mov      A,#00011100b
; display right
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Cursor_Home:
         Push     ACC
         Mov      A,#00000010b
; display right
         Lcall    LCD_Command
         Pop      ACC
EndSub
Subroutine LCD_Hexa:
         Push     07h
         Push     ACC

         Mov      7,A
         Anl      A,#0f0h
         Swap     A
         Orl      A,#30h
         Lcall    tes_huruf_
         Call     LCD_Data
         Mov      A,7
         Anl      A,#0fh
         Orl      A,#30h
         Lcall    tes_huruf_
         Lcall    LCD_Data

         Pop      ACC
         Pop      07h
         Ret

tes_huruf :
         Cjne
A,#3ah,tes_huruf_1
         Mov      A,#'a'
         Ret
tes_huruf_1:
         Cjne
A,#3bh,tes_huruf_2
         Mov      A,#'b'
         Ret
tes_huruf_2:
         Cjne
A,#3ch,tes_huruf_3
         Mov      A,#'c'
         Ret
tes_huruf_3:
         Cjne
A,#3dh,tes_huruf_4
         Mov      A,#'d'
         Ret
tes_huruf_4:
         Cjne
A,#3eh,tes_huruf_5
         Mov      A,#'e'
```

```
        Ret
tes_huruf_5:
        Cjne
A,#3fh,tes_huruf_6
        Mov     A,#'f'
        Ret
tes_huruf_6:
        Ret
EndSub
ppi_porta       Equ     0e000h
ppi_portb       Equ     0e001h
ppi_portc       Equ     0e002h
ppi_portcw      Equ     0e003h

Subroutine PPI_Get_Port_A:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_porta
        Movx    A,@DPTR
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Get_Port_B:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_portb
        Movx    A,@DPTR
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Get_Port_C:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_portc
        Movx    A,@DPTR
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Initialization:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_portcw
        Movx    @DPTR,A
; kirim ke port control ppi
8255
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Send_Port_A:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_porta
        Movx    @DPTR,A
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Send_Port_B:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_portb
        Movx    @DPTR,A
        Pop     DPL
        Pop     DPH
EndSub
Subroutine PPI_Send_Port_C:
        Push    DPH
        Push    DPL
        Mov
DPTR,#ppi_portc
        Movx    @DPTR,A
        Pop     DPL
        Pop     DPH
EndSub
Subroutine Serial_Receive:
        Jnb     RI,$
        Clr     RI
        Mov     A,SBUF
        Clr     RI
EndSub
Subroutine Serial_Transmit:
        Mov     SBUF,A
        Jnb     TI,$
        Clr     TI
EndSub
Subroutine
Serial_Initialization:
        Mov     SCON,#50H
        Mov     TH1,#250
; BAUDRATE 9600
        ;Mov     TH1,#208
; BAUDRATE 1200
        Mov     87H,#00H
; PCON
        Mov     TMOD,#21H
        Mov
TCON,#01010000B ; RUN T1 AND
T0
        ;Mov     IE,#10000010B
        ;Mov     IP,#00001000B
EndSub
Subroutine Serial_HEX:
        Push    07h
        Push    ACC

        Mov     7,A
```

```
        Anl     A,#0f0h
        Swap    A
        Orl     A,#30h
        Call    tes_huruf_
        Call
Serial_Transmit
        Mov     A,7
        Anl     A,#0fh
        Orl     A,#30h
        Lcall   tes_huruf_
        Lcall
Serial_Transmit

        Pop     ACC
        Pop     07h
        Ret

EndSub

Subroutine
Serial_Transmit_String:
        Push    ACC
        Push    DPL
        Push    DPH
xgetcarl:
        Clr     A           ;
mengambil data dari eprom
        Movc    A,@A+DPTR
        Cjne    A,#0,xtammpill
; tes apakah data habis?
        Ljmp    xmettul

xtammpill:                  ;
keluarkan data ke lcd
        Lcall
Serial_Transmit
        Inc     DPTR            ;
naikkan dptr
        Ljmp    xgetcarl
xmettul:
        Pop     DPH
        Pop     DPL
        Pop     ACC
EndSub
Subroutine
Serial_Transmit_String_Ram:
        Push    ACC
        Push    DPL
        Push    DPH
axgetcarl:
        Movx    A,@DPTR

        Cjne
A,#0,axtammpill     ; tes
apakah data habis?
        Ljmp    axmettul
```

```
axtammpill:                 ;
keluarkan data ke lcd

        Call
Serial_Transmit
        Inc     DPTR            ;
naikkan dptr
        Ljmp    axgetcarl
axmettul:
        Pop     DPH
        Pop     DPL
        Pop     ACC
EndSub
;================================
==========================
;===================== MODUL
KEYPAD ===================
keypad_id_keypad            Equ
02h     ;register r2
keypad_cek_key_88           Equ
03h     ;register r3
keypad_temp_baris           Equ
04h     ;register r4
keypad_temp_data            Equ
05h     ;register r5
key1x                       Equ
06h
keydelay                    Equ
07h
keydat1                     Equ
08h
keydat2                     Equ
09h
keypad_curpos               Equ
10h
keypad_port_a               Equ
0e000h
keypad_port_b               Equ
0e001h
keypad_port_c               Equ
0e002h
keypad_port_cw              Equ
0e003h
keypad_shift                Equ
06h
Keypad_cons_1:              Db
'1^',0
Keypad_cons_2:              Db
'ABC2',0
Keypad_cons_3:              Db
'DEF3',0
Keypad_cons_4:              Db
'GHI4',0
Keypad_cons_5:              Db
'JKL5',0
```

```
Keypad_cons_6:                Db
'MNO6',0
Keypad_cons_7:                Db
'PQRS7',0
Keypad_cons_8:                Db
'TUV8',0
Keypad_cons_9:                Db
'WXYZ9',0
Keypad_cons_0:                Db
'0 ',0
Keypad_cons_10:               Db
'.;?!()"',0
Keypad_dphh                   Equ
0ah
Keypad_dpll                   Equ
0bh
Keypad_dphh1                  Equ
0ch
Keypad_dpll1                  Equ
0dh
Keypad_dphh2                  Equ
0eh
Keypad_dpll2                  Equ
0fh
Subroutine
Keypad_Initialization:
        Push    DPL
        Push    DPH
        Mov     A,R0
        Cjne
A,#1,keypad_88_init
keypad_end_initx:
        Mov     A,#81h
;pa:out pb:out pcl:in pch:out
        Mov
DPTR,#keypad_port_cw
        Movx    @DPTR,A
        Jmp
keypad_end_init
keypad_88_init:
        Cjne
A,#2,keypad_end_initx
        Mov     A,#82h
;pa:out pb:in pc:out
        Mov
DPTR,#keypad_port_cw
        Movx    @DPTR,A
keypad_end_init:
        Pop     DPH
        Pop     DPL
EndSub
Subroutine Keypad_Alpa:
        Mov
keypad_shift,#0
keypad_alpa1:
        Mov     R0,#2
        Call    Keypad_Get
        Cjne
A,#'*',tampilkan_data_keypad
        Inc     keypad_shift
        Mov     A,keypad_shift
        Cjne
A,#5,keypad_alpa1
        Mov
keypad_shift,#0
        Jmp     keypad_alpa1
tampilkan_data_keypad:
        Cjne
A,#'1',proses_key_2
        Mov
DPTR,#Keypad_cons_1
        Jmp
proses_keypad_alpha
proses_key_2:
        Cjne
A,#'2',proses_key_3
        Mov
DPTR,#Keypad_cons_2
        Jmp
proses_keypad_alpha
proses_key_3:
        Cjne
A,#'3',proses_key_4
        Mov
DPTR,#Keypad_cons_3
        Jmp
proses_keypad_alpha
proses_key_4:
        Cjne
A,#'4',proses_key_5
        Mov
DPTR,#Keypad_cons_4
        Jmp
proses_keypad_alpha
proses_key_5:
        Cjne
A,#'5',proses_key_6
        Mov
DPTR,#Keypad_cons_5
        Jmp
proses_keypad_alpha
proses_key_6:
        Cjne
A,#'6',proses_key_7
        Mov
DPTR,#Keypad_cons_6
        Jmp
proses_keypad_alpha
proses_key_7:
        Cjne
A,#'7',proses_key_8
```

PROGRAM LISTING MIKROKONTROLER

```
        Mov
DPTR,#Keypad_cons_7
        Jmp
proses_keypad_alpha
proses_key_8:
        Cjne
A,#'8',proses_key_9
        Mov
DPTR,#Keypad_cons_8
        Jmp
proses_keypad_alpha
proses_key_9:
        Cjne
A,#'9',proses_key_0
        Mov
DPTR,#Keypad_cons_9
        Jmp
proses_keypad_alpha
proses_key_0:
        Cjne
A,#'0',proses_key_a
        Mov
DPTR,#Keypad_cons_0
        Jmp
proses_keypad_alpha
proses_key_a:
        Mov     B,A
        Call    LCD_Data
        Mov     A,B
        Ret
proses_keypad_alpha:
        Mov     A,keypad_shift
        Movc    A,@A+DPTR
        Cjne
A,#0,proses_keypad_alphaxx
        Ret
proses_keypad_alphaxx:
        Call
Delay_Fix_100ms
        Mov     B,A
        Call    LCD_Data
        Mov     A,B
        Ret
EndSub
Subroutine Keypad_Get:
        Call    keypad_get_44
        Mov     B,A
        Mov     A,R2
        Cjne
A,#01,Keypad_Get
        Call
Delay_Fix_100ms
        Call
Delay_Fix_100ms
        Call
Delay_Fix_100ms
        Mov     A,B
EndSub
Subroutine
Keypad_Get_Matrix_44:
;----------------
keypad_get_matrik_44:
keypad_loop_awal_44:
keypad_get_44:
        Mov     A,#80h
;kolom satu
        Mov
keypad_temp_baris,A     ;kolom
keypad_loopstart_44:
        Call
keypad_get_port_cl
        Anl     A,#00001111b
        Cjne
A,#0,keypad_loopstart_44   ;cek
keypad
        Mov
A,keypad_temp_baris     ;kolom

keypad_loop_44:
        Call
keypad_send_port_ch
        Mov
keypad_temp_baris,A     ;kolom
        Call
keypad_get_port_cl
        Anl     A,#00001111b
        Cjne
A,#0,keypad_baris1_44   ;cek
keypad ditekan atau tidak
        Mov
keypad_id_keypad,#0     ;simpan
id keypad (0:tak ditekan)
        Mov
A,keypad_temp_baris     ;kolom
        Rr      A
;naikan kolom
        Cjne
A,#08h,keypad_loop_44   ;cek
jika > 4 baris
        Ret

;------------------------------
--------------------
;               BARIS SATU
;------------------------------
--------------------
keypad_baris1_44:
        Mov
keypad_id_keypad,#1
;simpan id keypad
(<>0:ditekan)
```

```
        Cjne
A,#00000001b,keypad_baris2_44
;baris 1
        Mov
A,keypad_temp_baris
        Cjne
A,#10000000b,keypad_kolom21_44
;kolom 1
        Mov     A,#'1'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom21_44:
        Cjne
A,#01000000b,keypad_kolom31_44
;kolom 2
        Mov     A,#'2'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom31_44:
        Cjne
A,#00100000b,keypad_kolom41_44
;kolom 3
        Mov     A,#'3'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom41_44:
        Cjne
A,#00010000b,keypad_end_baris1
_44    ;kolom 5
        Mov     A,#'A'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_end_baris1_44:
        Jmp
keypad_end_matrik1_44
;------------------------------
--------------------
;                  BARIS DUA
;------------------------------
--------------------
keypad_baris2_44:
        Cjne
A,#00000010b,keypad_baris3_44
;baris 2
        Mov
A,keypad_temp_baris
        Cjne
A,#10000000b,keypad_kolom22_44
;kolom 1
        Mov     A,#'4'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom22_44:
        Cjne
A,#01000000b,keypad_kolom32_44
;kolom 2
        Mov     A,#'5'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom32_44:
        Cjne
A,#00100000b,keypad_kolom42_44
;kolom 3
        Mov     A,#'6'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom42_44:
        Cjne
A,#00010000b,keypad_end_baris2
_44    ;kolom 5
        Mov     A,#'B'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_end_baris2_44:
        Jmp
keypad_end_matrik1_44

;------------------------------
--------------------
;                  BARIS TIGA
;------------------------------
--------------------
keypad_baris3_44:
        Cjne
A,#00000100b,keypad_baris4_44
;baris 3
        Mov
A,keypad_temp_baris
        Cjne
A,#10000000b,keypad_kolom23_44
;kolom 1
        Mov     A,#'7'
        Mov
keypad_temp_data,A
```

```
        Jmp
keypad_end_matrik_44
keypad_kolom23_44:
        Cjne
A,#01000000b,keypad_kolom33_44
;kolom 2
        Mov     A,#'6'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom33_44:
        Cjne
A,#00100000b,keypad_kolom43_44
;kolom 3
        Mov     A,#'9'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom43_44:
        Cjne
A,#00010000b,keypad_end_baris3
_44   ;kolom 5
        Mov     A,#'C'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_end_baris3_44:
        Jmp
keypad_end_matrik1_44

;-----------------------------
--------------------
;               BARIS EMPAT
;-----------------------------
--------------------
keypad_baris4_44:
        Cjne
A,#00001000b,keypad_baris4_44
;baris 4
        Mov
A,keypad_temp_baris
        Cjne
A,#10000000b,keypad_kolom24_44
;kolom 1
        Mov     A,#'*'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom24_44:
        Cjne
A,#01000000b,keypad_kolom34_44
;kolom 2
        Mov     A,#'0'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom34_44:
        Cjne
A,#00100000b,keypad_kolom44_44
;kolom 3
        Mov     A,#'#'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_kolom44_44:
        Cjne
A,#00010000b,keypad_end_baris4
_44   ;kolom 4
        Mov     A,#'D'
        Mov
keypad_temp_data,A
        Jmp
keypad_end_matrik_44
keypad_end_baris4_44:
        Jmp
keypad_end_matrik1_44
;----- end deteksi keypad ----
-------------

keypad_end_matrik_44:
        Call
keypad_syarat_key_88
        Jmp
keypad_end_matrik2_44
keypad_end_matrik1_44:
        Jmp
keypad_loop_awal_44
keypad_end_matrik2_44:
        Mov
A,keypad_temp_data
        Ret
;-------------------
; send port c high
;-------------------
keypad_send_port_ch:
        Mov
DPTR,#keypad_port_c
        Movx    @DPTR,A
        Ret
;---- end of send port ch

;--------------------
; get port c low
;--------------------
keypad_get_port_cl:
```

```
        Mov
DPTR,#keypad_port_c
        Movx    A,@DPTR
        Ret
;--- end of get port cl
;-----------------------
; Syarat tampilkan data
;-----------------------
keypad_syarat_key_88:
        Mov     A,R0
;syarat tampilkan karakter
        Cjne
A,#0,keypad_sandi_data_88
        Mov
A,keypad_temp_data
        Jmp
keypad_tampil_data_88
keypad_sandi_data_88:
        Cjne
A,#1,keypad_hide_data_88
        Mov     A,R1    ;sandi
        Jmp
keypad_tampil_data_88
keypad_hide_data_88:
        Jmp
keypad_ora_tampil_88
keypad_tampil_data_88:
        Lcall   LCD_Data
keypad_ora_tampil_88:
        Ret
key1:
        Push    DPH
        Push    DPL
        Mov     key1x,#2
        Mov     keydelay,#50
keyin1:
        Mov     R0,key1x
        Call    keypad_get_44
        Mov     keydat1,A
        Mov     A,R2
        Cjne    A,#01,keyin1
        Mov     A,keydat1
        Jmp     keyin12
keyin12:
        Cjne    A,#'1',keyin13
        Mov
DPTR,#Keypad_cons_1
        Jmp     keyin_proses
keyin13:
        Cjne    A,#'2',keyin14
        Mov
DPTR,#Keypad_cons_2
        Jmp     keyin_proses
keyin14:
        Cjne    A,#'3',keyin15
        Mov
DPTR,#Keypad_cons_3
        Jmp     keyin_proses
keyin15:
        Cjne    A,#'4',keyin16
        Mov
DPTR,#Keypad_cons_4
        Jmp     keyin_proses
keyin16:
        Cjne    A,#'5',keyin17
        Mov
DPTR,#Keypad_cons_5
        Jmp     keyin_proses
keyin17:
        Cjne    A,#'6',keyin18
        Mov
DPTR,#Keypad_cons_6
        Jmp     keyin_proses
keyin18:
        Cjne    A,#'7',keyin19
        Mov
DPTR,#Keypad_cons_7
        Jmp     keyin_proses
keyin19:
        Cjne    A,#'8',keyin20
        Mov
DPTR,#Keypad_cons_8
        Jmp     keyin_proses
keyin20:
        Cjne    A,#'9',keyin21
        Mov
DPTR,#Keypad_cons_9
        Jmp     keyin_proses
keyin21:
        Cjne    A,#'0',keyin22
        Mov
DPTR,#Keypad_cons_0
        Jmp     keyin_proses
keyin22:
        Cjne    A,#'*',keyin23
        Mov
DPTR,#Keypad_cons_10
        Jmp     keyin_proses

keyin23:
        Jmp     keyin_end
;-----------
keyin_proses:
        Mov
Keypad_dphh,DPH
        Mov
Keypad_dpll,DPL
        Mov
Keypad_dphhl,DPH
        Mov
Keypad_dplll,DPL
```

```
        Mov
DPH,Keypad_dphh
        Mov
DPL,Keypad_dpll
        Clr     A
        Movc    A,@A+DPTR
        Mov     B,A
        Mov
R0,keypad_curpos
        Call
LCD_Cursor_Position
        Mov     A,B
        Call    LCD_Data
        Mov     A,#0
        Call
PPI_Send_Port_C
        Mov     R0,keydelay
        Call    Delay_Var_10ms
keyin100:
        Mov     R0,#2
        Call    keypad_get_44
        Mov     keydat2,A
        Mov     A,R2
        Cjne
A,#01,keyin101a
        Jmp     keyin_proses1
keyin101a:
        Mov     A,#0
        Call
PPI_Send_Port_C
        Mov
DPH,Keypad_dphh
        Mov
DPL,Keypad_dpll
        Clr     A
        Movc    A,@A+DPTR
        Cjne    A,#0,keyin_end
        Mov
DPH,Keypad_dphh1
        Mov
DPL,Keypad_dpll1
        Clr     A
        Movc    A,@A+DPTR
        Jmp     keyin_end
keyin_proses1:
        Mov     A,#0
        Call
PPI_Send_Port_C
        Mov     A,keydat1
        Cjne
A,keydat2,keyin101a
        Mov
DPH,Keypad_dphh
        Mov
DPL,Keypad_dpll
        Inc     DPTR
        Clr     A
        Movc    A,@A+DPTR
        Cjne    A,#0,keyin_200
        Mov
Keypad_dphh,Keypad_dphh1
        Mov
Keypad_dpll,Keypad_dpll1
        Mov
R0,keypad_curpos
        Call
LCD_Cursor_Position
        Mov
DPH,Keypad_dphh
        Mov
DPL,Keypad_dpll
        Clr     A
        Movc    A,@A+DPTR
        Call    LCD_Data
        Call    LCD_Blink_On
        Mov     R0,keydelay
        Call    Delay_Var_10ms
        Jmp     keyin100
keyin_200:
        Mov
Keypad_dphh,DPH
        Mov
Keypad_dpll,DPL
        Mov     B,A
        Mov
R0,keypad_curpos
        Call
LCD_Cursor_Position
        Mov     A,B
        Call    LCD_Data
        Call    LCD_Blink_On
        Mov     R0,keydelay
        Call    Delay_Var_10ms
        Jmp     keyin100


;----------
keyin_end:

        Call    LCD_Blink_On
        Pop     DPL
        Pop     DPH
        Ret

EndSub
Subroutine EEPROM_Write:
        Push    01
        Mov     01,A
save_03:
        Movx    @DPTR,A
save_01:
        Movx    A,@DPTR
```

```
        Cjne    A,01h,save_01
        Pop     01
EndSub
Subroutine EEPROM Read:
        Movx    A,@DPTR
EndSub


;================== DELAY
==============================
--
Subroutine Delay_Var_1ms:
        Call    Delay_Fix_1ms
        Djnz
R0,Delay_Var_1ms
EndSub
Subroutine Delay_Var 10ms:
        Call    Delay_Fix_10ms
        Djnz
R0,Delay_Var_10ms
EndSub
Subroutine Delay_Var_100ms:
        Call    Delay_Fix_1ms
        Djnz
R0,Delay_Var_100ms
EndSub
Subroutine Delay_Var_1s:
        Call    Delay_Fix_1s
        Djnz
R0,Delay Var 1s
EndSub
Subroutine Delay_Var_10s:
        Call    Delay_Fix_10s
        Djnz
R0,Delay_Var_10s
EndSub
Subroutine Delay Var 10us:
        Call    Delay_Fix_10us
        Djnz
R0,Delay_Var_10us
EndSub
Subroutine Delay_Fix_10us:
        Push    1
        Mov     1,#20
        Djnz    1,$
        Pop     1
EndSub
Subroutine Delay_Fix_10s:
        Push    1
        Mov     1,#100
delay_fix_10s_1:
        Call
Delay_Fix_100ms
        Djnz
1,delay_fix_10s_1
        Pop     1
EndSub
Subroutine Delay_Fix_1s:
        Push    1
        Mov     1,#100
delay fix 1000ms_1:
        Call    Delay_Fix_10ms
        Djnz
1,delay_fix_1000ms_1
        Pop     1
EndSub
Subroutine Delay_Fix_100ms:
        Push    1
        Mov     1,#10
delay_fix_100ms_1:
        Call    Delay_Fix_10ms
        Djnz
1,delay_fix_100ms_1
        Pop     1
EndSub
Subroutine Delay_Fix_10ms:
        Mov
TMOD,#00000001b    ; Timer 1
bekerja pada mode 1
        Mov     TL0,#3dh    ;
siapkan waktu tunda 50 mili-
detik
        Mov     TH0,#0b0h
        Clr     TF0
; me-nol-kan bit limpahan
        Setb    TR0
; timer mulai bekerja
        Jnb     TF0,$
; tunggu di sini sampai
melimpah
        Clr     TR0
; timer berhenti kerja
        Ret
EndSub
Subroutine Delay_Fix_1ms:
        Mov
TMOD,#00000001b    ; Timer 1
bekerja pada mode 1
        Mov     TL0,#0edh    ;
siapkan waktu tunda 50 mili-
detik
        Mov     TH0,#78h
        Clr     TF0
; me-nol-kan bit limpahan
        Setb    TR0
; timer mulai bekerja
        Jnb     TF0,$
; tunggu di sini sampai
melimpah
        Clr     TR0
; timer berhenti kerja
        Ret
```

```
EndSub

txtMasukKartu:  Db
'Masukkan Kartu',0
txtMasukPin:    Db
'Masukkan PIN',0
txtSalah:       Db       'PIN
Salah',0
txtStatus:      Db
'Status:',0
txtMasuk:       Db
'Masuk',0
txtKeluar:      Db
'Keluar',0
txtSilakanQuit: Db
'Keluarkan Kartu',0
```

PROGRAM LISTING VISUAL BASIC V6.0

```
'-------------------------------------------------------
' VBTerm - This is a demonstration program for the MSComm
' communications ActiveX control.
'
' Copyright (c) 1994, Crescent Software, Inc.
' by Don Malin and Carl Franklin.
'
' Updated by Mike Maddox
'-------------------------------------------------------
Option Explicit
Dim HasAccess As Boolean
Dim DataPilih As DataPilihX
Dim MKPilih As MKPilihX

Dim QuitNow As Boolean
Dim DataSerial As String
Dim PesanEnd As Boolean
Dim Ret As Integer        ' Scratch integer.
Dim Temp As String        ' Scratch string.
Dim hLogFile As Integer ' Handle of open log file.
Dim StartTime As Date    ' Stores starting time for port timer
Dim TimeOut As Boolean
Public rcd As New Recordset
Public rcd1 As New Recordset
Public rcd2 As New Recordset
Public cnn As New Connection

Dim sPath As String
Dim OldFile, NewFile
Dim rsImage As Recordset
Dim campo As Field
Dim SSql

Private Sub Combo1_Click()
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim D4, D3, D2, D1, D0


End Sub



Private Sub Command2_Click()
CMData.OpenFile App.Path + "\data.txt"
End Sub

Private Sub Form_Load()
Dim CommPort As String, Handshaking As String, Settings As String
Dim EA As Variant
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim s0, s1, s2, s3, s4, s5, s6, s7, s8, s9
Dim xx As MSComctlLib.ListItem
Dim s10, s12, s13, s14, s15, s16, s17, s18, s19
Dim s20, s22, s23, s24, s25, s26, s27, s28, s29
```

```
Dim s30, s32, s33, s34, s35, s36, s37, s38, s39
Dim s40, s42, s43, s44, s45, s46, s47, s48, s49
Dim who
Dim fs
Set fs = CreateObject("Scripting.FileSystemObject")

'On Error Resume Next

CMData.OpenFile App.Path + "\data.txt"
MSFIsian.ColWidth(0) = 500
MSFIsian.ColWidth(1) = 3000
MSFIsian.ColWidth(2) = 1300
MSFIsian.ColWidth(3) = 1300
MSFIsian.ColWidth(4) = 1300
MSFIsian.ColWidth(5) = 1200

MSFIsian.TextMatrix(0, 0) = "No"
MSFIsian.TextMatrix(0, 1) = "Nama"
MSFIsian.TextMatrix(0, 2) = "NIP"
MSFIsian.TextMatrix(0, 3) = "Jam Masuk"
MSFIsian.TextMatrix(0, 4) = "Jam Pulang"
MSFIsian.TextMatrix(0, 5) = "Reader Box"
    On Error Resume Next
    'Combo1.Text = "1"
    ' Set the default color for the terminal
    txtTerm.SelLength = Len(txtTerm)
    txtTerm.SelText = ""
    txtTerm.ForeColor = vbBlue

    ' Set Title
    App.Title = "Visual Basic Terminal"

    ' Set up status indicator light
    imgNotConnected.ZOrder

    ' Center Form
    frmTerminal.Move (Screen.Width - Width) / 2, (Screen.Height -
Height) / 2

    ' Load Registry Settings

    Settings = GetSetting(App.Title, "Properties", "Settings", "")
' frmTerminal.MSComm1.Settings]\
    If Settings <> "" Then
        MSComm1.Settings = Settings
        If Err Then
            MsgBox Error$, 48
            Exit Sub
        End If
    End If

    CommPort = GetSetting(App.Title, "Properties", "CommPort", "")
' frmTerminal.MSComm1.CommPort
    If CommPort <> "" Then MSComm1.CommPort = CommPort
```

```
    Handshaking      =      GetSetting(App.Title,      "Properties",
"Handshaking", "") 'frmTerminal.MSComm1.Handshaking
    If Handshaking <> "" Then
        MSComm1.Handshaking = Handshaking
        If Err Then
            MsgBox Error$, 48
            Exit Sub
        End If
    End If

    Echo = GetSetting(App.Title, "Properties", "Echo", "") ' Echo
    On Error GoTo 0

a = MSComm1.CommPort
b = MSComm1.Settings
MSComm1.Settings = "9600,n,8,1"
MSComm1.PortOpen = True
MSComm1.DTREnable = True
End Sub

Private Sub Form_Resize()
   ' Resize the Term (display) control
'      txtTerm.Move  0,  tbrToolBar.Height,  frmTerminal.ScaleWidth,
frmTerminal.ScaleHeight - sbrStatus.Height - tbrToolBar.Height

   ' Position the status indicator light
'    Frame1.Left = ScaleWidth - Frame1.Width * 1.5
End Sub

Private Sub Form_Unload(Cancel As Integer)
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
    Dim Counter As Long

    If MSComm1.PortOpen Then
       ' Wait 10 seconds for data to be transmitted.
       Counter = Timer + 10
       Do While MSComm1.OutBufferCount
          Ret = DoEvents()
          If Timer > Counter Then
             Select Case MsgBox("Data cannot be sent", 34)
                ' Cancel.
                Case 3
                   Cancel = True
                   Exit Sub
                ' Retry.
                Case 4
                   Counter = Timer + 10
                ' Ignore.
                Case 5
                   Exit Do
             End Select
          End If
       Loop

       MSComm1.PortOpen = 0
```

```
    End If

    ' If the log file is open, flush and close it.

QuitNow = True
End
End Sub


' The OnComm event is used for trapping communications events and
errors.
Private Static Sub MSComm1_OnComm()
    Dim EVMsg$
    Dim ERMsg$
    Select Case MSComm1.CommEvent
        Case comEvReceive
            Dim buffer As Variant
            Dim data As String
            Dim Data1
            buffer = MSComm1.Input
            'LedReceive.BackColor = &HC000&
            'TimerReceive.Enabled = True
            data = StrConv(buffer, vbUnicode)
            DataSerial = DataSerial & buffer
            ShowData txtTerm, data
    End Select
End Sub

Private Static Sub ShowData(Term As Control, data As String)
    On Error GoTo Handler
    Const MAXTERMSIZE = 16000
    Dim TermSize As Long, i

    ' Make sure the existing text doesn't get too large.
    TermSize = Len(Term.Text)
    If TermSize > MAXTERMSIZE Then
       Term.Text = Mid$(Term.Text, 4097)
       TermSize = Len(Term.Text)
    End If

    ' Point to the end of Term's data.
    Term.SelStart = TermSize

    ' Filter/handle BACKSPACE characters.
    Do
       i = InStr(data, Chr$(8))
       If i Then
          If i = 1 Then
             Term.SelStart = TermSize - 1
             Term.SelLength = 1
             data = Mid$(data, i + 1)
          Else
             data = Left$(data, i - 2) & Mid$(data, i + 1)
          End If
       End If
    Loop While i
```

```
    ' Eliminate line feeds.
    Do
       i = InStr(data, Chr$(10))
       If i Then
          data = Left$(data, i - 1) & Mid$(data, i + 1)
       End If
    Loop While i

    ' Make sure all carriage returns have a line feed.
    i = 1
    Do
       i = InStr(i, data, Chr$(13))
       If i Then
          data = Left$(data, i) & Chr$(10) & Mid$(data, i + 1)
          i = i + 1
       End If
    Loop While i

    ' Add the filtered data to the SelText property.
    Term.SelText = data

    ' Log data to file if requested.
    If hLogFile Then
       i = 2
       Do
          Err = 0
          Put hLogFile, , data
          If Err Then
             i = MsgBox(Error$, 21)
             If i = 2 Then

             End If
          End If
       Loop While i <> 2
    End If
    Term.SelStart = Len(Term.Text)
Exit Sub

Handler:
    'msgboxError$
    Resume Next
End Sub

Private Sub txtTerm_Change()
Dim a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u,
v, w, X, Y, z
Dim PIN, Kartu
Dim DataKartu, DataNama, DataNIP, DataPIN
Dim EA As Variant
Dim DataAda As Boolean
Dim NoBox

If InStr(1, txtTerm.Text, "@") <> 0 Then
    a = InStr(1, txtTerm.Text, "@")
    Kartu = Mid(txtTerm.Text, a - 1, 1)
```

```
            MSFIsian.TextMatrix(MSFIsian.Rows   -   2,   5)   =   "1
"
        Else
            MSFIsian.TextMatrix(MSFIsian.Rows       2,   5)   =   "2
"
        End If
        MSComm1.Output = "M"
        txtTerm.Text = ""
        Exit Sub
    End If
    a = 3
    txtTerm.Text = ""
ElseIf InStr(1, txtTerm.Text, "*") = 1 Then
    LblBox1.Caption = "READY"
    txtTerm.Text = ""
ElseIf InStr(1, txtTerm.Text, "&") = 1 Then
    LblBox2.Caption = "READY"
    txtTerm.Text = ""
End If

End Sub
```

```
    b = InStrRev(txtTerm.Text, "~", a)
    PIN = Mid(txtTerm.Text, b + 1, 6)
    c = CMData.GetLine(CDec(Kartu) - 1)
    EA = Split(c, "/")
    DataKartu = EA(1)
    DataPIN = EA(2)
    DataNama = EA(3)
    DataNIP = EA(4)
    If Kartu <> DataKartu Or PIN <> DataPIN Then
        MSComm1.Output = "E"
        txtTerm.Text = ""
        Exit Sub
    End If
    DataAda = False
    For i = 1 To MSFIsian.Rows - 1
        a = MSFIsian.TextMatrix(i, 2)
        b = MSFIsian.TextMatrix(i, 4)
        If a = DataNIP And b = "" Then
            DataAda = True
            Exit For
        End If
    Next i
    If DataAda = False Then
        MSFIsian.Rows = MSFIsian.Rows + 1
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 0) = MSFIsian.Rows
- 2
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 1) = DataNama
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 2) = DataNIP
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 3) = Time
        a = txtTerm.Text
        If InStr(1, txtTerm.Text, "*") <> 0 Then
            MSFIsian.TextMatrix(MSFIsian.Rows    -    2,    5)    =    "1
"
        Else
            MSFIsian.TextMatrix(MSFIsian.Rows    -    2,    5)    =    "2
"
        End If
        MSComm1.Output = "M"
        txtTerm.Text = ""
        Exit Sub
    End If
    a = MSFIsian.TextMatrix(i, 4)
    If a = "" Then
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 4) = Time
        MSComm1.Output = "K"
        txtTerm.Text = ""
        Exit Sub
    Else
        MSFIsian.Rows = MSFIsian.Rows + 1
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 0) = MSFIsian.Rows
- 2
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 1) = DataNama
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 2) = DataNIP
        MSFIsian.TextMatrix(MSFIsian.Rows - 2, 3) = Time
        If InStr(1, txtTerm.Text, "*") <> 0 Then
```

Features

- Compatible with MCS-51® Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

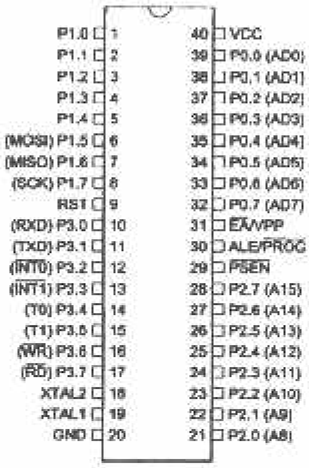
8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

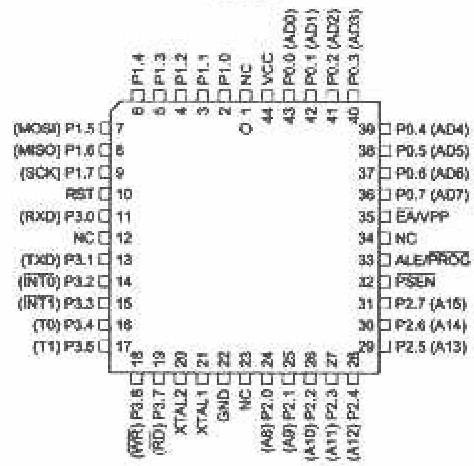
Rev. 2487A–10/01

Pin Configurations

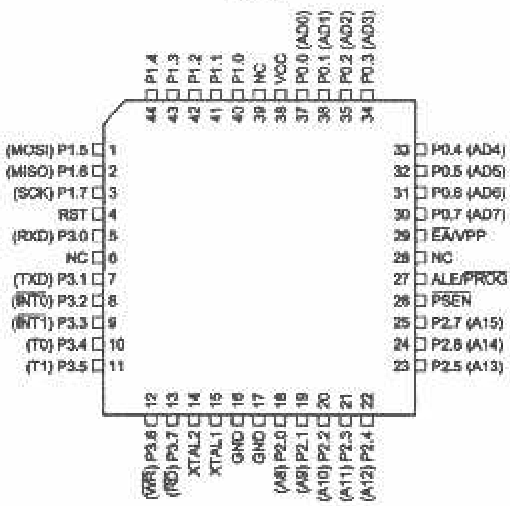
PDIP



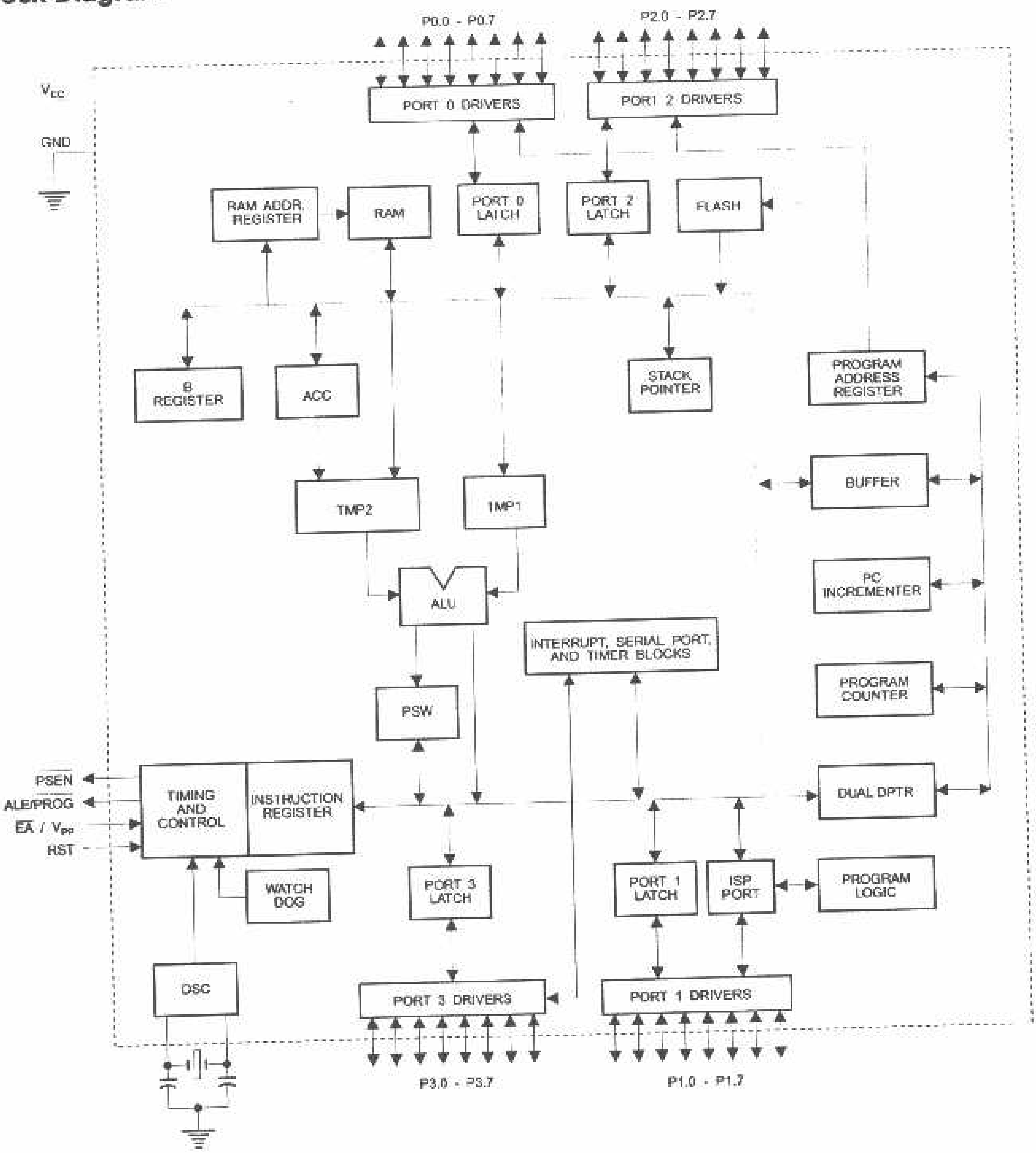
PLCC



TQFP



Block Diagram



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups.

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

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

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

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

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

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups 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 pull-ups.

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

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

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. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{PROG}$

Address Latch Enable (ALE) 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 ($\overline{PSEN}$) is the read strobe to external program memory.

When the AT89S51 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 program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier

Special Function Registers

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

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

Table 1. AT89S51 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

User 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.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the five interrupt sources in the IP register.

Table 2. AUXR: Auxiliary Register

AUXR Address = 8EH Reset Value = XXX00XX0B

Not Bit Addressable

	–	–	–	WDIDLE	DISRTO	–	–	DISALE
Bit	7	6	5	4	3	2	1	0

–	Reserved for future expansion
DISALE	Disable/Enable ALE
	DISALE Operating Mode
	0 ALE is emitted at a constant rate of 1/6 the oscillator frequency
	1 ALE is active only during a MOVX or MOVC instruction
DISRTO	Disable/Enable Reset out
	DISRTO
	0 Reset pin is driven High after WDT times out
	1 Reset pin is input only
WDIDLE	Disable/Enable WDT in IDLE mode
WDIDLE	
0	WDT continues to count in IDLE mode
1	WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal 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 AUXR1 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 rest under software control and is not affected by reset.

Table 3. AUXR1: Auxiliary Register 1

AUXR1
Address = A2H
Reset Value = XXXXXXX0B

Not Bit Addressable

	–	–	–	–	–	–	–	DPS
Bit	7	6	5	4	3	2	1	0

– Reserved for future expansion

DPS Data Pointer Register Select

DPS	
0	Selects DPTR Registers DP0L, DP0H
1	Selects DPTR Registers DP1L, DP1H

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the $\overline{EA}$ pin is connected to GND, all program fetches are directed to external memory.

On the AT89S51, if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through FFFH are directed to internal memory and fetches to addresses 1000H through FFFFH are directed to external memory.

Data Memory

The AT89S51 implements 128 bytes of on-chip RAM. The 128 bytes are accessible via direct and indirect addressing modes. Stack operations are examples of indirect addressing, so the 128 bytes of data RAM are available as stack space.

Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 98xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt, which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S51 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S51 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

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

Note that Table 4 shows that bit position IE.6 is unimplemented. In the AT89S51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle

Table 4. Interrupt Enable (IE) Register

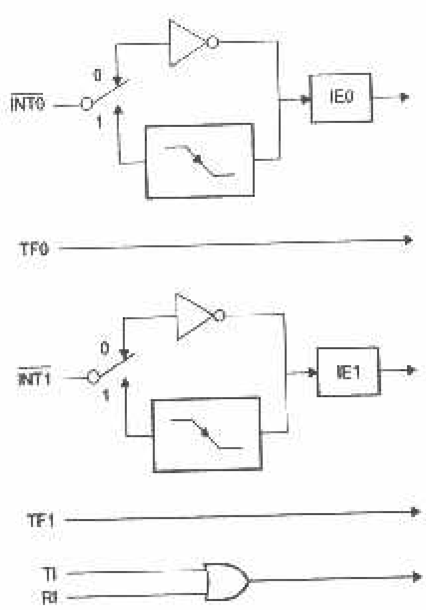
(MSB)							(LSB)
EA	–	–	ES	ET1	EX1	ET0	EX0

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
–	IE.5	Reserved
ES	IE.4	Serial Port 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 reserved bits, because they may be used in future AT89 products.		

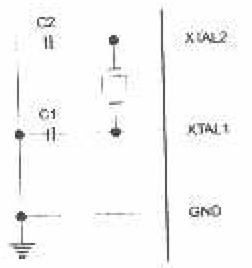
Figure 1. Interrupt Sources



Oscillator Characteristics

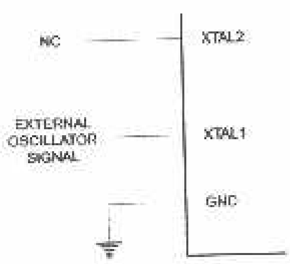
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



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

Figure 3. External Clock Drive Configuration



dle Mode

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

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

ower-down lode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by activation of an enabled external interrupt into $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Table 5. Status of External Pins During Idle and Power-down Modes

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

Program Memory Lock Bits

The AT89S51 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.

Table 6. Lock Bit Protection Modes

	Program Lock Bits			
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
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 is disabled
3	P	P	U	Same as mode 2, but verify is also disabled
4	P	P	P	Same as mode 3, but external execution is also disabled

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.

Programming the Flash – Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S51, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 µs. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89S51 features $\overline{\text{Data}}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.0 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 data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 51H indicates 89S51
(200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/$\overline{PROG}$ low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

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

rogramming ıe Flash – erial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be 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 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

ırial ogramming gorithm

To program and verify the AT89S51 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 33 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 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that 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 device 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.

$\overline{\text{Data}}$ Polling: The $\overline{\text{Data}}$ Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

erial rogramming ıstruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 8 on page 18.

'rogramming ıterface – 'arallel Mode

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

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

ıble 7. Flash Programming Modes

lode	V_{CC}	RST	$\overline{\text{PSEN}}$	ALE/ $\overline{\text{PROG}}$	$\overline{\text{EA}}$/ V_{pp}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.3-0 Address	P1.7-0 Address
/rite Code Data	5V	H	L	⊔ (2)	12V	L	H	H	H	H	D_{IN}	A11-8	A7-0
ead Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A11-8	A7-0
/rite Lock Bit 1	5V	H	L	⊔ (3)	12V	H	H	H	H	H	X	X	X
/rite Lock Bit 2	5V	H	L	⊔ (3)	12V	H	H	H	L	L	X	X	X
/rite Lock Bit 3	5V	H	L	⊔ (3)	12V	H	L	H	H	L	X	X	X
ead Lock Bits 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
hip Erase	5V	H	L	⊔ (1)	12V	H	L	H	L	L	X	X	X
ead Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	0000	00H
ead Device ID	5V	H	L	H	H	L	L	L	L	L	51H	0001	00H
ead Device ID	5V	H	L	H	H	L	L	L	L	L	06H	0010	00H

tes:
1. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Chip Erase.
2. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Write Code Data.
3. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Write Lock Bits.
4. RDY/$\overline{\text{BSY}}$ signal is output on P3.0 during programming.
5. X = don't care.

Figure 4. Programming the Flash Memory (Parallel Mode)

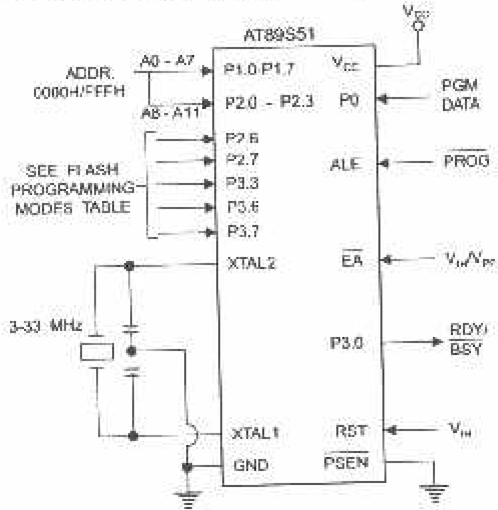
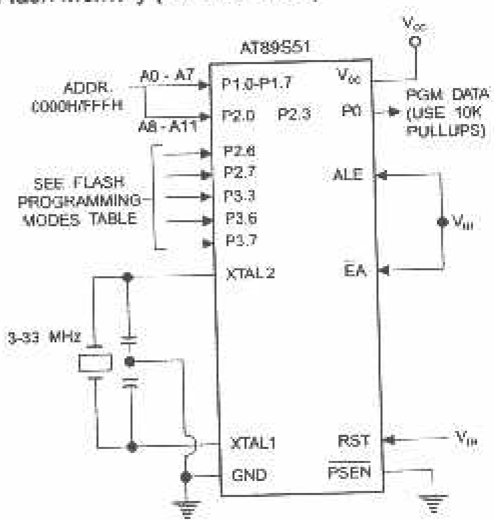


Figure 5. Verifying the Flash Memory (Parallel Mode)

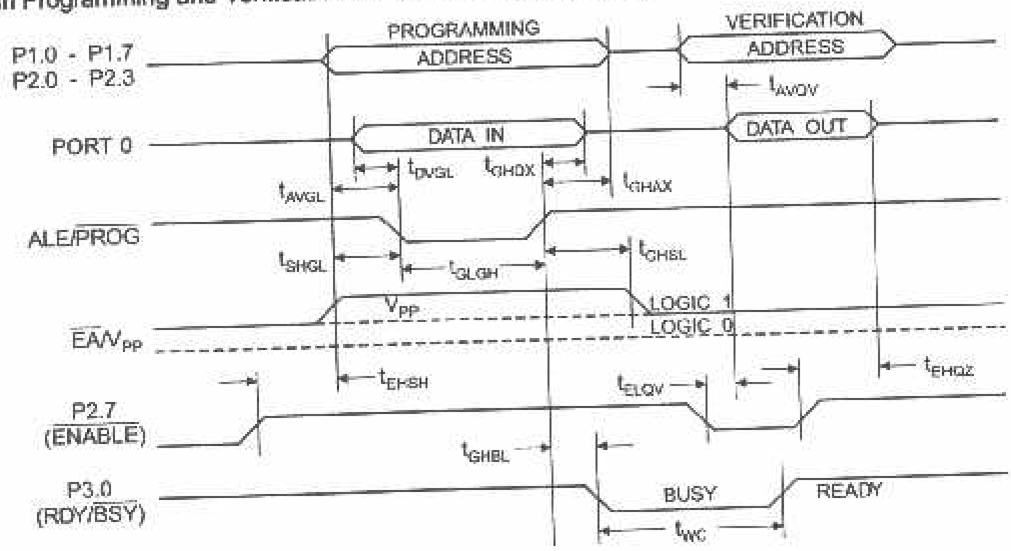


Flash Programming and Verification Characteristics (Parallel Mode)

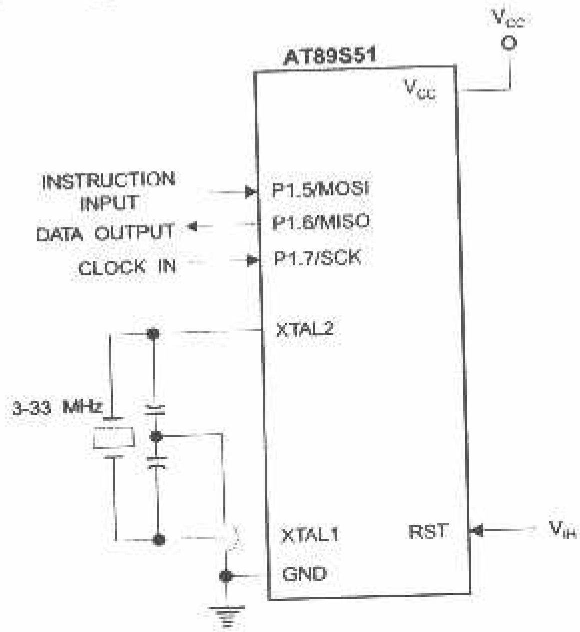
T_A = 20°C to 30°C, V_{CC} = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
$1/t_{CLCL}$	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		µs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		50	µs

igure 6. Flash Programming and Verification Waveforms – Parallel Mode



Igure 7. Flash Memory Serial Downloading



'ash Programming and Verification Waveforms – Serial Mode

jure 8. Serial Programming Waveforms

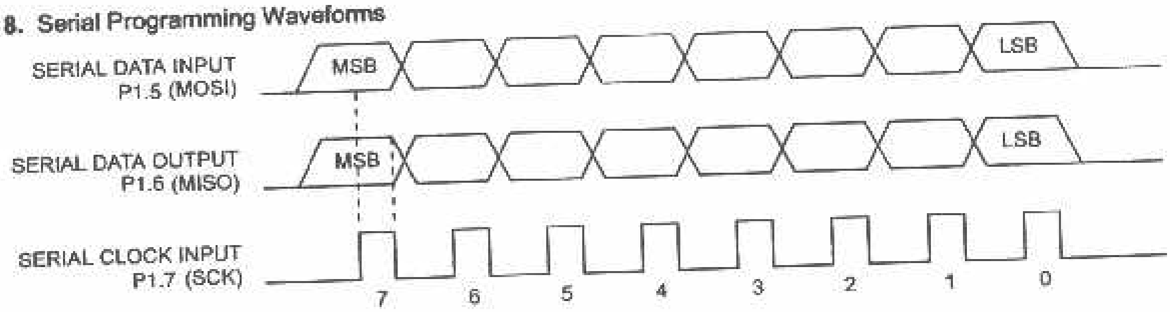


Table 8. Serial Programming Instruction Set

Instruction	Instruction Format Byte 1	Byte 2	Byte 3	Byte 4	Operation
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[2]	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes[1]	0010 1000	xxx A5 A4 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Notes: 1. The signature bytes are not readable in Lock Bit Modes 3 and 4.

2. B1 = 0, B2 = 0 → Mode 1, no lock protection
B1 = 0, B2 = 1 → Mode 2, lock bit 1 activated
B1 = 1, B2 = 0 → Mode 3, lock bit 2 activated
B1 = 1, B1 = 1 → Mode 4, lock bit 3 activated

} Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

After Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at XTAL1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

Serial Programming Characteristics

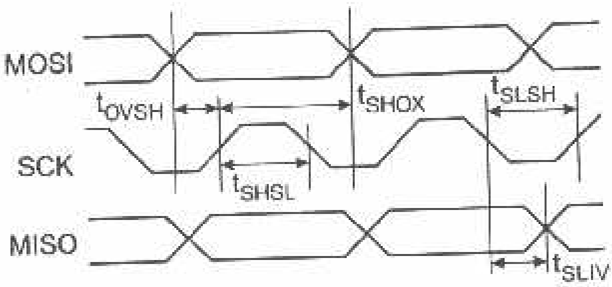


Figure 9. Serial Programming Timing

Table 9. Serial Programming Characteristics, T_A = -40° C to 85° C, V_{CC} = 4.0 - 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	8 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	8 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	μs

Absolute Maximum Ratings*

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

DC Characteristics

he values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High 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
V_{OH1}	Output High 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
L	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
L	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ±10%		-650	µA
I	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RST	Reset Pulldown Resistor		50	300	KΩ
IO	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
C	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode(2)	V_{CC} = 5.5V		50	µA

tes: 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.

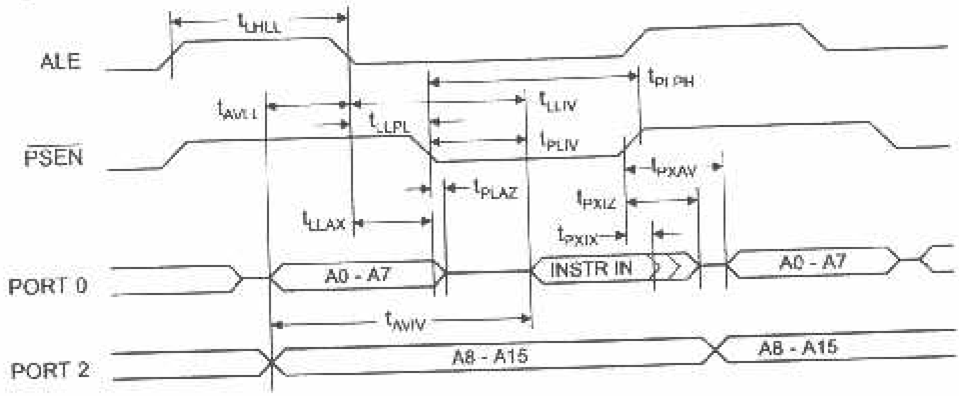
AC Characteristics

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

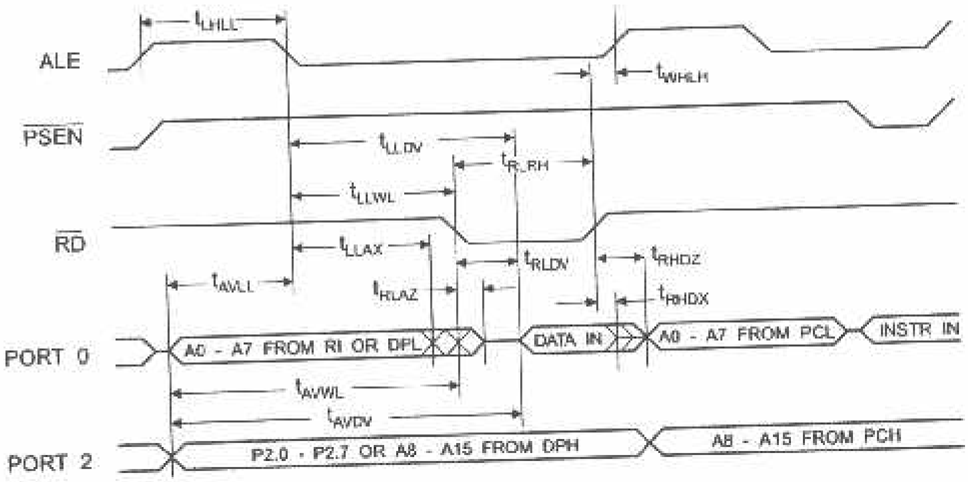
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-45		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-130		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

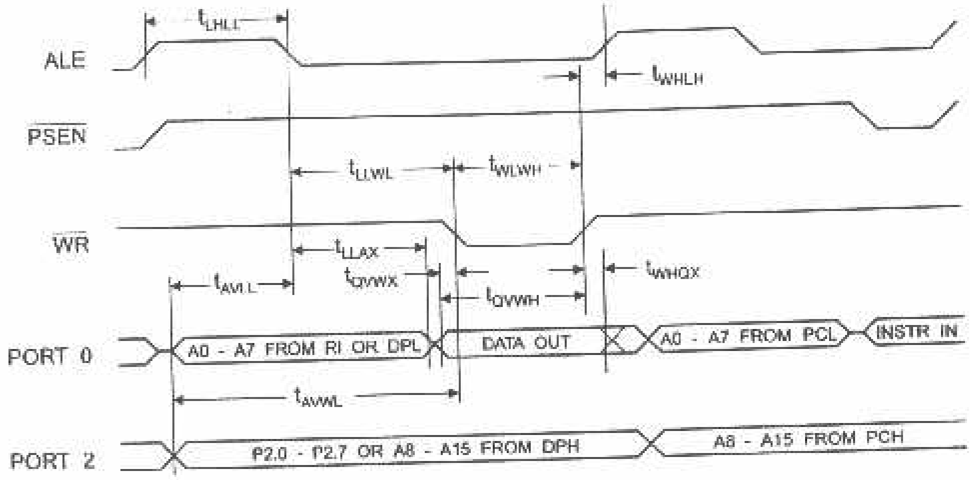
External Program Memory Read Cycle



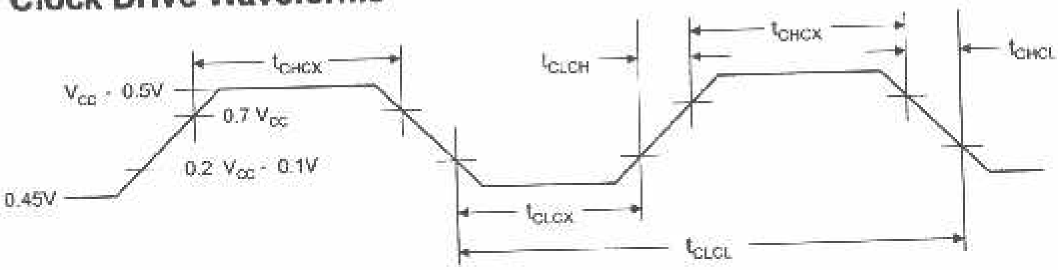
External Data Memory Read Cycle



xternal Data Memory Write Cycle



xternal Clock Drive Waveforms



ternal Clock Drive

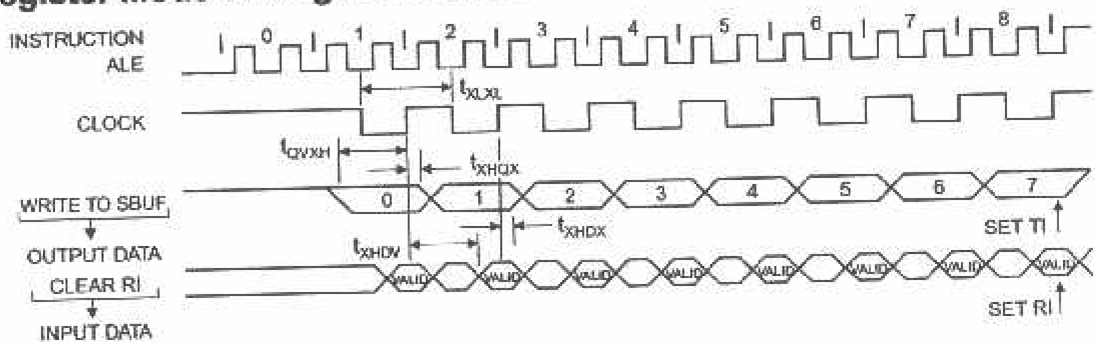
ymbol	Parameter	Min	Max	Units
t_{CLCL}	Oscillator Frequency	0	33	MHz
CL	Clock Period	30		ns
HCX	High Time	12		ns
CX	Low Time	12		ns
CH	Rise Time		5	ns
HCL	Fall Time		5	ns

;erial Port Timing: Shift Register Mode Test Conditions

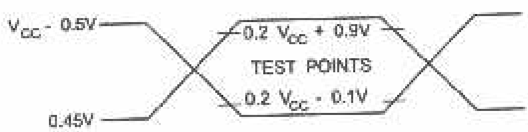
he values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
XLXL	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
QVXH	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
XHQX	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-80		ns
KHDX	Input Data Hold After Clock Rising Edge	0		0		ns
XHDV	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

hift Register Mode Timing Waveforms



: Testing Input/Output Waveforms(1)



e: 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.

oat Waveforms(1)

e: 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.

rdering Information

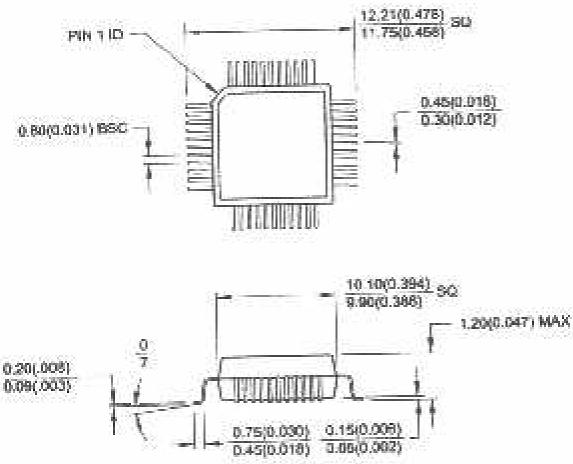
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AC AT89S51-24JC AT89S51-24PC	44A 44J 40P6	Commercial (0° C to 70° C)
		AT89S51-24AI AT89S51-24JI AT89S51-24PI	44A 44J 40P6	Industrial (-40° C to 85° C)
33	4.5V to 5.5V	AT89S51-33AC AT89S51-33JC AT89S51-33PC	44A 44J 40P6	Commercial (0° C to 70° C)

= Preliminary Availability

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

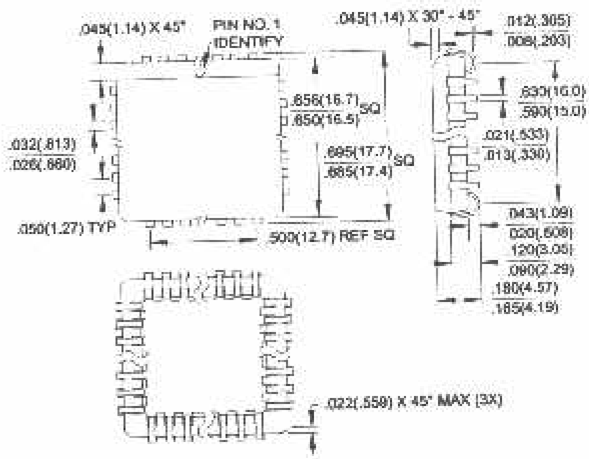
ackaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Millimeters and (Inches)*

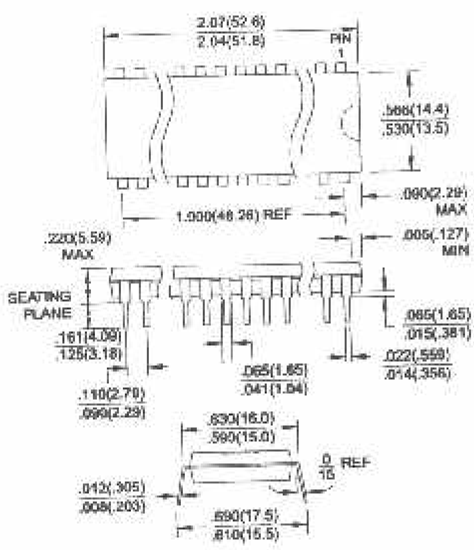


*Controlling dimension: millimeters

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)



40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AC



ɩtmel Headquarters

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

urope
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

sia
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 Product Operations

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

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

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

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

2487A–10/01/xM

Features

- Low-voltage and Standard-voltage Operation
  - 2.7 (V_{CC} = 2.7 to 5.5V)
  - 1.8 (V_{CC} = 1.8 to 5.5V)
- Low-power Devices (I_{SB} = 6 µA at 5.5V) Available
- Internally Organized 8192 x 8
- 2-Wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bi-directional Data Transfer Protocol
- 400 kHz Clock Rate
- Write Protect Pin for Hardware Data Protection
- 32-Byte Page Write Mode (Partial Page Writes Allowed)
- Self-Timed Write Cycle (5 ms max)
- High Reliability
  - Endurance: 1 Million Write Cycles
  - Data Retention: 100 Years
- Lead-free/Halogen-free Devices Available
- 8-lead PDIP, 8-lead JEDEC SOIC and 8-lead TSSOP Packages

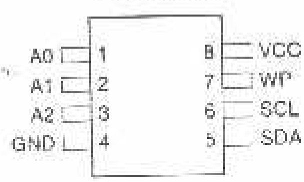
Description

The AT24C64B provides 65,536 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 8192 words of 8 bits each. The device's cascadable feature allows up to 8 devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The AT24C64B is available in space saving 8-lead PDIP, 8-lead JEDEC SOIC and 8-lead TSSOP packages and is accessed via a 2-wire serial interface. In addition, the entire family is available in 2.7V (2.7 to 5.5V) and 1.8V (1.8 to 5.5V) versions.

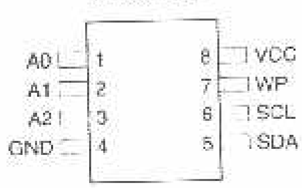
Pin Configurations

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

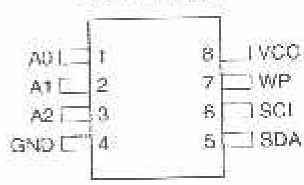
8-lead SOIC



8-lead PDIP



8-lead TSSOP



2-Wire Serial EEPROM

64K (8192 x 8)

AT24C64B

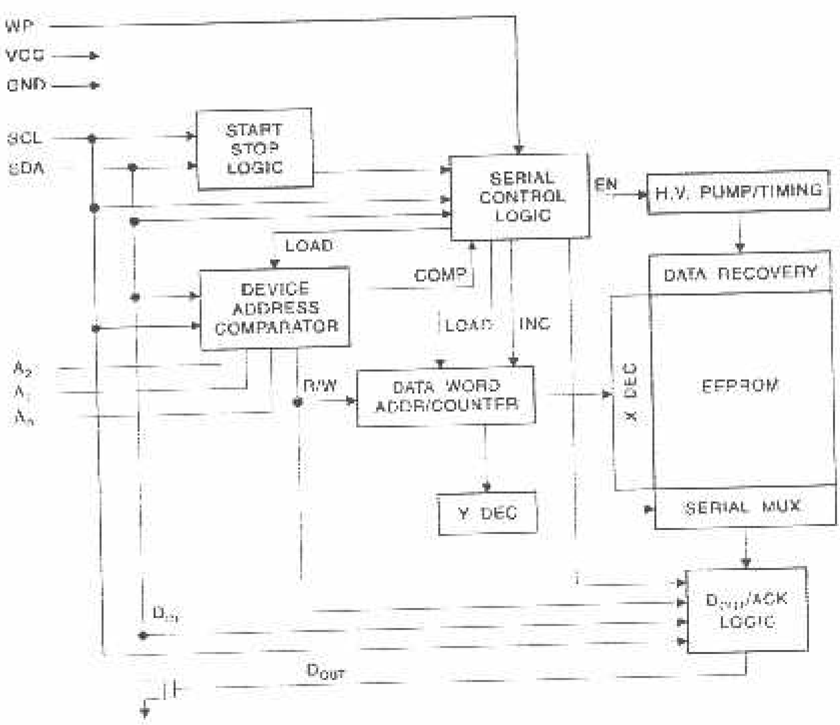
3350C-SEEPR-5/04

Absolute Maximum Ratings*

Operating Temperature	-55 to +125°C
Storage Temperature	-65 to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 to +7.0V
Maximum Operating Voltage	6.25V
DC Output Current	5.0 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.

Block Diagram



Pin Description

SERIAL CLOCK (SCL): The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

SERIAL DATA (SDA): The SDA pin is bidirectional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open collector devices.

DEVICE/ADDRESSES (A2, A1, A0): The A2, A1 and A0 pins are device address inputs that are hard wired or left not connected for hardware compatibility with other AT24CXX devices. When the pins are hardwired, as many as eight 64K devices may be addressed on a single bus system (device addressing is discussed in detail under the Device Addressing section). If the pins are left floating, the A2, A1 and A0 pins will be internally pulled down to GND if the capacitive coupling to the circuit board V_{CC} plane is <3pF. If coupling is >3pF, Atmel recommends connecting the address pins to GND.

WRITE PROTECT (WP): The write protect input, when connected to GND, allows normal write operations. When WP is connected high to V_{CC}, all write operations to the upper quandrant (16K bits) of memory are inhibited. If the pin is left floating, the WP pin will be internally pulled down to GND if the capacitive coupling to the circuit board V_{CC} plane is <3pF. If coupling is >3pF, Atmel recommends connecting the pin to GND.

Memory Organization

AT24C64B, 64K SERIAL EEPROM: The 64K is internally organized as 256 pages of 32 bytes each. Random word addressing requires a 13 bit data word address.

Pin Capacitance[1]

Applicable over recommended operating range from T_A = 25°C, f = 1.0 MHz, V_{CC} = +1.8V.

Symbol	Test Condition	Max	Units	Conditions
$C_{I/O}$	Input/Output Capacitance (SDA)	8	pF	$V_{I/O}$ = 0V
C_{IN}	Input Capacitance (A_0, A_1, A_2, SCL)	6	pF	V_{IN} = 0V

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

DC Characteristics

Applicable over recommended operating range from: T_{AI} = -40 to +85°C, V_{CC} = +1.8 to +5.5V (unless otherwise noted).

Symbol	Parameter	Test Condition		Min	Typ	Max	Units
V_{CC1}	Supply Voltage			1.8		5.5	V
V_{CC2}	Supply Voltage			2.7		5.5	V
V_{CC3}	Supply Voltage			4.5		5.5	V
I_{CC1}	Supply Current	V_{CC} = 5.0V	READ at 400 kHz		0.4	1.0	mA
I_{CC2}	Supply Current	V_{CC} = 5.0V	WRITE at 400 kHz		2.0	3.0	mA
I_{SB1}	Standby Current (1.8V option)	V_{CC} = 1.8V	V_{IN} = V_{CC} or V_{SS}			1.0	µA
I_{SB2}	Standby Current (2.7V option)	V_{CC} = 2.7V	V_{IN} = V_{CC} or V_{SS}			2.0	µA
I_{SB3}	Standby Current (5V option)	V_{CC} = 4.5 - 5.5V	V_{IN} = V_{CC} or V_{SS}			6.0	µA
I_{LI}	Input Leakage Current	V_{IN} = V_{CC} or V_{SS}			0.10	3.0	µA
I_{LO}	Output Leakage Current	V_{OUT} = V_{CC} or V_{SS}			0.05	3.0	µA
V_{IL}	Input Low Level[1]			-0.6		V_{CC} x 0.3	V
V_{IH}	Input High Level[1]			V_{CC} x 0.7		V_{CC} + 0.5	V
V_{OL2}	Output Low Level	V_{CC} = 3.0V	I_{OL} = 2.1 mA			0.4	V
V_{OL1}	Output Low Level	V_{CC} = 1.8V	I_{OL} = 0.15 mA			0.2	V

Note: 1. V_{IL} min and V_{IH} max are reference only and are not tested.

AC Characteristics

Applicable over recommended operating range from T_{AI} = -40°C to +85°C, V_{CC} = +1.8V to +5.5V, CL = 1 TTL Gate and 100 pF (unless otherwise noted).

Symbol	Parameter	AT24C64B				Units
		1.8V – 3.6V		5.0V		
		Min	Max	Min	Max	
f_{SCL}	Clock Frequency, SCL		400		400	kHz
t_{LOW}	Clock Pulse Width Low	1.3		1.2		µs
t_{HIGH}	Clock Pulse Width High	0.6		0.6		µs
t_{I}	Noise Suppression Time[1]		100		50	ns
t_{AA}	Clock Low to Data Out Valid	0.2	0.9	0.1	0.9	µs
t_{BUF}	Time the bus must be free before a new transmission can start[2]	1.3		1.2		µs
$t_{HD.STA}$	Start Hold Time	0.6		0.6		µs
$t_{SU.STA}$	Start Set-up Time	0.6		0.6		µs
$t_{HD.DAT}$	Data In Hold Time	0		0		µs
$t_{SU.DAT}$	Data In Set-up Time	100		100		ns
t_{R}	Inputs Rise Time[2]		0.3		0.3	µs
t_{F}	Inputs Fall Time[2]		300		300	ns
$t_{SU.STO}$	Stop Set-up Time	0.6		0.6		µs
t_{DH}	Data Out Hold Time	200		50		ns
t_{WR}	Write Cycle Time		5		5	ms
Endurance[1]	5.0V, 25°C, Page Mode	1M		1M		Write Cycles

Notes: 1. This parameter is characterized and is not 100% tested (T_A = 25°C)
2. This parameter is characterized and is not 100% tested.

Device Operation

CLOCK and DATA TRANSITIONS: The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods (refer to Data Validity timing diagram). Data changes during SCL high periods will indicate a start or stop condition as defined below.

START CONDITION: A high-to-low transition of SDA with SCL high is a start condition which must precede any other command (refer to Start and Stop Definition timing diagram).

STOP CONDITION: A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the stop command will place the EEPROM in a standby power mode (refer to Start and Stop Definition timing diagram).

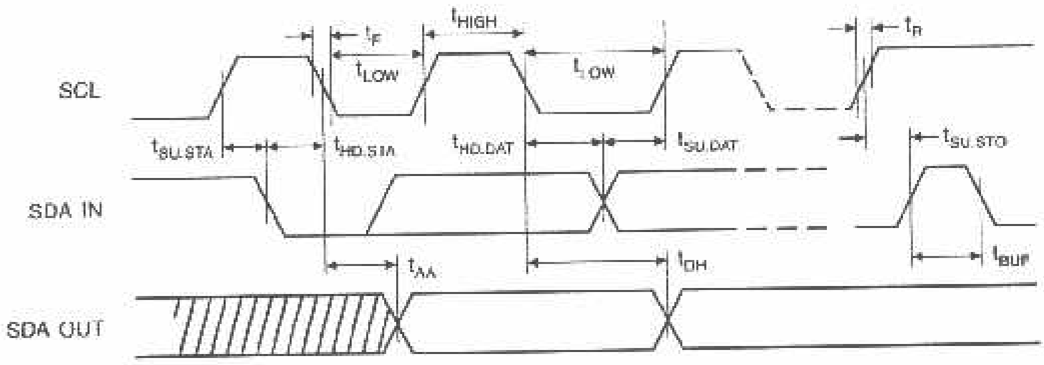
ACKNOWLEDGE: All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero during the ninth clock cycle to acknowledge that it has received each word.

STANDBY MODE: The AT24C64B features a low power standby mode which is enabled: a) upon power-up and b) after the receipt of the STOP bit and the completion of any internal operations.

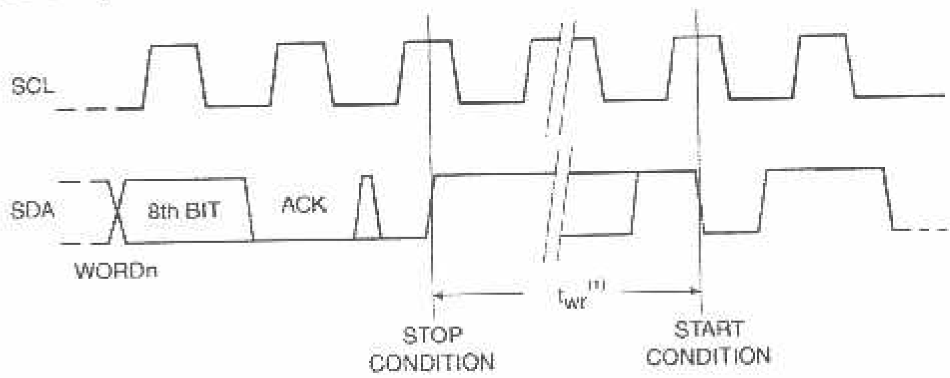
MEMORY RESET: After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

(a) Clock up to 9 cycles, (b) look for SDA high in each cycle while SCL is high and then (c) create a start condition as SDA is high.

Bus Timing
SCL: Serial Clock, SDA: Serial Data I/O

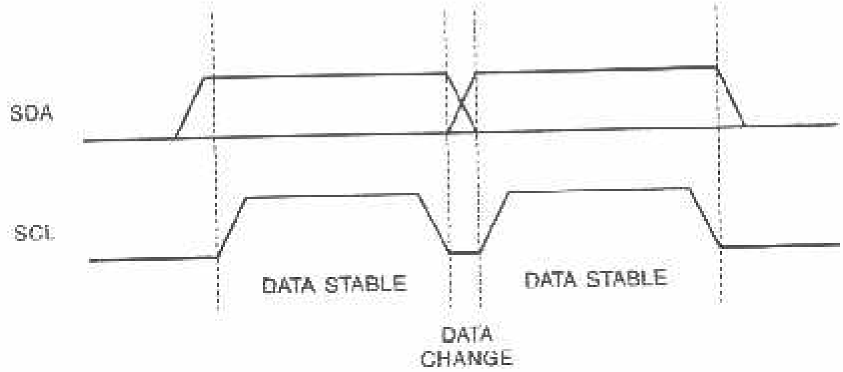


Write Cycle Timing
SCL: Serial Clock, SDA: Serial Data I/O

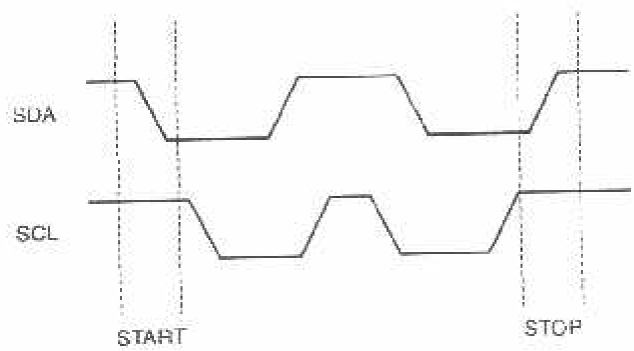


Note: 1. The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

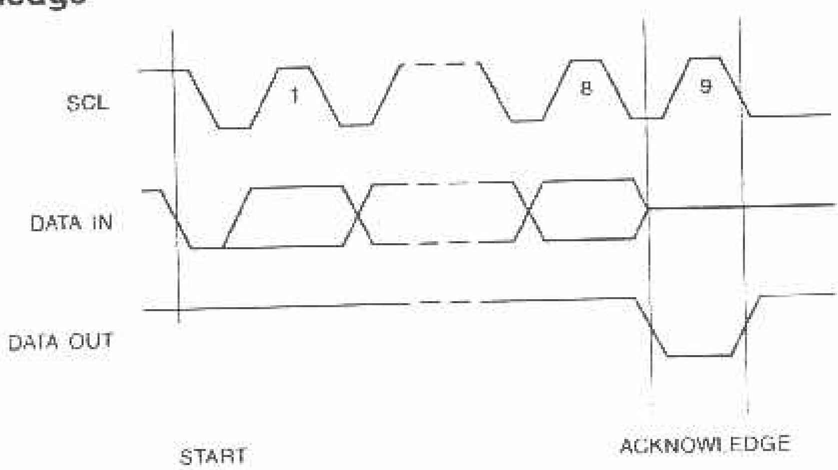
Data Validity



Start and Stop Definition



Output Acknowledge



Device Addressing

The 64K EEPROM requires an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 1). The device address word consists of a mandatory one, zero sequence for the first four most significant bits as shown. This is common to all 2-wire EEPROM devices.

The 64K uses the three device address bits A2, A1, A0 to allow as many as eight devices on the same bus. These bits must compare to their corresponding hardwired input pins. The A2, A1, and A0 pins use an internal proprietary circuit that biases them to a logic low condition if the pins are allowed to float.

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

Upon a compare of the device address, the EEPROM will output a zero. If a compare is not made, the device will return to standby state.

NOISE PROTECTION: Special internal circuitry placed on the SDA and SCL pins prevent small noise spikes from activating the device. A low-V_{CC} detector (5-volt option) resets the device to prevent data corruption in a noisy environment.

DATA SECURITY: The AT24C64B has a hardware data protection scheme that allows the user to write protect the upper quadrant (16K bits) of memory when the WP pin is at V_{CC}.

Write Operations

BYTE WRITE: A write operation requires two 8-bit data word addresses following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero and the addressing device, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle, t_{WR}, to the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (refer to Figure 2).

PAGE WRITE: The 64K EEPROM is capable of 32-byte page writes.

A page write is initiated the same way as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to 31 more data words. The EEPROM will respond with a zero after each data word received. The microcontroller must terminate the page write sequence with a stop condition (refer to Figure 3).

The data word address lower 5 bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than 32 data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten.

ACKNOWLEDGE POLLING: Once the internally-timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero, allowing the read or write sequence to continue.

Read Operations

Read operations are initiated the same way as write operations with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

CURRENT ADDRESS READ: The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "roll over" during read is from the last byte of the last memory page, to the first byte of the first page. The address "roll over" during write is from the last byte of the current page to the first byte of the same page.

Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero but does generate a following stop condition (refer to Figure 4).

RANDOM READ: A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 5).

SEQUENTIAL READ: Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledge. As long as the EEPROM receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will "roll over" and the sequential read will continue. The sequential read operation is terminated when the microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 6).

Figure 1. Device Address

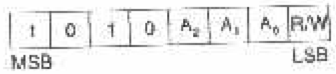


Figure 2. Byte Write

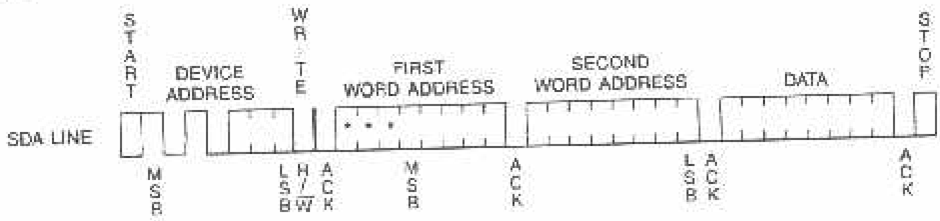
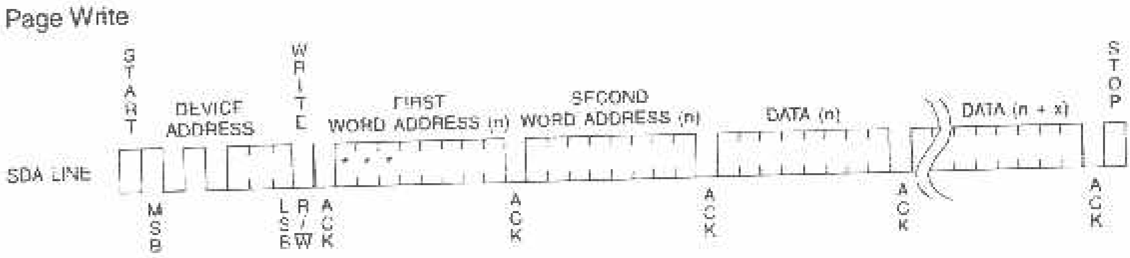


Figure 3. Page Write



Note: 1. * = DON'T CARE bits

Figure 4. Current Address Read

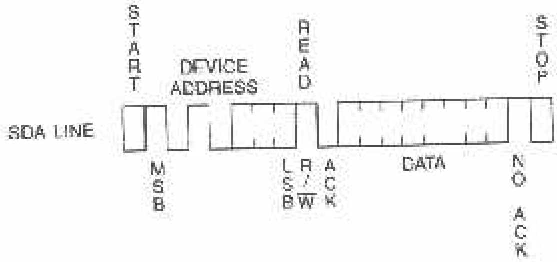
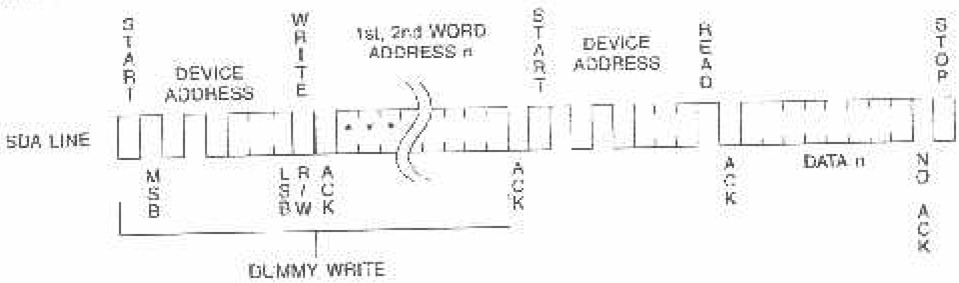
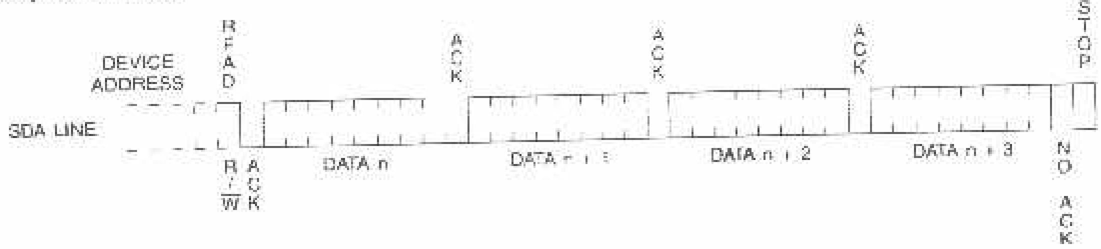


Figure 5. Random Read



Note: 1. * = DON'T CARE bits

Figure 6. Sequential Read



AT24C64B Ordering Information

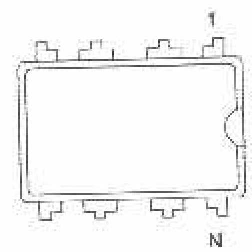
Ordering Code	Package	Operation Range
AT24C64B-10PI-2.7 AT24C64BN-10SI-2.7 AT24C64B-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C64B-10PI-1.8 AT24C64BN-10SI-1.8 AT24C64B-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C64BN-10SU-2.7 AT24C64BN-10SU-1.8 AT24C64B-10TU-2.7 AT24C64B-10TU-1.8	8S1 8S1 8A2 8A2	Lead-free/Halogen-free Industrial Temperature (-40°C to 85°C)

Note: For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics tables.

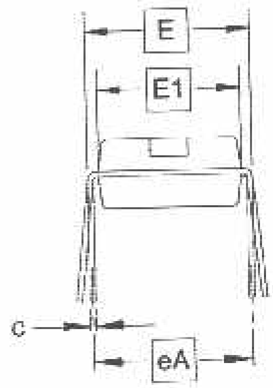
Package Type	
8P3	8-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
Options	
-2.7	Low Voltage (2.7V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

Packaging Information

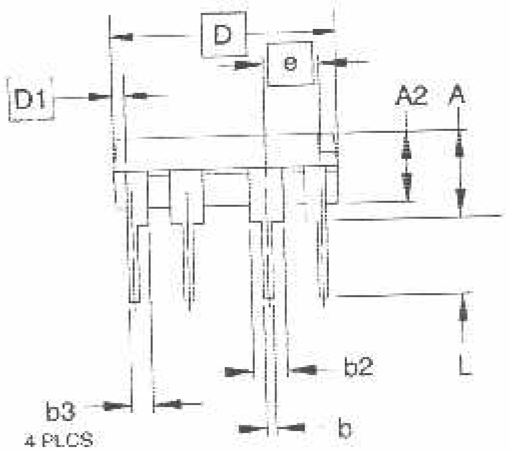
8P3 – PDIP



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = inches)

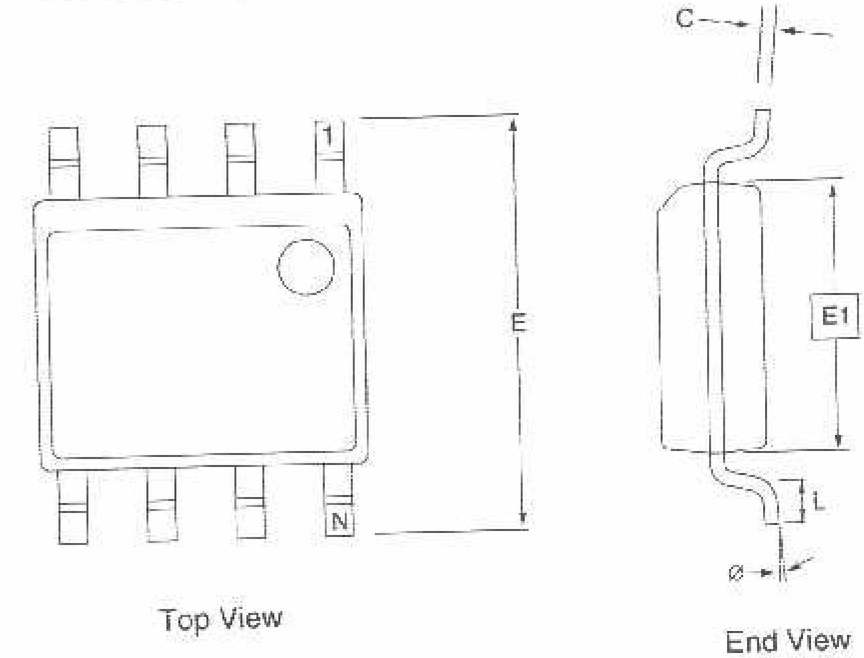
SYMBOL	MIN	NOM	MAX	NOTE
A			0.210	2
A2	0.115	0.130	0.195	
b	0.014	0.018	0.022	5
b2	0.045	0.060	0.070	6
b3	0.030	0.039	0.045	6
c	0.008	0.010	0.014	
D	0.355	0.365	0.400	3
D1	0.005			3
E	0.300	0.310	0.325	4
E1	0.240	0.250	0.280	3
e		0.100 BSC		
eA		0.300 BSC		4
L	0.115	0.130	0.150	2

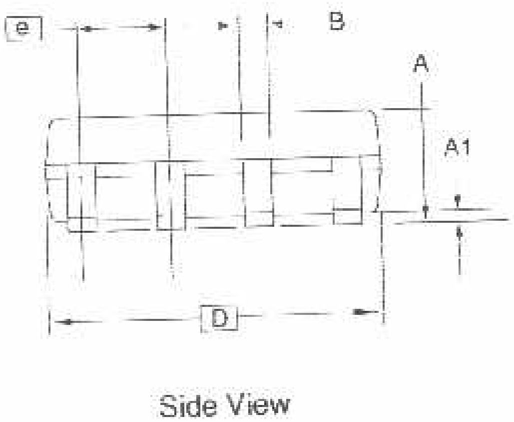
Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-001, Variation BA for additional information.
2. Dimensions A and L are measured with the package seated in JEDEC seating plane Gauge GS-3.
3. D, D1 and E1 dimensions do not include mold Flash or protrusions. Mold Flash or protrusions shall not exceed 0.010 inch.
4. E and eA measured with the leads constrained to be perpendicular to datum.
5. Pointed or rounded lead tips are preferred to ease insertion.
6. b2 and b3 maximum dimensions do not include Dambar protrusions. Dambar protrusions shall not exceed 0.010 (0.25 mm).

01/09/02

	TITLE	DRAWING NO.	REV.
ATMEL 2325 Orchard Parkway San Jose, CA 95131	**8P3**, 8-lead, 0.300" Wide Body, Plastic Dual In-line Package (PDIP)	8P3	B

8S1 – JEDEC SOIC





COMMON DIMENSIONS
(Unit of Measure = mm)

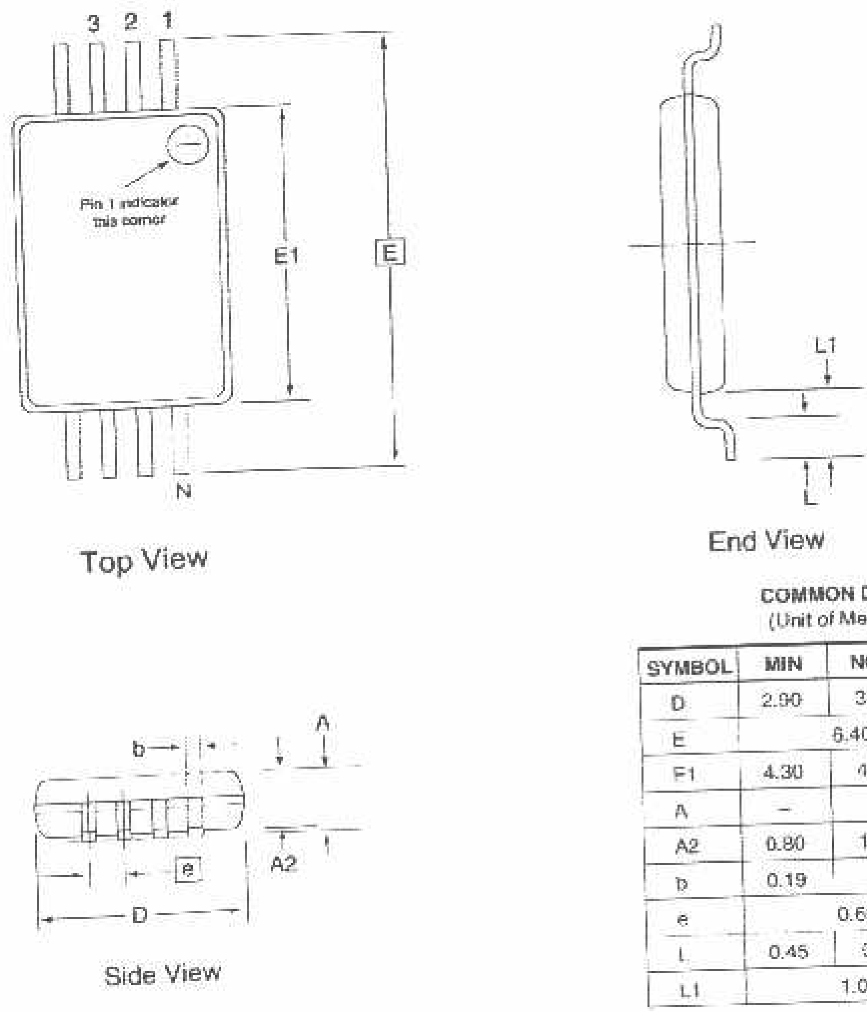
SYMBOL	MIN	NOM	MAX	NOTE
A	1.35	–	1.75	
A1	0.10	–	0.25	
b	0.31	–	0.51	
C	0.17	–	0.25	
D	4.80	–	5.00	
E1	3.81	–	3.99	
E	5.79	–	6.20	
e	1.27 BSC			
L	0.40	–	1.27	
Ø	0°	–	8°	

Note: These drawings are for general information only. Refer to JEDEC Drawing MS-012, Variation AA for proper dimensions, tolerances, datums, etc.

10/7/03

	TITLE	DRAWING NO.	REV.
ATMEL 1150 E. Cheyenne Mtn. Blvd. Colorado Springs, CO 80906	**8S1**, 8-lead (0.150" Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)	8S1	B

8A2 – TSSOP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
D	2.90	3.00	3.10	2, 5
E		6.40 BSC		
E1	4.30	4.40	4.50	3, 5
A	–	–	1.20	
A2	0.80	1.00	1.05	
b	0.19	–	0.30	4
e		0.65 BSC		
L	0.45	0.60	0.75	
L1		1.00 REF		

Notes:
1. This drawing is for general information only. Refer to JEDEC Drawing MO-153, Variation AA, for proper dimensions, tolerances, datums, etc.
2. Dimension D does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006 in) per side.
3. Dimension E1 does not include inter-lead Flash or protrusions. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010 in) per side.
4. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08 mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius of the foot. Minimum space between protrusion and adjacent lead is 0.07 mm.
5. Dimension D and E1 to be determined at Datum Plane H.

5/30/02

	TITLE	DRAWING NO.	REV.
ATMEL 2325 Orchard Parkway San Jose, CA 95131	**8A2**, 8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)	8A2	B

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
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

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

3350C–SEEPR–5/04 xM

HARRIS SEMICONDUCTOR

82C55A

CMOS Programmable Peripheral Interface

August 1996

Features

- Pin Compatible with NMOS 8255A
- 24 Programmable I/O Pins
- Fully TTL Compatible
- High Speed, No "Wait State" Operation with 5MHz and 8MHz 80C86 and 80C88
- Direct Bit Set/Reset Capability
- Enhanced Control Word Read Capability
- Scaled SAJI IV CMOS Process
- 2.5mA Drive Capability on All I/O Ports
- Low Standby Power (ICCSB) 10μA

Ordering Information

PACKAGE	TEMPERATURE RANGE	5MHz	8MHz	PKG. NO.
Plastic DIP	0°C to +70°C	CP82C55A-5	CP82C55A	E40.6
	-40°C to +85°C	IP82C55A-5	IP82C55A	E40.6
PLCC	0°C to +70°C	CS82C55A-5	CS82C55A	N44.65
	-40°C to +85°C	IS82C55A-5	IS82C55A	N44.65
CERDIP	0°C to +70°C	CD82C55A-5	CD82C55A	F40.6
	-40°C to +85°C	ID82C55A-5	ID82C55A	F40.6
	-55°C to +125°C	MD82C55A-5/B	MD82C55A/B	F40.6
SMD#		8406601QA	8406602QA	F40.6
CLCC	-55°C to +125°C	MR82C55A-5/B	MR82C55A/B	J44.A
SMD#		8406601XA	8406602XA	J44.A

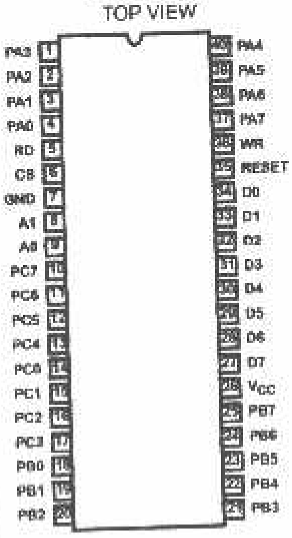
Description

The Harris 82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance and industry standard configuration of the 82C55A make it compatible with the 80C86, 80C88 and other microprocessors.

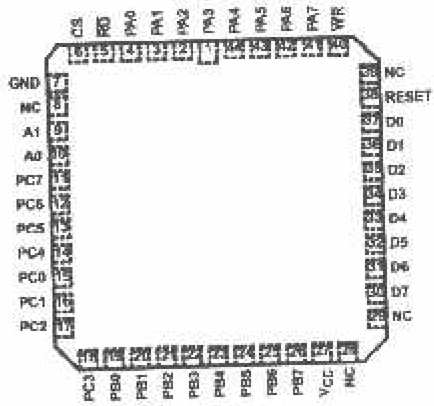
Static CMOS circuit design insures low operating power. TTL compatibility over the full military temperature range and bus hold circuitry eliminate the need for pull-up resistors. The Harris advanced SAJI process results in performance equal to or greater than existing functionally equivalent products at a fraction of the power.

Pinouts

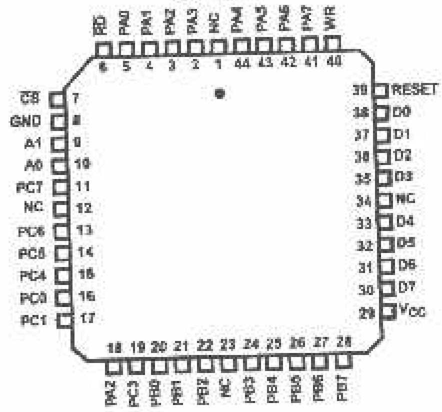
40 LEAD DIP
TOP VIEW



44 LEAD CLCC
TOP VIEW



44 LEAD PLCC
TOP VIEW



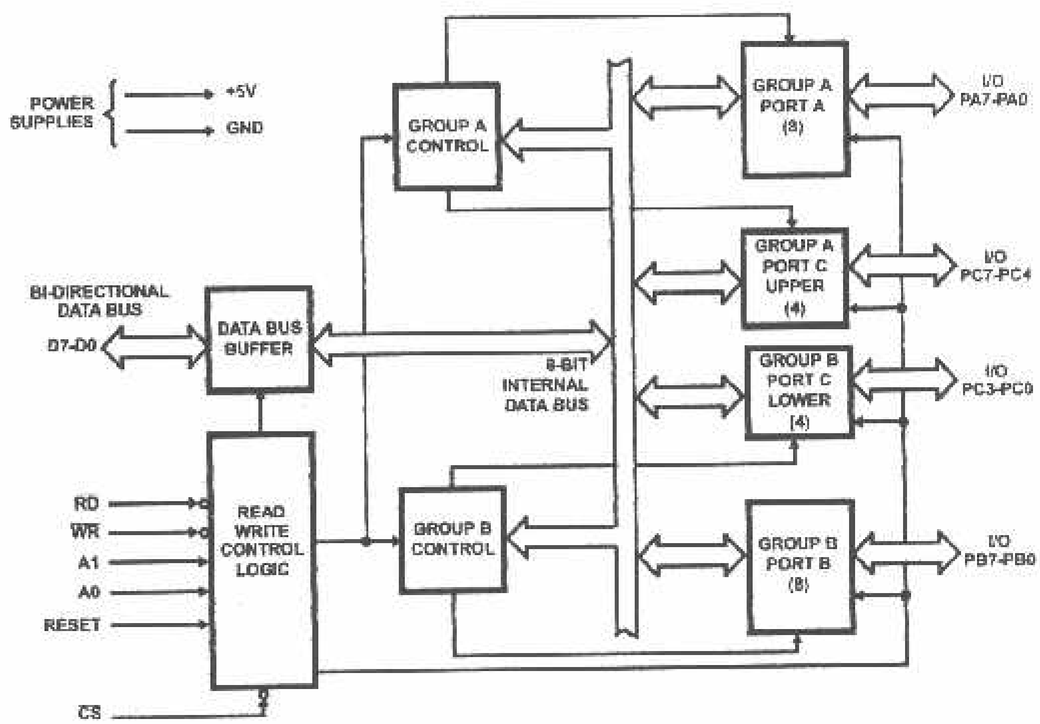
File Number 2969.1

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

Pin Description

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
V_{CC}	26		V_{CC}: The +5V power supply pin. A 0.1μF capacitor between pins 26 and 7 is recommended for decoupling.
GND	7		GROUND
D0-D7	27-34	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	35	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the "Bus Hold" circuitry turned on.
$\overline{CS}$	6	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
$\overline{RD}$	5	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	36	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	8, 9	I	ADDRESS: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	1-4, 37-40	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	18-25	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	10-17	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

Functional Description



Functional Description

Data Bus Buffer

This three-state bi-directional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(CS) Chip Select. A "low" on this input pin enables the communcation between the 82C55A and the CPU.

(RD) Read. A "low" on this input pin enables 82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 82C55A.

(WR) Write. A "low" on this input pin enables the CPU to write data or control words into the 82C55A.

(A0 and A1) Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word register. They are normally connected to the least significant bits of the address bus (A0 and A1).

82C55A BASIC OPERATION

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					DISABLE FUNCTION
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

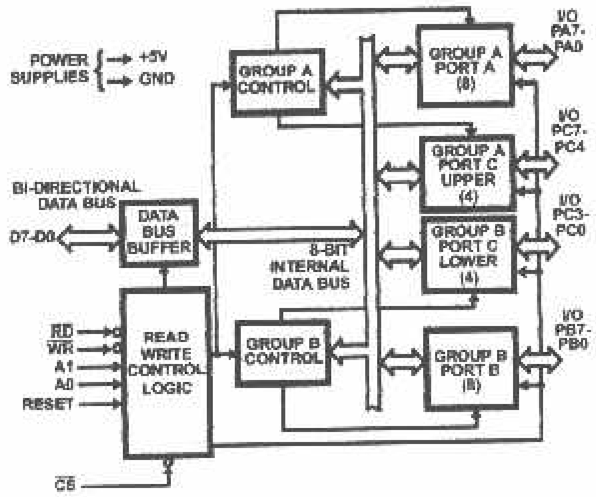


FIGURE 1. 82C55A BLOCK DIAGRAM. DATA BUS BUFFER, READ/WRITE, GROUP A & B CONTROL LOGIC FUNCTIONS

(RESET) Reset. A "high" on this input initializes the control register to 9Bh and all ports (A, B, C) are set to the input mode. "Bus hold" devices internal to the 82C55A will hold the I/O port inputs to a logic "1" state with a maximum hold current of 400μA.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7 - C4)

Control Group B - Port B and Port C lower (C3 - C0)

The control word register can be both written and read as shown in the "Basic Operation" table. Figure 4 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus-hold devices are present on Port A. See Figure 2A.

Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer. See Figure 2B.

Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal output and status signal inputs in conjunction with ports A and B. See Figure 2B.

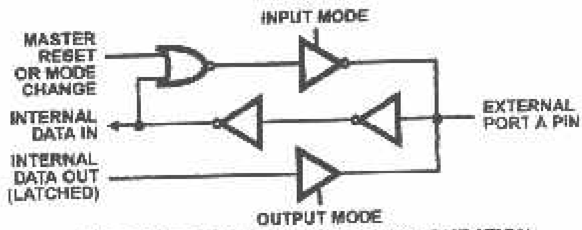


FIGURE 2A. PORT A BUS-HOLD CONFIGURATION

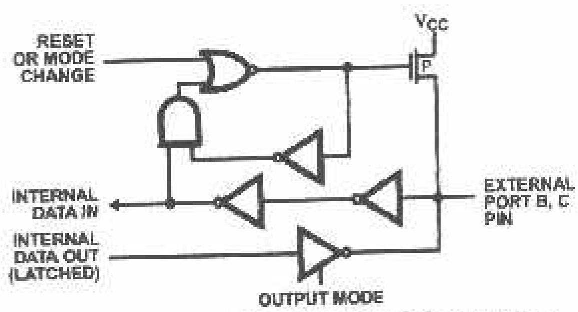


FIGURE 2B. PORT B AND C BUS-HOLD CONFIGURATION

FIGURE 2. BUS-HOLD CONFIGURATION

Operational Description

Mode Selection

There are three basic modes of operation than can be selected by the system software:

Mode 0 - Basic Input/Output
Mode 1 - Strobed Input/Output
Mode 2 - Bi-directional Bus

When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need to pullup or pull-down resistors in all-CMOS designs. The control word register will contain 9Bh. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine. Any port programmed as an output port is initialized to all zeros when the control word is written.

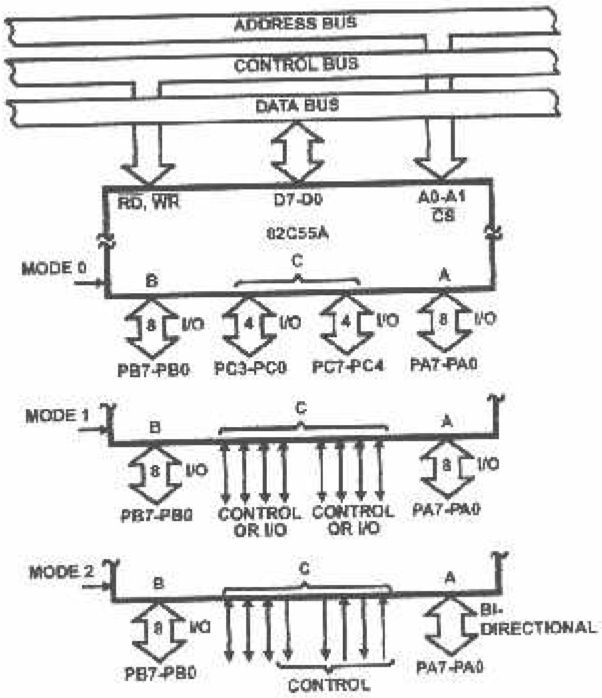


FIGURE 3. BASIC MODE DEFINITIONS AND BUS INTERFACE

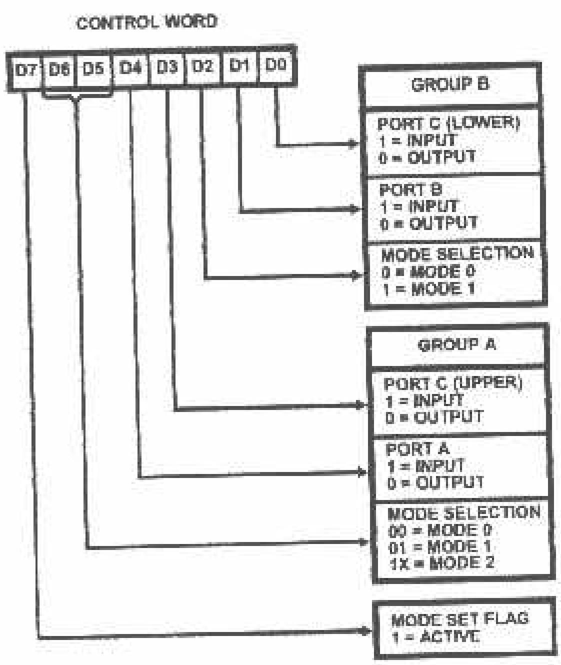


FIGURE 4. MODE DEFINITION FORMAT

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance: Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

The mode definitions and possible mode combinations may seem confusing at first, but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs. PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature (Figure 5)

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were output ports.

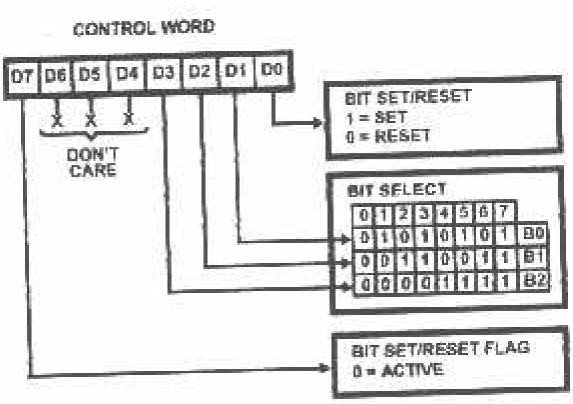


FIGURE 5. BIT SET/RESET FORMAT

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

INTE Flip-Flop Definition

(BIT-SET)-INTE is SET - Interrupt Enable

(BIT-RESET)-INTE is Reset - Interrupt Disable

NOTE: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

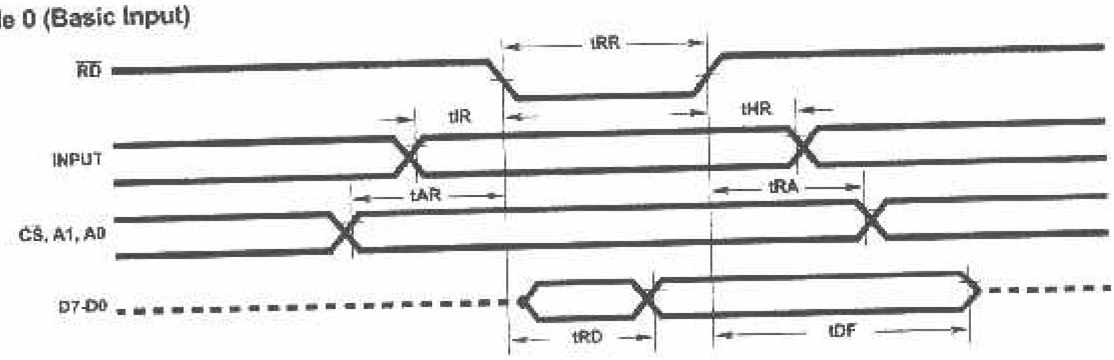
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports
- Any Port can be input or output
- Outputs are latched
- Input are not latched
- 16 different Input/Output configurations possible

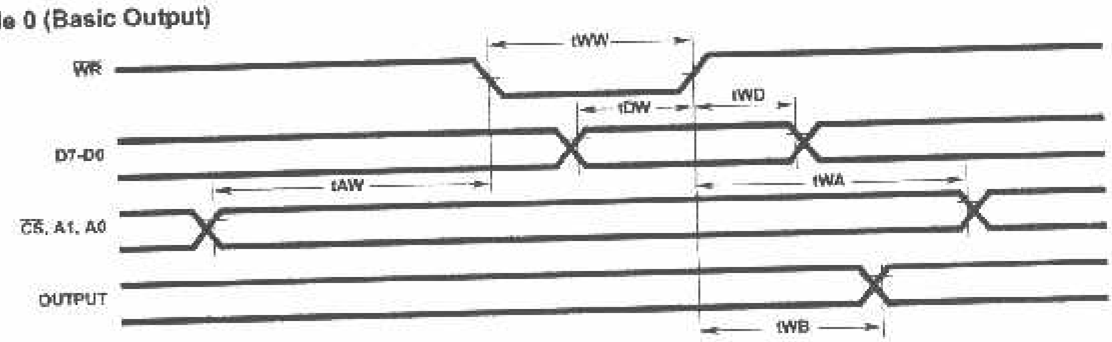
MODE 0 PORT DEFINITION

A		B		GROUP A			GROUP B	
D4	D3	D1	D0	PORT A	PORT C (Upper)	#	PORT B	PORT C (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

Mode 0 (Basic Input)

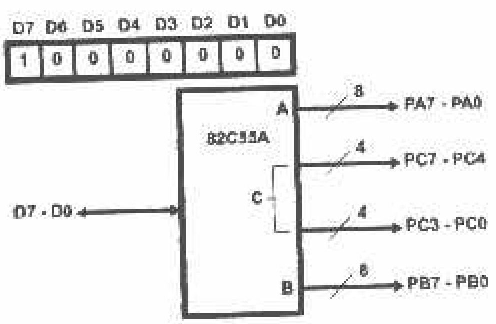


Mode 0 (Basic Output)

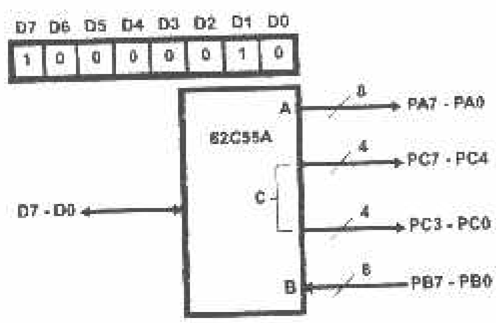


Mode 0 Configurations

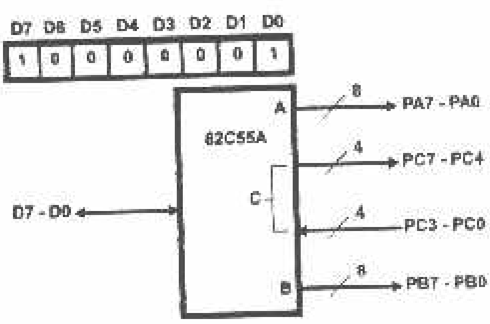
CONTROL WORD #0



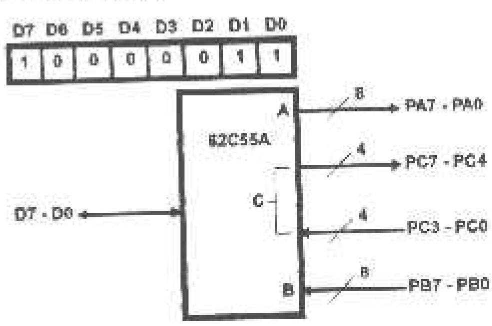
CONTROL WORD #2



CONTROL WORD #1

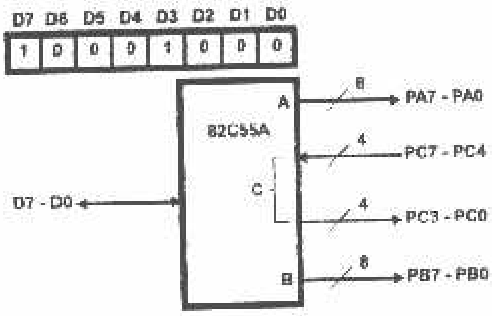


CONTROL WORD #3

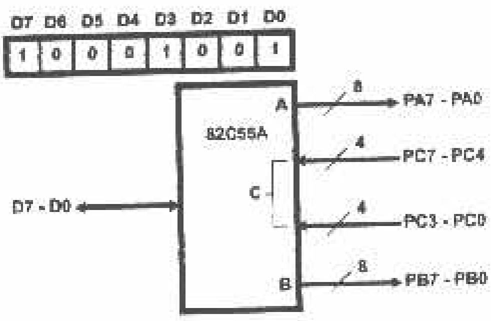


Mode 0 Configurations (Continued)

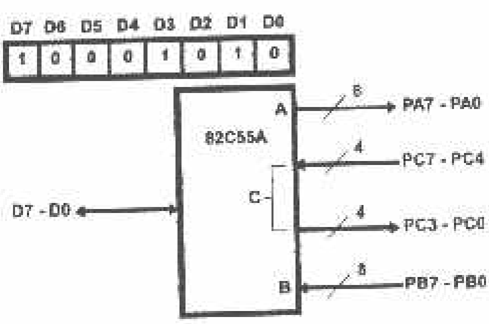
CONTROL WORD #4



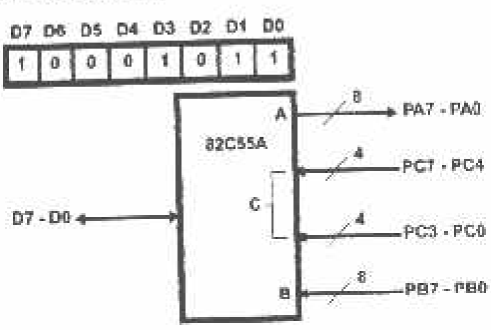
CONTROL WORD #5



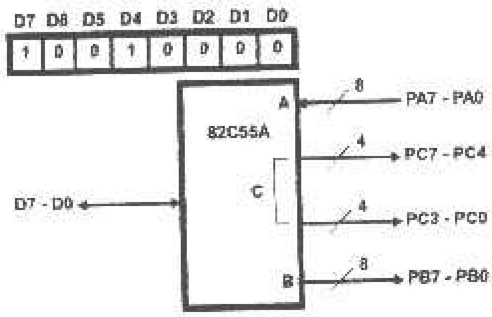
CONTROL WORD #6



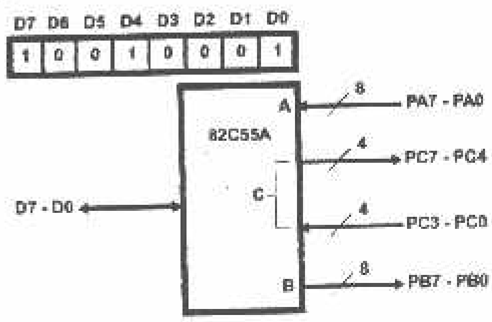
CONTROL WORD #7



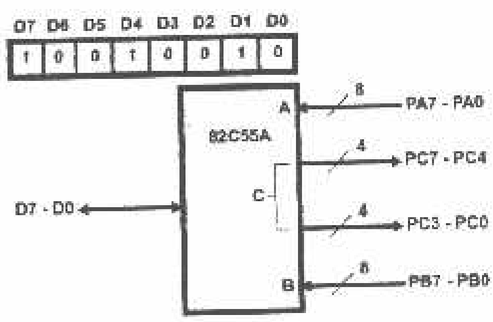
CONTROL WORD #8



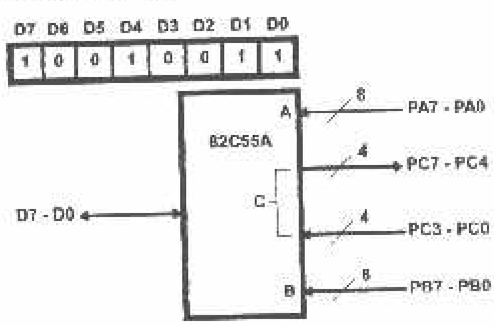
CONTROL WORD #9



CONTROL WORD #10

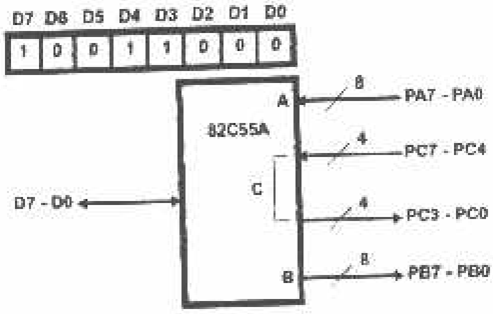


CONTROL WORD #11

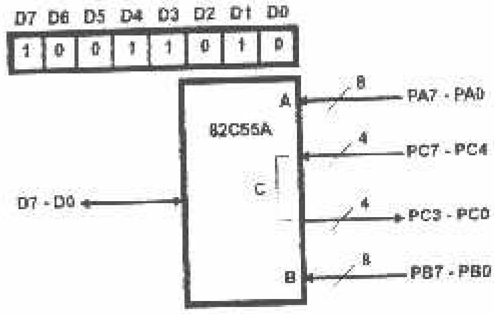


Mode 0 Configurations (Continued)

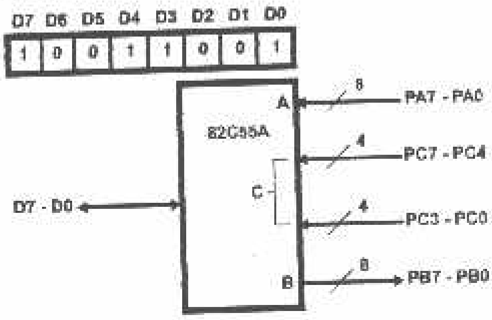
CONTROL WORD #12



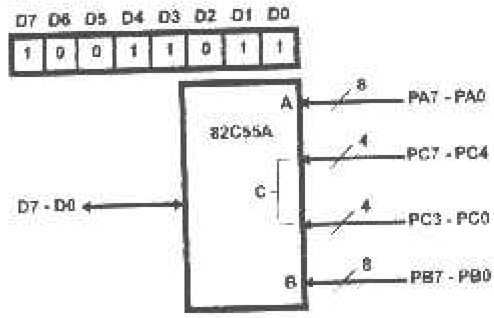
CONTROL WORD #14



CONTROL WORD #13



CONTROL WORD #15



Operating Modes

Mode 1 - (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "hand shaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "hand shaking" signals.

Mode 1 Basic Function Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit port and one 4-bit control/data port
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit port.

Input Control Signal Definition

(Figures 6 and 7)

STB (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch: in essence, and acknowledgment. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

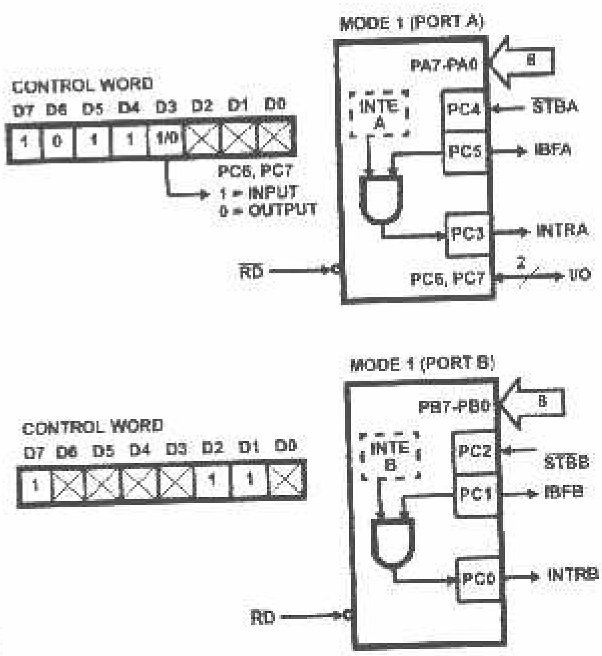


FIGURE 6. MODE 1 INPUT

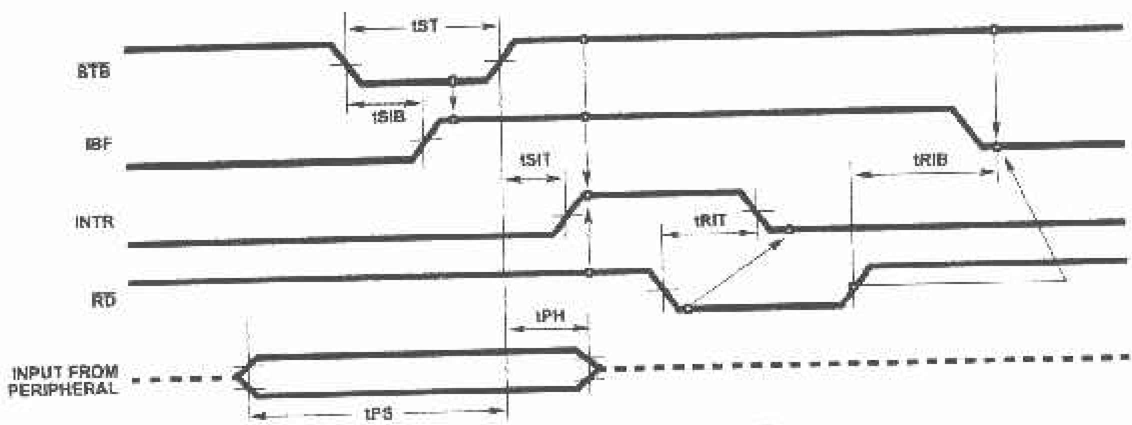


FIGURE 7. MODE 1 (STROBED INPUT)

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when and input device is requesting service. INTR is set by the condition: $\overline{\text{STB}}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{\text{RD}}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC4.

INTE B

Controlled by bit set/reset of PC2.

Output Control Signal Definition

(Figure 8 and 9)

$\overline{\text{OBF}}$ - Output Buffer Full F/F). The $\overline{\text{OBF}}$ output will go "low" to indicate that the CPU has written data out to be specified port. This does not mean valid data is sent out of the part at this time since $\overline{\text{OBF}}$ can go true before data is available. Data is guaranteed valid at the rising edge of $\overline{\text{OBF}}$, (See Note 1). The $\overline{\text{OBF}}$ F/F will be set by the rising edge of the $\overline{\text{WR}}$ input and reset by $\overline{\text{ACK}}$ input being low.

$\overline{\text{ACK}}$ - Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B is ready to be accepted. In essence, a response from the peripheral device indicating that it is ready to accept data, (See Note 1).

INTR - (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{\text{ACK}}$ is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{\text{WR}}$.

INTE A

Controlled by Bit Set/Reset of PC6.

INTE B

Controlled by Bit Set/Reset of PC2.

NOTE: To strobe data into the peripheral device, the user must operate the strobe line in a hand shaking mode. The user needs to send $\overline{\text{OBF}}$ to the peripheral device, generates an $\overline{\text{ACK}}$ from the peripheral device and then latch data into the peripheral device on the rising edge of $\overline{\text{OBF}}$.

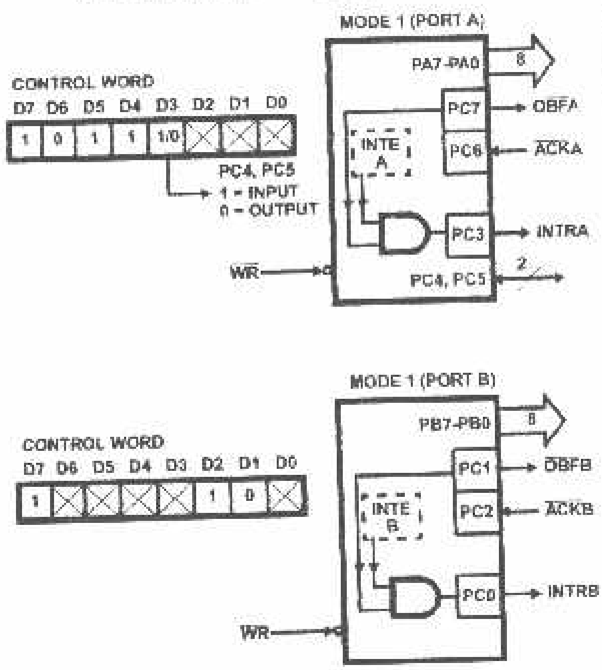


FIGURE 8. MODE 1 OUTPUT

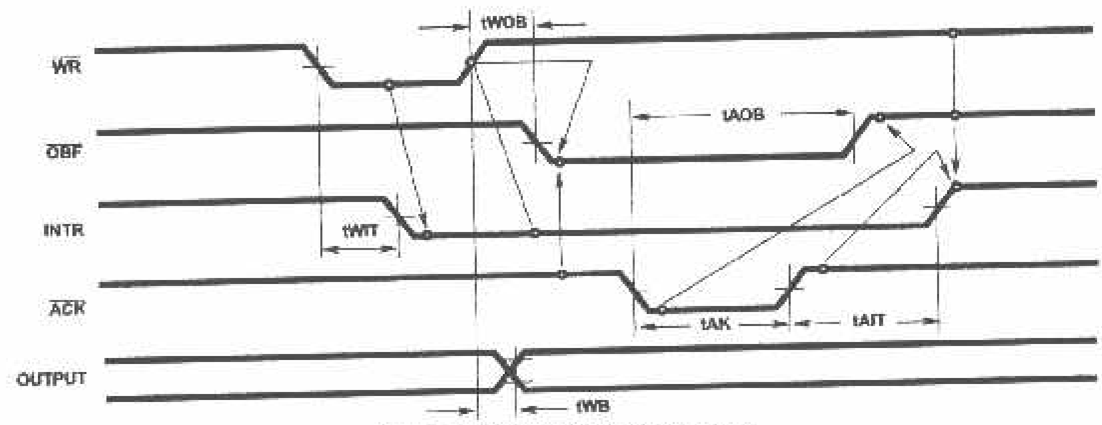
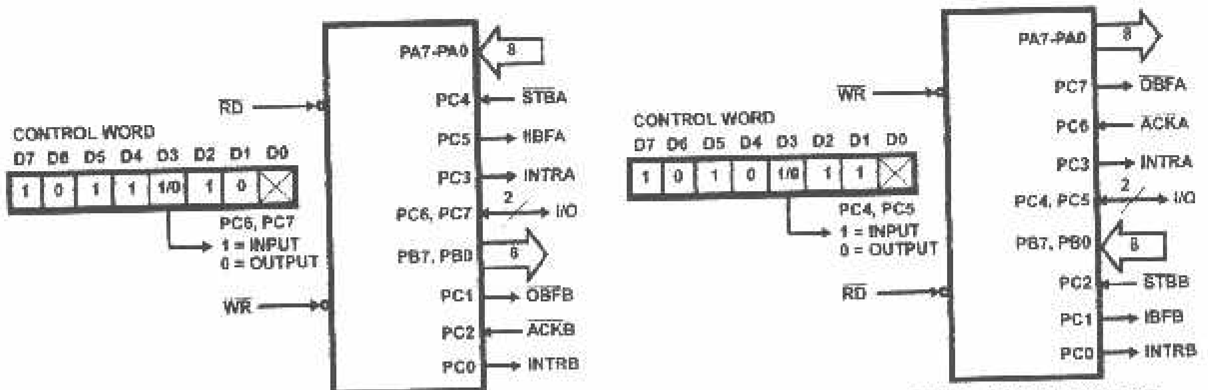


FIGURE 9. MODE 1 (STROBED OUTPUT)



PORT A - (STROBED INPUT)
PORT B - (STROBED OUTPUT)

PORT A - (STROBED OUTPUT)
PORT B - (STROBED INPUT)

Combinations of Mode 1: Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

FIGURE 10. COMBINATIONS OF MODE 1

Operating Modes

Mode 2 (Strobed Bi-Directional Bus I/O)

The functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Hand shaking" signals are provided to maintain proper bus flow discipline similar to Mode 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions:

- Used in Group A only
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C)
- Both inputs and outputs are latched
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A)

Bi-Directional Bus I/O Control Signal Definition

(Figures 11, 12, 13, 14)

INTR - (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

OBF - (Output Buffer Full). The OBF output will go "low" to indicate that the CPU has written data out to port A.

ACK - (Acknowledge). A "low" on this input enables the three-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 - (The INTE flip-flop associated with OBF). Controlled by bit set/reset of PC4.

Input Operations

STB - (Strobe Input). A "low" on this input loads data into the input latch.

IBF - (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 - (The INTE flip-flop associated with IBF). Controlled by bit set/reset of PC4.

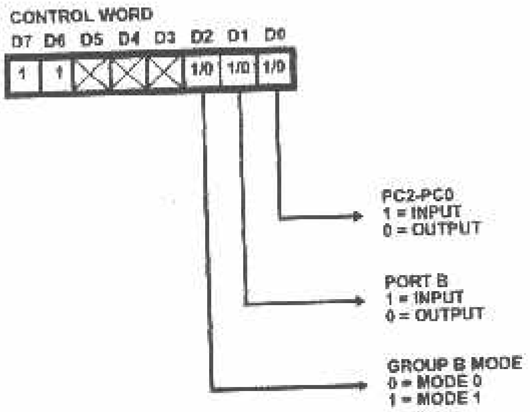


FIGURE 11. MODE CONTROL WORD

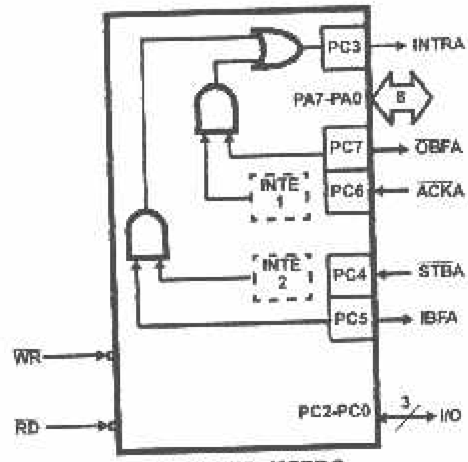
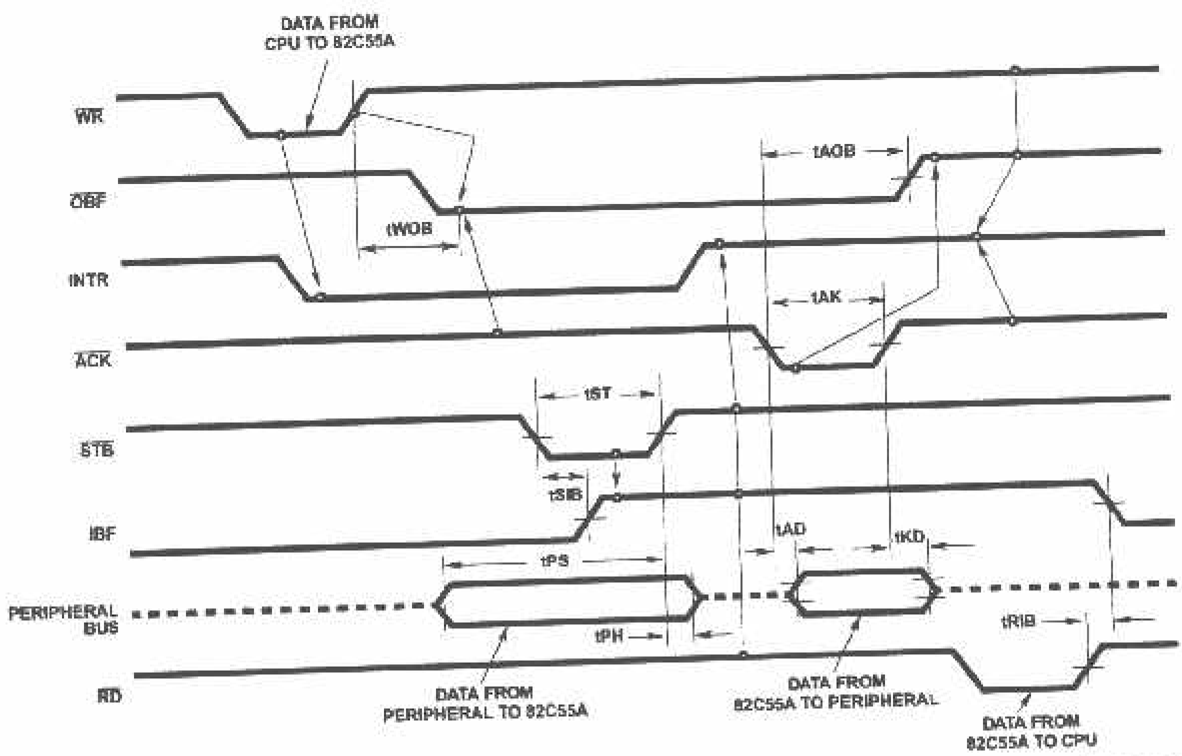


FIGURE 12. MODE 2



NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before RD is permissible. (INTR = IBF • MASK • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • MASK • $\overline{ACK}$ • $\overline{WR}$)

FIGURE 13. MODE 2 (BI-DIRECTIONAL)

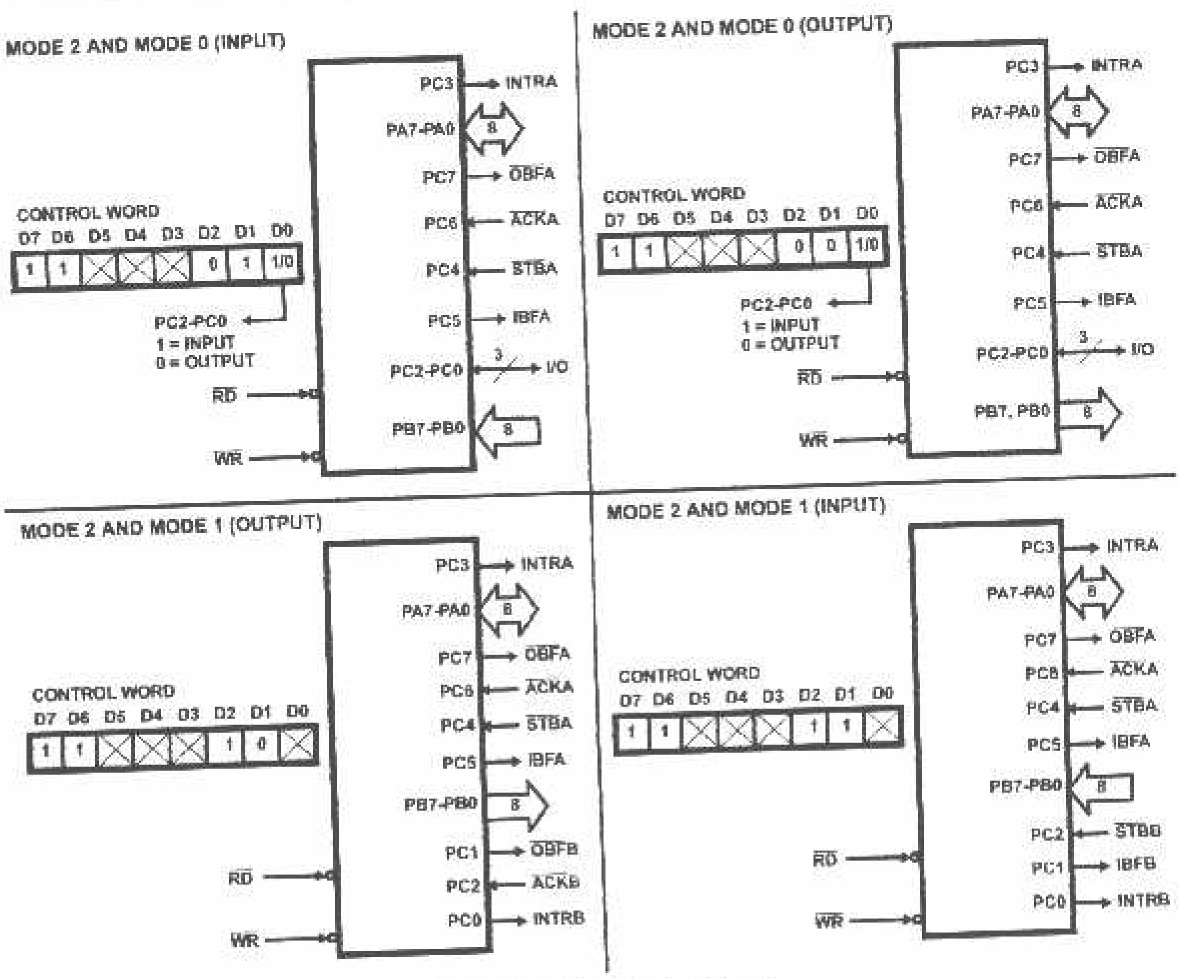


FIGURE 14. MODE 2 COMBINATIONS

MODE DEFINITION SUMMARY

	MODE 0		MODE 1		MODE 2
	IN	OUT	IN	OUT	GROUP A ONLY
PA0	In	Out	In	Out	↔
PA1	In	Out	In	Out	↔
PA2	In	Out	In	Out	↔
PA3	In	Out	In	Out	↔
PA4	In	Out	In	Out	↔
PA5	In	Out	In	Out	↔
PA6	In	Out	In	Out	↔
PA7	In	Out	In	Out	↔
PB0	In	Out	In	Out	Mode 0 or Mode 1 Only
PB1	In	Out	In	Out	
PB2	In	Out	In	Out	
PB3	In	Out	In	Out	
PB4	In	Out	In	Out	
PB5	In	Out	In	Out	
PB6	In	Out	In	Out	
PB7	In	Out	In	Out	
PC0	In	Out	INTRB	INTRB	I/O
PC1	In	Out	IBFB	$\overline{\text{OBFB}}$	I/O
PC2	In	Out	$\overline{\text{STBB}}$	$\overline{\text{ACKB}}$	I/O
PC3	In	Out	INTRA	INTRA	INTRA
PC4	In	Out	$\overline{\text{STBA}}$	I/O	$\overline{\text{STBA}}$
PC5	In	Out	IBFA	I/O	IBFA
PC6	In	Out	I/O	$\overline{\text{ACKA}}$	$\overline{\text{ACKA}}$
PC7	In	Out	I/O	$\overline{\text{OBFA}}$	$\overline{\text{OBFA}}$

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{\text{ACK}}$ and $\overline{\text{STB}}$ lines, will be placed on the data bus. In place of the $\overline{\text{ACK}}$ and $\overline{\text{STB}}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 17.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including IBF and $\overline{\text{OBF}}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{\text{ACK}}$ and $\overline{\text{STB}}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{\text{ACK}}$ and $\overline{\text{STB}}$ lines with the "Set Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 17.

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBFA	INTEA	INTRA	INTEB	IBFB	INTRB
GROUP A					GROUP B		

OUTPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
OBFA	INTEA	I/O	I/O	INTRA	INTEB	OBFB	INTRB
GROUP A					GROUP B		

FIGURE 15. MODE 1 STATUS WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
OBFA	INTE1	IBFA	INTE2	INTRA	X	X	X
GROUP A					GROUP B		

(Defined by Mode 0 or Mode 1 Selection)

FIGURE 16. MODE 2 STATUS WORD FORMAT

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status (Figures 15 and 16)

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is not special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

INTERRUPT ENABLE FLAG	POSITION	ALTERNATE PORT C PIN SIGNAL (MODE)
INTE B	PC2	$\overline{ACKB}$ (Output Mode 1) or $\overline{STBB}$ (Input Mode 1)
INTE A2	PC4	$\overline{STBA}$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACKA}$ (Output Mode 1 or Mode 2)

FIGURE 17. INTERRUPT ENABLE FLAGS IN MODES 1 AND 2

Applications of the 82C55A

The 82C55A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 82C55A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 82C55A to exactly "fit" the application. Figures 18 through 24 present a few examples of typical applications of the 82C55A.

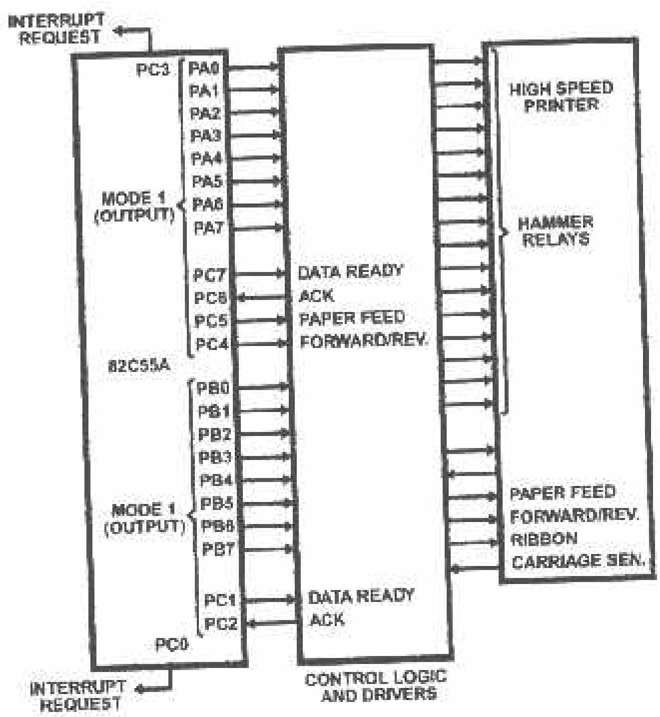


FIGURE 18. PRINTER INTERFACE

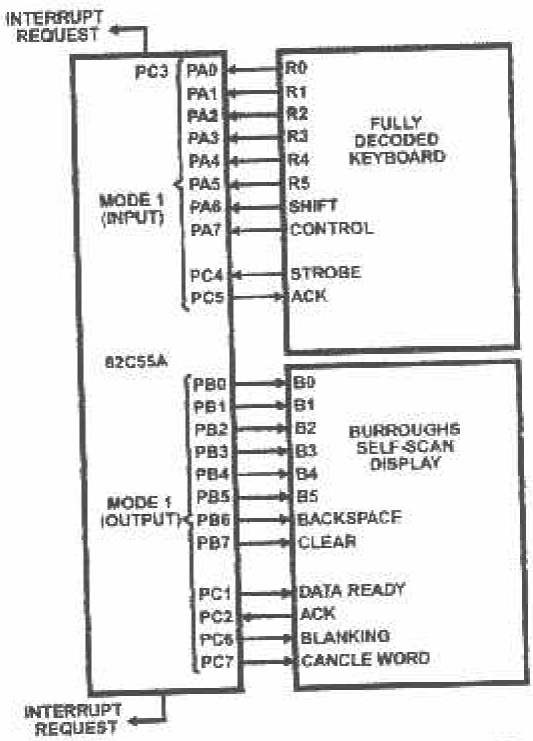


FIGURE 19. KEYBOARD AND DISPLAY INTERFACE

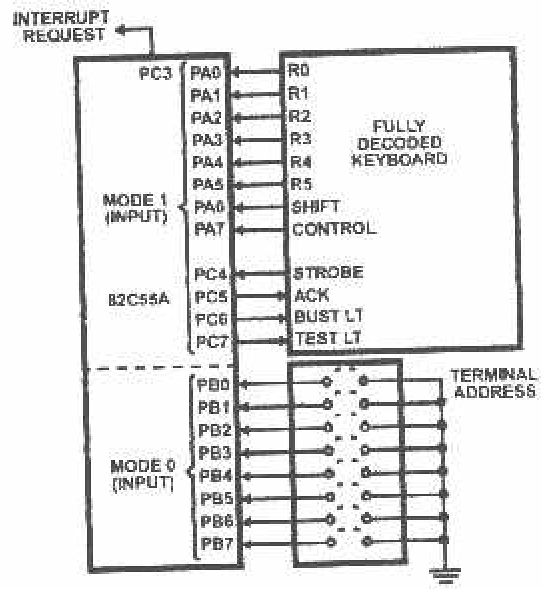


FIGURE 20. KEYBOARD AND TERMINAL ADDRESS INTERFACE

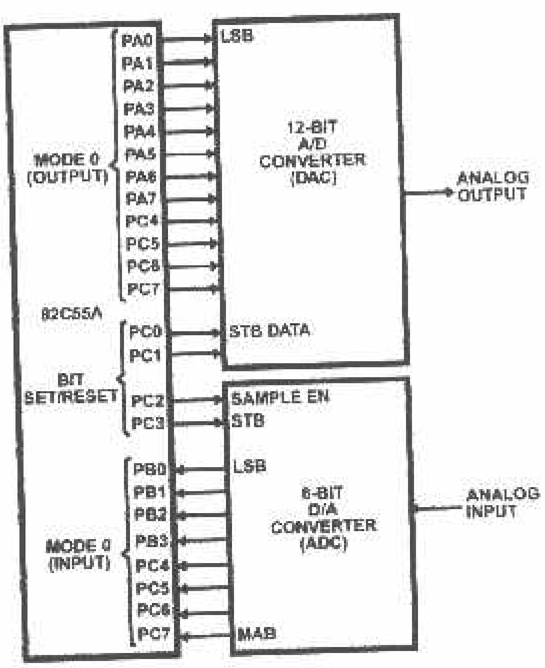


FIGURE 21. DIGITAL TO ANALOG, ANALOG TO DIGITAL

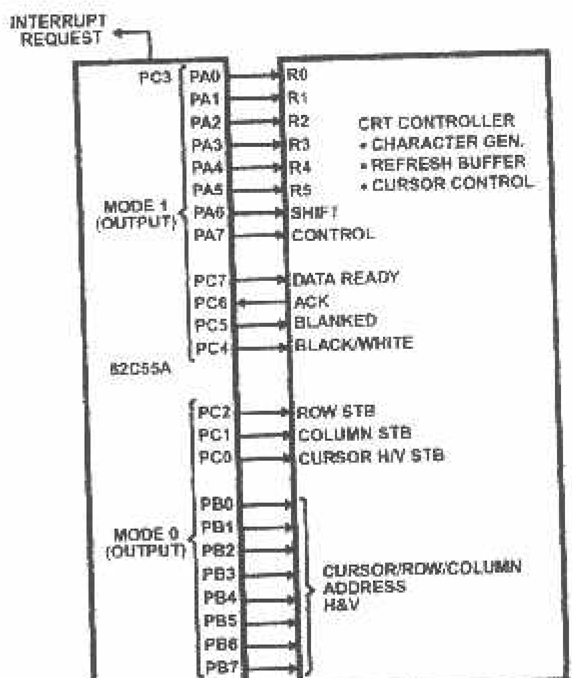


FIGURE 22. BASIC CRT CONTROLLER INTERFACE

Timing Waveforms (Continued)

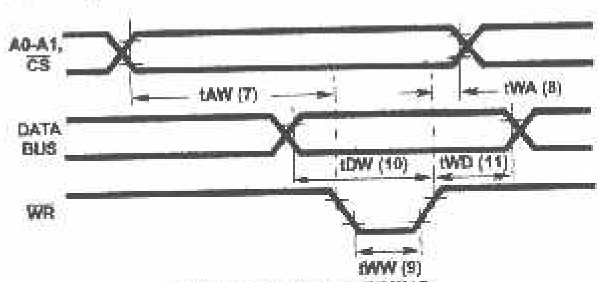


FIGURE 30. WRITE TIMING

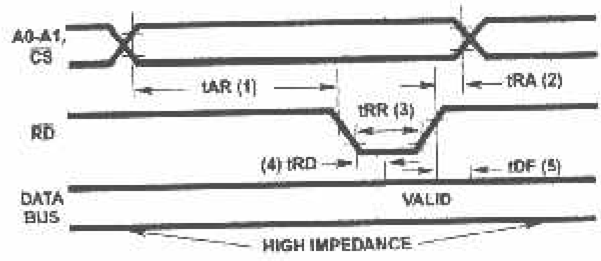
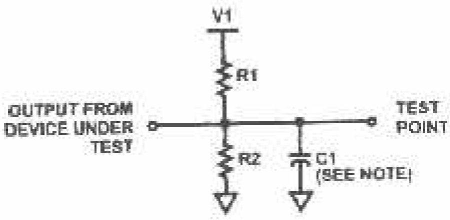


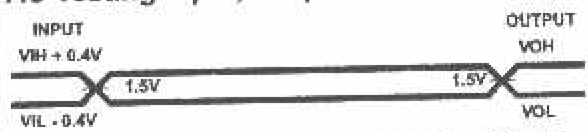
FIGURE 31. READ TIMING

AC Test Circuit



NOTE: Includes STRAY and JIG Capacitance

AC Testing Input, Output Waveforms



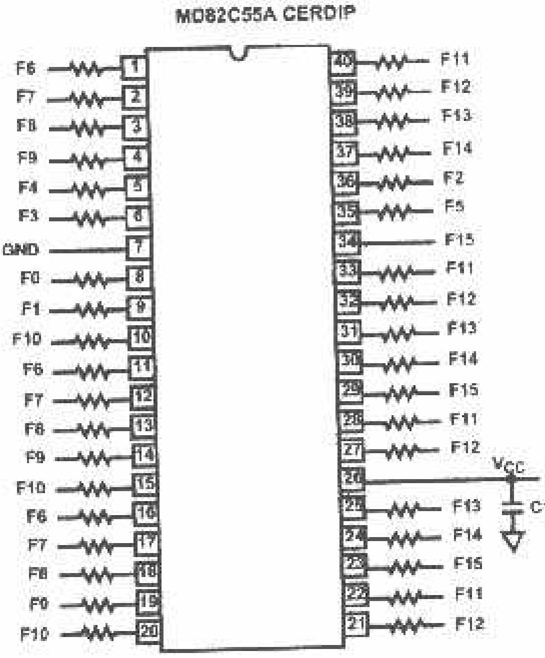
AC Testing: All AC Parameters tested as per test circuits. Input RISE and FALL times are driven at 1ns/V.

TEST CONDITION DEFINITION TABLE

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523Ω	Open	150pF
2	V_{CC}	2kΩ	1.7kΩ	50pF
3	1.5V	750Ω	Open	50pF

Burn-In Circuits

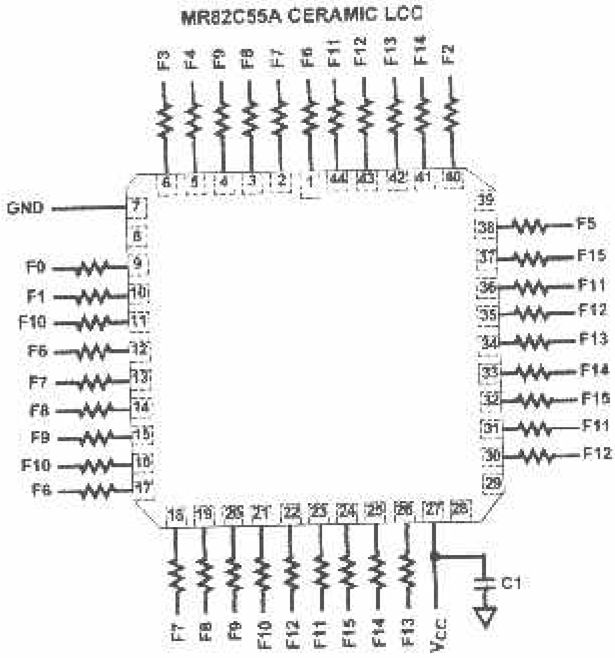
MD82C55A CERDIP



NOTES:

1. V_{CC} = 5.5V ± 0.5V
2. VIH = 4.5V ± 10%
3. VIL = -0.2V to 0.4V
4. GND = 0V

MR82C55A CERAMIC LCC



NOTES:

1. C1 = 0.01μF minimum
2. All resistors are 47kΩ ± 5%
3. f0 = 100kHz ± 10%
4. f1 = f0 ÷ 2; f2 = f1 ÷ 2, . . . ; f15 = f14 ÷ 2

Timing Waveforms (Continued)

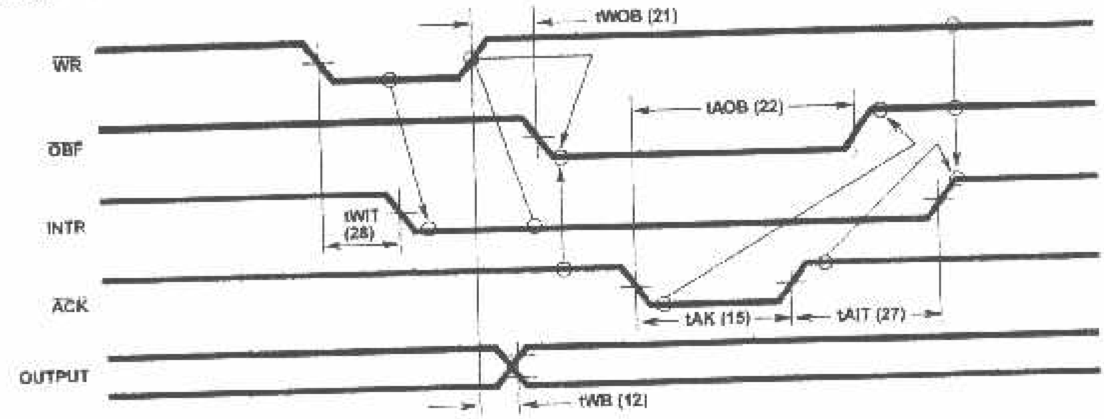


FIGURE 28. MODE 1 (STROBED OUTPUT)

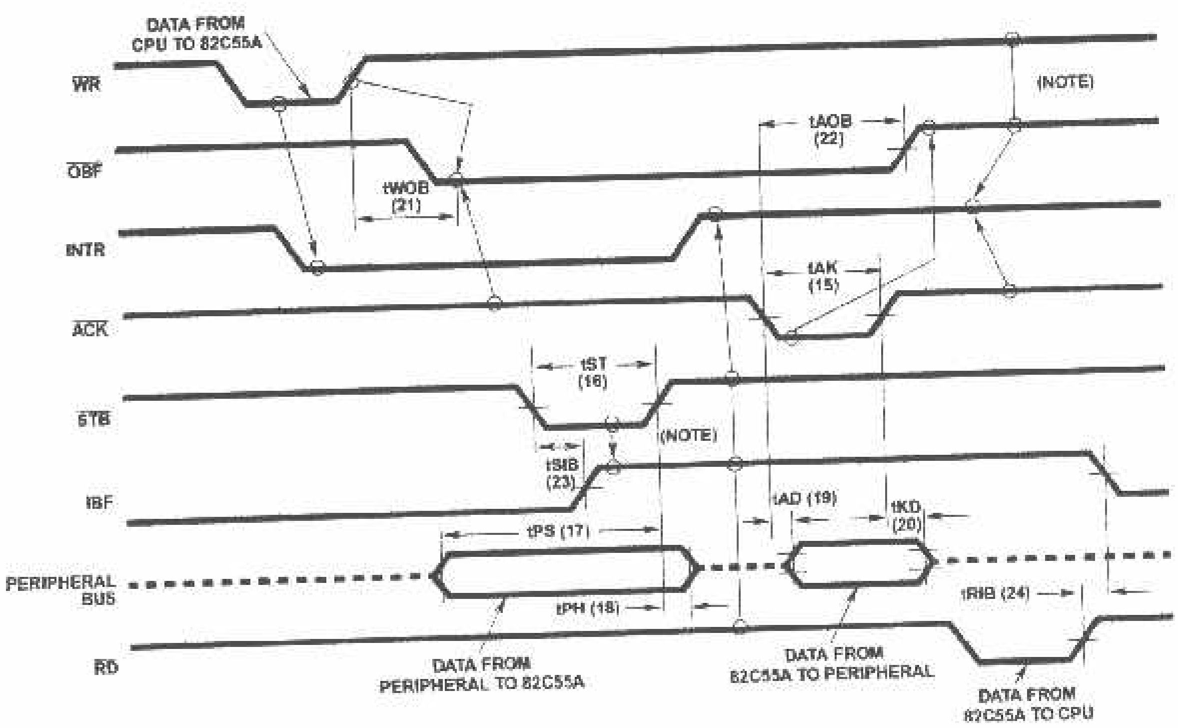


FIGURE 29. MODE 2 (BI-DIRECTIONAL)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible. (INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ • $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

Timing Waveforms

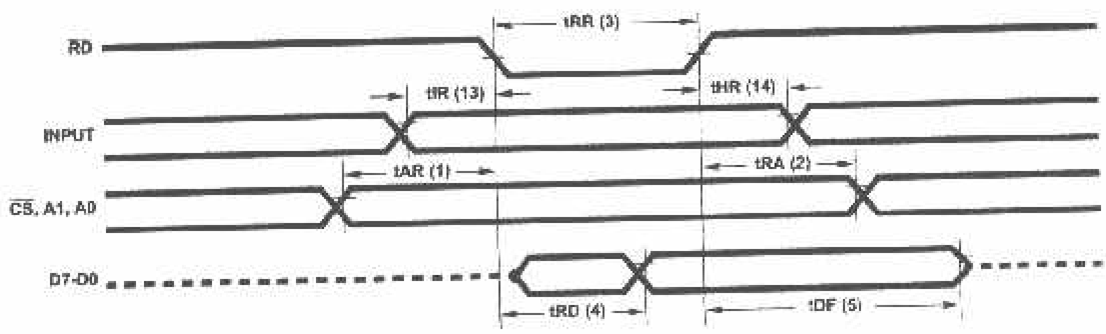


FIGURE 25. MODE 0 (BASIC INPUT)

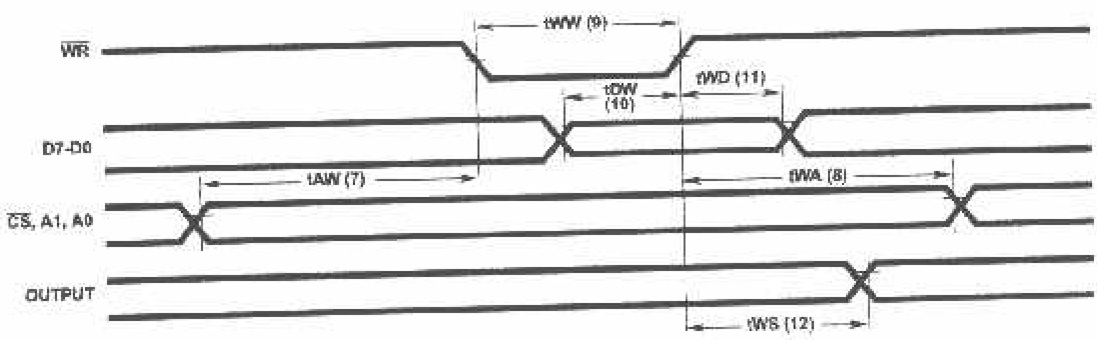


FIGURE 26. MODE 0 (BASIC OUTPUT)

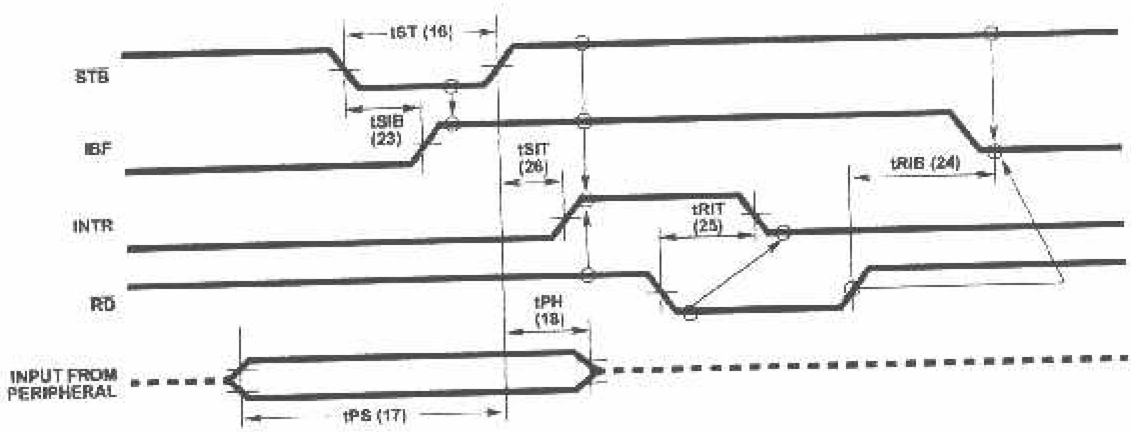


FIGURE 27. MODE1 (STROBED INPUT)

AC Electrical Specifications V_{CC} = +5V± 10%, GND = 0V; T_A = -55°C to +125°C (M82C55A) (M82C55A-5);
T_A = -40°C to +85°C (I82C55A) (I82C55A-5);
T_A = 0°C to +70°C (C82C55A) (C82C55A-5)

SYMBOL	PARAMETER	82C55A-5		82C55A		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
READ TIMING							
(1) tAR	Address Stable Before $\overline{RD}$	0	-	0	-	ns	
(2) tRA	Address Stable After $\overline{RD}$	0	-	0	-	ns	
(3) tRR	$\overline{RD}$ Pulse Width	250	-	150	-	ns	
(4) tRD	Data Valid From $\overline{RD}$	-	200	-	120	ns	1
(5) tDF	Data Float After $\overline{RD}$	10	75	10	75	ns	2
(6) tRV	Time Between $\overline{RD}$s and/or $\overline{WR}$s	300	-	300	-	ns	
WRITE TIMING							
(7) tAW	Address Stable Before $\overline{WR}$	0	-	0	-	ns	
(8) tWA	Address Stable After $\overline{WR}$	20	-	20	-	ns	
(9) tWW	$\overline{WR}$ Pulse Width	100	-	100	-	ns	
(10) tDW	Data Valid to $\overline{WR}$ High	100	-	100	-	ns	
(11) tWD	Data Valid After $\overline{WR}$ High	30	-	30	-	ns	
OTHER TIMING							
(12) tWB	$\overline{WR}$ = 1 to Output	-	350	-	350	ns	1
(13) tIR	Peripheral Data Before $\overline{RD}$	0	-	0	-	ns	
(14) tHR	Peripheral Data After $\overline{RD}$	0	-	0	-	ns	
(15) tAK	ACK Pulse Width	200	-	200	-	ns	
(16) tST	STB Pulse Width	100	-	100	-	ns	
(17) tPS	Peripheral Data Before STB High	20	-	20	-	ns	
(18) tPH	Peripheral Data After STB High	50	-	50	-	ns	
(19) tAD	ACK = 0 to Output	-	175	-	175	ns	1
(20) tKD	ACK = 1 to Output Float	20	250	20	250	ns	2
(21) tWOB	$\overline{WR}$ = 1 to OBF = 0	-	150	-	150	ns	1
(22) tAOB	ACK = 0 to OBF = 1	-	150	-	150	ns	1
(23) tSIB	STB = 0 to IBF = 1	-	150	-	150	ns	1
(24) tRIB	$\overline{RD}$ = 1 to IBF = 0	-	150	-	150	ns	1
(25) tRIT	$\overline{RD}$ = 0 to INTR = 0	-	200	-	200	ns	1
(26) tSIT	STB = 1 to INTR = 1	-	150	-	150	ns	1
(27) tAIT	ACK = 1 to INTR = 1	-	150	-	150	ns	1
(28) tWIT	$\overline{WR}$ = 0 to INTR = 0	-	200	-	200	ns	1
(29) tRES	Reset Pulse Width	500	-	500	-	ns	1, (Note 1)

NOTE:
1. Period of initial Reset pulse after power-on must be at least 50μsec. Subsequent Reset pulses may be 500ns minimum.

Absolute Maximum Ratings

Supply Voltage . . . . +8.0V
Input, Output or I/O Voltage . . . . . GND-0.5V to V_{CC}+0.5V
ESD Classification . . . . Class 1

Operating Conditions

Operating Voltage Range . . . . +4.5V to +5.5V
Operating Temperature Range
- C82C55A . . . . 0°C to +70°C
- I82C55A . . . . -40°C to +85°C
- M82C55A . . . . -55°C to +125°C

Thermal Information

Thermal Resistance (Typical) Note 1	θ_{JA}	θ_{JC}
CERDIP Package	55°C/W	20°C/W
Ceramic LCC Package	70°C/W	16°C/W
Plastic DIP Package	50°C/W	-
PLCC	46°C/W	-

Storage Temperature Range . . . . -65°C to +150°C
Maximum Junction Temperature
- Ceramic Package . . . . +175°C
- Plastic Package . . . . +150°C

Maximum Lead Temperature (Soldering 10s) . . . . . +300°C
(Lead Tips Only for Surface Mount Packages)

Die Characteristics

Gate Count . . . . 1000 Gates

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.

DC Electrical Specifications

V_{CC} = 5.0V ±10%; T_A = 0°C to +70°C (C82C55A);
T_A = -40°C to +85°C (I82C55A);
T_A = -55°C to +125°C (M82C55A)

SYMBOL	PARAMETER	LIMITS MIN	LIMITS MAX	UNITS	TEST CONDITIONS
V_{IH}	Logical One Input Voltage	2.0 2.2	-	V	I82C55A, C82C55A, M82C55A
V_{IL}	Logical Zero Input Voltage	-	0.8	V	
V_{OH}	Logical One Output Voltage	3.0 V_{CC} -0.4	-	V	I_{OH} = -2.5mA, I_{OH} = -100µA
V_{OL}	Logical Zero Output Voltage	-	0.4	V	I_{OL} +2.5mA
I_I	Input Leakage Current	-1.0	+1.0	µA	V_{IN} = V_{CC} or GND, DIP Pins: 5, 6, 8, 9, 35, 36
IO	I/O Pin Leakage Current	-10	+10	µA	VO = V_{CC} or GND DIP Pins: 27 - 34
IBHH	Bus Hold High Current	-50	-400	µA	VO = 3.0V. Ports A, B, C
IBHL	Bus Hold Low Current	50	400	µA	VO = 1.0V. Port A ONLY
IDAR	Darlington Drive Current	-2.5	Note 1, 3	mA	Ports A, B, C. Test Condition 3
ICCSB	Standby Power Supply Current	-	10	µA	V_{CC} = 5.5V, V_{IN} = V_{CC} or GND. Output Open
ICCOP	Operating Power Supply Current	-	1	mA/MHz	T_A = +25°C, V_{CC} = 5.0V, Typical (See Note2)

NOTES:
1. No internal current limiting exists on Port Outputs. A resistor must be added externally to limit the current.
2. ICCOP = 1mA/MHz of Peripheral Read/Write cycle time. (Example: 1.0µs I/O Read/Write cycle time = 1mA).
3. Tested as V_{OH} at -2.5mA.

Capacitance T_A = 25°C

SYMBOL	PARAMETER	TYPICAL	UNITS	TEST CONDITIONS
CIN	Input Capacitance	10	pF	FREQ = 1MHz, All Measurements are referenced to device GND
CI/O	I/O Capacitance	20	pF	

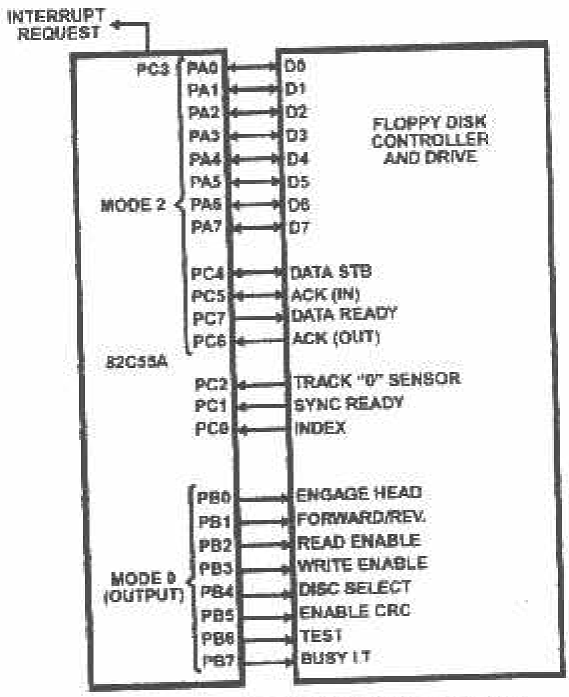


FIGURE 23. BASIC FLOPPY DISC INTERFACE

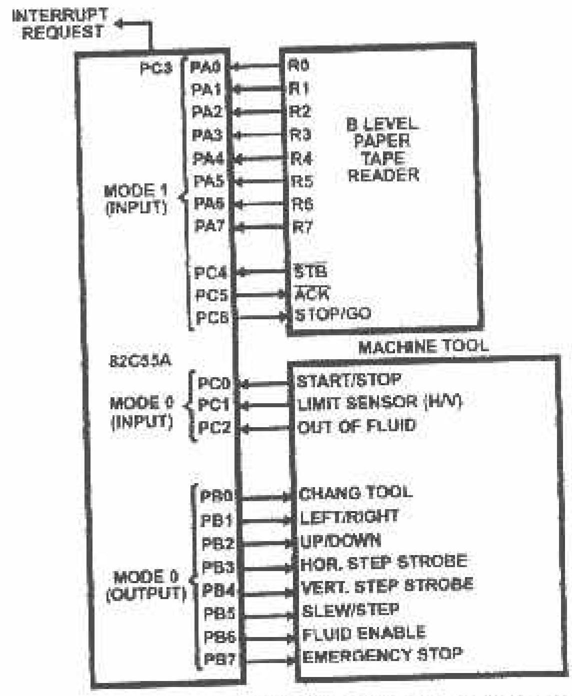


FIGURE 24. MACHINE TOOL CONTROLLER INTERFACE

Die Characteristics

DIE DIMENSIONS:
95 x 100 x 19 ±1mils

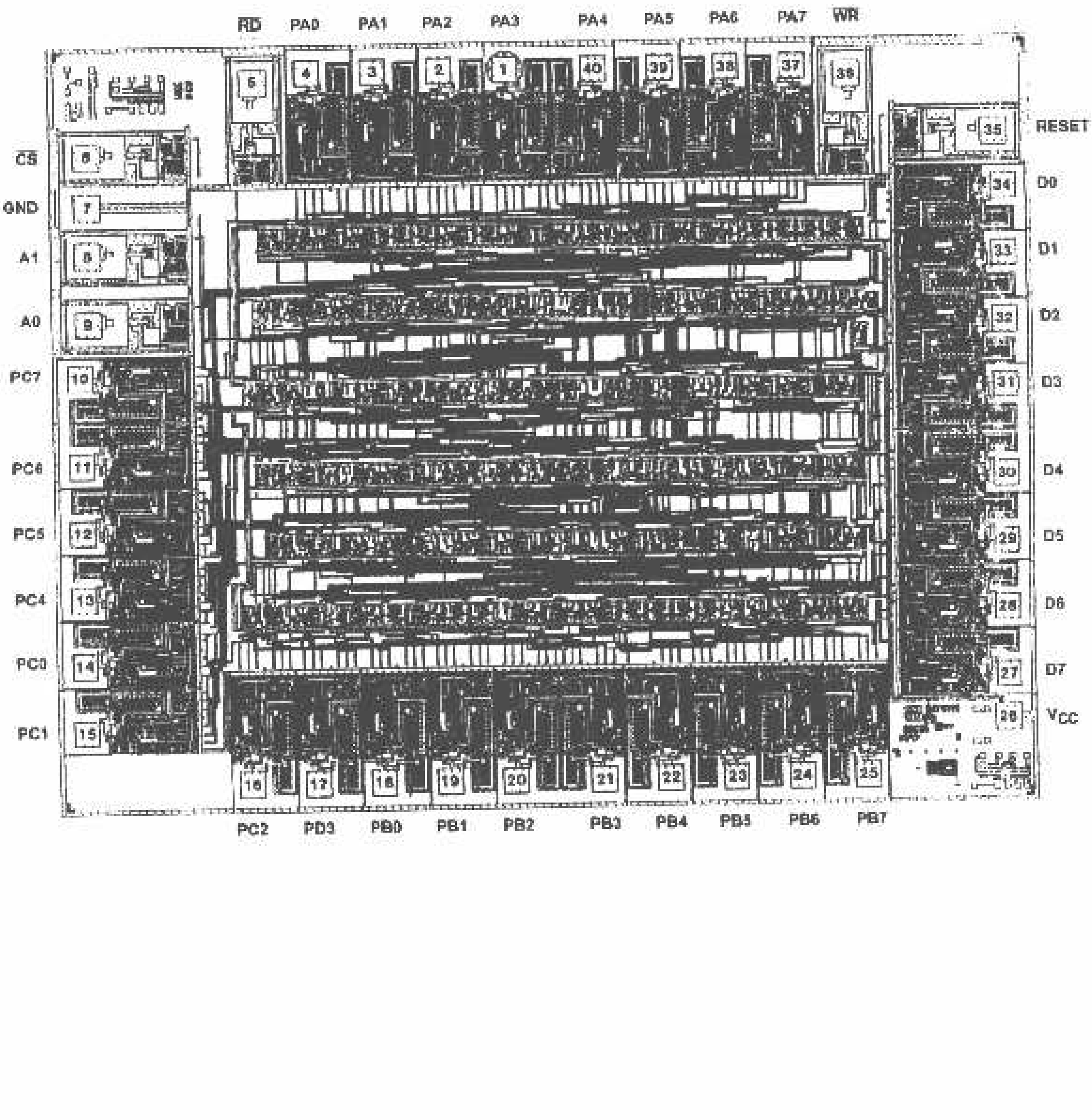
METALLIZATION:
Type: Silicon - Aluminum
Thickness: 11kÅ ±1kÅ

GLASSIVATION:
Type: SiO_2
Thickness: 8kÅ ±1kÅ

WORST CASE CURRENT DENSITY:
0.78×10^5 A/cm^2

Metallization Mask Layout

82C55A



19-4323; Rev 12; 10/03

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220–MAX249

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5μW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multidrop RS-232 Networks

Next-Generation Device Features

- ♦ **For Low-Voltage, Integrated ESD Applications: MAX3222E/MAX3232E/MAX3237E/MAX3241E/ MAX3246E: +3.0V to +5.5V, Low-Power, Up to 1Mbps, True RS-232 Transceivers Using Four 0.1μF External Capacitors (MAX3246E Available in a UCSP™ Package)**
- ♦ **For Low-Cost Applications: MAX221E: ±15kV ESD-Protected, +5V, 1μA, Single RS-232 Transceiver with AutoShutdown™**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No		120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) .... -0.3V to +6V
Input Voltages
  T_{IN} .... -0.3V to (V_{CC} - 0.3V)
  R_{IN} (Except MAX220) .... ±30V
  R_{IN} (MAX220) .... ±25V
  T_{OUT} (Except MAX220) (Note 1) .... ±15V
  T_{OUT} (MAX220) .... ±13.2V
Output Voltages
  T_{OUT} .... ±15V
  R_{OUT} .... -0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND .... Continuous
Continuous Power Dissipation (T_A = +70°C)
  16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
  18-Pin Plastic DIP (derate 11.11mW/°C above +70°C) 889mW
  20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) 440mW
  16-Pin Narrow SO (derate 8.70mW/°C above +70°C) 696mW
  16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
  18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
  20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
  20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
  16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
  18-Pin CERDIP (derate 10.53mW/°C above +70°C) 842mW
Operating Temperature Ranges
  MAX2_ _AC_ _, MAX2_ _C_ _ .... 0°C to +70°C
  MAX2_ _AE_ _, MAX2_ _E_ _ .... -40°C to +85°C
  MAX2_ _AM_ _, MAX2_ _M_ _ .... -55°C to +125°C
Storage Temperature Range .... -65°C to +160°C
Lead Temperature (soldering, 10s) .... +300°C

Note 1: Input voltage measured with T_{OUT} in high impedance state, $\overline{SHDN}$ or V_{CC} = 0V
Note 2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS					
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND	±5	±8		V
Input Logic Threshold Low			1.4	0.8	V
Input Logic Threshold High	All devices except MAX220	2	1.4		V
	MAX220: V_{CC} = 5.0V	2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation		5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown, MAX220		±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242		±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V		±0.01	±10	
Data Rate			200	116	kbps
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V	300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V	±7	±22		mA
RS-232 RECEIVERS					
RS-232 Input Voltage Operating Range				±30	V
RS-232 Input Threshold Low	V_{CC} = 5V, All except MAX243 $R2_{IN}$	0.8	1.3		V
	V_{CC} = 5V, MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V, All except MAX243 $R2_{IN}$		1.8	2.4	V
	V_{CC} = 5V, MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.	0.2	0.5	1	V
	MAX243		1		
RS-232 Input Resistance		3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA		0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA	3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND	-2	-10		mA
	Shrinking V_{OUT} = V_{CC}	10	30		

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

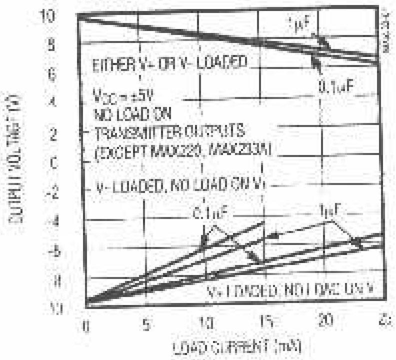
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA
$\overline{EN}$ Input Threshold Low	MAX242			1.4	0.8	V
$\overline{EN}$ Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V T_A = +25°C measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	µs
	t_{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ Goes High), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ Goes Low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 3: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ is ≥ 0V or is floating.

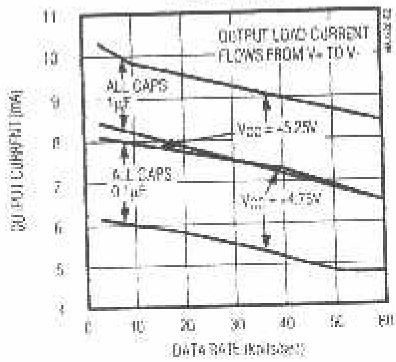
Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243

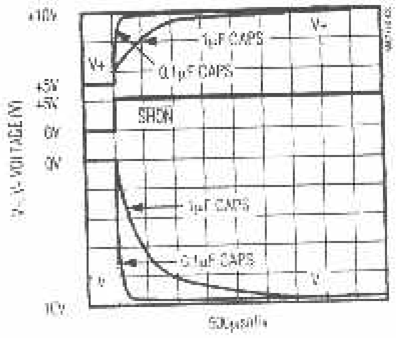
OUTPUT VOLTAGE vs. LOAD CURRENT



AVAILABLE OUTPUT CURRENT vs. DATA RATE



MAX222/MAX242 ON-TIME EXITING SHUTDOWN



ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

V_{CC} .... -0.3V to +6V
V+ .... (V_{CC} - 0.3V) to +14V
V- .... +0.3V to -14V
Input Voltages
  T_{IN} .... -0.3V to (V_{CC} + 0.3V)
  R_{IN} .... ±30V
Output Voltages
  T_{OUT} .... (V+ + 0.3V) to (V- - 0.3V)
  R_{OUT} .... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_{OUT} .... Continuous
Continuous Power Dissipation (T_A = +70°C)
  14-Pin Plastic DIP (derate 10.00mW/°C above +70°C) ... 800mW
  16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) ... 842mW
  20-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ... 889mW
  24-Pin Narrow Plastic DIP
    (derate 13.33mW/°C above +70°C) 1.07W
  24-Pin Plastic DIP (derate 9.09mW/°C above +70°C) 500mW
  16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
  20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
  24-Pin Wide SO (derate 11.76mW/°C above +70°C) 941mW
  28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
  44-Pin Plastic FP (derate 11.11mW/°C above +70°C) 889mW
  14-Pin CERDIP (derate 9.09mW/°C above +70°C) 727mW
  16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
  20-Pin CERDIP (derate 11.11mW/°C above +70°C) 889mW
  24-Pin Narrow CERDIP
    (derate 12.50mW/°C above +70°C) 1W
  24-Pin Sidebraze (derate 20.0mW/°C above +70°C) 1.6W
  28-Pin SSOP (derate 9.52mW/°C above +70°C) 762mW
Operating Temperature Ranges
  MAX2 _ _ C _ _ .... 0°C to +70°C
  MAX2 _ _ E _ _ .... -40°C to +85°C
  MAX2 _ _ M _ _ .... -55°C to +125°C
Storage Temperature Range .... -65°C to +160°C
Lead Temperature (soldering, 10s) .... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5.0	±7.3		V
V_{CC} Power-Supply Current	No load, T_A = +25°C	MAX232/233		5	10	mA
		MAX223/230/234–238/240/241		7	15	
		MAX231/239		0.4	[illegible]	
V+ Power-Supply Current		MAX231		1.8	5	mA
		MAX239		5	15	
Shutdown Supply Current	T_A = +25°C	MAX223		15	50	µA
		MAX230/235/236/240/241		1	10	
Input Logic Threshold Low	T_{IN}; EN, $\overline{SHDN}$ (MAX233), $\overline{EN}$, SHDN (MAX230/235–241)				0.8	V
Input Logic Threshold High	T_{IN}		2.0			V
	EN, $\overline{SHDN}$ (MAX223); $\overline{EN}$, SHDN (MAX230/235/236/240/241)		2.4			
Logic Pull-Up Current	T_{IN} = 0V			1.5	200	µA
Receiver Input Voltage Operating Range			-30		30	V

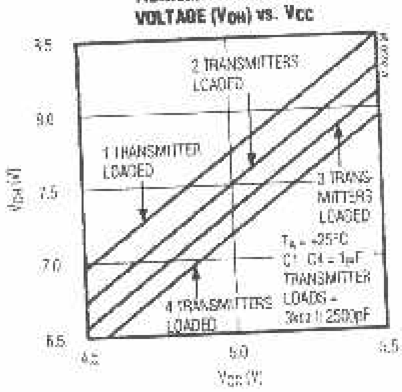
ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

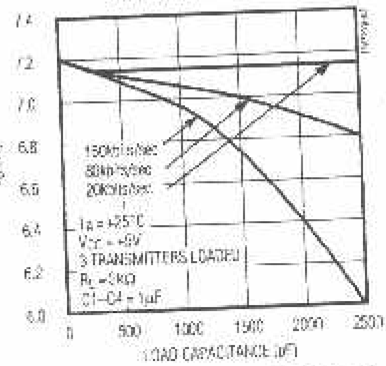
(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%, V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

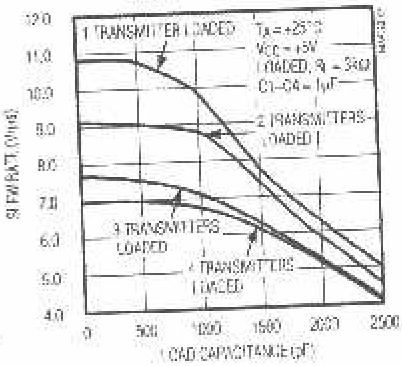
PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	T_A = +25°C, V_{CC} = 5V; Normal operation $\overline{SHDN}$ = 5V (MAX223), SHDN = 0V (MAX235/236/240/241)	0.8	1.2		V
	T_A = +25°C, V_{CC} = 5V; Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)	0.6	1.5		V
RS-232 Input Threshold High	T_A = +25°C, V_{CC} = 5V; Normal operation $\overline{SHDN}$ = 5V (MAX223), SHDN = 0V (MAX235/236/240/241)		1.7	2.4	V
	T_A = +25°C, V_{CC} = 5V; Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)		1.5	2.4	V
RS-232 Input Hysteresis	V_{CC} = 5V, no hysteresis in shutdown	0.2	0.5	1.0	V
RS-232 Input Resistance	T_A = +25°C, V_{CC} = 5V	3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 1.6mA (MAX231/232/233, I_{OUT} = 3.2mA)			0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1mA	3.5	V_{CC} - 0.4		V
TTL/CMOS Output Leakage Current	0V ≤ R_{OUT} ≤ V_{CC}; EN = 0V (MAX223); $\overline{EN}$ = V_{CC} (MAX235–241)		0.05	±10	µA
Receiver Output Enable Time	Normal operation; MAX223		600		ns
	Normal operation; MAX235/236/239/240/241		400		ns
Receiver Output Disable Time	Normal operation; MAX223		900		ns
	Normal operation; MAX235/236/239/240/241		250		ns
Propagation Delay	RS-232 IN to TTL/CMOS OUT, C_L = 150pF; Normal operation		0.5	10	µs
	RS-232 IN to TTL/CMOS OUT, C_L = 150pF; $\overline{SHDN}$ = 0V (MAX223), t_{PHLS}		4	40	µs
	RS-232 IN to TTL/CMOS OUT, C_L = 150pF; $\overline{SHDN}$ = 0V (MAX223), t_{PLHS}		6	40	µs
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V	3	5.1	30	V/µs
	MAX231/MAX232/MAX233, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V		4	30	V/µs
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V	300			Ω
Transmitter Output Short-Circuit Current			±10		mA

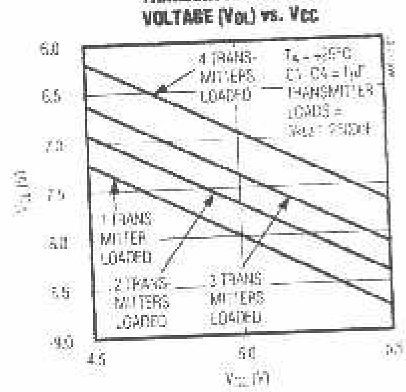
Typical Operating Characteristics

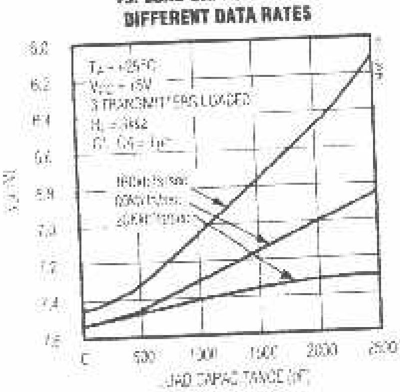
MAX223/MAX230–MAX241

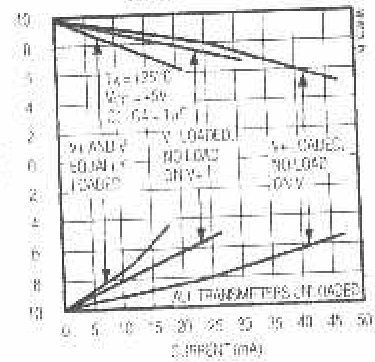


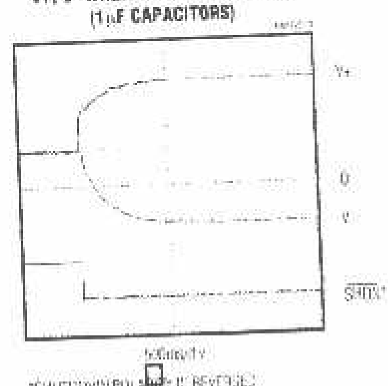












*SHUTDOWN POLARITY IS REVERSED FOR NON MAX241 PARTS

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX225/MAX244–MAX249

Supply Voltage (V_{CC}) .... -0.3V to +6V
Input Voltages
T_{IN}, $\overline{ENA}$, $\overline{ENB}$, $\overline{ENR}$, $\overline{ENT}$, $\overline{ENRA}$,
$\overline{ENRB}$, $\overline{ENTA}$, $\overline{ENTB}$........................ 0.3V to (V_{CC} + 0.3V)
R_{IN} .... ±25V
T_{OUT} (Note 3).. ±15V
R_{OUT} .... -0.3V to (V_{CC} + 0.3V)
Short Circuit (one output at a time)
T_{OUT} to GND .... Continuous
R_{OUT} to GND .... Continuous

Continuous Power Dissipation (T_A = +70°C)
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
40-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ... 611mW
44-Pin PLCC (derate 13.33mW/°C above +70°C) 1.07W
Operating Temperature Ranges
MAX225C_ _, MAX24_C_ _ .... 0°C to +70°C
MAX225E_ _, MAX24_E_ _ .... -40°C to +85°C
Storage Temperature Range .... -65°C to +160°C
Lead Temperature (soldering, 10s) .... +300°C

Note 4: Input voltage measured with transmitter output in a high-impedance state, shutdown, or V_{CC} = 0V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249

(MAX225, V_{CC} = 5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF, T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Tables 1a–1d	Normal operation		10	50	µA
		Shutdown		±0.01	±1	
Data Rate	Tables 1a–1d, normal operation			120	64	kbps
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±7.5		V
Output Leakage Current (Shutdown)	Tables 1a–1d	$\overline{ENA}$, $\overline{ENB}$, $\overline{ENT}$, $\overline{ENTA}$, $\overline{ENTB}$ = V_{CC}, V_{OUT} = ±15V		±0.01	±25	µA
		V_{CC} = 0V, V_{OUT} = ±15V		±0.01	±25	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V (Note 4)		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V		±7	±30		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±25	V
RS-232 Input Threshold Low	V_{CC} = 5V		0.8	1.3		V
RS-232 Input Threshold High	V_{CC} = 5V			1.8	2.4	V
RS-232 Input Hysteresis	V_{CC} = 5V		0.2	0.5	1.0	V
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	Normal operation, outputs disabled, Tables 1a–1d, 0V ≤ V_{OUT} ≤ V_{CC}, $\overline{ENR\_}$ = V_{CC}			±0.05	±0.10	µA

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249 (continued)

($MAX225$, V_{CC} = 5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF; $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
POWER SUPPLY AND CONTROL LOGIC						
Operating Supply Voltage		MAX225	4.75		5.25	V
		MAX244–MAX249	4.5		5.5	
V_{CC} Supply Current (Normal Operation)	No load	MAX225		10	20	mA
		MAX244–MAX249		11	30	
	3kΩ loads on all outputs	MAX225		40		
		MAX244–MAX249		57		
Shutdown Supply Current	T_A = +25°C			8	25	µA
	$T_A = T_{MIN}$ to T_{MAX}				50	
Control Input	Leakage current				±1	µA
	Threshold low			1.4	0.8	V
	Threshold high		2.4	1.4		
AC CHARACTERISTICS						
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V		5	10	30	V/µs
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t_{PHLT}			1.3	3.5	µs
	t_{PLHT}			1.5	3.5	
Receiver Propagation Delay TLL to RS-232 (Normal Operation), Figure 2	t_{PHLR}			0.6	1.5	µs
	t_{PLHR}			0.6	1.5	
Receiver Propagation Delay TLL to RS-232 (Low-Power Mode), Figure 2	t_{PHLS}			0.6	10	µs
	t_{PLHS}			3.0	10	
Transmitter + to - Propagation Delay Difference (Normal Operation)	$t_{PHLT} - t_{PLHT}$			350		ns
Receiver + to - Propagation Delay Difference (Normal Operation)	$t_{PHLR} - t_{PLHR}$			350		ns
Receiver-Output Enable Time, Figure 3	t_{ER}			100	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}			100	500	ns
Transmitter Enable Time	t_{ET}	MAX246–MAX249 (excludes charge-pump startup)		5		µs
		MAX225/MAX245–MAX249 (includes charge-pump startup)		10		ms
Transmitter Disable Time, Figure 4	t_{DT}			100		ns

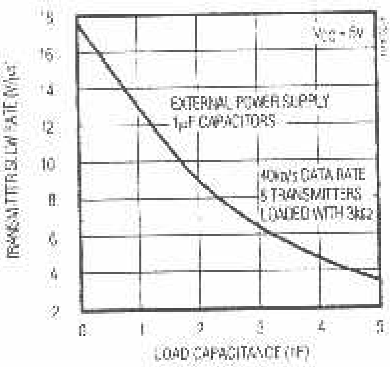
Note 6: The 300Ω minimum specification complies with EIA/TIA-232E, but the actual resistance when in shutdown mode or V_{CC} = 0V is 10MΩ as is implied by the leakage specification.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

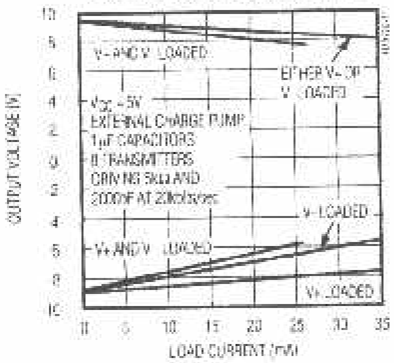
Typical Operating Characteristics

MAX225/MAX244–MAX249

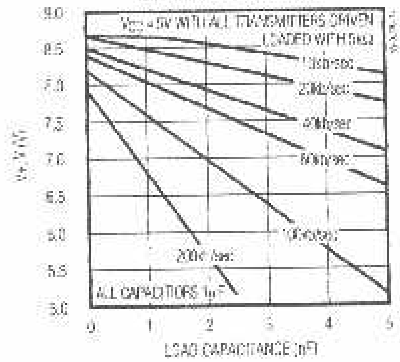
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



OUTPUT VOLTAGE vs. LOAD CURRENT FOR V+ AND V-



TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



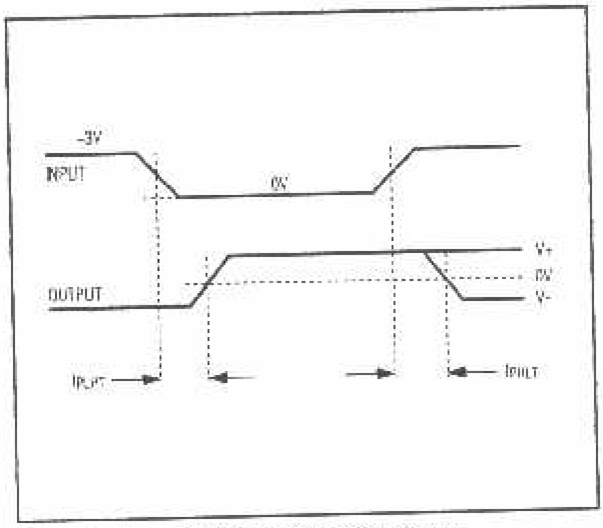


Figure 1. Transmitter Propagation-Delay Timing

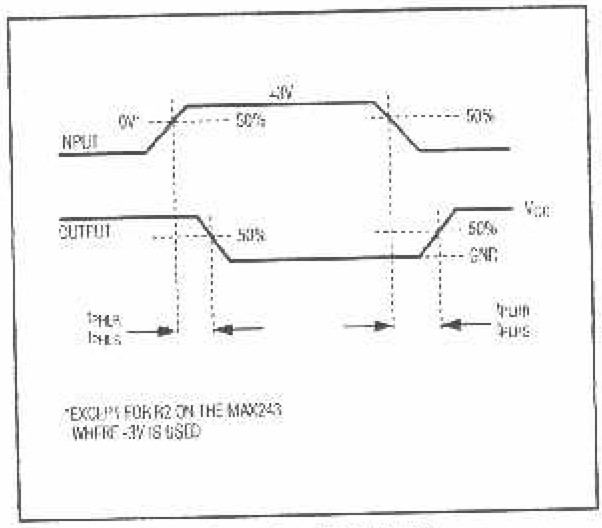


Figure 2. Receiver Propagation-Delay Timing

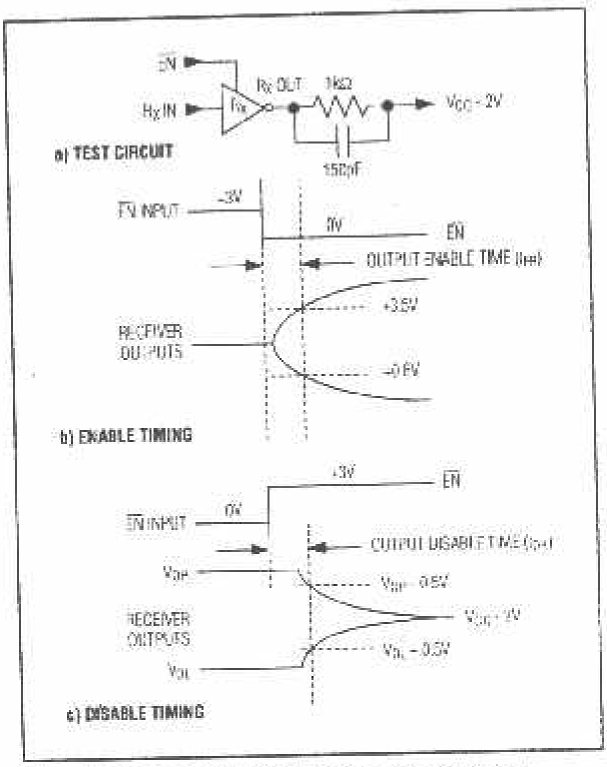


Figure 3. Receiver-Output Enable and Disable Timing

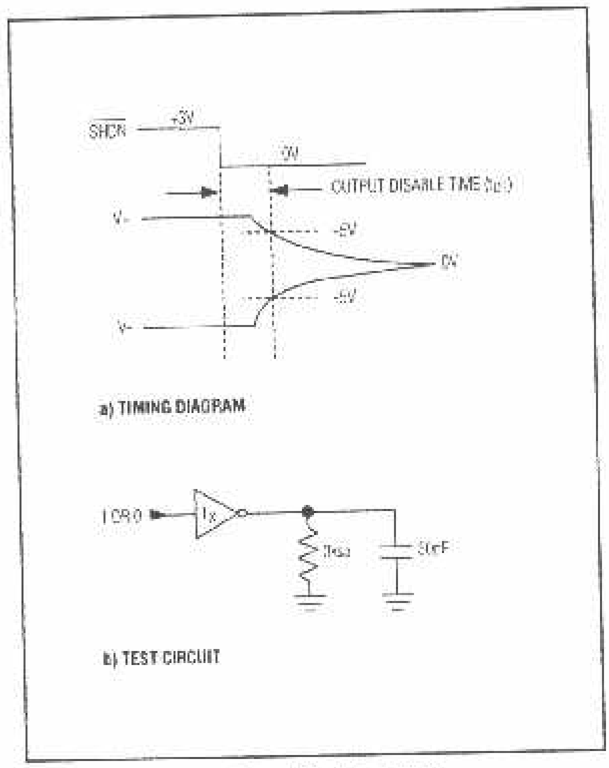


Figure 4. Transmitter-Output Disable Timing

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Table 1a. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS	RECEIVERS
0	0	Normal Operation	All Active	All Active
0	1	Normal Operation	All Active	All 3-State
1	0	Shutdown	All 3-State	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State

Table 1b. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All Active	RA1–RA4 3-State, RA5 Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All 3-State	All Low-Power Receive Mode	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RB5 Low-Power Receive Mode

Table 1c. MAX246 Control Pin Configurations

$\overline{ENA}$	$\overline{ENB}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All 3-State	All Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All Active	RA1–RA4 3-State, RA5 Active	All Active
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RA5 Low-Power Receive Mode

Table 1d. MAX247/MAX248/MAX249 Control Pin Configurations

$\overline{ENTA}$	$\overline{ENTB}$	$\overline{ENRA}$	$\overline{ENRB}$	OPERATION STATUS		TRANSMITTERS		RECEIVERS	
					MAX247	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB5
					MAX248	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB4
					MAX249	TA1–TA3	TB1–TB3	RA1–RA5	RB1–RB5
0	0	0	0	Normal Operation		All Active	All Active	All Active	All Active
0	0	0	1	Normal Operation		All Active	All Active	All Active	All 3-State, except RB5 stays active on MAX247
0	0	1	0	Normal Operation		All Active	All Active	All 3-State	All Active
0	0	1	1	Normal Operation		All Active	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
0	1	0	0	Normal Operation		All Active	All 3-State	All Active	All Active
0	1	0	1	Normal Operation		All Active	All 3-State	All Active	All 3-State, except RB5 stays active on MAX247
0	1	1	0	Normal Operation		All Active	All 3-State	All 3-State	All Active
0	1	1	1	Normal Operation		All Active	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247
1	0	0	0	Normal Operation		All 3-State	All Active	All Active	All Active
1	0	0	1	Normal Operation		All 3-State	All Active	All Active	All 3-State, except RB5 stays active on MAX247
1	0	1	0	Normal Operation		All 3-State	All Active	All 3-State	All Active
1	0	1	1	Normal Operation		All 3-State	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
1	1	0	0	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	Low-Power Receive Mode
1	1	0	1	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	All 3-State, except RB5 stays active on MAX247
1	1	1	0	Shutdown		All 3-State	All 3-State	All 3-State	Low-Power Receive Mode
1	1	1	1	Shutdown		All 3-State	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Detailed Description

The MAX220–MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

Dual Charge-Pump Voltage Converter

The MAX220–MAX249 have two internal charge-pumps that convert +5V to ±10V (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see the *Typical Operating Characteristics* section), except on the MAX225 and MAX245–MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum ±5V EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245–MAX249, avoid using V+ and V- to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the $\overline{SHDN}$ pin must be tied to V_{CC}. This is because V+ is internally connected to V_{CC} in shutdown mode.

RS-232 Drivers

The typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver and V_{CC} = +5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28 specification, which calls for ±5V minimum driver output levels under worst-case conditions. These include a minimum 3kΩ load, V_{CC} = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400kΩ input pull-up resistors to V_{CC} are built in (except for the MAX220). The pull-up resistors force the outputs of unused drivers low because all drivers invert. The internal input pull-up resistors typically source 12µA, except in shutdown mode where the pull-ups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25µA)—when in shutdown mode, in three-state mode, or when device power is removed. Outputs can be driven to ±15V. The power-supply current typically drops to 8µA in shutdown mode. The MAX220 does not have pull-up resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or V_{CC}.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239–MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1µA with the driver output pulled to ground. The driver output leakage remains less than 1µA, even if the transmitter output is backdriven between 0V and (V_{CC} + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also zener clamped to approximately V_{CC} + 6V, with a series impedance of 1kΩ.

The driver output slew rate is limited to less than 30V/µs as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/µs unloaded and 10V/µs loaded with 3Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels.

The receiver inputs withstand an input overvoltage up to ±25V and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	$\overline{SHDN}$	EN	EN(R)	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

nominal 5kΩ values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.

Low-Power Receive Mode

The low-power receive-mode feature of the MAX223, MAX242, and MAX245–MAX249 puts the IC into shutdown mode but still allows it to receive information. This is important for applications where systems are periodically awakened to look for activity. Using low-power receive mode, the system can still receive a signal that will activate it on command and prepare it for communication at faster data rates. This operation conserves system power.

Negative Threshold—MAX243

The MAX243 is pin compatible with the MAX232A, differing only in that RS-232 cable fault protection is removed on one of the two receiver inputs. This means that control lines such as CTS and RTS can either be driven or left floating without interrupting communication. Different cables are not needed to interface with different pieces of equipment.

The input threshold of the receiver without cable fault protection is -0.8V rather than +1.4V. Its output goes positive only if the input is connected to a control line that is actively driven negative. If not driven, it defaults to the 0 or "OK to send" state. Normally, the MAX243's other receiver (+1.4V threshold) is used for the data line (TD or RD), while the negative threshold receiver is connected to the control line (DTR, DTS, CTS, RTS, etc.).

Other members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA-232E specifications. This means a receiver output goes high whenever its input is driven negative, left floating, or shorted to ground. The high output tells the serial communications IC to stop sending data. To avoid this, the control lines must either be driven or connected with jumpers to an appropriate positive voltage level.

Shutdown—MAX222–MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown. On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5μs for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ($\overline{EN}$ for the MAX242 and EN for the MAX223) that allows receiver output control independent of $\overline{SHDN}$ (SHDN for MAX241). With all other devices, $\overline{SHDN}$ (SHDN for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25μA and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the $\overline{ENR}$ input. On the MAX245, eight of the receiver outputs are controlled by the $\overline{ENR}$ input, while the remaining two receivers (RA5 and RB5) are always active. RA1–RA4 and RB1–RB4 are put in a three-state mode when $\overline{ENR}$ is a logic high.

Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245–MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Tables 1a–1d define the control states. The MAX244 has no control pins and is not included in these tables.

The MAX246 has ten receivers and eight drivers with two control pins, each controlling one side of the device. A logic high at the A-side control input ($\overline{ENA}$) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input ($\overline{ENB}$) causes the four B-side drivers and receivers to go into a three-state mode. As in the MAX245, one A-side and one B-side receiver (RA5 and RB5) remain active at all times. The entire device is put into shutdown mode when both the A and B sides are disabled ($\overline{ENA}$ = $\overline{ENB}$ = +5V).

The MAX247 provides nine receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs each control four drivers. The ninth receiver (RB5) is always active. The device enters shutdown mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX248 provides eight receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control four drivers each. This part does not have an always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX249 provides ten receivers and six drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control five receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$. In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kbits/sec.

Applications Information

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, V_{CC} should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.

FAIRCHILD SEMICONDUCTOR™

March 1998

DM74LS573
Octal D Latch with 3-STATE Outputs

General Description

The 'LS573 is a high speed octal latch with buffered common Latch Enable (LE) and buffered common Output Enable (OE) inputs.

This device is functionally identical to the 'LS373, but has different pinouts. For truth tables, discussion of operations and AC and DC specifications, please refer to the 'LS373 data sheet.

Features

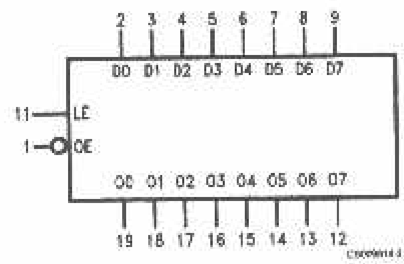
- Inputs and outputs on opposite sides of package allowing easy interface with microprocessors
- Useful as input or output port for microprocessors
- Functionally identical to 'LS373
- Input clamp diodes limit high speed termination effects
- Fully TTL and CMOS compatible

Connection Diagram

Dual-In-Line Package

Order Number DM74LS573WM or DM74LS573N
See Package Number M20B or N20A

Logic Symbol



V_{CC} = Pin 20
GND = Pin 10

Pin Names	Description
D0–D7	Data Inputs
LE	Latch Enable Input (Active HIGH)
OE	3-STATE Output Enable Input (Active LOW)
O0–O7	3-STATE Latch Outputs

Function Table

OUTPUT Enable	Latch Enable	D	Output O
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

L = Low State, H = High State, X = Don't Care
Z = High Impedance State
Q_0 = Previous Condition of O

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	DM74LS			Units
		Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			V
V_{IL}	Low Level Input Voltage			0.8	V
I_{OH}	High Level Input Current			−2.6	mA
I_{OL}	Low Level Output Current			24	mA
T_A	Free Air Operating Temperature	0		70	°C

Note 1: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Electrical Characteristics

Over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	2.7	3.4		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 3)	−30		−130	mA
I_{CC}	Supply Current	V_{CC} = Max			50	mA
I_{OZH}	3-STATE Output off Current High	V_{CC} = V_{CCH}, V_{OZH} = 2.7V			20	μA
I_{OZL}	3-STATE Output off Current Low	V_{CC} = V_{CCH}, V_{OZL} = 0.4V			−20	μA

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

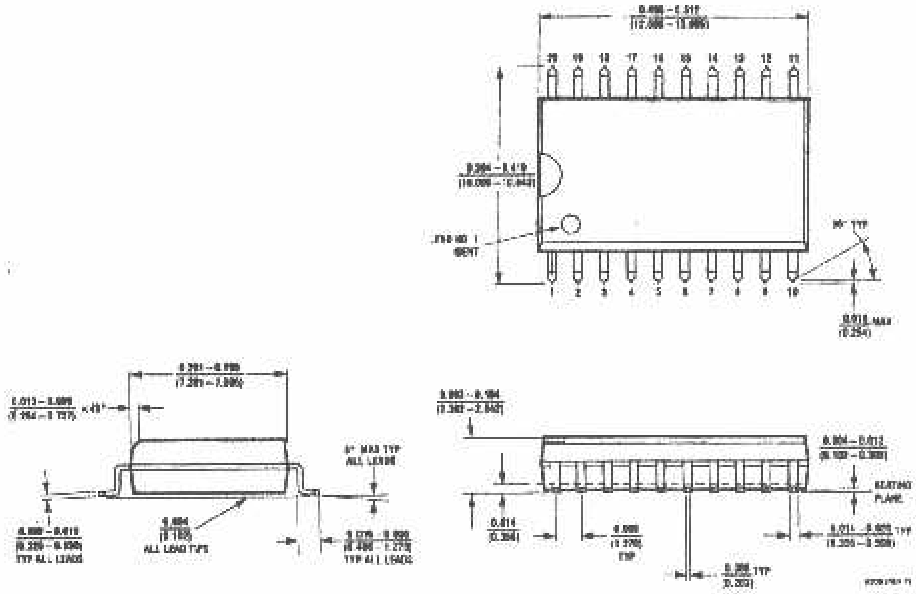
Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics

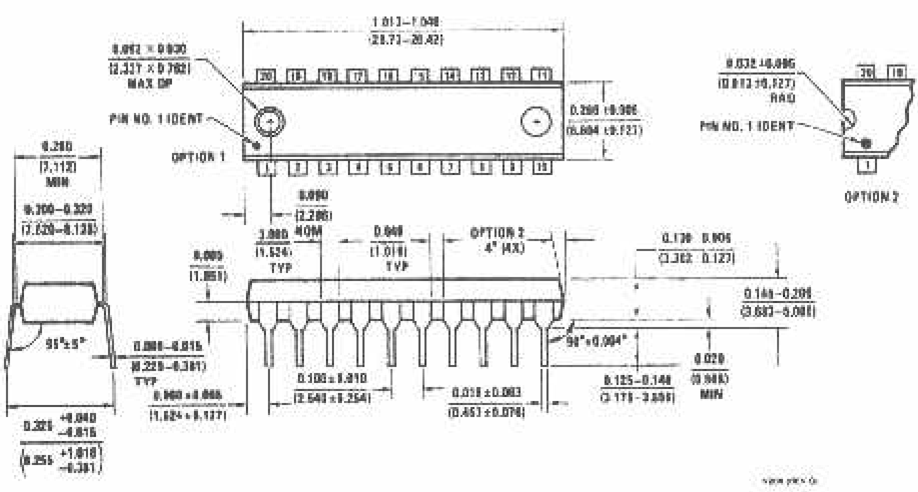
at V_{CC} = 5V and T_A = 25°C (see Section 1 for Test Waveforms and output loading)

Symbol	Parameter	R_L = 2 kΩ, C_L = 50 pF		Units
		Min	Max	
t_{PLH} t_{PHL}	Propagation Delay Data to Q		27 18	ns
t_{PLH} t_{PHL}	Propagation Delay LE to Q		36 25	ns
t_{PZH} t_{PZL}	3-STATE Enable Time $\overline{OE}$ to Q		20 25	ns
t_{PHZ} t_{PLZ}	3-STATE Enable Time $\overline{OE}$ to Q		20 25	ns
t_s(H) t_s(L)	Setup Time (High/Low) Data to LE	3 7		ns
t_h(H) t_h(L)	Hold Time (High/Low) Data to LE	10 10		ns
t_w(H)	Pulse Width (High) Data to LE	15		ns

Physical Dimensions inches (millimeters) unless otherwise noted



20-Lead Wide Small Outline Molded Package (M)
Order Number DM74LS573WM
Package Number M20B



20-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS573N
Package Number N20A

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD 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 (c) 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 in 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.

Fairchild Semiconductor Corporation
Americas
Customer Response Center
Tel: 1-888-522-5372

Fairchild Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 8 141-35-0
English Tel: +44 (0) 1 793-85-68-56
Italy Tel: +39 (0) 2 57 5631

Fairchild Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: +852 2737-7200
Fax: +852 2314-0061

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179

www.fairchildsemi.com

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS138 • DM74LS139 Decoder/Demultiplexer

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The DM74LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The DM74LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

Features

- Designed specifically for high speed:
  - Memory decoders
  - Data transmission systems
- DM74LS138 3-to-8-line decoders incorporates 3 enable inputs to simplify cascading and/or data reception
- DM74LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
  - DM74LS138 21 ns
  - DM74LS139 21 ns
- Typical power dissipation
  - DM74LS138 32 mW
  - DM74LS139 34 mW

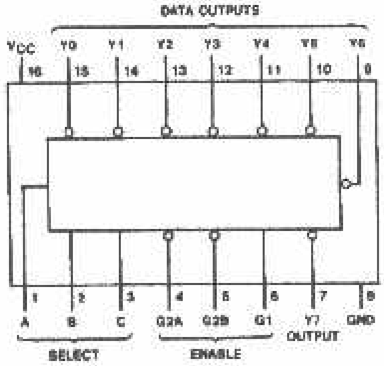
Ordering Code:

Order Number	Package Number	Package Description
DM74LS138M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS138SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS138N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS139M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS139SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS139N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

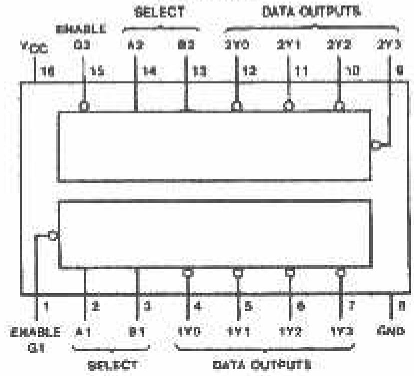
Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagrams

DM74LS138



DM74LS139



Function Tables

DM74LS138

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

DM74LS139

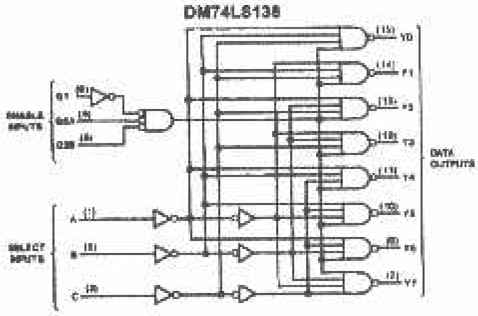
Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care

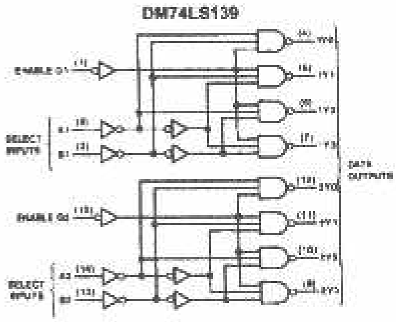
Note 1: G2 = G2A + G2B

Logic Diagrams

DM74LS138



DM74LS139



Absolute Maximum Ratings(Note 2)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note 2: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

DM74LS138 Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage	2			V
V_{IL}	LOW Level Input Voltage			0.8	V
I_{OH}	HIGH Level Output Current			−0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

DM74LS138 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 3)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 4)	−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max (Note 5)		6.3	10	mA

Note 3: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 4: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 5: I_{CC} is measured with all outputs enabled and OPEN.

DM74LS138 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	From (Input) To (Output)	Levels of Delay	R_L = 2 kΩ				Units
				C_L = 15 pF		C_L = 50 pF		
				Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output	2		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output	2		27		40	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output	3		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output	3		27		40	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output	2		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output	2		24		40	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output	3		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output	3		28		40	ns

DM74LS139 Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage	2			V
V_{IL}	LOW Level Input Voltage			0.8	V
I_{OH}	HIGH Level Output Current			−0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

DM74LS139 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 6)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 7)	−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max (Note 8)		6.8	11	mA

Note 6: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 7: Not more than one output should be shorted at a time, and the duration should not exceed one second.

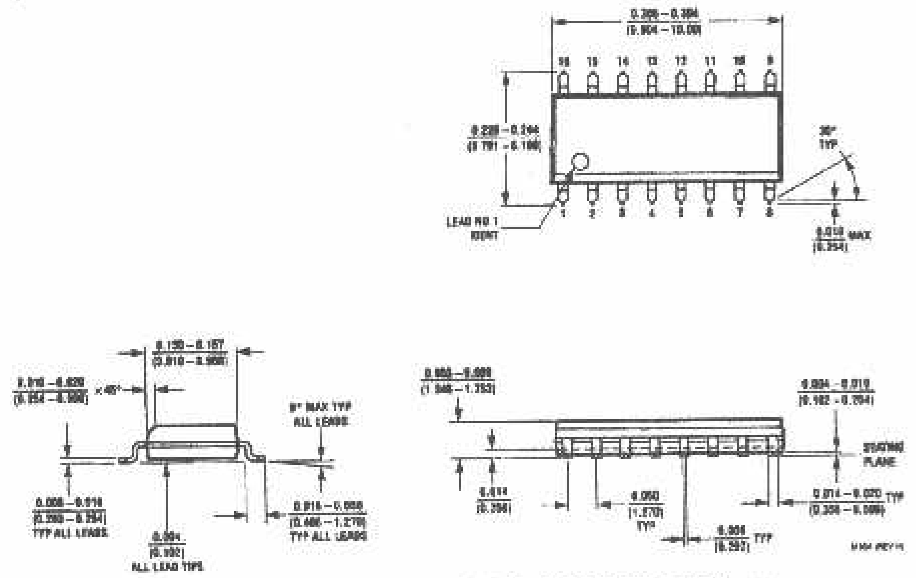
Note 8: I_{CC} is measured with all outputs enabled and OPEN.

DM74LS139 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

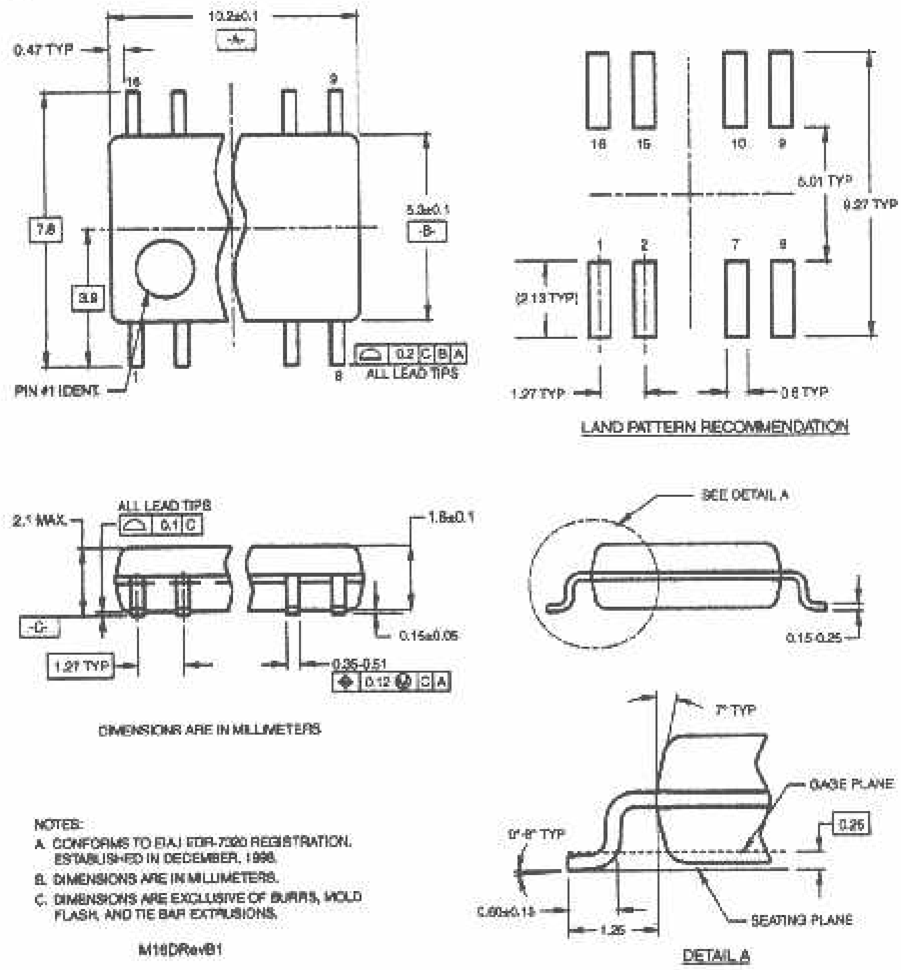
Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output		27		40	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output		24		40	ns

Physical Dimensions inches (millimeters) unless otherwise noted



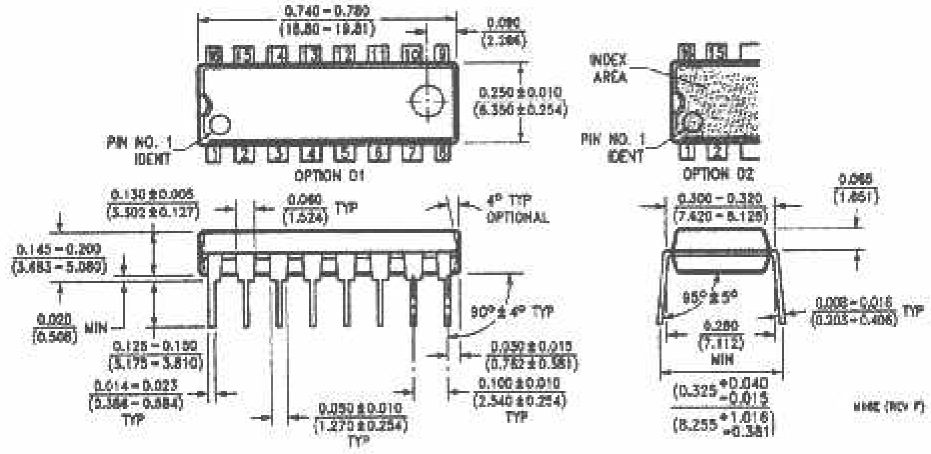
16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
Package Number M16A

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



16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
Package Number M16D

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



16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N16E