TUGAS AKHIR

PENGAMAN MOBIL MENGGUNAKAN RFID CARD BERBASIS AT89C51

Disusun Oleh :

Nama : Aditya Dwiananda

Nim : 02.52.052/P

KONSENTRASI TEKNIK ENERGI LISTRIK

JURUSAN TEKNIK ELEKTRO DIPLOMA III

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

MARET 2007

TUGAS AKHIR

PERANCANGAN MODUL MEDOURNAVAN PELIP CAKRA
UNTUK ANAK ATROGI ?

Nama : [illegible]
NIM : [illegible]

[illegible]
[illegible]

LEMBAR PERSETUJUAN

PENGAMAN MOBIL MENGGUNAKAN RFID CARD BERBASIS AT89C51

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Diploma Tiga (DIII) Pada Jurusan Elektro Energi Listrik

Disusun Oleh :

NAMA : Aditya Dwiananda
NIM : 02.52.052/P

Mengetahui, Ketua jurusan Teknik Elektro D-III	Menyetujui, Dosen Pembimbing
(H.Choirul Saleh,MT) NIP Y : 1018800190	(Ir. Taufik Hidayat, MT) NIP Y : 1018700151

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

KATA PENGANTAR

Dengan mengucapkan puja dan puji syukur kehadirat Tuhan atas berkat rahmat, hidayah dan segala karunia-Nya, maka Laporan Tugas Akhir ini dapat terselesaikan. Tugas Akhir ini dikerjakan sebagai salah satu syarat untuk menyelesaikan studi di Teknik Elektro Energi Listrik Diploma- III, Institut Teknologi Nasional Malang.

Harapan kami semoga dengan adanya Tugas Akhir ini kami lakukan dapat bermanfaat bagi kami dan perkembangan ilmu di ITN Malang pada masa yang akan datang.

Tugas Akhir ini tidak akan terwujud tanpa adanya partisipasi dari berbagai pihak. Oleh karena itu sudah sepatutnya bila pada kesempatan ini penulis mengucapkan terima kasih yang tulus kepada semua pihak yang telah membantu menyelesaikan Tugas Akhir ini diantaranya:

1. Bapak Dr.Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku Dekan FTI ITN Malng.
3. Bapak Ir. Choirul Saleh, MT selaku Kepala Jurusan Elektro-DIII.
4. Bapak Ir. Taufik Hidayat, MT selaku Dosen Pembimbing Tugas Akhir
5. Keluaragaku yang telah memberikan dukungan serta doanya untuk menyelesaikan Tugas Akhir ini.
6. Rekan-rekan yang telah banyak membantu dalam penyusunan Tugas Akhir ini.

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

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

Malang, Maret 2007

Penulis

ABSTRAKSI

"Pengaman Mobil Menggunakan RFID *Card* Berbasis AT89C51", Aditya Dwiananda, Nim : 0252052/P, Jurusan Teknik Elektro Diploma tiga (D-III). Program studi Teknik Energi Listrik, Fakultas Teknik Industri, Dosen Pembimbing Ir. Taufik Hidayat, MT. Institut Teknologi Nasional Malang.

Kata Kunci : *RFID,* pengaman , mobil, pengapian, coil, pemutus, relay, barcode, mikrokontroller, *card,* atmel, bermotor, *hardware, interface,* pengendalian.

RFID ***(Radio Frekuensi Identification)*** merupakan sebuah teknologi compact wireless untuk transformasi dunia komersial. Sebagai suksesor dari barcode. RFID dapat melakukan kontrol otomatis untuk banyak hal. Sistem-sistem RFID menawarkan peningkatan efisiensi dalam pengendalian inventaris, manajemen supply chain, smart card, sucurity access dan masih banyak lagi. Dengan keamanan yang di punyai RFID card, kami membuat sebuah perangkat (interface) dengan mengunakan mikrokontroller Atmel AT 89C51 agar RFID card ini bisa digunakan sebagai gerbang untuk mengakses hardware mobil (pengapian) yang berfungsi sebagai pengaman mobil. Karena akhir-akhir ini banyak terjadi pencurian kendaraan bermotor. Yang sedikit banyak merugikan pemilik kendaraan bermotor. Tujuan pengujian adalah untuk mengetahui keadaan masukan atau keadaan keluaran dari tiap blok rangkaian yang direncanakan, sehingga dengan pengujian ini dapat diketahui apakah alat yang direncanakan dapat berfungsi dengan baik dan sesuai dengan yang diharapkan atau tidak.

Implementasi RFID Card pada sistem pengamanan kendaraan bormotor khususnya pada mobil dengan menggunakan interface berupa mikrokontroller Atmel AT 89C51. Mesin baru bisa dinyalakan bila RFID Card didekatkan dengan reader, apabila kartu sesuai dengan data yang ada di mikrokontroller, mikrokontroller akan memberi signal dan relay aktif. Karena tegangan kerja alat DC +5V, sedangkan sumber tegangan pada mobil dihasilkan dari Accu yang mempunyai output tegangan DC +12V maka dibutuhkan regulator selain sebagai proteksi pada rangkaian juga sebagai tenurun tegangan dari 12V menjadi 5V. Ini menggunakan LM7805CT.

Aplikasi keunggulan **RFID card** dalam bidang pengaman kendaraan bermotor dengan memutus arus power dari coil pengapian. Hasil rancangan ini diharapkan dapat mengurangi pencurian kendaraan bermotor.

DAFTAR ISI

BAB IV PENGUJIAN ALAT

BAB V PENUTUP

KETERANGAN GAMBAR

No Halaman

KETERANGAN TABEL

No Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi Radio Frekuensi Identification (RFID) merupakan teknologi yang diharapkan dapat menggantikan barcode optik di masa yang akan datang. Kelebihan RFID dapat melakukan many-to-many communication (banyak reader dapat membaca satu kartu, maupun satu reader dapat membaca banyak kartu), transmisi data secara wireless, kecepatan perhitungan, serta identifikasi item secara individu. Dengan kelebihan-kelebihannya, sistem RFID menjanjikan prospek untuk berbagai aplikasi, terutama untuk kalangan industri, seperti manajemen perpustakaan, manajemen inventory farmasi, manajemen supply chain, smart card, sucurity access dan masih banyak lagi.

Dengan keamanan yang di punyai RFID card, kami membuat sebuah perangkat (interface) dengan mengunakan mikrokontroller Atmel AT 89C51 agar RFID card ini bisa digunakan sebagai gerbang untuk mengakses hardware mobil (pengapian) yang berfungsi sebagai pemutus rangkaian pengapian yang terhubung dengan kunci kontak.

1.2 Rumusan Masalah

Berdasarkan hal tersebut diatas maka timbul permasalahan seperti :

- Bagaimana merencanakan serta membuat interface agar RFID card bisa untuk mengakses hardware pada kendaraan bermotor.
- Bagaimana cara mengimplementasikan RFID card ini untuk memutuskan rangkaian pengapian kendaraan bermotor yang terhubung dengan kontak.

1.3 Tujuan Penulisan

Merancang untuk mengaplikasikan **RFID card** dalam bidang pengaman kendaraan bermotor dengan memutus arus power dari coil rangkaian pengapian. Diharap dapat mengurangi terjadinya pencurian kendaraan bermotor.

1.4 Batasan Masalah

Agar permasalahan tidak terlalu luas maka penulis membatasi hanya pada hal-hal berikut :

- Tidak membahas secara detail mikrokontroler yang digunakan.
- Ditekankan pada system kerja alat.
- Tidak membahas cara memasukkan data ke dalam kartu.
- Tidak membahas detail aplikasi yang lain.

1.5 Metodologi Penulisan

Untuk mencapai tujuan yang direncanakan dengan hasil optimal, maka dalam pengerjaannya laporan akhir ini dilakukan secara bertahap dengan langkah-langkah sebagai berikut :

1. *Studi literature* tentang *mikrokontroler*
2. *Survey* tentang komponen yang memenuhi
3. Desain rangkaian dan PCB
4. Memasukan *software* ke dalam AT 89C51
5. Pengujian dan perakitan secara menyeluruh

1.6 Sistematika Penulisan

Sistematika penulisan dalam proses penyelesaian penulisan dan pembuatan alat ini penulis melakukan dalam tahap-tahap yang sesederhana mungkin untuk mempermudah pemahaman dan penguasaan teori aplikasi peralatan ini secara praktis. Langkah awal proses terebut adalah studi kepustakaan serta penguasaan teori yang disusul dengan perancangan rangkaian. Selanjutnya diikuti dengan pembuatan laporan tugas akhir yang berupa buku merupakan akhir dari pembuatan tugas akhir. Langkah-langkah diatas dapat dibuat sistematika pembahasan dari buku ini menjadi lima bab, yakni:

BAB I PENDAHULUAN, pada bab I ini berisi tentang hal-hal yang mendasari penulis mengangkat permasalahan yang antara lain:

- Latar belakang permasalahan
- Rumusan Masalah

- Batasan Masalah
- Tujuan penulisan
- Metodologi
- Sistematika pembahasan.

BAB II LANDASAN TEORI, pada bab II ini penulis mencoba mengangkat teori-teori dasar komponen sebagai penunjang dari permasalahan yang diambil. Adapun komponen-komponen yang dimaksud adalah:

1. *Mikrokontroller* AT89C51.
2. *Relay.*
3. *Transistor sebagai driver.*
4. *Reader ID-12.*
5. *RFID Card.*

BAB III PERENCANAAN SISTEM DAN PEMBUATAN ALAT

Dalam bab ini merupakan pembahasan dari komponen-komponen yang digunakan dan pemilihannya menurut perencanaan rangkaian yang telah terintegrasi menjadi suatu sistem kontrol yang berbasis *Mikrokontroller* untuk diimplementasikan dalam bidang pengaman kendaraan bermotor. Isi dari bab ini antara lain:

- Pendahuluan
- Perencanaan Rangkaian Perangkat Keras (*Hardware*)
- Perencanaan Rangkaian Sistem *Mikrokontroller* AT89C51
- Perencanaan RFID Card

- Perencanaan Rangkaian Reader
- Perencanaan Perangkat Lunak

BAB IV PENGUJIAN ALAT

Pada bab ini berisi tentang bagaimana alat ini bekerja dengan baik dibandingkan dengan cara konvensional. Adapun isinya yaitu:

- Pengujian alat.
- Tujuan pengujian.
- Pengujian *cat daya*
- Pengujian *reader/RFID*

BAB V PENUTUP

Bab ini akan membahas **kesimpulan** dan **saran** yang diperoleh dari kekurangan dan kesalahan yang muncul pada pembuatan alat.

BAB II

LANDASAN TEORI

Landasan teori sangat membantu untuk dapat memahami suatu sistem. Selain dari pada itu dapat juga dijadikan sebagai bahan acuan di dalam merencanakan suatu system. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan selanjutnya. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi mikrokontroler AT89C51, *Countactless Reader ID12, RFid Card*

2.1. Sistem Mikrokontroller AT89C51

Mikrokrontroller berbeda dengan mikroprosessor karena selain memiliki CPU juga dilengkapi dengan memori dan input-input yang merupakan kelengkapan sistem dalam mikrokomputer dalam keping tunggal *(Single Chip Mikrokomputer)* yang dapat berdiri sendiri.

2.1.1. Mikrokontroller AT89C51

Perbedaan mendasar antara mikrokontroller dan mikroprosesor adalah mikrokontroller selain memiliki CPU juga dilengkapi dengan memori *input-output* yang merupakan kelengkapan sebagai *system* minimum mikrokomputer sehingga sebuah mikrokontoller dapat dikatakan sebagai mikrokomputer dalam keping tunggal *(single chip Microcomputer)* yang dapat berdiri sendiri.

Mikrokontroller AT89C51 adalah *mikrokontroller* ATMEL yang *kompatibel* penuh dengan mikrokontroller keluarga MCS-51, membutuhkan daya yang rendah, memiliki performa yang tinggi dan merupakan *mikrokomputer* 8 bit yang dilengkapi 4 Kbyte EPROM (*Erasable and Programable Read Only Memori*) dan 128 byte RAM internal. Program memori dapat diprogram ulang dalam sistem atau dengan menggunakan Program *Nonvolately Memory Konvensional.*

Dalam sistem *mikrontroller* terdapat dua hal yang mendasar, yaitu: perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung. Berikut ini adalah tabel keluarga *mikrokontroller* MCS- 51, dapat dilihat bahwa *mikrokontroller* 8031 merupakan *versi* tanpa EPROM dari *mikrokontroller* 8051

PART NUMBER	ON- CHIP CODE MEMORY	ON CHIP DATA MEMORY	TIMER
8051	4K ROM	128 BYTES	2
8031	0K	128 BYTES	2
8751	4K EOROM	128 BYTES	2
8052	8KROM	256 BYTES	3
8032	0K	256 BYTES	3
8752	8KEPROM	256 BYTES	3
AT89C51	4K EPROM	128 BYTES	2

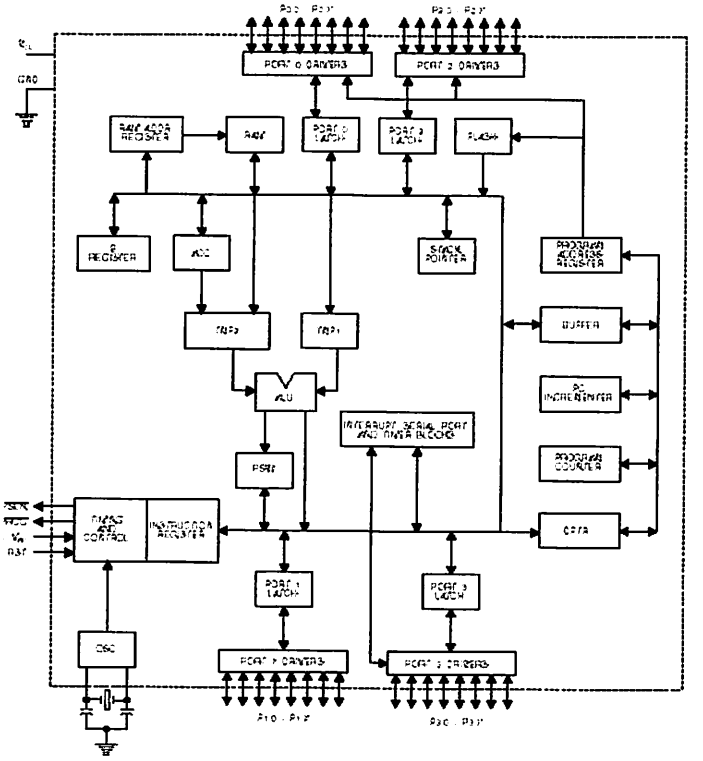
Tabel 2.1. Keluarga *Mikrokontoller MCS- 51*

2.1.2 Arsitektur AT89C51

Sebagai *single chip* yaitu suatu system mikroprosesor yang terintegrasi, mikrokontroller AT89C51 mempunyai konfigurasi sebagai berikut:

1. *CPU* 8 bit termasuk keluarga *MCS-51.*
2. 4 *Kbyte* alamat untuk *memory program internal* (EEPROM).
3. 128 *byte memory* data dalam (*Internal Data memory*/ RAM).
4. 8 *bit* program status *word* (PSW).
5. 8 *bit stack pointer* (SP).
6. 32 pin I/O tersusun yaitu port 0-port 3 @ 8 bit.
7. 2 buah *timer/ counter* 16 bit.
8. Data serial *full dupleks.*
9. *Control register.*
10. 5 sumber *interrupt.*
11. Rangkaian *osilator* dan *clock.*

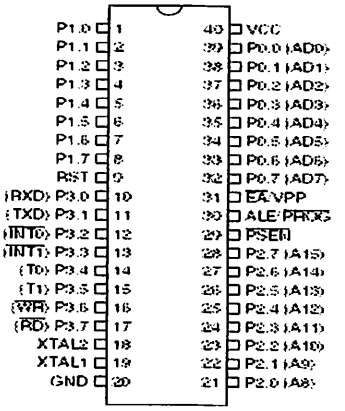
Arsitektur dasar dari *mikrokontroller* AT89C51 seperti diagram blok berikut ini:



Gambar 2.1. Blok Diagram AT 89C51 [1]

2.1.3 Fungsi Pin Mikrokontroller AT89C51

Susunan pin-pin *mikrokontroller* AT89C51 diperlihatkan pada Gambar 2.2, dan penjelasan dari masing-masing pin adalah sebagai berikut:



Gambar 2.2. *Pin*/kaki dari IC AT 89C51 [1]

1. Port 0

Port 0 merupakan port dua fungsi yang berada pada pin 32-39 dari IC AT 89C51. Merupakan port I/O 8 bit dua arah yang serba guna port ini dapat digunakan sebagai *multlipleks bus* data dan bus alamat rendah untuk pengaksesan memori eksternal.

2. Port 1

Port 1 merupakan port I/O yang berada pada pin 1-8. Port ini dapat bekerja dengan baik untuk operasi bit maupun *byte*,tergantung dari pengaturan pada *software*.

3. Port 2

Port 2 merupakan port I/O serba guna yang berada pada pin 21- 28, port ini dapat juga digunakan sebagai bus alamat byte tinggi untuk rancangan yang melibatkan pengaksesan *memori* eksternal.

4. Port 3

Port 3 merupakan port I/O yang memiliki dua fungsi yang berada pada pin 10-17, port ini mempunyai multi fungsi, seperi yang terdapat pada Tabel 2.2 berikut:

BIT	NAMA	BIT ADDRES	FUNGSI ALTERNATIF
P3.0	RXD	B0H	Penerima data pada port serial
P3.1	TXD	B1H	Pemancar data pada port serial
P3.2	INT0	B2H	Eksternal interupsi 0
P3.3	INT 1	B3H	Eksternal interuposi 1
P3.4	T0	B4H	Input Timer/ counter eksternal
P3.5	T1	B5H	Input Timer / counter
P3.6	WR	B6H	Sinyal pembacaan memori data eksternal
P3.7	RD	B7H	Sinyal penulisan memori data eksternal

Tabel 2.2. Fungsi Alternarif Port 3

5. PSEN (*Programable Store Enable*)

PSEN adalah sebuah sinyal keluaran yang terdapat pada pin 29. Fungsinya adalah sebagai sinyal kontrol untuk memungkinkan mikrokontroller membaca program (code) dari memori eksternal atau dapat dikatakan sebagai sinyal kontrol yang menghubungkan memori *program* eksternal dengan bus selama pengaksesan.

6. ALE (*Address Latch Enable*)

Sinyal output ALE yang berada pada pin 30 fungsinya sama dengan ALE pada *mikroprosesor* INTEL 8085 atau 8088. Sinyal ALE dipergunakan untuk demultlipleks bus alamat dan bus data. Dan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.

7. EA (*External Acces*)

Maksudnya sinyal EA terdapat pada pin 31 yang dapat diberikan logika rendah (ground) atau logika tinggi (+ 5 V). Jika EA diberikan logika tinggi maka mikrokontroller akan mengakses program dari ROM internal (EEPROM/ *flash memori).*Jika EA diberi logika rendah maka mikrokontroller akan mengakses program dari memori eksternal.

8. RST (*Reset*)

Input *reset* pada pin 9 adalah reset master untuk AT89C51. Perubahan tegangan dari rendah ke tinggi akan merest AT 89C51.

9. Power

AT89C51 dioperasikan dengan tegangan supply +5v, pin Vcc berada pada pin 40 dan Vss(*ground*) pada pin 20.

2.1.4 Organisasi Memori

Mikrokontroller AT89C51 *mengimplementasikan* ruang memori yang terpisah antara program (*code*) dan data. Seperti ditunjukkan pada Tabel 2.3, program data keduanya bisa merupakan memori internal, tetapi keduanya dapat diperluas dengan memori eksternal sampai 64 Kb memori program dan 64 Kb memori data.

Memori internal terdiri dari ROM/ flash memori dan RAM data didalam chip. RAM berisi susunan *general purposes storage, bit addressable storage, register bank dan special function register.* Ruang internal pada mikrokontroller AT89C51 dibagi menjadi:

1. *Register* bank (00H-1FH), *bit addressable.*
2. *Bit adresable RAM* (20H-2FH).
3. *General Purpose RAM* (30H-7FH).
4. *Special Fungction register* (80H-FFH).

2.1.5 Timer dan Counter

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

- Sebagai *timer/ counter* 8 *bit*

T= (255-TLx) * 1/(F osc/12)

Dimana *TLX* adalah *register TLO* atau *TL1*

- Sebagai *timer / counter* 16 *bit*

T= (65535-THx TLx) * 1 /(Fosc/12)

Dimana :

THx = isi *register TH0* atau *TH1*

TLx = isi *register TLO* atau *TL1*

Pengontrolan kerja timer atau *counter* adalah pada *register timer control* (TCON). Adapun definisi dari bit- bit pada timer control adalah sebagai berikut:

Simbol	Posisi	Fungsi
TF1	TCON. 7	Timer 1 *over flow flag*, diset oleh perangkat keras saat *timer/ counter* menghasilkan *over flow*
TR1	TCON. 6	Bit untuk menjalankan *timer* 1. diset oleh *software* untuk membuat *timer ON/OFF.*
TF 0	TCON. 5	*Timer 0 over flag.* Diset oleh *hardware*
TR 0	TCON. 4	Bit untuk menjalankan *timer* 0. Diset / clear oleh *software* untuk membuat *timer ON* atau *OFF.*
IE 1	TCON. 3	*Eksternal interupt 1 Edge.*
IT 1	TCON. 2	*Interupt 1 type control bit.* Diset/ *clear* oleh *software* untuk menspesifikasi sisi turun/ level rendah dari intrupsi eksternal.
IE 0	TCON. 1	*Eksternal interrupt 0 edge flaf.*
IT 0	TCON. 0	Interupt 0 *type control bit.*

Tabel 2.3. Keterangan Register TCON

MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

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

MSB LSB

GATE	C/T	M1	M0	GATE	C/T	M1	M0

Keterangan :

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

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

M1 : Bit pemilih mode 1

M0 : Bit pemilh mode 0

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

Tabel 2.4. Kombinasi MO dan M1 pada register TMOD[1]

Dibawah ini akan dijelaskan tentang pengertian tentang *mode* yang akan digunakan pada *register* TMOD, sebagai berikut:

- Mode 0

  Dalam kode ini register timer disusun sebagai *register* 13 bit setelah semua perhitungan selesai, mikrokontroller akan mengeset *timer Interupt Flag* (TF1).

Dengan membuat GATE = 1,timer dapat dikontrol oleh masukan liar INT 1,untuk fasilitas pengukuran lebar pulsa.

- Mode 1

Mode 1 sama dengan mode 0 kecuali *register timer* akan bekerja dalam *register* 16-bit.

- Mode 2

Mode 2 menyusun *register timer* sebagai 8-bit counter. Over flow dari TL1 tidak hanya mengeset TF1 tetapi juga mengisi TL1 dengan isi TH 1 yang diatur secara *software*. Pengisian ini tidak mengubah TH1.

- Mode 3

Timer 1 dalam mode 3 semata-mata memegang hitungan. Efeknya sama seperti mengeset TR=0. timer 0 dalam mode 3 menetapkan TL 0 dan TH0 sebagai 2 *counter* terpisah. TL0 menggunakan *control bit timer* 0,yaitu C/T, GATE, TR0, INT0, DAN TF0, TH0 ditetapkan sebagai fungsi *TIMER.*

2.1.6 SFR *(Special Function Register)*

Register internal 8051 tersusun sebagai bagian dari *RAM* internal mikrokontroller. Tentunya setiap register mempunyai sebuah alamat. Special *Function Register* (SFR) berjumlah 21 yang terletak pada bagian atas *RAM internal*,yaitu yang beralamat 80H - ffH. Dapat diperlihatkan seperti table berikut ini:

SIMBOL	*NAME*	ADDRES
ACC	*ACCUMULATOR*	0E0H
B	*B REGISTER*	0F0H
PSW	*PROGRAM STATUS WORD*	0D0H
IP	*INTERUPT PRIORITY CONTROL*	0B8H
IE	*INTERUPT ENABLE CONTROL*	0A8H
P3	*PORT 3*	0B0H
P2	*PORT 2*	0A0H
P1	*PORT 1*	90H
P0	*PORT 0*	80H
SBUF	*SERIAL DATA BUFFER*	99H
SCON	*SERIAL CONTROL*	98H
TH1	*TIMER/ COUNTER 1 HIGH CONTROL*	8DH
TH0	*TIMER/ COUNTER 0 HIGH CONTROL*	8CH
TL1	*TIMER/ COUNTER1 LOW CONTROL*	8BH
TL0	*TIMER/ COUNTER 0 LOW CONTROL*	8AH
TMOD	*TIMER/ COUNTER MODE CINTROL*	89H
TCON	*TIMER/ COUNTER CONTROL*	88H
PCON	*POWER CINTROL*	87H
DPH	*HIGH BYTE*	83H
DPL	*LOW BIYTE*	82H
SP	*STACK POINTER*	80H

Tabel 2.5. *Special Function Register* (SFR)

2.1.7 Program Status Word

Untuk mendefinisikan program status word ini dapat dilakukan perbyte maupun secara keseluruhan dari register ini, terletak dialamat D0H yang berisi bit status. Selengkapnya terdapat pada tabel berikut:

BIT	SIMBOL	ADDRES	BIT DESCRIPTION
PSW. 7	CY	D7 H	*Carry Flag*
PSW. 6	AC	D6 H	*Auxciliaricary Flaf*
PSW. 5	F0	D5 H	*Flag 0*
PSW. 4	RS1	D4 H	*Register bank select 1*
PSW. 3	RS0	D3 H	*Register bank select 0* *00 = bank 0; addresses 00H – 07H* *01 = bank 1; addresses 08 H- 0FH* *10 = bank 2; addresses 10 H- 17 H* *11 = bank 3; addresses 18 H- 1FH*
PSW. 2	0V	D2 H	*Over Flow Flag*
PSW. 1	-	D1 H	*Reserved*
PSW. 0	P	D0 H	*Even Parity flag*

Tabel 2.6. *Program Status Word* (PSW) [1]

2.1.8 Power Register Control

PCON terletak pada alamat 87 H yang berisi beberapa bit control dan dirangkum pada tabel berikut ini.

BIT	SIMBOL	DISKRIPSI
7	SMOD	*Double – baud rate bit*; jika diset maka *baud rate didouble* dan berlaku pada mode serial p[ort 1,2 dan 3
6	-	Tidak didefinisikan

5	-	Tidak didefinisikan
4	-	Tidak didefinisikan
3	GF1	*General purpose flag bit 1*
2	GF2	*General purpose flag bit 0*
1*	PD	*Power down*; kondisi set untuk mengaktifkan mode *power down*, keluar dari mode ini hanya dengan *reset.*
0*	IDL	*Mode idle*; kondisi set untuk mengaktifkan *mode idle*, keluar dari mode ini hanya dengan *interrupt* atau sistem reset

Tabel 2.7. *Power Control Register*[1]

2.1.9 Sistem Interupsi

Mikrokontroller 8051 mempunyai 5 buah sumber interupt yang dapat membangkitkan *interrupt reguest:*

- INT0 : permintaan *interrupt* luar dari kaki P3.2
- INT 1 : Permintaan *interrupt* luar dari kaki P3.3
- Timer/ counter 0 : bila terjadi *overflow*
- *Timer/ Counter* 1 : Bila terjadi *overflow*
- Port serial : Bila Pengiriman/ Penerimaan satu *frame* telah Lengkap

Saat terjadi *interrupt* mikrokontroller secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interrupt service selesai dikerjakan, mikrokontroller akan mengerjakan program semula. Dua sumber merupakan sumber *interupsi eksternal*, INT1. Kedua interupsi eksternal dapat aktif level aktif

transisi tergantung isi ITO dan IT1. Pada register TCON interupsi timer 1dan timer 0 aktif pada saat timer yang sesuai mengalami *rool-over. Interupt* serial dibangkitkan dengan melakukan operasi OR pada R1 dan T1. setiap sumber interupsi dapat *enable* atau *disable* secara *software.*

Tingkat prioritas semua sumber interupsi dapat diprogram sendiri- sendiri dengan set atau *clear* bit pada SFR IP (*Interupt Priority).* Interupsi tingkat rendah dapat diinterupsi oleh interupsi yang mempunyai tingkat interupsi yang lebih tinggi, tetapi tidak sebaliknya. Walaupun demikian, interupsi yang tingkat interupsi nya lebih tinggi tidak bisa menginterupsi sumber interupsi yang lain.

2.1.10 Metode Pengalamatan

Metode pengalamatan pada AT 89C51 adalah sebagai berikut:

a. Pengamatan tak langsung : *Operand* pengalamatan tak langsung menunjuk kearah sebuah register yang berisi lokasi alamat memori yang akan digunakan dalam operasi. Lokasi yang nyata tergantung pada isi *register* saat instruksi dijalankan. Untuk melaksanakan pengalamatan tak langsung digunakan symbol @. Berikut ini diberikan beberapa contoh:

ADD A, @ R0	: Tambahan isi RAM yang lokasinya ditunjuk oleh register R0 ke akumulator
DEC @R1	: Kurangilah dengan satu, isi RAM yang alamatnya ditunjukan oleh register R1.
MOVX @ DPTR,A	: Pindahkan isi akumullator ke memori luar yang lokasinya ditunjukkan oleh *data pointer* (DPTR).

b.Pengalamatan langsung

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung. Untuk melaksanakan hal tersebut digunakan tanda #. Sebagai contoh:

MOVA, # 01 H: isi akumulator dengan bilangan 01 H

MOV DPTR, # 19 ABH: Isi register DPTR dengan bilangan 19AB h

Pengalamatan data langsung dari 0 sampai 127 akan mengakses RAM internal. Sedang pengalamatan dari 128 sampai 255 akan mengakses *register* perangkat keras sebagai contoh:

MOV P3, A : Pindahkan isi akumulator ke alamat data B0 H

(BOH adalah alamat Port 3)

c. Pengalamatan bit

Pengalamatan bit adalah penunjukan alamat lokasi bit baik dalam RAM internal, (byte 32 sampai 47) maupun bit perangkat keras. Untuk melakukan pengalamatan bit digunakan simbol titik misalnya :

SETB 88 H. 6: set bit pad lokasi 88H (Timer 1ON)

d. Pengalamatan kode

Ada tiga macam instruksi yang dibutuhkan dalam pengalamatan kode, yaitu *relative jump, in- blockjump atau call, dan long jump.*

2.2. On/Off (Relay)

Relay adalah suatu perangkat *switch* (saklar) yang dioperasikan oleh gaya *elektromagnetik* yang dihasilkan oleh kumparan yang berada di dalamnya. *Relay*

ini pada umumnya digunakan untuk menyambung atau memutuskan hubungan antara suatu bagian yang lain dalam suatu rangkaian elektronik, selain itu juga dimaksudkan untuk mengisolasi *switching* antara tegangan catu daya tinggi dengan tegangan catu daya rendah. Kerugian yang ditemui pada *relay* yaitu adanya tanggapan waktu *respon (response time)* saat *on* maupun *off* yang relative lambat serta adanya efek induksi balik sesaat setelah *relay off.* Oleh sebab itu maka antara IC pengendali dan relay perlu diisolasi dengan suatu rangkaian isolasi

2.2.1. Transistor Sebagai Driver (switching)

Suatu transistor bila ingin digunakan sebagai saklar elektronika *(switching)* maka harus dioperasikan dalam keadaan saturasi dan *cutoff.* Pada keadaan saturasi tegangan antara kaki kolektor dan emitor dianggap 0 volt (ideal) atau $V_{CE} \approx 0$ Volt dan arus basis sama dengan $I_{B(Sat)}$ serta arus kolektor adalah maksimum.

Persamaan-persamaan yang dapat digunakan dalam perhitungan yaitu :

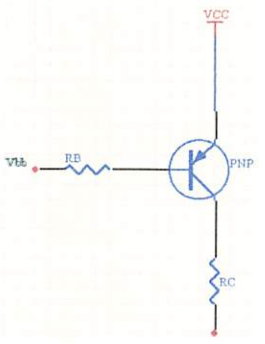
$$I_{C(Sat)} \quad \frac{Vcc}{Rc}$$

$$Vbb \quad Ib \quad Vbe$$

$$Hfe \quad \frac{I_{C(Sat)}}{I_{B(Sat)}}$$

Pada keadaan *cutoff* dimana I_B 0 dan arus kolektor kecil sehingga dapat diabaikan (hanya arus bocoran I_{CEO} yang ada)

Jika arus basis lebih besar dari pada $I_{B(Sat)}$, arus kolektor tak dapat bertambah karena dioda kolektor tidak lagi dibaca *reverse*. Dengan perkataan lain hal ini keadaan *saturasi* yang sama. [3]



Gambar 2.3 Transistor Sebagai Saklar[6]

2.3 Countactless Reader ID12

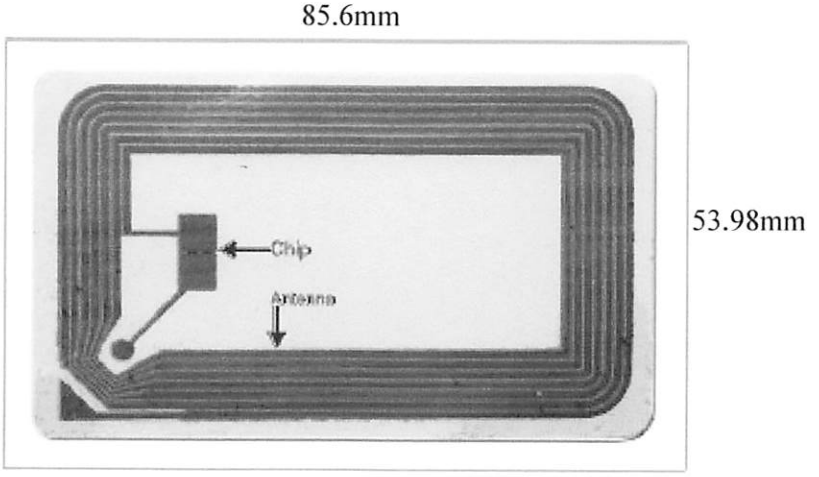
Pada pembuatan alat ini kami menggunakan Reader merek Elatec tipe ID-12, untuk merek ini elatec mengeluarkan berbagai tipe antara lain ID-2, ID-12, ID-20.

Untuk ID-12 dan ID-20 menggunakan internal antenna dan hanya mempunyai jarak baca 12cm – 16cm, sedangkan untuk tipe ID-2 menggunakan external antenna dan sanggup membaca diatas 25cm. Semua reader ini bekerja pada frekuensi 125kHz. Bekerja pada tegangan +5V DC dan pada suhu ruangan -10°C sampai +50°C. Dimensi reader ID-12 26mm x 25mm x 7mm. [3]

2.4 RFID Card

Sebuah kartu RFID atau transponder (tag), terdiri atas mikrochip dan sebuah antenna (gambar 2), microchip itu bisa sebesar butiran pasir. Chip tersebut

menyimpan nomer seri yang unik atau informasi yang lainnya tergantung tipe memorinya.



Gambar 2.4 RFID Card [11]

Transponder sangat bervariasi bentuknya, yang kami gunakan ini berbentuk kartu yang ukurannya sudah mengikuti standarisasi 85.6mm x 53.98mm. transponder ada beberapa jenis :

Tag pasif, yaitu tag yang tidak mempunyai catu daya sendiri serta tidak menginisiasi komunikasi dengan reader. Sebagai gantinya tag merespon emisi frekuensi dan menurunkan dayanya dari gelombang – gelombang energi yang dipancarkan oleh reader.sebuah tag pasif minimum mempunyai nomor seri unik. Data tambahan dimungkinkan ditambahkan tergantung dari kapasitas penyimpanan. Bekerja pada frekuensi rendah (low frequency, LF). Tag pasif lebih murah dibandingkan tag yang lainnya.

Tag semipasif, adalah versi tag yang mempunyai catu daya sendiri tetapi tidak dapat menginisiasi komunikasi dengan reader.dalam hal ini baterai digunakan oleh tag sebagai catu daya untuk melakukan fungsi yang lain seperti pemantauan

keadaan lingkungan dan mencatu bagian elektronik internal tag. Dan memfasilitasi penyimpanan informasi.tag versi ini tidak secara aktif memancarkan signal ke reader.

	Tag pasif	*Tag* semipasif	*Tag* aktif
Catu daya	eksternal (dari *reader*)	baterai internal	baterai internal
Rentang baca	dapat mencapai 20 kaki	dapat mencapai 100 kaki	dapat mencapai 750 kaki
Tipe memori	umumnya read-only	read-write	read-write
Harga	5.20 hingga beberapa dolar	$2 hingga $10	$20 atau lebih
Usia *tag*	dapat mencapai 20 tahun	2 sampai 7 tahun	5 sampai 10 tahun

Tabel 2.8 Karakteristik umum Tag RFID [11]

Tag aktif, tag jenis ini selain mempunyai antenna dan chip juga memiliki catu daya dan pemancar serta mengirim signal kontinyu. Tag ini biasanya memiliki kemampuan baca tulis.dalam hal ini data tag dapat ditulis ulang.tag ini dapat menginisiasi komunikasi dan dapat berkomunikasi pada jaraj yang lebih jauh, hingga 750 kaki. Tag ini merupakan yang paling mahal. [11]

Gelombang	Frekuensi	Rentang dan laju baca	Contoh penggunaan
LF	125 KHz	--1.5 kaki kecepatan baca rendah	Access control, animal tracking, point of sale applications
HF	13.56 MHz	--3 kaki: kecepatan baca sedang	Access control, smart cards, item-level tracking
UHF	860-930 MHz	up to 15 kaki kecepatan baca tinggi	Pallet tracking, supply chain management
Gelombang mikro	2.45/5.8 GHz	--3 kaki: kKecepatan baca tinggi	Supply chain management

Tabel 2.9 Frekuensi RFID yang umum beroperasi pada tag Pasif [11]

BAB III

METODOLOGI

3.1 Perancangan Alat

Dalam Bab ini akan dibahas pembuatan seluruh sistem perangkat yang ada pada alat System Pengamanan kendaraan bermotor, secara garis besar terdapat dua bagian perangkat yang ada yaitu:

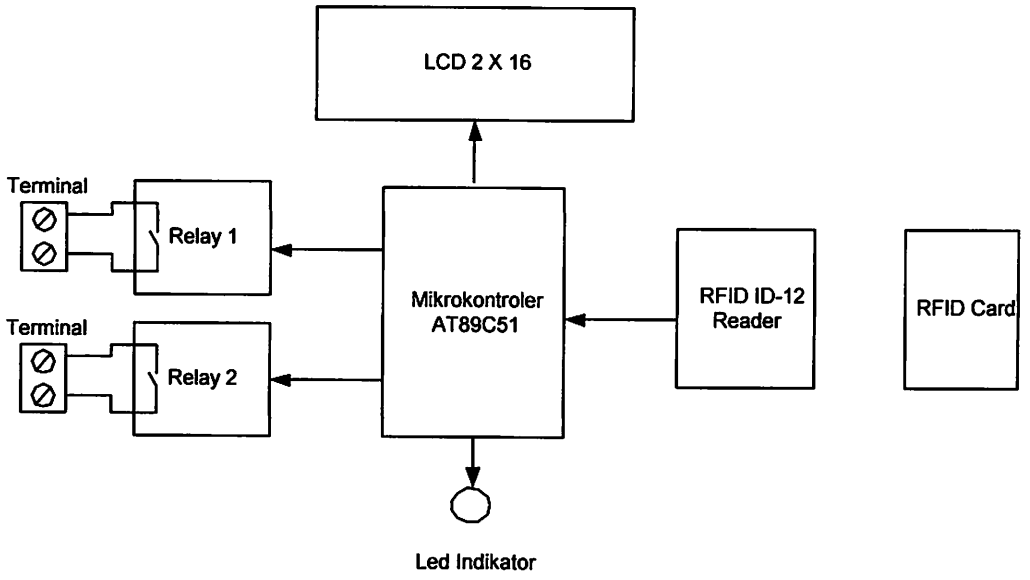
1. Perencanaan perangkat keras
2. Perencanaan perangkat lunak

Pada perencanaan perangkat keras akan meliputi penjelasan dari perencanaan diagram blok sistem dan juga perencanaan *minimum* sistem mikrokontroller AT89C51 beserta peripheral yang digunakan pada perencanaan perangkat lunak yang juga digunakan pada minimum sistem *mikrokontroller* AT89C51. Akan tetapi perangkat tersebut dalam kerjanya akan saling mendukung satu dengan lainnya sehingga alat yang direncanakan dapat berjalan sesuai dengan perencanaannya.

3.2 Perancangan Perangkat Keras

Dalam Tugas Akhir ini perencanaan dan pembuatan Alat System Pengaman kendaraan bermotor berbasis *mikrokontroller* AT89C51 sebagai kontrol utama dan menggunakan komponen lain sebagai komponen pendukung. Sebelum membuat perangkat keras keras terlebih dahulu direncanakan blok diagram yang

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



Gambar 3.1 Diagram Blok secara keseluruhan

Gambar diatas adalah diagram blok dari rangkaian system. Rangkaian alat pengaman ini terdiri dari RFID card, reader, mikrokontroller AT 89C51 dan relay.

3.2.1 Cara Kerja Rangkaian Secara Keseluruhan

Pertama Card didekatkan ke RFID reader kemudian data ID yang diterima dari RFID Card dikirimkan ke Mikrokontroler melalui jalur serial dengan baut rate 9600bps dengan format data

02	10 ASCII Data Characters	Checksum	CR	LF	03

Seluruh data dalam bentuk Byte. 02 adalah start transmisi data, 10 Ascii Data Character adalah Data Tipe dan Nomor Kartu Format Hexadesimal, Ceksum adalah ceksum dari 10 data Ascii, CR adalah #13, LF adalah #10 dan 03 adalah End of transmisi.

Data yang diolah oleh mikrokontroler adalah 10 Ascii Data Character. Dari 10 data tersebut data ke 1-2 adalah data tipe kartu sedangkan 3-10 adalah Nomor kartu.

Mikrokontroler mengkonversikan Hexadesimal data ke 3-10 menjadi bilangan decimal yang terdapat pada kartu.

Kemudian data nomor kartu yang telah dikonversikan dalam bentuk decimal tersebut dibandingkan dengan 2 buah nomor kartu yang telah diinputkan kedalam program memori mikro, Jika Data kartu cocok dengan nomor kartu yang ada didalam mikro, maka relay akan ON (jika relay pada posisi ON maka akan di OFFkan /toggle) dan Led indicator akan menyala.

Sebaliknya jika data nomor tersebut tidak cocok terhadap kedua nomor yang telah diinputkan, maka dilayar LCD akan tampil NOMOR TIDAK DIKENAL.

Mikro mengaktifkan relay dengan memberikan logika HIGH terhadap basis transistor driver Relay, sehingga transistor tersebut ON dan mengalirkan arus dari +5V melewati kumparan relay sehingga terjadi medan magnet dan akan menarik saklar yang ada didalamnya sehingga terjadi kontak.

3.2.2 Fungsi Komponen dari Rangkaian Sistem

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

1. Mikrokontroller AT89C51

   Digunakan sebagai kotrol utama untuk mengendalikan system dengan bantuan *software*.

2. RFID card

   Sebagai kunci (akses) agar sistem dapat berfungsi sebagai pengaman.

3. Reader ID-12

   Digunakan untuk membaca RFID card dan kemudian memerintahkan mikrokontroller.

4. LCD

   Untuk menampilkan apabila kartu sesuai akan tertuliskan no kartu dan bila tidak sesuai akan tampil *nomor tidak dikenal.*

5. Relay

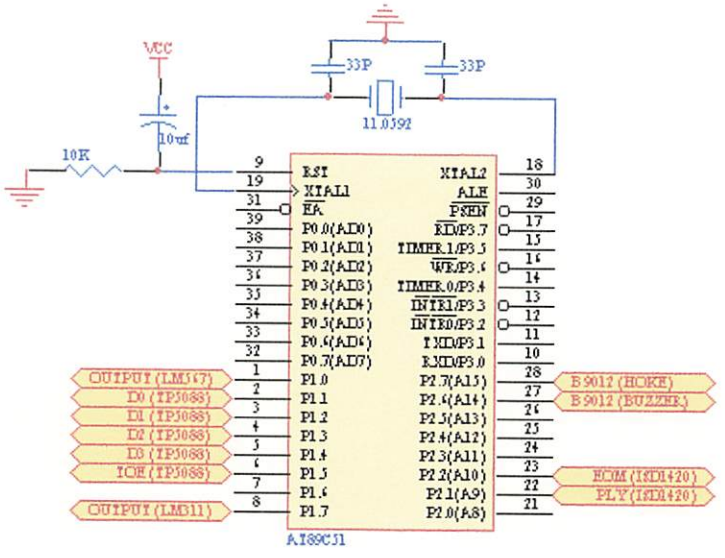
   Mikro mengaktifkan relay dengan memberikan logika HIGH terhadap basis transistor driver Relay, sehingga transistor tersebut ON dan mengalirkan arus dari +0.77V melewati kumparan relay sehingga terjadi medan magnet dan akan menarik saklar yang ada didalamnya sehingga terjadi kontak.

3.3 Perencanaan Rangkaian System Mikrokontroller AT89C51

Dalam perencanaan rangkaian *mikrokontroller* AT89C51 mengikuti ketentuan-ketentuan yang menjadi aturan minimum sistem yang telah ditentukan oleh pabrik pembuatnya.

3.3.1 Perencanaan Rangkaian Minimum Sistem MCU

Rangkaian *minimum mikrokontroller* AT89C51 dan penyemat (pin) yang digunakan dalam perencanaan alat ini ditunjukkan pada gambar berikut ini:



Gambar 3.2. Rangkaian *Minimum Mikrokontroller* AT89C51

Penyemat X1 dan X2 dihubungkan dengan *kristal* yang berfungsi sebagai pembentuk sebuah *isolator* bagi *mikrokontroller*. *Kristal* 12MHz ini didukung dua *capasitor* keramik C1 dan C2 yang nilainya sama. Apabila terjadi beda *potensial*

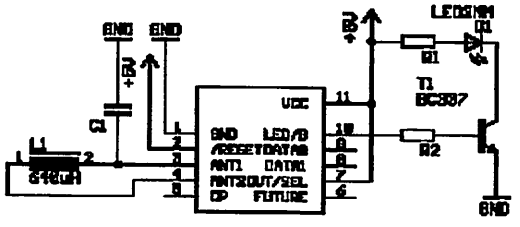
pada kedua *kapasitor* tersebut maka *kristal* akan *berosilasi*. *Pulsa* yang keluar adalah berbentuk gigi gergaji dan akan dikuatkan oleh rangkaian internal pembangkit rangkaian pulsa pada *mikrokontroller* sehingga akan berubah menjadi pulsa clock. Untuk pembagian dari *frekuensi internal mikrokontroller* itu sendiri yang *diinisialisasi* dengan *program*.

Penyemat *Reset* dihubungkan dengan *saklar* yang digunakan untuk me-*Reset mikrokontroller*. Karena kaki *reset* ini aktif *berlogic* tinggi maka diperlukan Resistor R1 yang nilainya 10K, yang dihubungkan dengan tegangan 0 Volt untuk memastikan penyemat *Reset berlogic* rendah saat sistem ini bekerja. Kapasitor C1=10µF berfungsi untuk meredam adanya pelentingan akibat penekanan *saklar Reset*.

3.4 Perencanaan Rangkaian Reader

Untuk membaca kartu dibutuhkan reader. Pada pembuatan alat ini kami menggunakan Reader merek Elatec tipe ID-12.

Untuk ID-12 menggunakan internal antenna dan hanya mempunyai jarak baca 12cm – 16cm, Semua reader ini bekerja pada frekuensi 125kHz. Bekerja pada tegangan +5V DC dan pada suhu ruangan -10°C sampai +50°C. Dimensi reader ID-12 26mm x 25mm x 7mm.



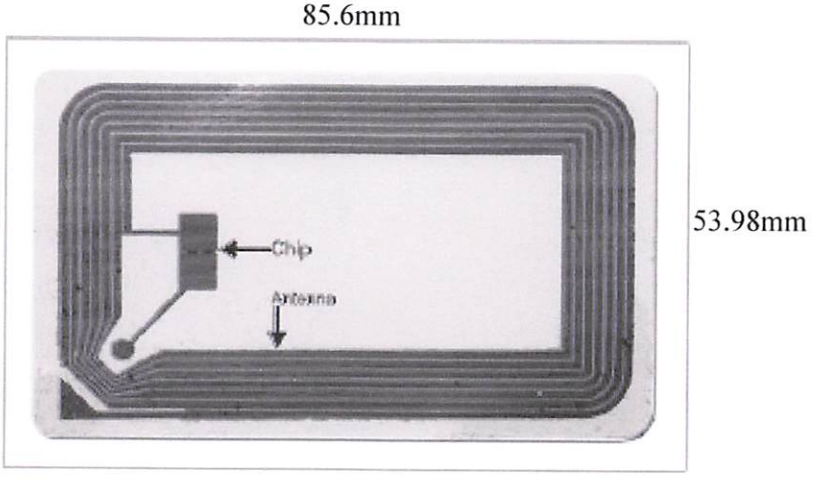
Gambar 3.3 Rangkaian reader

Pin No.	Description	ASCII	Magnetic Emulation	Wiegand 26
Pin 1	Zero Volts	GND	GND	GND
Pin 2	Strap to +5V	Reset bar	Reset Bar	Reset bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present	No function
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to 5V
Pin8	Data 1	CMOS	Data	One Output
Pin 9	Data 0	TTL Data interval	Clock	Zero Output
Pin 10	3.1 kHz Logic	Beeper/LED	Beeper/LED	Beeper/LED
Pin 11	DC Voltage Supply	+5V	+5V	+5V

Tabel 3.1 pin Reader

3.5 Perencanaan RFID Card

Sebuah kartu RFID atau transponder (tag), terdiri atas mikrochip dan sebuah antenna (gambar 2), microchip itu bisa sebesar butiran pasir. Chip tersebut menyimpan nomer seri yang unik atau informasi yang lainnya tergantung tipe memorinya.



Gambar 3.4 RFID Card

3.6 LCD

Untuk mempermudah pengamatan kami gunakan LCD 2 garis 16 karakter

Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

Tabel 3.2 Pin LCD

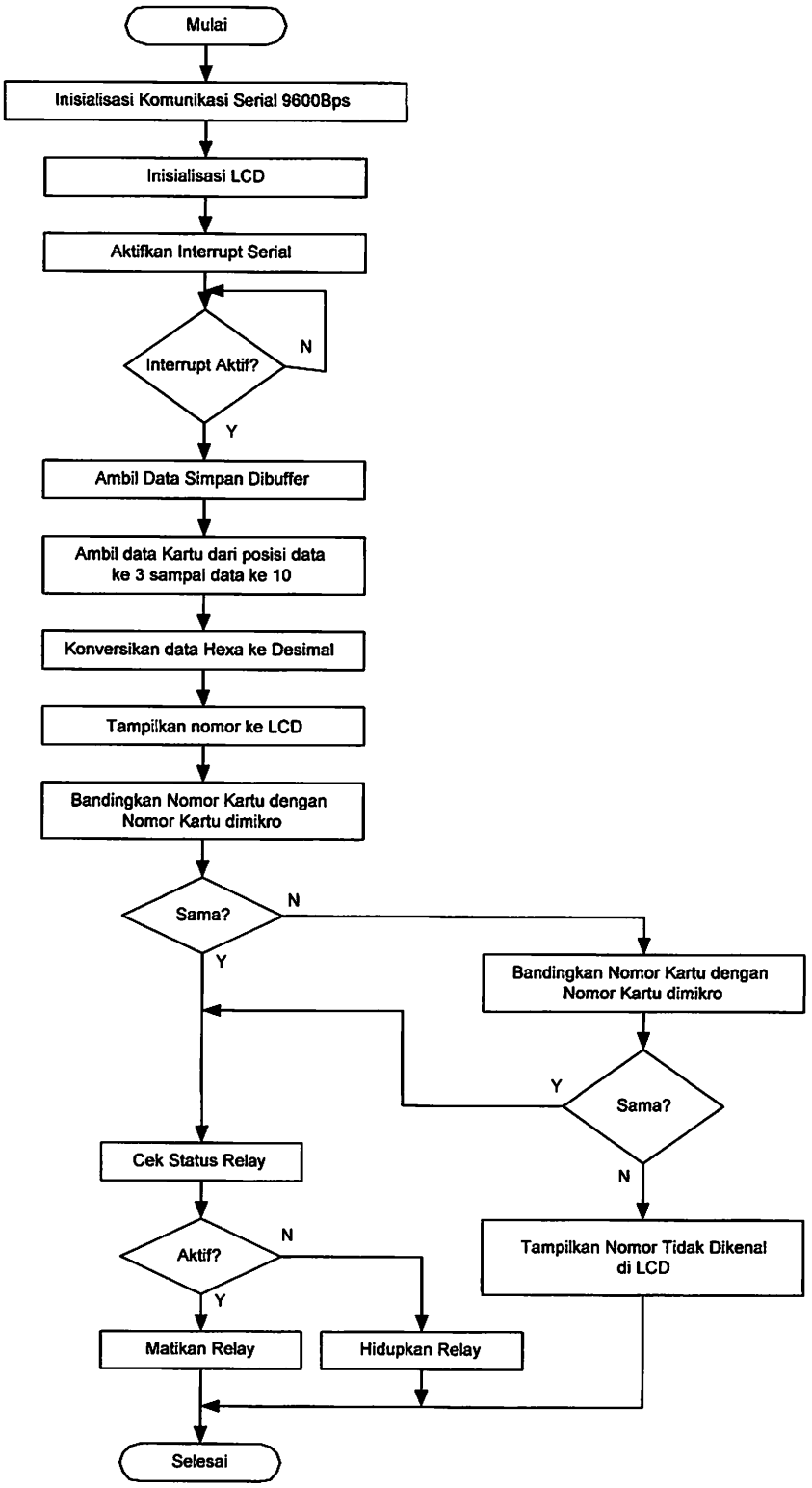
3.7 Relay

Untuk memutus arus Coil pengapian mobil dibutuhkan relay yang di driver dari transistor.

3.8 Perencanaan Perangkat Lunak

Pembuatan perangkat keras tidak banyak gunanya apabila tidak didukung oleh perangkat lunak. Perangkat lunak yang digunakan disini menggunakan bahasa program *assembler.* Dan sebelum menyusun program assembler terlebih dahulu kita harus membuat diagram alir *(flowchart)* dari program tersebut. Diagram alir akan mempermudah menentukan program yang akan dibuat.

Dibawah ini dapat dilihat perencanaan diagram alir program dari alat yang direncanakan dan pada lampiran dapat dilihat susunan/*listing* program bahasa *Assembler.*


Mulai
Inisialisasi Komunikasi Serial 9600Bps
Inisialisasi LCD
Aktifkan Interrupt Serial
Interrupt Aktif?
N
Y
Ambil Data Simpan Dibuffer
Ambil data Kartu dari posisi data ke 3 sampai data ke 10
Konversikan data Hexa ke Desimal
Tampilkan nomor ke LCD
Bandingkan Nomor Kartu dengan Nomor Kartu dimikro
Sama?
N
Y
Bandingkan Nomor Kartu dengan Nomor Kartu dimikro
Y
Sama?
N
Cek Status Relay
Aktif?
N
Y
Tampilkan Nomor Tidak Dikenal di LCD
Matikan Relay
Hidupkan Relay
Selesai

BAB IV

PENGUJIAN ALAT

Tujuan pengujian adalah untuk mengetahui keadaan masukan atau keadaan keluaran dari tiap blok rangkaian yang direncanakan, sehingga dengan pengujian ini dapat diketahui apakah alat yang direncanakan dapat berfungsi dengan baik dan sesuai dengan yang diharapkan atau tidak. Untuk tujuan ini, pengujian dilakukan dengan urutan rangkaian sebagai berikut :

1. Pengujian Reader
2. Pengujian LCD
3. Pengujian Regulator
4. Pengujian Relay

4.1 Pengujian Reader

4.1.1. Tujuan

Untuk mengetahui berfungsi atau tidaknya reader yang digunakan harus dilakukan pengetesan.

4.1.2 Peralatan Yang Digunakan

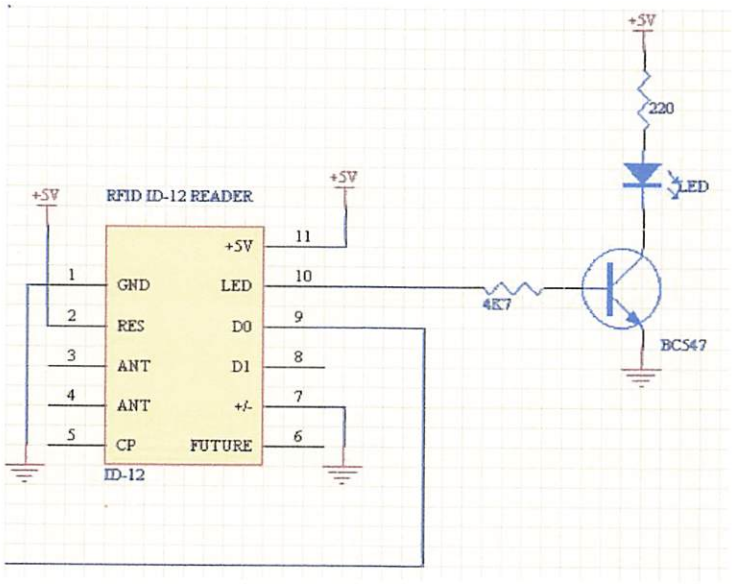
1. Rangkaian Regulator
2. LED (Light Emitting Diode)

4.1.3 Langkah Pengujian

Langkah-langkah pengujian Reader :

1. Menyusun rangkaian Regulator

2. Keluaran Regulator dihubungkan dengan pin 11 dan 2
3. ground pada pin 1 dan 7
4. LED pada pin 10



Gambar 4.1 Rangkaian Pengujian Reader

4.1.4 Hasil Pengujian

Pengujian rangkaian reader dengan cara menghubungkan dengan tegangan DC +5V, sedangkan keluarannya diberi indikator LED.Setelah itu kartu didekatkan pada reader

Gambar 4.2 Pengujian Reader dengan output LED

Data Hasil Percobaan Pengujian Reader

No	Posisi Kartu	LED
1	dekat	nyala
2	jauh	mati
3	dekat	nyala
4	jauh	mati
5	dekat	nyala
6	jauh	mati

Tabel 4.1 hasil percobaan pengujian Reader

4.2 Pengujian LCD

4.2.1 Tujuan

Pengujian rangkaian untuk mengetahui LCD yang kita gunakan berfungsi untuk tampilan diterima atau tidaknya kartu yang dipakai.

4.2.2 Peralatan Yang Digunakan

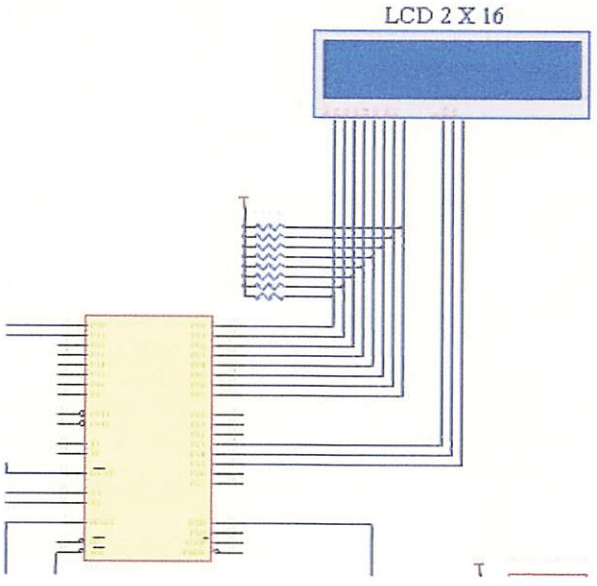
1. Rangkaian Regulator

2. Rangkaian LCD

4.2.3 Langkah Pengujian

Langkah pengujian LCD Adalah sebagai berikut :

1. Hubungkan pin 1 dengan ground
2. Hubungkan pin 2 dengan DC +5V
3. Hubungkan pin 7-14 ke data pada mikro



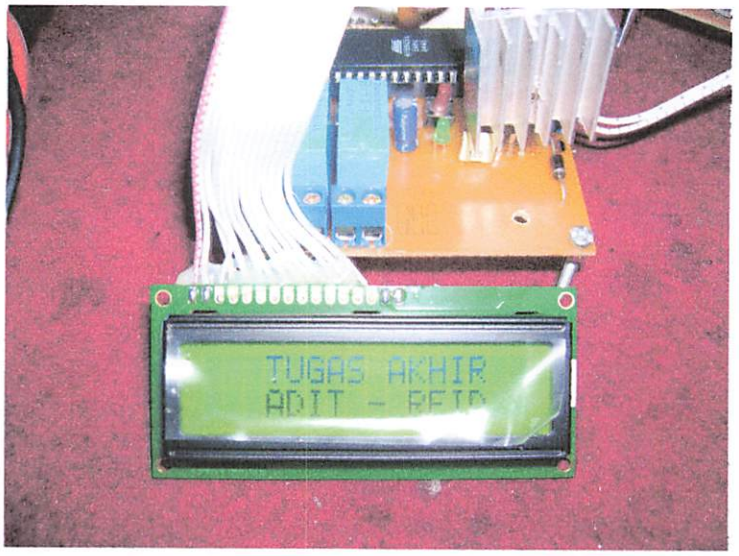
Gambar 4.3 Rangkaian Pengujian LCD

Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable *(Strobe)*
7	D0	I/O	Data *LSB*
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data *MSB*

Tabel 4.2 Pin LCD

4.2.4 Hasil Pengujian

Dengan menghubungkan pin-pin LCD pada jalurnya, apabila pin 2 masuk tegangan DC +5V maka LCD akan menyala.



Gambar 4.4 Pengujian LCD

4.3 Pengujian Regulator

4.3.1 Tujuan

Karena tegangan kerja alat DC +5V, sedangkan sumber tegangan pada mobil dihasilkan dari Accu yang mempunyai output tegangan DC +12V maka dibutuhkan regulator selain sebagai proteksi pada rangkaian juga sebagai tenurun tegangan dari 12V menjadi 5V. Ini menggunakan LM7805CT.

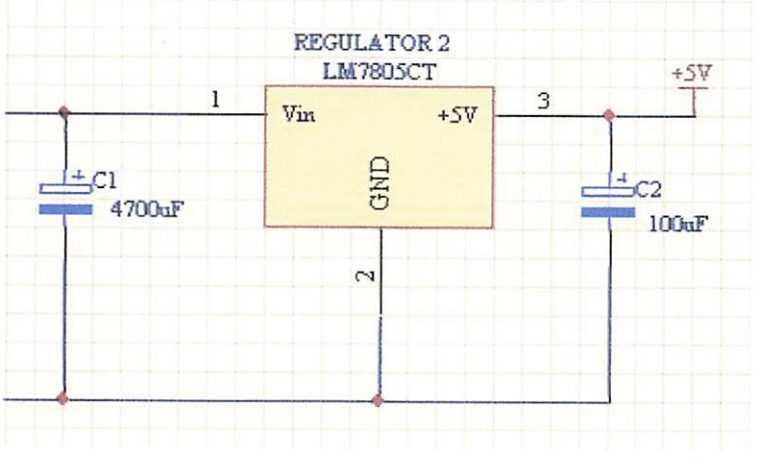
4.3.2 Peralatan Yang Digunakan

1. Rangkaian Regulator
2. Multitester Digital

4.3.3 Langkah Pengujian

Langkah pengujian Regulator adalah sebagai berikut :

1. Menyusun rangkaian seperti pada gambar 4.5
2. Menghubungka kaki 3 LM7805CT dengan + pada multitester

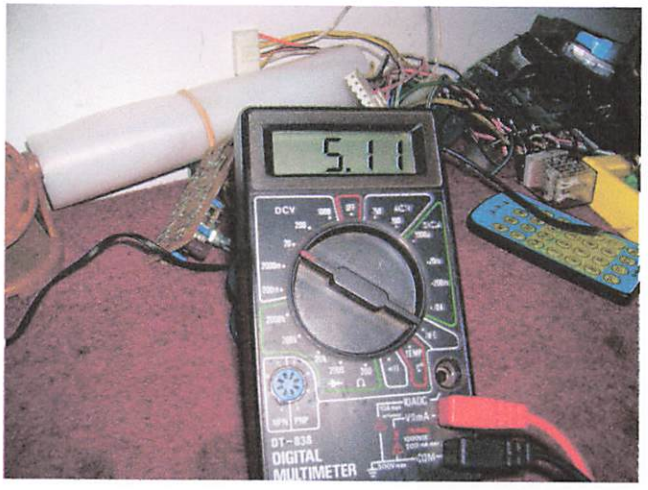


Gambar 4.5 Pengujian Rangkaian Regulator

4.3.4 Hasil Pengujian

Pengujian rangkaian dengan menghubungkan kaki 1 ke + Accu, dan kaki 2 Pada ground. Sedangkan keluarannya dari LM7805CT pada kaki 3, itu kita ukur menggunakan multitester digital.

Dan hasil pengukuran dari keluaran LM7805CT ini adalah DC + 5.1V



Gambar 4.6 Hasil Pengukuran Regulator

Data hasil pengukuran mengunakan LM7805CT

No	Tegangan masuk (Volt)	Tegangan Keluar (Volt)
1	12	5.11
2	12	5.11
3	12	5.10
4	12	5.11
5	12	5.11
6	12	5.11

Tabel 4.3 Data Hasil pengukuran LM7805CT

4.4 Pengujian Relay

4.4.1 Tujuan

Pengujian relay ini untuk memutus dan menyambung Coil pengapian mobil, relay ini akan bekerja berdasarkan drive dari transistor C547

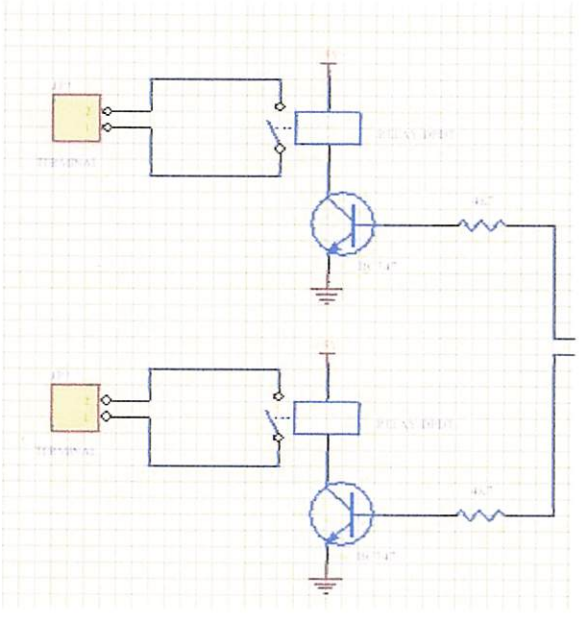
4.4.2 Peralatan Yang Digunakan

1. Rangkaian Relay
2. Multitester Digital

4.4.3 Langkah Pengujian

Langkah-langkah pengujian Relay adalah sebagai berikut :

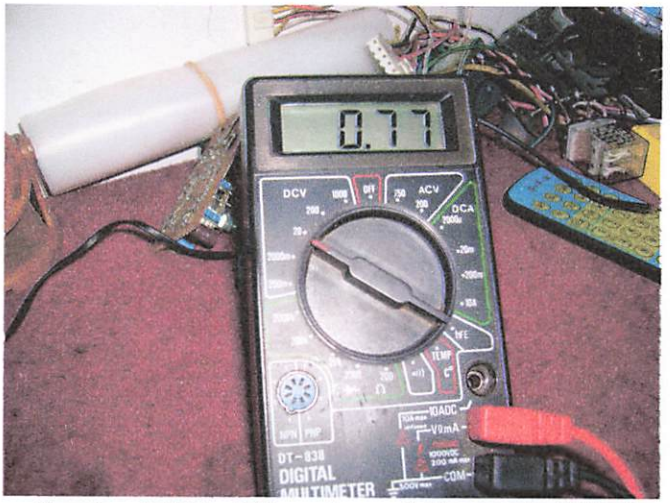
1. Menyusun rangkaian seperti pada gambar 4.7
2. Melihat dengan multitester digital antara terminal 1&2



Gambar 4.7 Pengujian Rangkaian Relay

4.4.4 Hasil Pengujian

Untuk mendrive Relay mengunakan transistor C547. pada hasil pengujian rangkaian ini didapat output dari C547 adalah DC +0.77V. dan tegangan ini yang digunakan untuk menggerakkan relay yang menghasilkan terminal 1&2 terhubung.



Gambar 4.8 Hasil Pengukuran keluaran dari C547

Data Hasil Percobaan Pengukuran Transistor C547

No	Tegangan masuk (Volt)	Tegangan Keluar (Volt)
1	1.2	0.77
2	1.2	0.77
3	1.2	0.76
4	1.2	0.76
5	1.2	0.77
6	1.2	0.77

Tabel 4.4 Data Hasil Percobaan Pengukuran C547

BAB V

PENUTUP

Bab ini merupakan kesimpulan dari aplikasi Mikrokontroller AT89C51 dengan Kartu RFID sebagai mengaman mobil. Bab ini juga diberikan saran yang mengkin bisa dilaksanakan untuk meningkatkan unjuk kerja piranti yang telah dibuat.

5.1. Kesimpulan

Setelah melalui beberapa tahap perencanaan alat yang kemudian dilanjutkan dengan pengujian alat untuk memberikan gambaran mengenai unjuk kerja alat yang telah dibuat. Dari hal pengujian dan analisis diperoleh kesimpulan sebagai berikut :

1. Dengan mengaplikasikan keunggulan **RFID card** dalam bidang pengaman kendaraan bermotor dengan memutus arus power dari coil pengapian. Dengan ini kami berharap bisa mengurangi memungkinan terjadinya pencurian kendaraan bermotor.
2. Dari hasil pengujian keluaran transistor C547 yang digunakan untuk mendrive relay didapat output sebesar +0.77V
3. Sedangakan pada pengujian rangkaian Regulator LM7805CT didapat output sebesar +5.1V ini digunakan untuk tegangan kerja alat.

5.2. SARAN

1. Dalam rancang bangun suatu peralatan hendaknya direncanakan secara matang dengan memperhatikan keterbatasan pengetahuan, ketrampilan dan pengalaman yang dimiliki serta sarana dan prasarana yang dibutuhkan. Observasi di lapangan perlu dilakukan untuk melengkapi data-data yang dibutuhkan dan memperluas wawasan.
2. Bahan dan komponen yang digunakan dalam perancangan dipilih setepat mungkin dengan harga yang terjangkau tetapi masih memenuhi syarat-syarat kenyamanan dan keamanan.
3. Dalam pengoperasian alat, hendaknya diperhatikan cara pengoperasian yang benar karena akan memberikan hasil yang maksimal sesuai dengan yang diharapkan.

DAFTAR PUSTAKA

1]. Data Sheet Mikrokontroller AT89C51. *Atmel Corp*

2]. Data Sheet ID-12

3]. Data Sheet LM7805

4]. Data Sheet LCD

5]. ATMEL Data Book, 1999

6]. Design Protel 98

7]. Ari Juels (2005),*RFID Security and Privacy:A Research Survey,* http://www.rsasecurity.com/rsalabs/staff/bios/ajuel/publication/pdfs/rfid_survey_28_09_05.pdf, 26 Januari 2007, 15.23 WIB

8]. Ari Juels and Paul Syverson and Dan Bailey (2005),*High-Power Proxies for Enhancing RFID privacy and Utility,* http://www.rsasecurity.com/rsalabs/staff/bios/ajuel/publication/pdfs/RFIDREP2.pdf , 26 Januari 2007, 15.23 WIB

9]. Ari Juels and Stephen A. Weis (2005), *Defining Strong Privacy for RFID,* http://www.rsasecurity.com/rsalabs/staff/bios/ajuel/publication/rfid_privacy/rfidprivacy.pdf , 26 Januari 2007, 15.23 WIB

10].United States Government Accountability Office (2005), *Informaton Security:Radio Frequency Identification Technology in the Federal Government*, http://www.gao.gov/new.items/d05551.pdf, 26 Januari 2007, 15.23 WIB

11]. Dedi Supriatna (2007), Studi Mengenai Aspek Privasi Pada Sistim RFID, Sekolah Tehnik Elektro dan Informatika Institut Teknologi Bandung

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-3
KONSENTRASI TEKNIK ELEKTRONIKA

LEMBAR ASISTENSI BIMBINGAN TUGAS AKHIR

Nama Mahasiswa : ADITYA DWIANANDA
NIM : 02.52.052/P
Masa bimbingan : 4 Januari 2007 – 4 April 2007
Judul TA : Pengaman Mobil Menggunakan RFID Card Berbasis AT89C51

No	Tanggal	Materi	Paraf
1	4 Januari	Buat rancangan alat, lengkapi	
2		dg spesifikasi komponennya	
3	10 pebruari	- buat alat ; kalibrasi penyempurnaan alat.	
4	15 pebruari.	- buat kesimpulan	
5	17 maret	- lanjutan, bab II teori2 penunjang	
6	20 maret.	- lanjutkan bab I, dan abstraksinya	
7	22 maret	Acc mengikuti ujian TA	

Malang,.....2007

Mengetahui,
Dosen Pembimbing

(Ir. Taufik Hidayat, MT)
1018700151

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-3
KONSENTRASI TEKNIK ELEKTRONIKA

LEMBAR PERBAIKAN TUGAS AKHIR

Nama Mahasiswa : Aditya Dwiananda
NIM : 02.52.052/P
Jurusan : Teknik Elektro D-III
Konsentrasi : Teknik Energi Listrik D-III
Hari / Tanggal :

No	Materi Perbaikan	Paraf
1	Resolusi RFID	
2	Abstrak	
3	BAB IV Analisa Data	
4		
5		
6		
7		

Telah Diperiksa / Disetujui

Penguji I

(Ir.Choirul Saleh, MT)

Penguji II

(Bambang Priyo Hartono, ST, MT)

Mengetahui,
Dosen Pembimbing

(Ir.H. Taufik Hidayat, MT)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO D-3
KONSENTRASI TEKNIK ELEKTRONIKA

BERITA ACARA UJIAN TUGAS AKHIR
FAKULTAS TEKNOLODI INDUSTRI

Nama : Aditya Dwiananda

NIM : 02.52.052/P

Jurusan : Teknik Elektro D-III

Konsentrasi : Teknik Energi Listrik D-III

Judul TA : PENGAMAN MOBIL MENGGUNAKAN RFID CARD BERBASIS AT89C51

Dipertahankan dihadapan Team Penguji Tugas Akhir Jenjang Diploma (D III) pada :

Hari : Kamis

Tanggal : 22 Maret 2007

Dengan Nilai : 78,25 (B+)

Panitia Ujian Tugas Akhir

(Ir. Mochtar Asroni, MSME)
Ketua

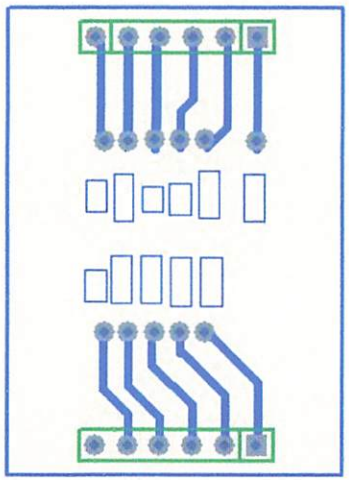
(Ir. Choirul Saleh, MT)
Sekretaris

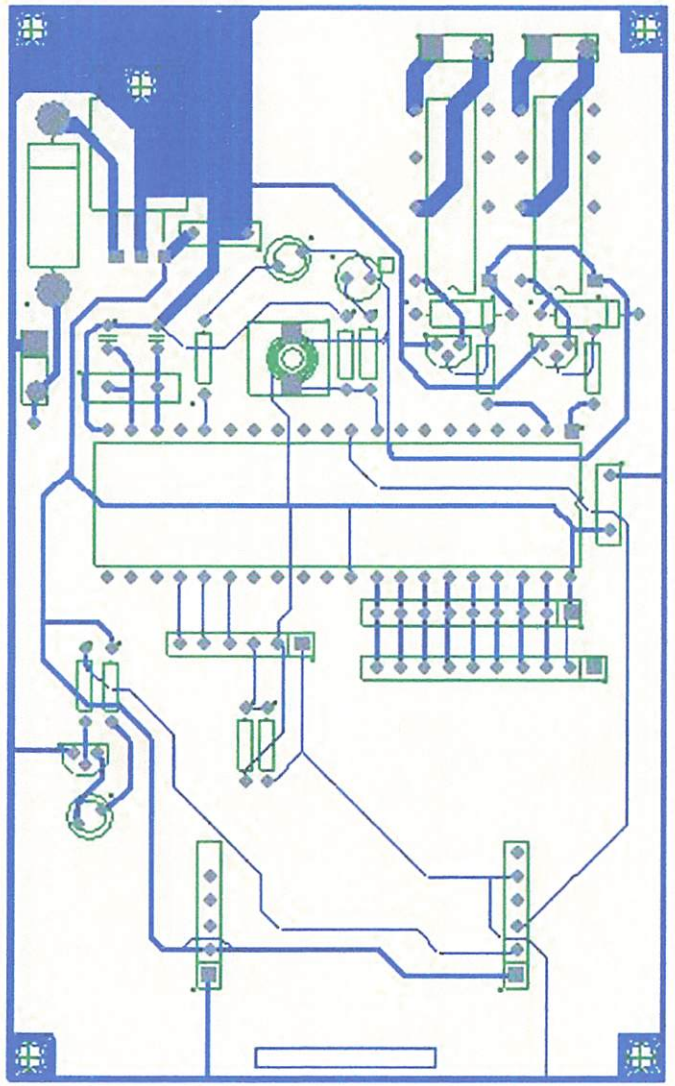
Anggota Penguji

(Ir.Choirul Saleh, MT)
Pertama

(Bambang Priyo Hartono, ST, MT)
Kedua

LAMPIRAN





```
#INCLUDE          "8051.H"
RELAY1            .EQU    P1.0
RELAY2            .EQU    P1.1
LED               .EQU    P3.6
RW_LCD            .EQU    P2.4
RS_LCD            .EQU    P2.5
E_LCD             .EQU    P2.3
PORTLCD           .EQU    P0

                  .ORG    $30
DATANOHEX1        .BLOCK  1
DATANOHEX2        .BLOCK  1
DATANOHEX3        .BLOCK  1
DATANOHEX4        .BLOCK  1
DATADECHASIL1     .BLOCK  1
DATADECHASIL2     .BLOCK  1
DATADECHASIL3     .BLOCK  1
DATADECHASIL4     .BLOCK  1
DATADECHASIL5     .BLOCK  1
DATAPENAMBAH1     .BLOCK  1
DATAPENAMBAH2     .BLOCK  1
DATAPENAMBAH3     .BLOCK  1
DATAPENAMBAH4     .BLOCK  1
DATAPENAMBAH5     .BLOCK  1
DATAS1            .BLOCK  1
DATAS2            .BLOCK  1
DATAS3            .BLOCK  1
DATAS4            .BLOCK  1
DATAS5            .BLOCK  1
DATAS6            .BLOCK  1
DATAS7            .BLOCK  1
DATAS8            .BLOCK  1
BUFDATANOMOR      .BLOCK  18
DATANOMORASCII1   .BLOCK  1
DATANOMORASCII2   .BLOCK  1
DATANOMORASCII3   .BLOCK  1
DATANOMORASCII4   .BLOCK  1
DATANOMORASCII5   .BLOCK  1
DATANOMORASCII6   .BLOCK  1
DATANOMORASCII7   .BLOCK  1
DATANOMORASCII8   .BLOCK  1
DATANOMORASCII9   .BLOCK  1
DATANOMORASCII10  .BLOCK  1
STATUSRELAY       .BLOCK  1
STBENAR           .BLOCK  1
STDATA            .BLOCK  1


                  .ORG    $0
                  LJMP    MULAI

                  .ORG    $23
                  LJMP    SERINT

                  .ORG    $100
MULAI:            MOV     SP,#$20
                  MOV     PSW,#0
                  MOV     R0,#BUFDATANOMOR
                  CLR     RELAY1
                  CLR     RELAY2
                  MOV     STATUSRELAY,#0
                  MOV     STDATA,#0
                  SETB    LED
                  LCALL   DELAY
                  LCALL   DELAY
                  LCALL   INIT_LCD
                  LCALL   INITSERIAL
                  MOV     DPTR,#TEXTAWAL1
```

```
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
                MOV     DPTR,#TEXTAWAL2
                LCALL   PROC_STRTOLCD
                LCALL   DELAY
                LCALL   DELAY

                SETB    ES
                SETB    EA

LOOP:           MOV     SP,#$20
                MOV     A,STDATA
                CJNE    A,#0,ADADATAMASUK
                LJMP    LOOP

ADADATAMASUK:   CLR     ES
                CLR     EA

                LCALL   PROC_CLEARLCD
                MOV     DPTR,#TEXTAWAL2
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
                MOV     DPTR,#TEXTAWAL3
                LCALL   PROC_STRTOLCD
                MOV     R1,#DATANOMORASCII1
                MOV     R2,#10
TAMPNO:         MOV     A,@R1
                LCALL   WRITE_DATALCD
                INC     R1
                DJNZ    R2,TAMPNO

;--------------- PROSES MEMBANDINGKAN
                MOV     STBENAR,#0
                MOV     R0,#DATANOMORASCII1
                MOV     DPTR,#NOKARTU1
                LCALL   BANDINGKARTU
                MOV     A,STBENAR
                CJNE    A,#$01,CEKKARTUKE2
                MOV     STDATA,#0
                MOV     R0,#BUFDATANOMOR
                SETB    ES
                SETB    EA
                LJMP    LOOP
CEKKARTUKE2:
                MOV     STBENAR,#0
                MOV     R0,#DATANOMORASCII1
                MOV     DPTR,#NOKARTU2
                LCALL   BANDINGKARTU
                MOV     A,STBENAR
                CJNE    A,#$01,TAMPILANERROR
                MOV     STDATA,#0
                MOV     R0,#BUFDATANOMOR
                SETB    ES
                SETB    EA
                LJMP    LOOP


TAMPILANERROR:
                LCALL   DELAY
                LCALL   PROC_HOMELCD
                LCALL   PROC_LFLCD
                MOV     DPTR,#TEXTERROR
                LCALL   PROC_STRTOLCD
                LCALL   DELAY
                LCALL   DELAY
                LCALL   PROC_HOMELCD
                MOV     DPTR,#TEXTAWAL1
                LCALL   PROC_STRTOLCD
                LCALL   PROC_LFLCD
```

```
                MOV     DPTR,#TEXTAWAL2
                LCALL   PROC_STRTOLCD
                MOV     R0,#BUFDATANOMOR
                MOV     STDATA,#0
                SETB    ES
                SETB    EA
                LJMP    LOOP

SERINT:         JBC     RI,GETDATASERIAL
                RETI

GETDATASERIAL:
                PUSH    ACC

                MOV     A,SBUF
                CLR     RI
                CJNE    A,#$03,ISIKEBUFFER
                LCALL   PROSES
                MOV     R0,#BUFDATANOMOR
                POP     ACC
                RETI

ISIKEBUFFER:    MOV     @R0,A
                INC     R0

                POP     ACC
                RETI

PROSES:         MOV     R0,#BUFDATANOMOR
                INC     R0
                INC     R0
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS1,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS2,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS3,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS4,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS5,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS6,A
                INC     R0
                MOV     A,@R0
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS7,A
                INC     R0
                MOV     A,@R0
```

```
                LCALL   CEKA_F
                ANL     A,#$0F
                MOV     DATAS8,A

;---------------PENGGABUNGAN
                MOV     A,DATAS1
                SWAP    A
                ANL     A,#$F0
                MOV     DATANOHEX1,A
                MOV     A,DATAS2
                ANL     A,#$0F
                ORL     A,DATANOHEX1
                MOV     DATANOHEX1,A

                MOV     A,DATAS3
                SWAP    A
                ANL     A,#$F0
                MOV     DATANOHEX2,A
                MOV     A,DATAS4
                ANL     A,#$0F
                ORL     A,DATANOHEX2
                MOV     DATANOHEX2,A

                MOV     A,DATAS5
                SWAP    A
                ANL     A,#$F0
                MOV     DATANOHEX3,A
                MOV     A,DATAS6
                ANL     A,#$0F
                ORL     A,DATANOHEX3
                MOV     DATANOHEX3,A

                MOV     A,DATAS7
                SWAP    A
                ANL     A,#$F0
                MOV     DATANOHEX4,A
                MOV     A,DATAS8
                ANL     A,#$0F
                ORL     A,DATANOHEX4
                MOV     DATANOHEX4,A

                LCALL   HEXTODES

                MOV     A,DATADECHASIL1
                LCALL   ANDF0
                MOV     DATANOMORASCII1,A
                MOV     A,DATADECHASIL1
                LCALL   AND0F
                MOV     DATANOMORASCII2,A

                MOV     A,DATADECHASIL2
                LCALL   ANDF0
                MOV     DATANOMORASCII3,A
                MOV     A,DATADECHASIL2
                LCALL   AND0F
                MOV     DATANOMORASCII4,A

                MOV     A,DATADECHASIL3
                LCALL   ANDF0
                MOV     DATANOMORASCII5,A
                MOV     A,DATADECHASIL3
                LCALL   AND0F
                MOV     DATANOMORASCII6,A

                MOV     A,DATADECHASIL4
                LCALL   ANDF0
                MOV     DATANOMORASCII7,A
                MOV     A,DATADECHASIL4
                LCALL   AND0F
```

```
                MOV     DATANOMORASCII8,A

                MOV     A,DATADECHASIL5
                LCALL   ANDF0
                MOV     DATANOMORASCII9,A
                MOV     A,DATADECHASIL5
                LCALL   AND0F
                MOV     DATANOMORASCII10,A

                MOV     STDATA,#1
                RET


BANDINGKARTU:
                MOV     R2,#10
ULANGCEK:
                CLR     A
                MOV     B,@R0
                MOVC    A,@A+DPTR
                CJNE    A,B,NOMORTIDAKCOCOK
                INC     R0
                INC     DPTR
                DJNZ    R2,ULANGCEK
DATABENAR:
                MOV     A,STATUSRELAY
                CJNE    A,#0,DATABENAR1
                MOV     STATUSRELAY,#1
                SETB    RELAY1
                SETB    RELAY2
                CLR     LED
                LJMP    DATABENAR2
DATABENAR1:     MOV     STATUSRELAY,#0
                CLR     RELAY1
                CLR     RELAY2
                SETB    LED
DATABENAR2:     MOV     STBENAR,#1
                RET

NOMORTIDAKCOCOK:
                MOV     STBENAR,#0

                RET

PROC_HOMELCD:
                MOV     A,#02H
                LCALL   WRITE_CTRLLCD
                RET
PROC_LFLCD:
                MOV     A,#0C0H
                LCALL   WRITE_CTRLLCD
                RET
PROC_CLEARLCD:
                MOV     A,#01H
                LCALL   WRITE_CTRLLCD
                RET

WRITE_DATALCD:
                MOV         PORTLCD,A
                CLR     RW_LCD
                SETB    RS_LCD
                CLR     E_LCD

                LCALL   DELAY_LCD

                SETB    RS_LCD
                SETB    E_LCD
                LCALL   DELAY_LCD

                SETB    RS_LCD
```

```
                CLR     E_LCD
                NOP
                CLR     RS_LCD
                CLR     E_LCD
                LCALL   DELAY_LCD
                RET

WRITE_CTRLLCD:
                MOV         PORTLCD,A
                CLR     RW_LCD
                CLR     RS_LCD
                CLR     E_LCD

                LCALL   DELAY_LCD
                CLR     RS_LCD
                SETB    E_LCD
                LCALL   DELAY_LCD
                CLR     RS_LCD
                CLR     E_LCD
                NOP
                CLR     RS_LCD
                CLR     E_LCD
                LCALL   DELAY_LCD


                RET


;-------------------------------------------------------
;       Routine Pengiriman String ke Display LCD
;-------------------------------------------------------
PROC_STRTOLCD:
                CLR     A
                MOVC    A,@A+DPTR
                CJNE    A,#00H,STRTOLCD
                RET
STRTOLCD:
                LCALL   WRITE_DATALCD
                INC     DPTR
                SJMP    PROC_STRTOLCD

;-------------------------------------------------------
;      INISIAL LCD
;-------------------------------------------------------
INIT_LCD
                MOV     A,#38H
                LCALL   WRITE_CTRLLCD
                MOV     A,#0EH
                LCALL   WRITE_CTRLLCD
                MOV     A,#0CH
                LCALL   WRITE_CTRLLCD
                MOV     A,#06H
                LCALL   WRITE_CTRLLCD
                MOV     A,#01H
                LCALL   WRITE_CTRLLCD
                RET


ANDF0:          SWAP    A
                ANL     A,#$0F
                ADD     A,#$30
                RET
AND0F:
                ANL     A,#$0F
                ADD     A,#$30
                RET
```

```
CEKA_F:
CEKA:       CJNE    A,#'A',CEKB
            MOV     A,#$0A
            ADD     A,#$30
            RET
CEKB:       CJNE    A,#'B',CEKC
            MOV     A,#$0B
            ADD     A,#$30
            RET
CEKC:       CJNE    A,#'C',CEKD
            MOV     A,#$0C
            ADD     A,#$30
            RET
CEKD:       CJNE    A,#'D',CEKE
            MOV     A,#$0D
            ADD     A,#$30
            RET
CEKE:       CJNE    A,#'E',CEKF
            MOV     A,#$0E
            ADD     A,#$30
            RET
CEKF:       CJNE    A,#'F',CEKPA
            MOV     A,#$0F
            ADD     A,#$30
CEKPA:
            RET


HEXTODES:
            MOV     DATADECHASIL1,#0
            MOV     DATADECHASIL2,#0
            MOV     DATADECHASIL3,#0
            MOV     DATADECHASIL4,#0
            MOV     DATADECHASIL5,#0

            MOV     A,DATANOHEX4
KE1:        CLR     C
            RRC     A
            JNC     KE2
            LCALL   HITBIT1
            LCALL   HITUNG
KE2:        CLR     C
            RRC     A
            JNC     KE3
            LCALL   HITBIT2
            LCALL   HITUNG
KE3:        CLR     C
            RRC     A
            JNC     KE4
            LCALL   HITBIT3
            LCALL   HITUNG
KE4:        CLR     C
            RRC     A
            JNC     KE5
            LCALL   HITBIT4
            LCALL   HITUNG
KE5:        CLR     C
            RRC     A
            JNC     KE6
            LCALL   HITBIT5
            LCALL   HITUNG
KE6:        CLR     C
            RRC     A
            JNC     KE7
            LCALL   HITBIT6
            LCALL   HITUNG
KE7:        CLR     C
```

```
        RRC     A
        JNC     KE8
        LCALL   HITBIT7
        LCALL   HITUNG
KE8:    CLR     C
        RRC     A
        JNC     KE9
        LCALL   HITBIT8
        LCALL   HITUNG
KE9:
        MOV     A,DATANOHEX3
        CLR     C
        RRC     A
        JNC     KE10
        LCALL   HITBIT9
        LCALL   HITUNG
KE10:   CLR     C
        RRC     A
        JNC     KE11
        LCALL   HITBIT10
        LCALL   HITUNG
KE11:   CLR     C
        RRC     A
        JNC     KE12
        LCALL   HITBIT11
        LCALL   HITUNG
KE12:   CLR     C
        RRC     A
        JNC     KE13
        LCALL   HITBIT12
        LCALL   HITUNG
KE13:   CLR     C
        RRC     A
        JNC     KE14
        LCALL   HITBIT13
        LCALL   HITUNG
KE14:   CLR     C
        RRC     A
        JNC     KE15
        LCALL   HITBIT14
        LCALL   HITUNG
KE15:   CLR     C
        RRC     A
        JNC     KE16
        LCALL   HITBIT15
        LCALL   HITUNG
KE16:   CLR     C
        RRC     A
        JNC     KE17
        LCALL   HITBIT16
        LCALL   HITUNG
KE17:
        MOV     A,DATANOHEX2
        CLR     C
        RRC     A
        JNC     KE18
        LCALL   HITBIT17
        LCALL   HITUNG
KE18:   CLR     C
        RRC     A
        JNC     KE19
        LCALL   HITBIT18
        LCALL   HITUNG
KE19:   CLR     C
        RRC     A
        JNC     KE20
        LCALL   HITBIT19
        LCALL   HITUNG
KE20:   CLR     C
```

```
        RRC     A
        JNC     KE21
        LCALL   HITBIT20
        LCALL   HITUNG
KE21:   CLR     C
        RRC     A
        JNC     KE22
        LCALL   HITBIT21
        LCALL   HITUNG
KE22:   CLR     C
        RRC     A
        JNC     KE23
        LCALL   HITBIT22
        LCALL   HITUNG
KE23:   CLR     C
        RRC     A
        JNC     KE24
        LCALL   HITBIT23
        LCALL   HITUNG
KE24:   CLR     C
        RRC     A
        JNC     KE25
        LCALL   HITBIT24
        LCALL   HITUNG
KE25:
        MOV     A,DATANOHEX1
        CLR     C
        RRC     A
        JNC     KE26
        LCALL   HITBIT25
        LCALL   HITUNG
KE26:   CLR     C
        RRC     A
        JNC     KE27
        LCALL   HITBIT26
        LCALL   HITUNG
KE27:   CLR     C
        RRC     A
        JNC     KE28
        LCALL   HITBIT27
        LCALL   HITUNG
KE28:   CLR     C
        RRC     A
        JNC     KE29
        LCALL   HITBIT28
        LCALL   HITUNG
KE29:   CLR     C
        RRC     A
        JNC     KE30
        LCALL   HITBIT29
        LCALL   HITUNG
KE30:   CLR     C
        RRC     A
        JNC     KE31
        LCALL   HITBIT30
        LCALL   HITUNG
KE31:   CLR     C
        RRC     A
        JNC     KE32
        LCALL   HITBIT31
        LCALL   HITUNG
KE32:   CLR     C
        RRC     A
        JNC     KE33
        LCALL   HITBIT32
        LCALL   HITUNG
KE33:
```

```
                RET

HITUNG:         PUSH    ACC
                MOV     A,DATADECHASIL5
                ADD     A,DATAPENAMBAH5
                DA      A
                MOV     DATADECHASIL5,A

                MOV     A,DATADECHASIL4
                ADDC    A,DATAPENAMBAH4
                DA      A
                MOV     DATADECHASIL4,A

                MOV     A,DATADECHASIL3
                ADDC    A,DATAPENAMBAH3
                DA      A
                MOV     DATADECHASIL3,A

                MOV     A,DATADECHASIL2
                ADDC    A,DATAPENAMBAH2
                DA      A
                MOV     DATADECHASIL2,A

                MOV     A,DATADECHASIL1
                ADDC    A,DATAPENAMBAH1
                DA      A
                MOV     DATADECHASIL1,A
                POP     ACC
                RET

;---------------------------------------
HITBIT1:
                MOV     DATAPENAMBAH5,#$01
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT2:
                MOV     DATAPENAMBAH5,#$02
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT3:
                MOV     DATAPENAMBAH5,#$04
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT4:
                MOV     DATAPENAMBAH5,#$08
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT5:
                MOV     DATAPENAMBAH5,#$16
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
```

```
HITBIT6:
                MOV     DATAPENAMBAH5,#$32
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT7:
                MOV     DATAPENAMBAH5,#$64
                MOV     DATAPENAMBAH4,#$00
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT8:
                MOV     DATAPENAMBAH5,#$28
                MOV     DATAPENAMBAH4,#$01
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
;---------------------------------------
HITBIT9:
                MOV     DATAPENAMBAH5,#$56
                MOV     DATAPENAMBAH4,#$02
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT10:
                MOV     DATAPENAMBAH5,#$12
                MOV     DATAPENAMBAH4,#$05
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT11:
                MOV     DATAPENAMBAH5,#$24
                MOV     DATAPENAMBAH4,#$10
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT12:
                MOV     DATAPENAMBAH5,#$48
                MOV     DATAPENAMBAH4,#$20
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT13:
                MOV     DATAPENAMBAH5,#$96
                MOV     DATAPENAMBAH4,#$40
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT14:
                MOV     DATAPENAMBAH5,#$92
                MOV     DATAPENAMBAH4,#$81
                MOV     DATAPENAMBAH3,#$00
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT15:
                MOV     DATAPENAMBAH5,#$84
                MOV     DATAPENAMBAH4,#$63
```

```
                MOV     DATAPENAMBAH3,#$01
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT16:
                MOV     DATAPENAMBAH5,#$68
                MOV     DATAPENAMBAH4,#$27
                MOV     DATAPENAMBAH3,#$03
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
;---------------------------------------
HITBIT17:
                MOV     DATAPENAMBAH5,#$36
                MOV     DATAPENAMBAH4,#$55
                MOV     DATAPENAMBAH3,#$06
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT18:
                MOV     DATAPENAMBAH5,#$72
                MOV     DATAPENAMBAH4,#$10
                MOV     DATAPENAMBAH3,#$13
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT19:
                MOV     DATAPENAMBAH5,#$44
                MOV     DATAPENAMBAH4,#$21
                MOV     DATAPENAMBAH3,#$26
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT20:
                MOV     DATAPENAMBAH5,#$88
                MOV     DATAPENAMBAH4,#$42
                MOV     DATAPENAMBAH3,#$52
                MOV     DATAPENAMBAH2,#$00
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT21:
                MOV     DATAPENAMBAH5,#$76
                MOV     DATAPENAMBAH4,#$85
                MOV     DATAPENAMBAH3,#$04
                MOV     DATAPENAMBAH2,#$01
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT22:
                MOV     DATAPENAMBAH5,#$52
                MOV     DATAPENAMBAH4,#$71
                MOV     DATAPENAMBAH3,#$09
                MOV     DATAPENAMBAH2,#$02
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT23:
                MOV     DATAPENAMBAH5,#$04
                MOV     DATAPENAMBAH4,#$43
                MOV     DATAPENAMBAH3,#$19
                MOV     DATAPENAMBAH2,#$04
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT24:
                MOV     DATAPENAMBAH5,#$08
                MOV     DATAPENAMBAH4,#$86
                MOV     DATAPENAMBAH3,#$38
                MOV     DATAPENAMBAH2,#$08
```

```
                MOV     DATAPENAMBAH1,#$00
                RET
;----------------------------------------
HITBIT25:
                MOV     DATAPENAMBAH5,#$16
                MOV     DATAPENAMBAH4,#$72
                MOV     DATAPENAMBAH3,#$77
                MOV     DATAPENAMBAH2,#$16
                MOV     DATAPENAMBAH1,#$00
                RET

HITBIT26:
                MOV     DATAPENAMBAH5,#$32
                MOV     DATAPENAMBAH4,#$44
                MOV     DATAPENAMBAH3,#$55
                MOV     DATAPENAMBAH2,#$33
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT27:
                MOV     DATAPENAMBAH5,#$64
                MOV     DATAPENAMBAH4,#$88
                MOV     DATAPENAMBAH3,#$10
                MOV     DATAPENAMBAH2,#$67
                MOV     DATAPENAMBAH1,#$00
                RET
HITBIT28:
                MOV     DATAPENAMBAH5,#$28
                MOV     DATAPENAMBAH4,#$77
                MOV     DATAPENAMBAH3,#$21
                MOV     DATAPENAMBAH2,#$34
                MOV     DATAPENAMBAH1,#$01
                RET
HITBIT29:
                MOV     DATAPENAMBAH5,#$56
                MOV     DATAPENAMBAH4,#$54
                MOV     DATAPENAMBAH3,#$43
                MOV     DATAPENAMBAH2,#$68
                MOV     DATAPENAMBAH1,#$02
                RET
HITBIT30:
                MOV     DATAPENAMBAH5,#$12
                MOV     DATAPENAMBAH4,#$09
                MOV     DATAPENAMBAH3,#$87
                MOV     DATAPENAMBAH2,#$36
                MOV     DATAPENAMBAH1,#$05
                RET
HITBIT31:
                MOV     DATAPENAMBAH5,#$24
                MOV     DATAPENAMBAH4,#$18
                MOV     DATAPENAMBAH3,#$74
                MOV     DATAPENAMBAH2,#$73
                MOV     DATAPENAMBAH1,#$10
                RET

HITBIT32:
                MOV     DATAPENAMBAH5,#$48
                MOV     DATAPENAMBAH4,#$36
                MOV     DATAPENAMBAH3,#$48
                MOV     DATAPENAMBAH2,#$47
                MOV     DATAPENAMBAH1,#$21
                RET
;----------------------------------------

INITSERIAL:
            MOV     TMOD,#20H
            MOV     TCON,#41H
            MOV     TH1,#0FDH
            MOV     SCON,#50H
            RET
```

```
DELAY_LCD:
                MOV     R7,#04H
D_LCD1:         MOV     R6,#3FH         ;4F
D_LCD2:         DJNZ    R6,D_LCD2
                DJNZ    R7,D_LCD1
                RET
DELAY:          MOV     R5,#$03
DELAY1:         MOV     R6,#$FF
DELAY2:         MOV     R7,#$FF
DELAY3:         DJNZ    R7,DELAY3
                DJNZ    R6,DELAY2
                DJNZ    R5,DELAY1
                RET

TEXTAWAL1:      .BYTE   "   TUGAS AKHIR  ",0
TEXTAWAL2:      .BYTE   "   ADIT - RFID  ",0
TEXTAWAL3:      .BYTE   "NOMOR:",0
TEXTERROR:      .BYTE   "  TIDAK DIKENAL ",0
TEXTBLANK:      .BYTE   "                ",0
NOKARTU1:       .BYTE   "1644377311",0
NOKARTU2:       .BYTE   "1644370924",0
                .END
```

IS89C51

CMOS SINGLE CHIP 8-BIT MICROCONTROLLER with 4-Kbytes of FLASH

NOVEMBER 1998

FEATURES

- 80C51 based architecture
- 4-Kbytes of on-chip Reprogrammable Flash Memory
- 128 x 8 RAM
- Two 16-bit Timer/Counters
- Full duplex serial channel
- Boolean processor
- Four 8-bit I/O ports, 32 I/O lines
- Memory addressing capability
  – 64K ROM and 64K RAM
- Program memory lock
  – Lock bits (3)
- Power save modes:
  – Idle and power-down
- Six interrupt sources
- Most instructions execute in 0.3 µs
- CMOS and TTL compatible
- Maximum speed: 40 MHz @ Vcc = 5V
- Industrial temperature available
- Packages available:
  – 40-pin DIP
  – 44-pin PLCC
  – 44-pin PQFP

GENERAL DESCRIPTION

The *ISSI* IS89C51 is a high-performance microcontroller fabricated using high-density CMOS technology. The CMOS IS89C51 is functionally compatible with the industry standard 80C51 microcontrollers.

The IS89C51 is designed with 4-Kbytes of Flash memory, 128 x 8 RAM; 32 programmable I/O lines; a serial I/O port for either multiprocessor communications, I/O expansion or full duplex UART; two 16-bit timer/counters; an six-source, two-priority-level, nested interrupt structure; and an on-chip oscillator and clock circuit. The IS89C51 can be expanded using standard TTL compatible memory.

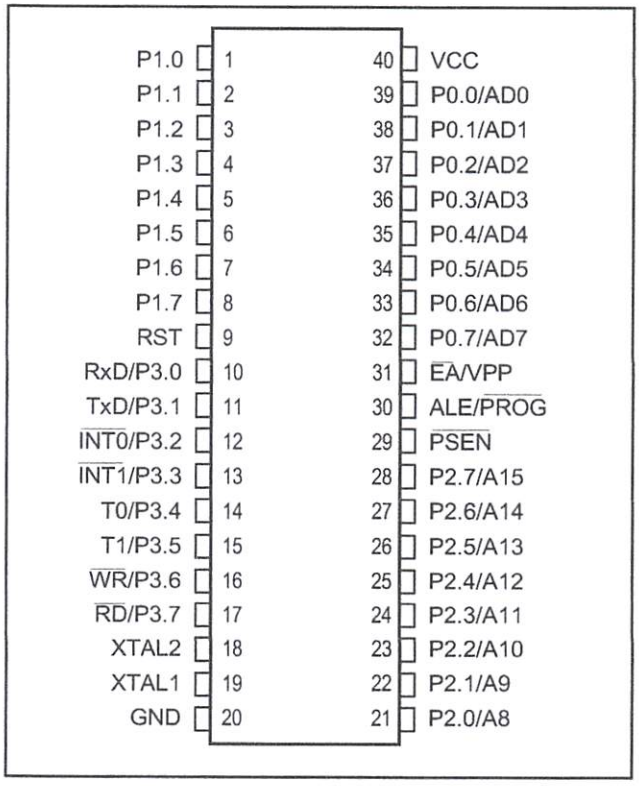


Figure 1. IS89C51 Pin Configuration: 40-pin PDIP

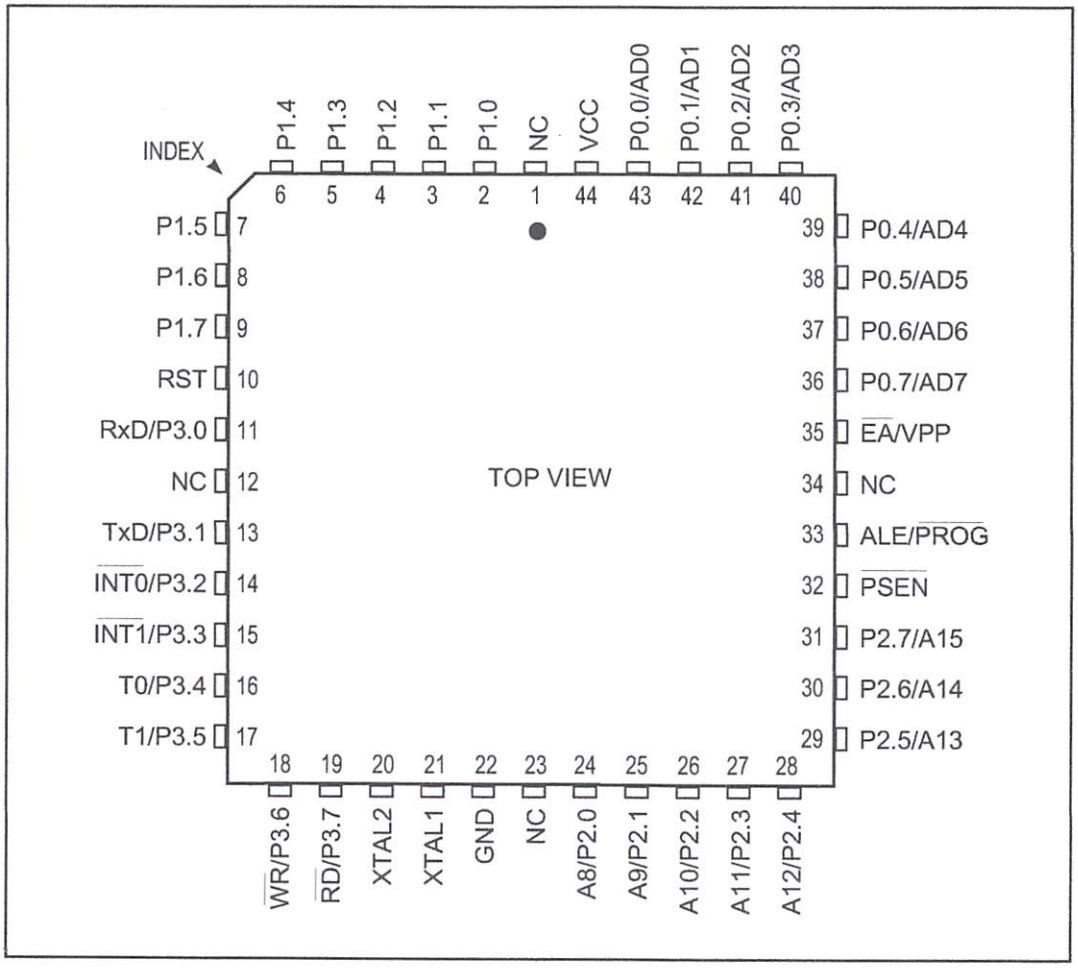


Figure 2. IS89C51 Pin Configuration: 44-pin PLCC

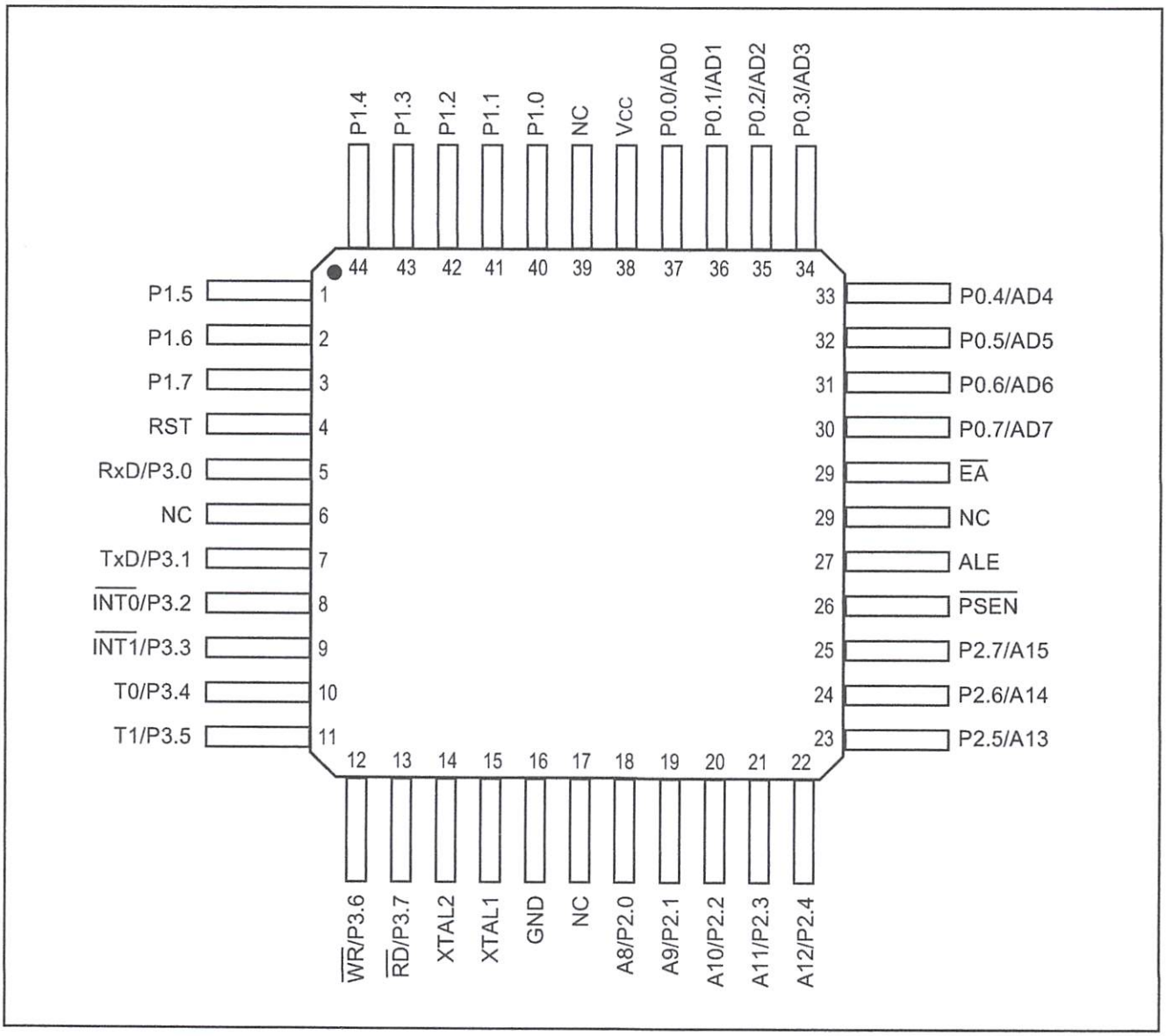


Figure 3. IS89C51 Pin Configuration: 44-pin PQFP

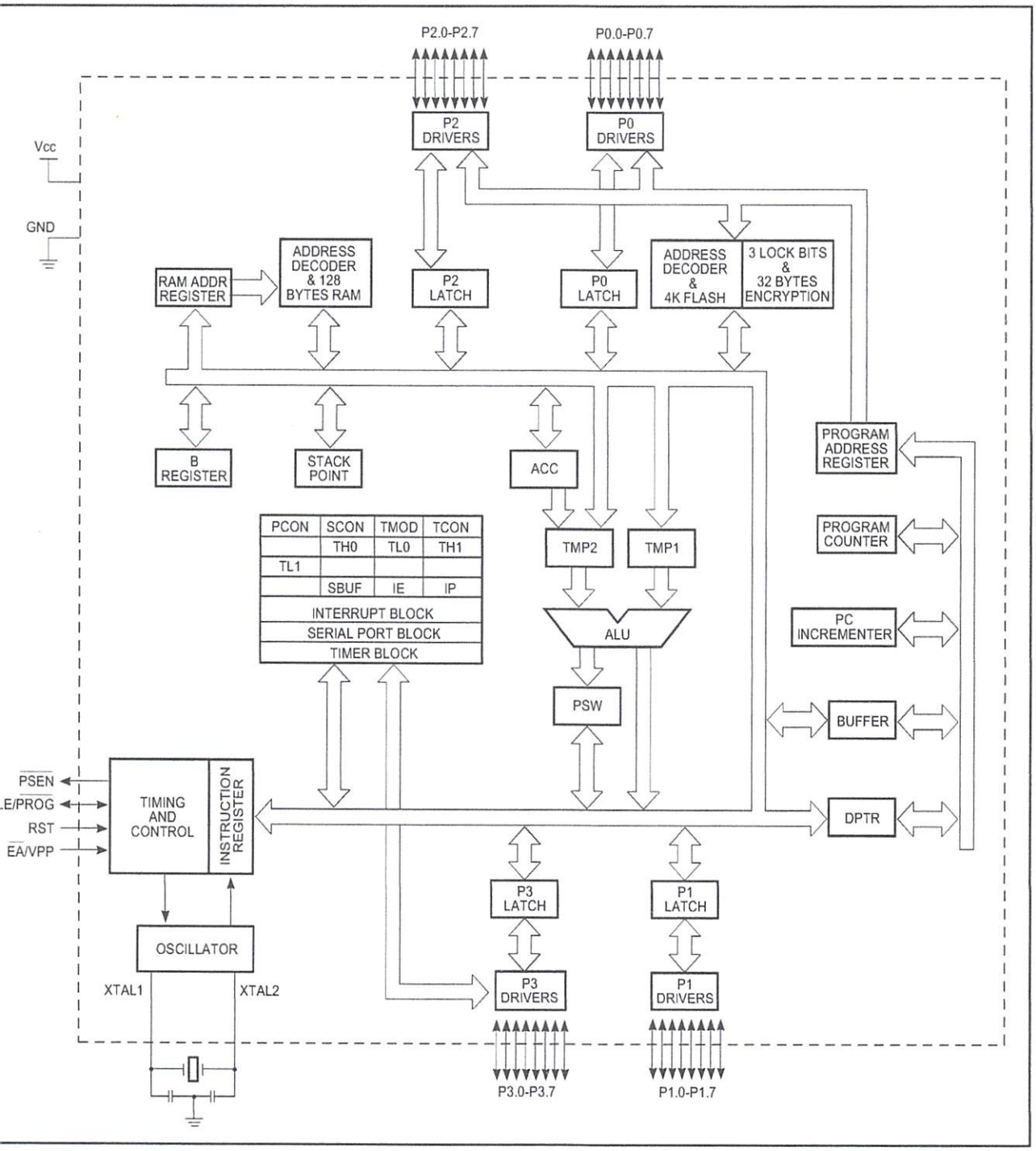


Figure 4. IS89C51 Block Diagram

Table 1. Detailed Pin Description

Symbol	PDIP	PLCC	PQFP	I/O	Name and Function
ALE/$\overline{PROG}$	30	33	27	I/O	**Address Latch Enable:** Output pulse for latching the low byte of the address during an address to the external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the Program Pulse input ($\overline{PROG}$) during Flash programming.
$\overline{EA}$/V_{PP}	31	35	29	I	**External Access enable:** $\overline{EA}$ must be externally held low to enable the device to fetch code from external program memory locations 0000H to FFFFH. If $\overline{EA}$ is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFH. This also receives the 12V programming enable voltage (V_{PP}) during Flash programming.
P0.0-P0.7	39-32	43-36	37-30	I/O	**Port 0:** Port 0 is an 8-bit open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pullups when emitting 1s. Port 0 also receives the code bytes during programmable memory programming and outputs the code bytes during program verification. External pullups are required during program verification.
P1.0-P1.7	1-8	2-9	40-44 1-3	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current because of the internal pullups. (See DC Characteristics: I_{IL}). The Port 1 output buffers can sink/source four TTL inputs. Port 1 also receives the low-order address byte during Flash programming and verification.
P2.0-P2.7	21-28	24-31	18-25	I/O	**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current because of the internal pullups. (See DC Characteristics: I_{IL}). Port 2 emits the high order address byte during fetches from external program memory and during accesses to external data memory that used 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ Ri [i = 0, 1]), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order bits and some control signals during Flash programming and verification. P2.6 and P2.7 are the control signals while the chip programs and erases.

ble 1. Detailed Pin Description *(continued)*

mbol	PDIP	PLCC	PQFP	I/O	Name and Function
0-P3.7	10-17	11, 13-19	5, 7-13	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally pulled low will source current because of the internal pullups. (See DC Characteristics: I_{IL}). Port 3 also serves the special features of the IS89C51, as listed below:
	10	11	5	I	**RxD (P3.0):** Serial input port.
	11	13	7	O	**TxD (P3.1):** Serial output port.
	12	14	8	I	**$\overline{INT0}$ (P3.2):** External interrupt 0.
	13	15	9	I	**$\overline{INT1}$ (P3.3):** External interrupt 1.
	14	16	10	I	**T0 (P3.4):** Timer 0 external input.
	15	17	11	I	**T1 (P3.5):** Timer 1 external input.
	16	18	12	O	**$\overline{WR}$ (P3.6):** External data memory write strobe.
	17	19	13	O	**$\overline{RD}$ (P3.7):** External data memory read strobe.
$\overline{EN}$	29	32	26	O	**Program Store Enable:** The read strobe to external program memory. When the device is executing code from the 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. $\overline{PSEN}$ is not activated during fetches from internal program memory.
T	9	10	4	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal MOS resistor to GND permits a power-on reset using only an external capacitor connected to Vcc.
AL 1	19	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
AL 2	18	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier.
D	20	22	16	I	**Ground:** 0V reference.
	40	44	38	I	**Power Supply:** This is the power supply voltage for operation.

OPERATING DESCRIPTION

The detail description of the IS89C51 included in this description are:

- **Memory Map and Registers**
- **Timer/Counters**
- **Serial Interface**
- **Interrupt System**
- **Other Information**
- **Flash Memory**

MEMORY MAP AND REGISTERS

Memory

The IS89C51 has separate address spaces for program and data memory. The program and data memory can be up to 64K bytes long. The lower 4K program memory can reside on-chip. Figure 5 shows a map of the IS89C51 program and data memory.

The IS89C51 has 128 bytes of on-chip RAM, plus numbers of special function registers. The lower 128 bytes can be accessed either by direct addressing or by indirect addressing. Figure 6 shows internal data memory organization and SFR Memory Map.

The lower 128 bytes of RAM can be divided into three segments as listed below and shown in Figure 7.

1. ***Register Banks 0-3:*** locations 00H through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers R0-R7. Reset initializes the stack point to location 07H, and is incremented once to start from 08H, which is the first register of the second register bank.
2. ***Bit Addressable Area:*** 16 bytes have been assigned for this segment 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH). Each of the 16 bytes in this segment can also be addressed as a byte.
3. ***Scratch Pad Area:*** 30H-7FH are available to the user as data RAM. However, if the data pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

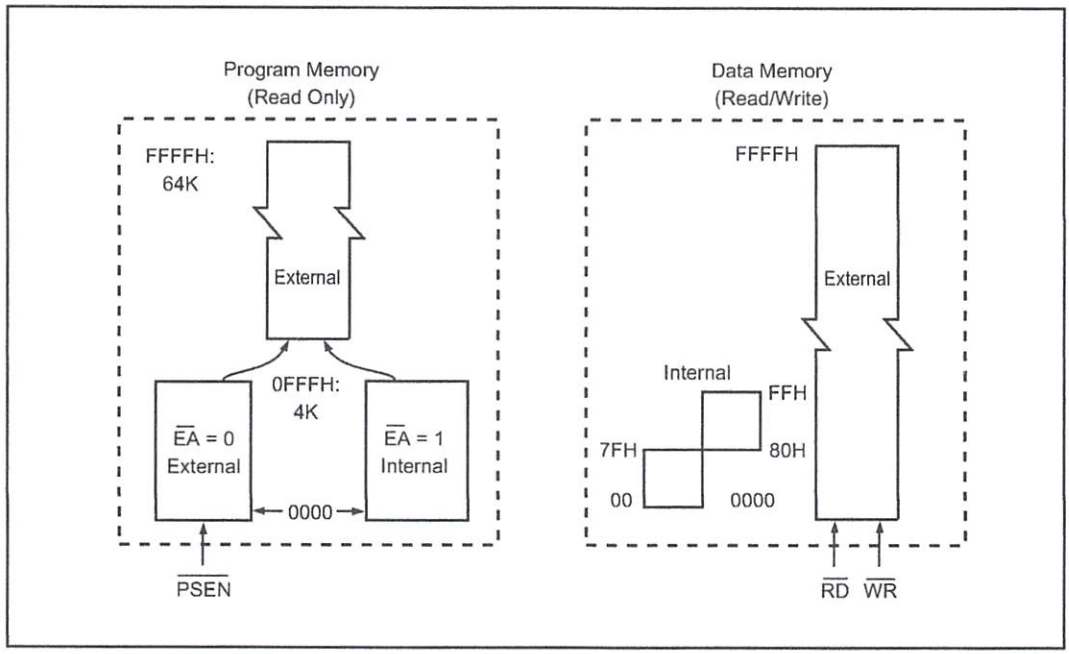


Figure 5. IS89C51 Program and Data Memory Structure

ECIAL FUNCTION REGISTERS

Special Function Registers (SFR's) are located in per 128 Bytes direct addressing area. The SFR Memory p in Figure 6 shows that.

all of the addresses are occupied. Unoccupied dresses are not implemented on the chip. Read accesses hese addresses in general return random data, and e accesses have no effect.

er software should not write 1s to these unimplemented ations, since they may be used in future microcontrollers nvoke new features. In that case, the reset or inactive ues of the new bits will always be 0, and their active ues will be 1.

The functions of the SFRs are outlined in the following sections, and detailed in Table 2.

Accumulator (ACC)
ACC is the Accumulator register. The mnemonics for Accumulator-specific instructions, however, refer to the Accumulator simply as A.

B Register (B)
The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Program Status Word (PSW). The PSW register contains program status information.

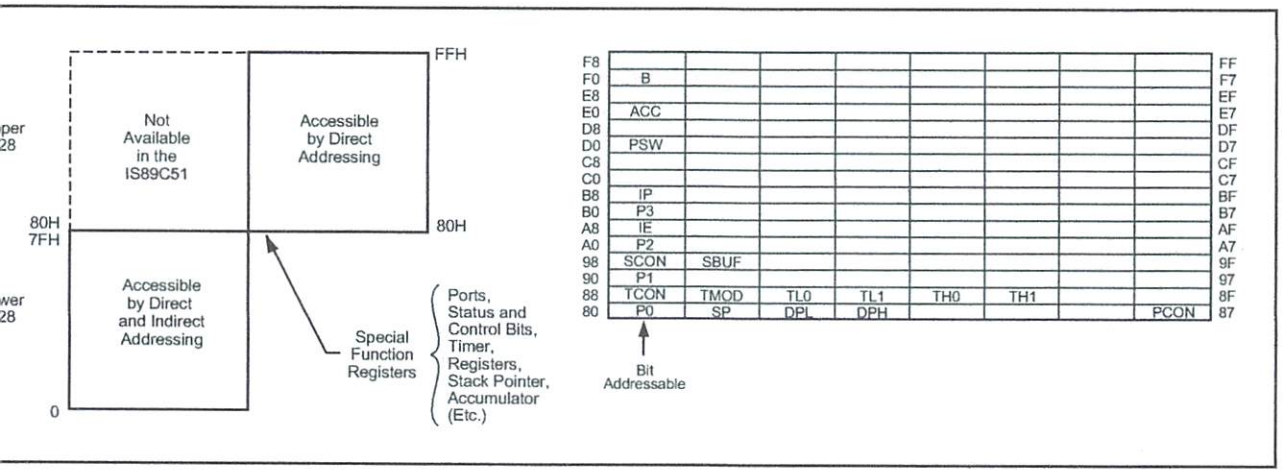


Figure 6. Internal Data Memory and SFR Memory Map

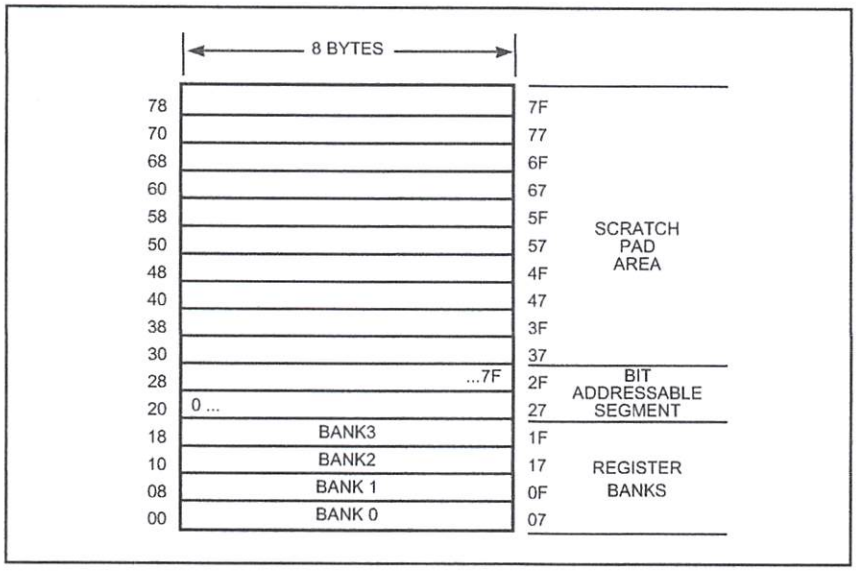


Figure 7. Lower 128 Bytes of Internal RAM

SPECIAL FUNCTION REGISTERS

(Continued)

Stack Pointer (SP)
The Stack Pointer Register is eight bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data Pointer (DPTR)
The Data Pointer consists of a high byte (DPH) and a low byte (DPL). Its function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 To 3
P0, P1, P2, and P3 are the SFR latches of Ports 0, 1, 2, and 3, respectively.

Serial Data Buffer (SBUF)
The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer, where it is held for serial transmission. (Moving a byte to SBUF initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

Timer Registers
Register pairs (TH0, TL0) and (TH1, TL1) are the 16-bit Counter registers for Timer/Counters 0 and 1, respectively.

Control Registers
Special Function Registers IP, IE, TMOD, TCON, SCON, and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. They are described in later sections of this chapter.

le 2. Special Function Registers

ymbol	Description	Direct Address	Bit Address, Symbol, or Alternative Port Function								Reset Value
CC(1)	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
(1)	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
PH	Data pointer (DPTR) high	83H									00H
PL	Data pointer (DPTR) low	82H									00H
E(1)	Interrupt enable	A8H	**AF** EA	**AE** —	**AD** —	**AC** ES	**AB** ET1	**AA** EX1	**A9** ET0	**A8** EX0	0XX00000B
P(1)	Interrupt priority	B8H	**BF** —	**BE** —	**BD** —	**BC** PS	**BB** PT1	**BA** PX1	**B9** PT0	**B8** PX0	XXX00000B
P0(1)	Port 0	80H	**87** P0.7 AD7	**86** P0.6 AD6	**85** P0.5 AD5	**84** P0.4 AD4	**83** P0.3 AD3	**82** P0.2 AD2	**81** P0.1 AD1	**80** P0.0 AD0	FFH
P1(1)	Port 1	90H	**97** P1.7	**96** P1.6	**95** P1.5	**94** P1.4	**93** P1.3	**92** P1.2	**91** P1.1	**90** P1.0	FFH
P2(1)	Port 2	A0H	**A7** P2.7 AD15	**A6** P2.6 AD14	**A5** P2.5 AD13	**A4** P2.4 AD12	**A3** P2.3 AD11	**A2** P2.2 AD10	**A1** P2.1 AD9	**A0** P2.0 AD8	FFH
P3(1)	Port 3	B0H	**B7** P3.7 $\overline{RD}$	**B6** P3.6 $\overline{WR}$	**B5** P3.5 T1	**B4** P3.4 T0	**B3** P3.3 $\overline{INT1}$	**B2** P3.2 $\overline{INT0}$	**B1** P3.1 TXD	**B0** P3.0 RXD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0XXX0000B
PSW(1)	Program status word	D0H	**D7** CY	**D6** AC	**D5** F0	**D4** RS1	**D3** RS0	**D2** OV	**D1** —	**D0** P	00H
BUF	Serial data buffer	99H									XXXXXXXXB
CON(1)	Serial controller	98H	**9F** SM0	**9E** SM1	**9D** SM2	**9C** REN	**9B** TB8	**9A** RB8	**99** TI	**98** RI	00H
P	Stack pointer	81H									07H
CON(1)	Timer control	88H	**8F** TF1	**8E** TR1	**8D** TF0	**8C** TR0	**8B** IE1	**8A** IT1	**89** IE0	**88** IT0	00H
MOD	Timer mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H
H0	Timer high 0	8CH									00H
H1	Timer high 1	8DH									00H
L0	Timer low 0	8AH									00H
L1	Timer low 1	8BH									00H

lotes:
1. Denotes bit addressable.

The detail description of each bit is as follows:

PSW:

Program Status Word. Bit Addressable.

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	—	P

Register Description:

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register bank selector bit 1.(1)
RS0	PSW.3	Register bank selector bit 0.(1)
OV	PSW.2	Overflow flag.
—	PSW.1	Usable as a general purpose flag
P	PSW.0	Parity flag. Set/Clear by hardware each instruction cycle to indicate an odd/even number of "1" bits in the accumulator.

Note:

1. The value presented by RS0 and RS1 selects the corresponding register bank.

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

PCON:

Power Control Register. Not Bit Addressable.

7	6	5	4	3	2	1	0
SMOD	—	—	—	GF1	GF0	PD	IDL

Register	Description:
SMOD	Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD=1, the baud rate is doubled when the serial port is used in modes 1, 2, or 3.
—	Not implemented, reserve for future use.(1)
—	Not implemented, reserve for future use.(1)
—	Not implemented, reserve for future use.(1)
GF1	General purpose flag bit.
GF0	General purpose flag bit.
PD	Power-down bit. Setting this bit activates power-down mode.
IDL	Idle mode bit. Setting this bit activates idle mode. If 1s are written to PD and IDL at the same time, PD takes precedence.

Note:

1. User software should not write 1s to reserved bits. These bits may be used in future products to invoke new features.

IE:

Interrupt Enable Register. Bit Addressable.

7	6	5	4	3	2	1	0
EA	—	—	ES	ET1	EX1	ET0	EX0

Register Description:

EA	IE.7	Disable all interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserve for future use.(5)
—	IE.5	Not implemented, reserve for future use.(5)
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

Note: To use any of the interrupts in the 80C51 Family, the following three steps must be taken:

1. Set the EA (enable all) bit in the IE register to 1.
2. Set the coresponding individual interrupt enable bit in the IE register to 1.
3. Begin the interrupt service routine at the corresponding Vector Address of that interrupt (see below).

Interrupt Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H

4. In addition, for external interrupts, pins $\overline{INT0}$ and $\overline{INT1}$ (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 0 or 1.
   ITX = 0 level activated (X = 0, 1)
   ITX = 1 transition activated
5. User software should not write 1s to reserved bits. These bits may be used in future products to invoke new features.

errupt Priority Register. Bit Addressable.

	6	5	4	3	2	1	0
	—	—	PS	PT1	PX1	PT0	PX0

gister Description:

	IP.7	Not implemented, reserve for future use(3)
	IP.6	Not implemented, reserve for future use(3)
	IP.5	Not implemented, reserve for future use(3)
S	IP.4	Defines Serial Port interrupt priority level
T1	IP.3	Defines Timer 1 interrupt priority level
X1	IP.2	Defines External Interrupt 1 priority level
T0	IP.1	Defines Timer 0 interrupt priority level
X0	IP.0	Defines External Interrupt 0 priority level

Notes:

1. In order to assign higher priority to an interrupt the coresponding bit in the IP register must be set to 1. While an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.
2. Priority within level is only to resolve simultaneous requests of the same priority level. From high-to-low, interrupt sources are listed below:
   IE0
   TF0
   IE1
   TF1
   RI or TI
3. User software should not write 1s to reserved bits. These bits may be used in future products to invoke new features.

TCON:

Timer/Counter Control Register. Bit Addressable

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Register Description:

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/Cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR0	TCON.4	Timer 0 run control bit. Set/Cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when the External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/Cleared by software specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when the External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/Cleared by software specify falling edge/low level triggered External Interrupt.

TMOD:

Timer/Counter Mode Control Register.
Not Bit Addressable.

Timer 1				Timer 0			
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

GATE	When TRx (in TCON) is set and GATE=1, TIMER/COUNTERx will run only while $\overline{INTx}$ pin is high (hardware control). When GATE=0, TIMER/COUNTERx will run only while TRx=1 (software control).
C/$\overline{T}$	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit.(1)
M0	Mode selector bit.(1)

Note 1:

M1	M0	Operating Mode
0	0	Mode 0. (13-bit Timer)
0	1	Mode 1. (16-bit Timer/Counter)
1	0	Mode 2. (8-bit auto-load Timer/Counter)
1	1	Mode 3. (Splits Timer 0 into TL0 and TH0. TL0 is an 8-bit Timer/Counter controller by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.)
1	1	Mode 3. (Timer/Counter 1 stopped).

SCON:

Serial Port Control Register. Bit Addressable.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Register Description:

SM0	SCON.7	Serial port mode specifier.(1)
SM1	SCON.6	Serial port mode specifier.(1)
SM2	SCON.5	Enable the multiprocessor communication feature in mode 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2=1 then RI will not be activated if valid stop bit was not received. In mode 0, SM2 should be 0.
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in mode 2 and 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 and 3, RB8 is the 9th data bit that was received. In mode 1, if SM2=0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the eighth bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the eighth bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

Note 1:

SM0	SM1	MODE	Description	Baud Rate
0	0	0	Shift register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fosc/64 or Fosc/32
1	1	3	9-bit UART	Variable

IER/COUNTERS

IS89C51 has two 16-bit Timer/Counter registers: er 0 and Timer 1. All two can be configured to operate er as Timers or event Counters.

Timer, the register is incremented every machine cycle. s, the register counts machine cycles. Since a machine e consists of 12 oscillator periods, the count rate is 1/12 ne oscillator frequency.

a Counter, the register is incremented in response to a -0 transition at its corresponding external input pin, T0 T1. The external input is sampled during S5P2 of every chine cycle. When the samples show a high in one cycle a low in the next cycle, the count is incremented. The count value appears in the register during S3P1 of the e following the one in which the transition was detected. ce two machine cycles (24 oscillator periods) are required ecognize a 1-to-0 transition, the maximum count rate is 4 of the oscillator frequency. There are no restrictions on duty cycle of the external input signal, but it should be l for at least one full machine cycle to ensure that a given el is sampled at least once before it changes.

ddition to the Timer or Counter functions, Timer 0 and er 1 have four operating modes: 13-bit timer, 16-bit timer, t auto-reload, split timer.

Timer 0 and Timer 1

The Timer or Counter function is selected by control bits C/$\overline{T}$ in the Special Function Regiser TMOD. These two Timer/Counters have four operating modes, which are selected by bit pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters, but Mode 3 is different. The four modes are described in the following sections.

Mode 0:

Both Timers in Mode 0 are 8-bit Counters with a divide-by-32 prescaler. Figure 8 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements. TR1 is a control bit in the Special Function Register TCON. Gate is in TMOD.

The 13-bit register consists of all eight bits of TH1 and the lower five bits of TL1. The upper three bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1, except that TR0, TF0 and $\overline{INT0}$ replace the corresponding Timer 1 signals in Figure 8. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

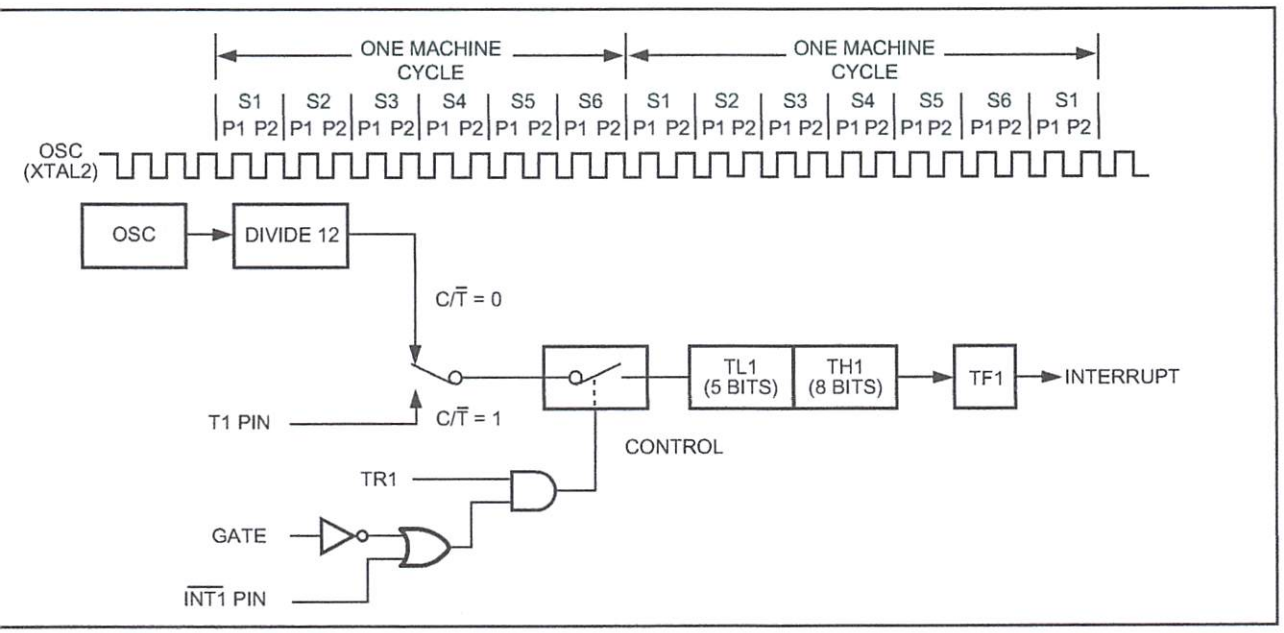


Figure 8. Timer/Counter 1 Mode 0: 13-Bit Counter

Mode 1:
Mode 1 is the same as Mode 0, except that the Timer register is run with all 16 bits. The clock is applied to the combined high and low timer registers (TL1/TH1). As clock pulses are received, the timer counts up: 0000H, 0001H, 0002H, etc. An overflow occurs on the FFFFH-to-0000H overflow flag. The timer continues to count. The overflow flag is the TF1 bit in TCON that is read or written by software (see Figure 9).

Mode 2:
Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 10. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves the TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

Mode 3:
Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 11. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt.

Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the IS89C51 can appear to have three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.

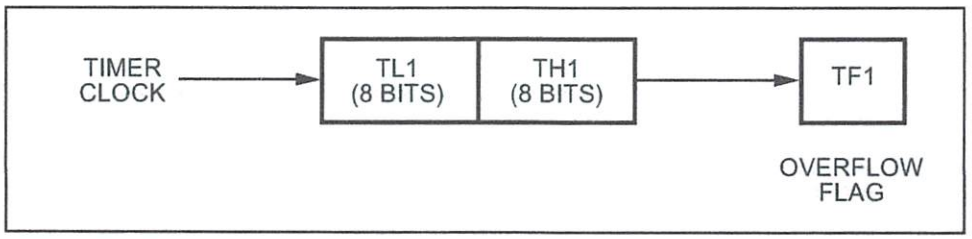


Figure 9. Timer/Counter 1 Mode 1: 16-Bit Counter

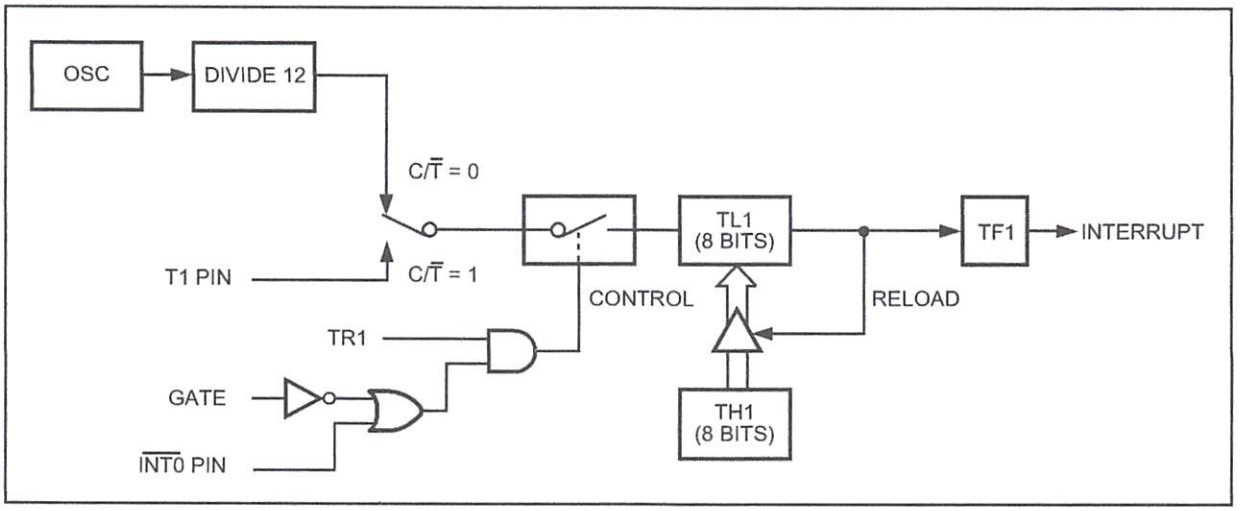


Figure 10. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

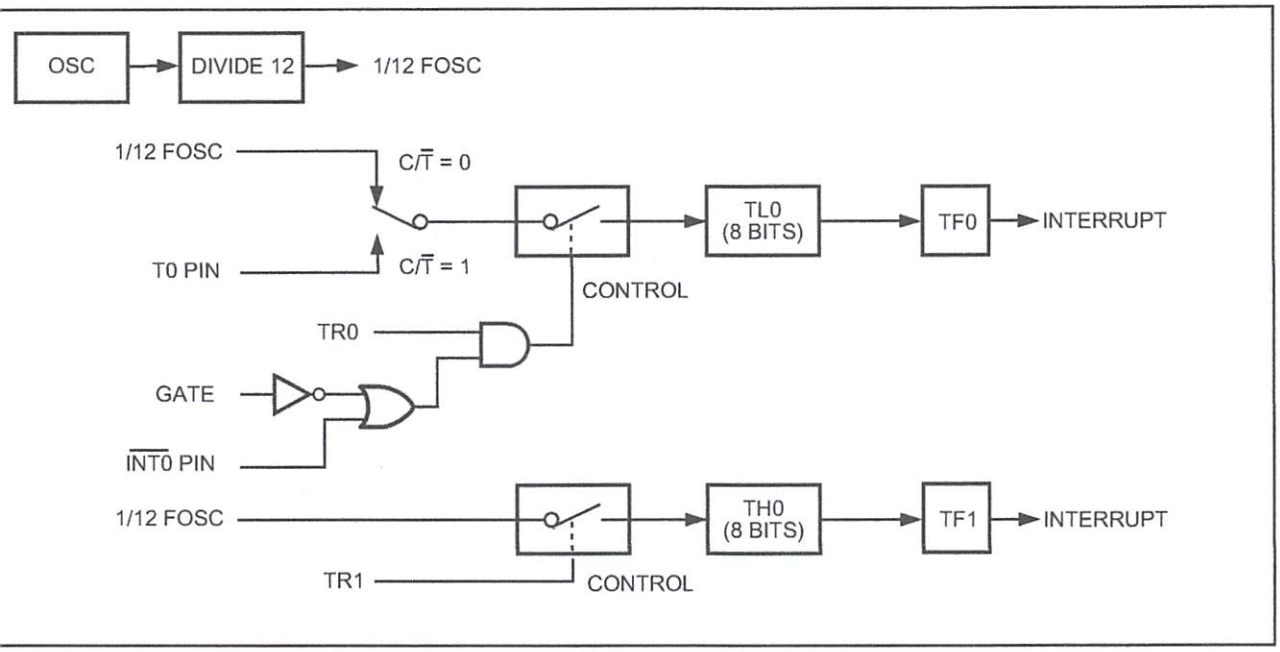


Figure 11. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Timer Setup

Tables 3 through 6 give TMOD values that can be used to set up Timers in different modes.

It assumes that only one timer is used at a time. If Timers 0 and 1 must run simultaneously in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if Timer 0 must run in Mode 1 GATE (external control), and Timer 1 must run in Mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user is not ready at this point to turn the timers on and will do so at another point in the program by setting bit TRx (in TCON) to 1.

Table 3. Timer/Counter 0 Used as a Timer

Mode	Timer 0 Function	TMOD Internal Control(1)	TMOD External Control(2)
0	13-Bit Timer	00H	08H
1	16-Bit Timer	01H	09H
2	8-Bit Auto-Reload	02H	0AH
3	Two 8-Bit Timers	03H	0BH

Table 4. Timer/Counter 0 Used as a Counter

Mode	Timer 0 Function	TMOD Internal Control(1)	TMOD External Control(2)
0	13-Bit Timer	04H	0CH
1	16-Bit Timer	05H	0DH
2	8-Bit Auto-Reload	06H	0EH
3	One 8-Bit Counter	07H	0FH

Notes:
1. The Timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

Table 5. Timer/Counter 1 Used as a Timer

Mode	Timer 1 Function	TMOD Internal Control(1)	TMOD External Control(2)
0	13-Bit Timer	00H	80H
1	16-Bit Timer	10H	90H
2	8-Bit Auto-Reload	20H	A0H
3	Does Not Run	30H	B0H

Table 6. Timer/Counter 1 Used as a Counter

Mode	Timer 1 Function	TMOD Internal Control(1)	TMOD External Control(2)
0	13-Bit Timer	40H	C0H
1	16-Bit Timer	50H	D0H
2	8-Bit Auto-Reload	60H	E0H
3	Not Available	—	—

Notes:
1. The Timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{INT1}$ (P3.3) when TR1 = 1 (hardware control).

RIAL INTERFACE

Serial port is full duplex, which means it can transmit receive simultaneously. It is also receive-buffered, ch means it can begin receiving a second byte before eviously received byte has been read from the receive ster. (However, if the first byte still has not been read n reception of the second byte is complete, one of the s will be lost.) The serial port receive and transmit sters are both accessed at Special Function Register JF. Writing to SBUF loads the transmit register, and ding SBUF accesses a physically separate receive ster.

serial port can operate in the following four modes:

de 0:
al data enters and exits through RXD. TXD outputs the clock. Eight data bits are transmitted/received, with LSB first. The baud rate is fixed at 1/12 the oscillator uency (see Figure 12).

de 1:
bits are transmitted (through TXD) or received (through D): a start bit (0), eight data bits (LSB first), and a stop 1). On receive, the stop bit goes into RB8 in Special ction Register SCON. The baud rate is variable (see ure 13).

Mode 2:
Eleven bits are transmitted (through TXD) or received (through RXD): a start bit (0), eight data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the ninth data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) can be moved into TB8. On receive, the ninth data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency (see Figure 14).

Mode 3:
Eleven bits are transmitted (through TXD) or received (through RXD): a start bit (0), eight data bits (LSB first), a programmable ninth data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which is variable in Mode 3 (see Figure 15).

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, nine data bits are received, followed by a stop bit. The ninth bit goes into RB8; then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt is activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON.

The following example shows how to use the serial interrupt for multiprocessor communications. When the master processor must transmit a block of data to one of several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the ninth bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave is interrupted by a data byte. An address byte, however, interrupts all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave clears its SM2 bit and prepares to receive the data bytes that follows. The slaves that are not addressed set their SM2 bits and ignore the data bytes.

SM2 has no effect in Mode 0 but can be used to check the validity of the stop bit in Mode 1. In a Mode 1 reception, if SM2 = 1, the receive interrupt is not activated unless a valid stop bit is received.

Baud Rates

The baud rate in Mode 0 is fixed as shown in the following equation.

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of the SMOD bit in Special Function Register PCON. If SMOD = 0 (the value on reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is 1/32 of the oscillator frequency, as shown in the following equation.

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD}}{64} \times (\text{Oscillator Frequency})$$

In the IS89C51, the Timer 1 overflow rate determines the baud rates in Modes 1 and 3.

Using the Timer 1 to Generate Baud Rates

When Timer 1 is the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD according to the following equation.

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{SMOD}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either timer or counter operation in any of its three running modes. In the most typical applications, it is configured for timer operation in auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the following formula.

$$\text{Mode 1,3 Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator Frequency}}{12x\,[256 - (TH1)]}$$

Programmers can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Table 7 lists commonly used baud rates and how they can be obtained from Timer 1.

le 7. Commonly Used Baud Rates Generated by Timer 1

			Timer 1		
Baud Rate	fosc	SMOD	C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

e About Mode 0

al data enters and exits through RXD. TXD outputs the clock. Eight data bits are transmitted/received, with LSB first. The baud rate is fixed at 1/12 the oscillator uency.

ıre 12 shows a simplified functional diagram of the al port in Mode 0 and associated timing.

ısmission is initiated by any instruction that uses SBUF a destination register. The "write to SBUF" signal at 2 also loads a 1 into the ninth position of the transmit register and tells the TX Control block to begin a smission. The internal timing is such that one full chine cycle will elapse between "write to SBUF" and vation of SEND.

ND transfer the output of the shift register to the rnate output function line of P3.0, and also transfers FT CLOCK to the alternate output function line of P3.1. FT CLOCK is low during S3, S4, and S5 of every chine cycle, and high during S6, S1, and S2. At S6P2 very machine cycle in which SEND is active, the tents of the transmit shift register are shifted one ition to the right.

data bits shift out to the right, 0s come in from the left. en the MSB of the data byte is at the output position of shift register, the 1 that was initially loaded into the h position is just to the left of the MSB, and all positions ne left of that contain 0s. This condition flags the TX ıtrol block to do one last shift, then deactivate SEND set TI. Both of these actions occur at S1P1 of the tenth chine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and RI = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register and activates RECEIVE in the next clock phase.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted on position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the tenth machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

More About Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), eight data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the IS89C51 the baud rate is determined by the Timer 1 overflow rate.

Figure 13 shows a simplified functional diagram of the serial port in Mode 1 and associated timings for transmit and receive.

Transmission is initiated by any instruction that uses SBUF as a destination register.

The "write to SBUF" signal also loads a 1 into the ninth bit position of the transmit shift register and flags the TX control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.

The transmission begins when SEND is activated, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, 0s are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the ninth position is just to the left of the MSB, and all positions to the left of that contain 0s. This condition flags the TX Control unit to do one last shift, then deactivate SEND and set TI. This occurs at the tenth divide-by-16 rollover after "write to SBUF".

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16th. At the seventh, eighth, and ninth counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least two of the three samples. This is done to reject noise. In order to reject false bits, if the value accepted during the first bit time is not 0, the receive circuits are reset and the unit continues looking for another 1-to-0 transition. If the start bit is valid, it is shifted into the input shift register, and reception of the rest of the frame proceeds.

As data bits come in from the right, 1s shift to the left. When the start bit arrives at the leftmost position in the shift register, (which is a 9-bit register in Mode 1), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8 and to set RI is generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0 and

2) Either SM2 = 0, or the received stop bit =1

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the eight data bits go into SBUF, and RI is activated. At this time, whether or not the above conditions are met, the unit continues looking for a 1-to-0 transition in RXD.

More About Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), eight data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the ninth data bit (TB8) can be assigned the value of 0 or 1. On receive, the ninth data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 14 and 15 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the ninth bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the ninth bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.

The transmission begins when SEND is activated, which puts the start bit at TXD. One bit timer later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the ninth bit position of the shift register. Thereafter, only 0s are clocked in. Thus, as data bits shift out to the right, 0s are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain 0s. This condition flags the TX Control unit to do one last shift, then deactivate SEND and set TI. This occurs at the eleventh divide-by-16 rollover after "write to SBUF".

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

e seventh, eighth, and ninth counter states of each bit , the bit detector samples the value of RXD. The value epted is the value that was seen in at least two of the e samples. If the value accepted during the first bit time t 0, the receive circuits are reset and the unit continues ing for another 1-to-0 transition. If the start bit proves l, it is shifted into the input shift register, and reception e rest of the frame proceeds.

lata bits come in from the right, 1s shift out to the left. en the start bit arrives at the leftmost position in the shift ster (which in Modes 2 and 3 is a 9-bit register), it flags RX Control block to do one last shift, load SBUF and 3, and set RI. The signal to load SBUF and RB8 and to RI is generated if, and only if, the following conditions met at the time the final shift pulse is generated:

I = 0, and

ither SM2 = 0 or the received ninth data bit = 1

her of these conditions is not met, the received frame etrievably lost, and RI is not set. If both conditions are , the received ninth data bit goes into RB8, and the first t data bits go into SBUF. One bit time later, whether the ve conditions were met or not, the unit continues ing for a 1-to-0 transition at the RXD input.

e that the value of the received stop bit is irrelevant to JF, RB8, or RI.

Table 8. Serial Port Setup

Mode	SCON	SM2Variation
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

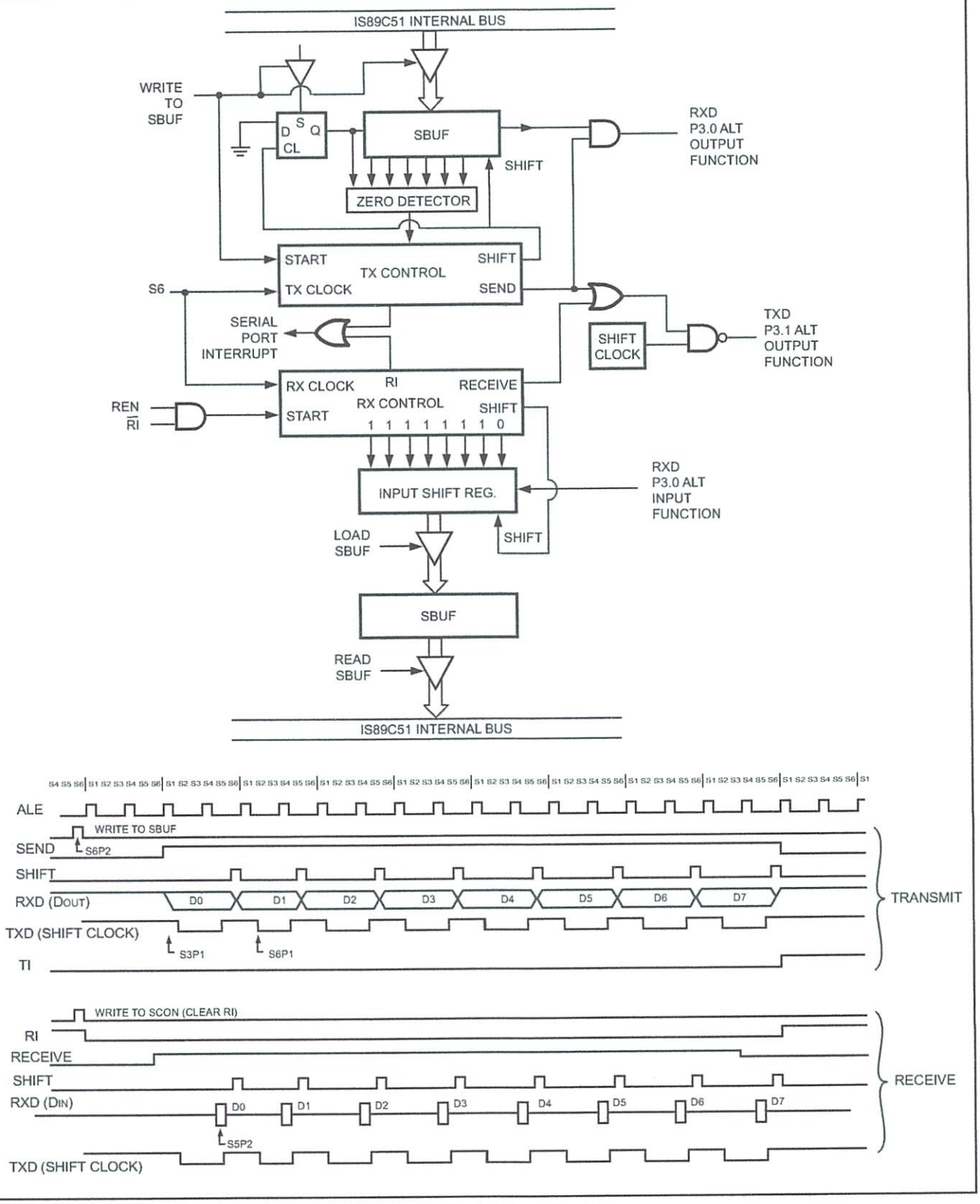


Figure 12. Serial Port Mode 0

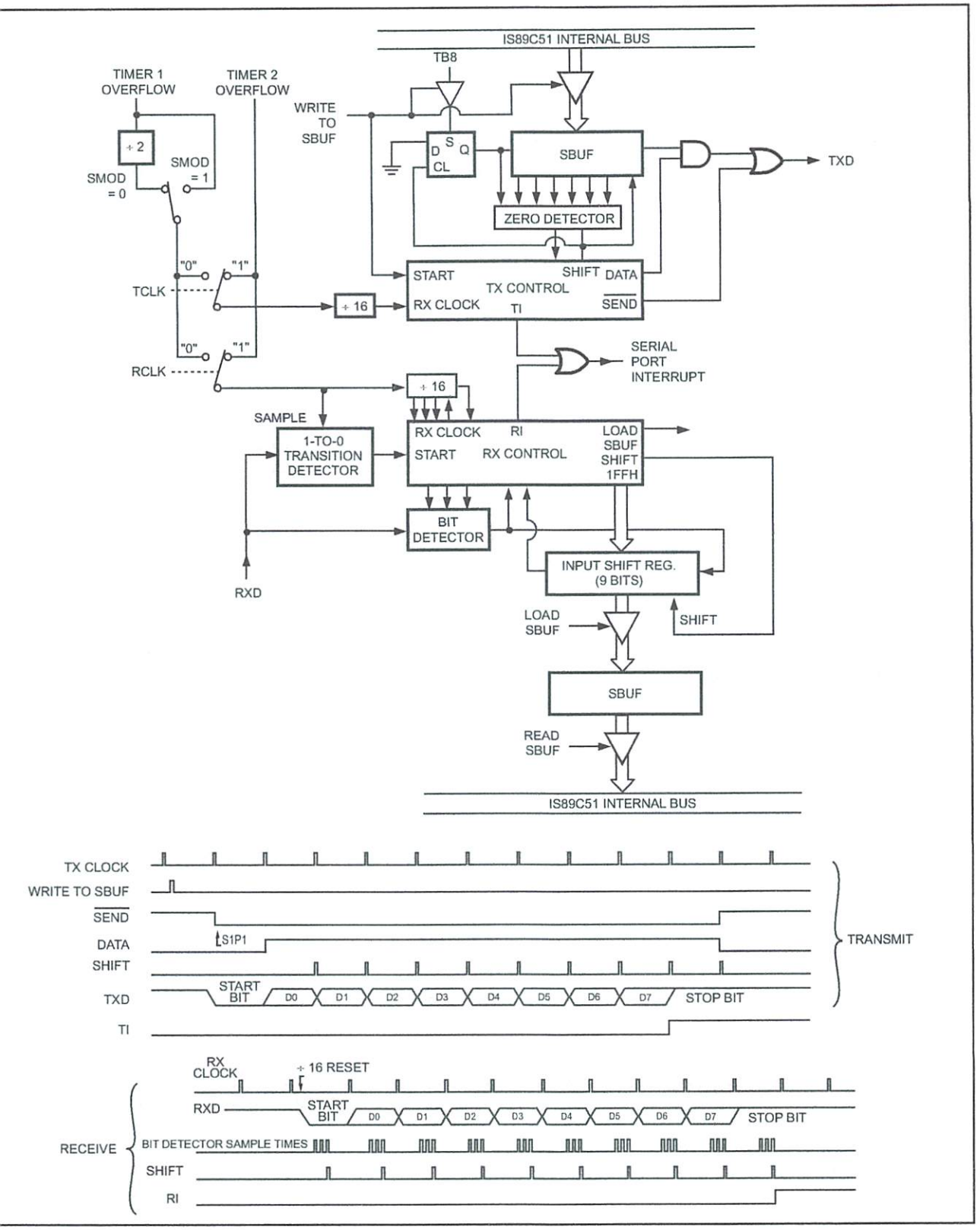


Figure 13. Serial Port Mode 1

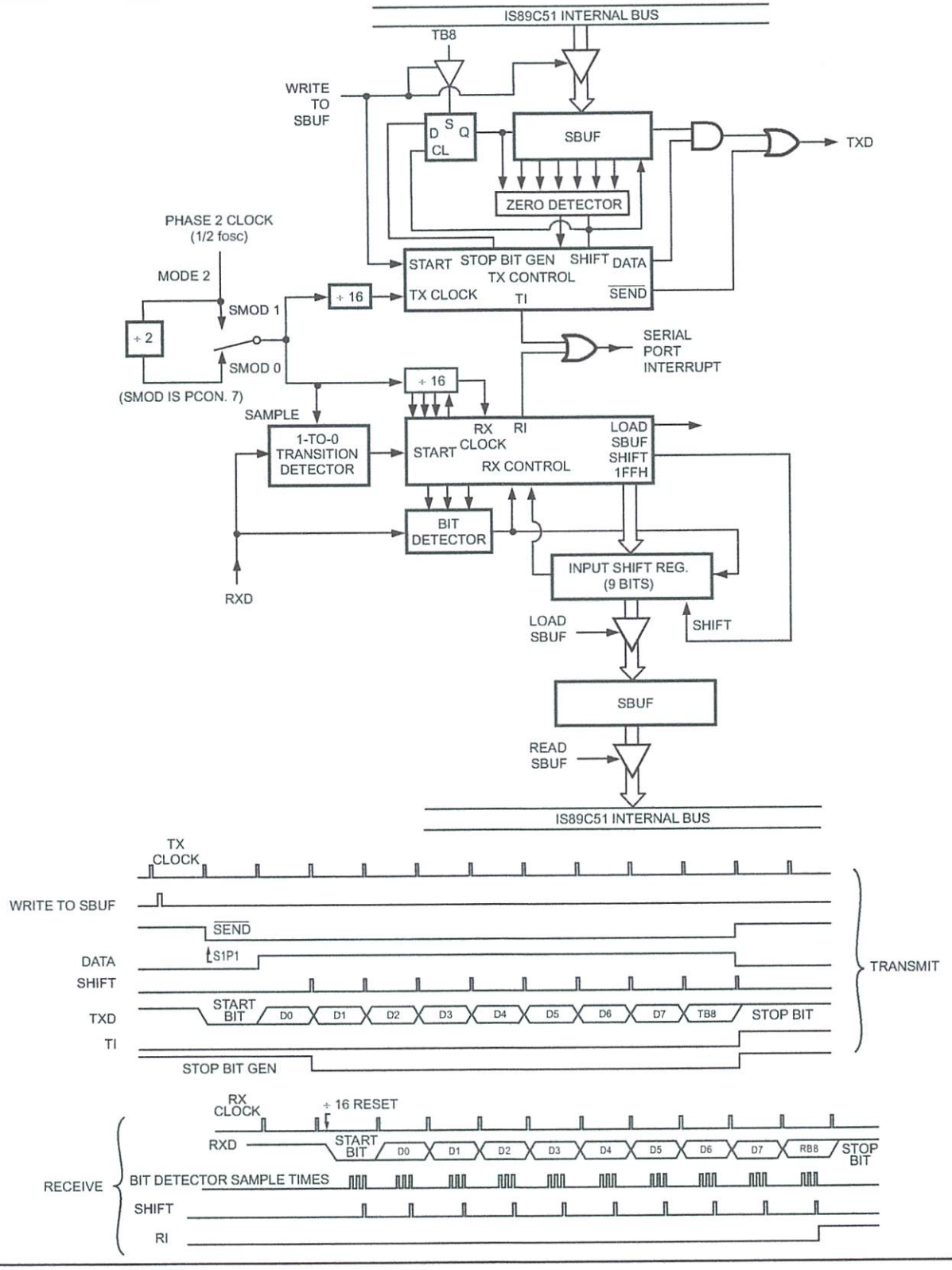


Figure 14. Serial Port Mode 2

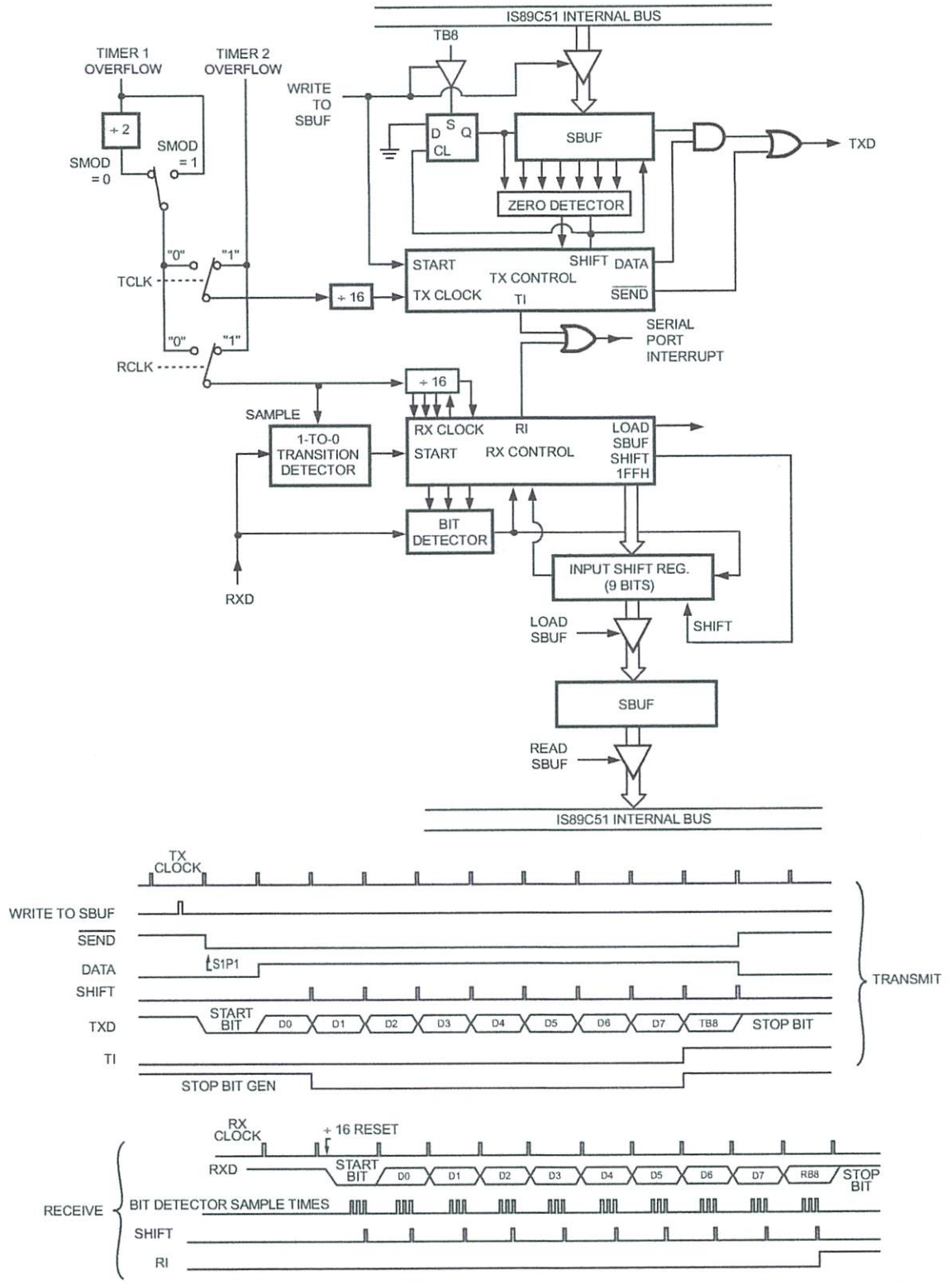


Figure 15. Serial Port Mode 3

INTERRUPT SYSTEM

The IS89C51 provides six interrupt sources: two external interrupts, two timer interrupts, and a serial port interrupt. These are shown in Figure 16.

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored, hardware clears the flag that generated an external interrupt only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI or TI generated the interrupt, and the bit must be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated and pending interrupts can be canceled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (interrupt enable) at address 0A8H. As well as individual enable bits for each interrupt source, there is a global enable/disable bit that is cleared to disable all interrupts or set to turn on interrupts (see SFR IE).

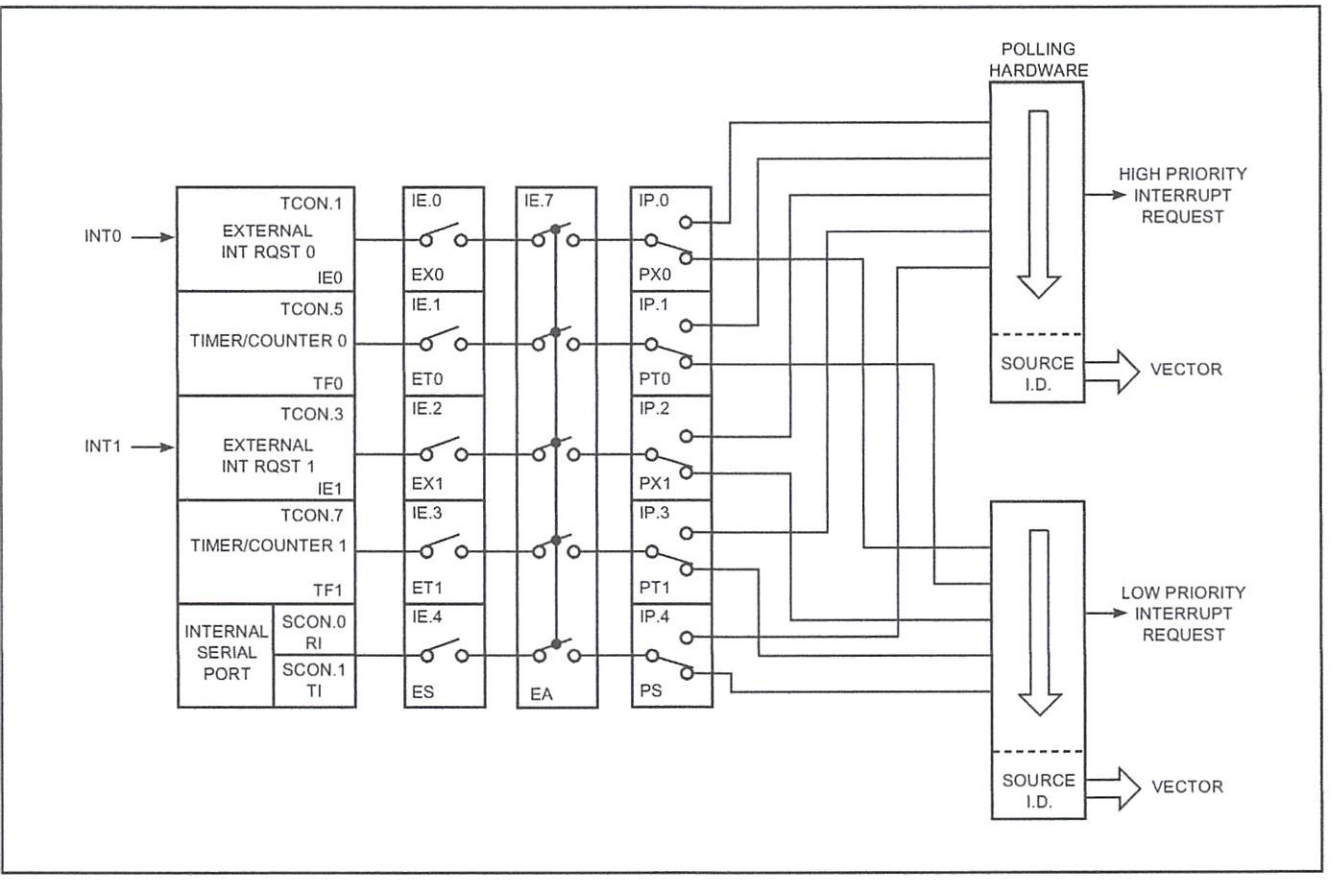


Figure 16. Interrupt System

ority Level Structure

h interrupt source can also be individually programmed ne of two priority levels by setting or clearing a bit in cial Function Register IP (interrupt priority) at address H. IP is cleared after a system reset to place all rrupts at the lower priority level by default. A low-rity interrupt can be interrupted by a high-priority rrupt but not by another low-priority interrupt. A high-rity interrupt can not be interrupted by any other rrupt source.

vo requests of different priority levels are received ultaneously, the request of higher priority level is iced. If requests of the same priority level are received ultaneously, an internal polling sequence determines ch request is serviced. Thus, within each priority level e is a second priority structure determined by the ng sequence, as follows:

Source	Priority Within Level
IE0	(Highest)
TF0	
IE1	
TF1	
RI + TI	(Lowest)

e that the "priority within level" structure is only used to lve *simultaneous requests of the same priority level.*

w Interrupts Are Handled

interrupt flags are sampled at S5P2 of every machine e. The samples are polled during the following machine e. If one of the flags was in a set condition at S5P2 of preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. If an active interrupt flag is not being serviced because of one of the above conditions and is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new. The polling cycle/LCALL sequence is illustrated in Figure 17.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 17, then in accordance with the above rules it will be serviced during C5 and C6, without any instruction of the lower priority routine having been executed.

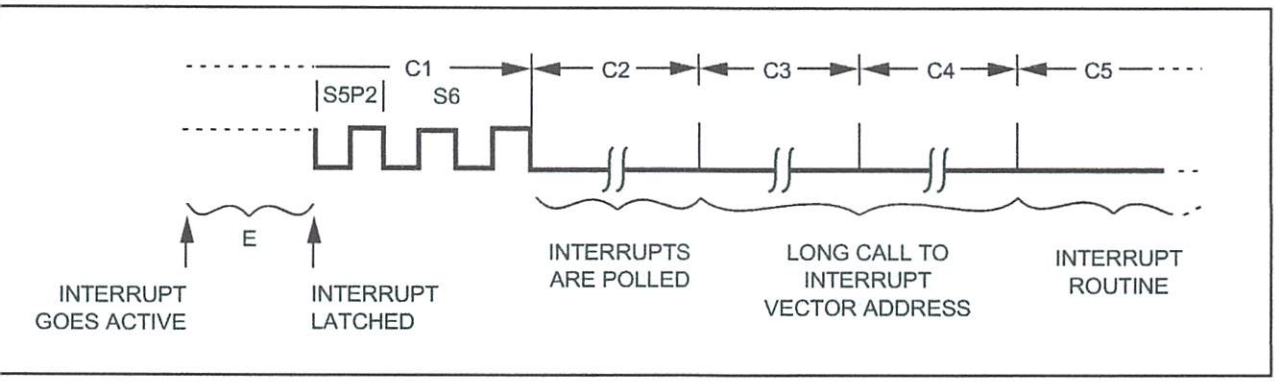


Figure 17. Interrupt Response Timing Diagram

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it does not. It never clears the Serial Port flags. This must be done in the user's software. The processor clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being serviced, as shown in the following table.

Interrupt Source	Interrupt Request Bits	Cleared by Hardware	Vector Address
$\overline{INT0}$	IE0	No (level) Yes (trans.)	0003H
Timer 0	TF0	Yes	000BH
$\overline{INT1}$	IE1	No (level) Yes (trans.)	0013H
Timer 1	TF1	Yes	001BH
Serial Port	RI, TI	No	0023H
System Reset	RST		0000H

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

Interrupt	Flag	SFR Register and Bit Position
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial Port	TI	SCON.1
Serial Port	RI	SCON.0

When an interrupt is accepted, the following action occurs:

1. The current instruction completes operation.
2. The PC is saved on the stack.
3. The current interrupt status is saved internally.
4. Interrupts are blocked at the level of the interrupts.
5. The PC is loaded with the vector address of the ISR (interrupts service routine).
6. The ISR executes.

The ISR executes and takes action in response to the interrupt. The ISR finishes with RETI (return from interrupt) instruction. This retrieves the old value of the PC from the stack and restores the old interrupt status. Execution of the main program continues where it left off.

External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx= 0, external interrupt x is triggered by a detected low at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{INTx}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one machine cycle, and then hold it low for at least one machine cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

sponse Time

INT0 and INT1 levels are inverted and latched into the rupt flags IE0 and IE1 at S5P2 of every machine cycle. larly, the Serial Port flags RI and TI are set at S5P2. values are not actually polled by the circuitry until the machine cycle.

Timer 0 and Timer 1 flags, TF0 and TF1, are set at 2 of the cycle in which the timers overflow. The values hen polled by the circuitry in the next cycle.

request is active and conditions are right for it to be owledged, a hardware subroutine call to the requested ice routine will be the next instruction executed. The itself takes two cycles. Thus, a minimum of three plete machine cycles elapsed between activation of xternal interrupt request and the beginning of execution e first instruction of the service routine. Figure 16 vs response timings.

nger response time results if the request is blocked by of the three previously listed conditions. If an interrupt qual or higher priority level is already in progress, the tional wait time depends on the nature of the other rupt's service routine. If the instruction in progress is n its final cycle, the additional wait time cannot be more three cycles, since the longest instructions (MUL and are only four cycles long. If the instruction in progress ETI or an access to IE or IP, the additional wait time not be more than five cycles (a maximum of one more e to complete the instruction in progress, plus four es to complete the next instruction if the instruction is . or DIV).

s, in a single-interrupt system, the response time is ys more than three cycles and less than nine cycles.

Single-Step Operation

The IS89C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be serviced while an interrupt of equal priority level is still in progress, nor will it be serviced after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-step operation is to program one of the external interrupts (for example, INT0) to be level-activated. The service routine for the interrupt will terminate with the following code:

```
JNB     P3.2,$      ;Wait Here Till INT0 Goes High
JB      P3.2,$      ;Now Wait Here Till it Goes Low
RETI                ;Go Back and Execute One
                     Instruction
```

If the INT0 pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until INT0 is pulsed (from low-to-high-to-low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

OTHER INFORMATION

Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger.

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running.* The CPU responds by generating an internal reset, with the timing shown in Figure 18.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 9 lists the SFRs and their reset values.

Then internal RAM is not affected by reset. On power-up the RAM content is indeterminate.

Table 9. Reset Values of the SFR's

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	FFH
IP	XXX00000B
IE	0XX00000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Indeterminate
PCON	0XXX0000B

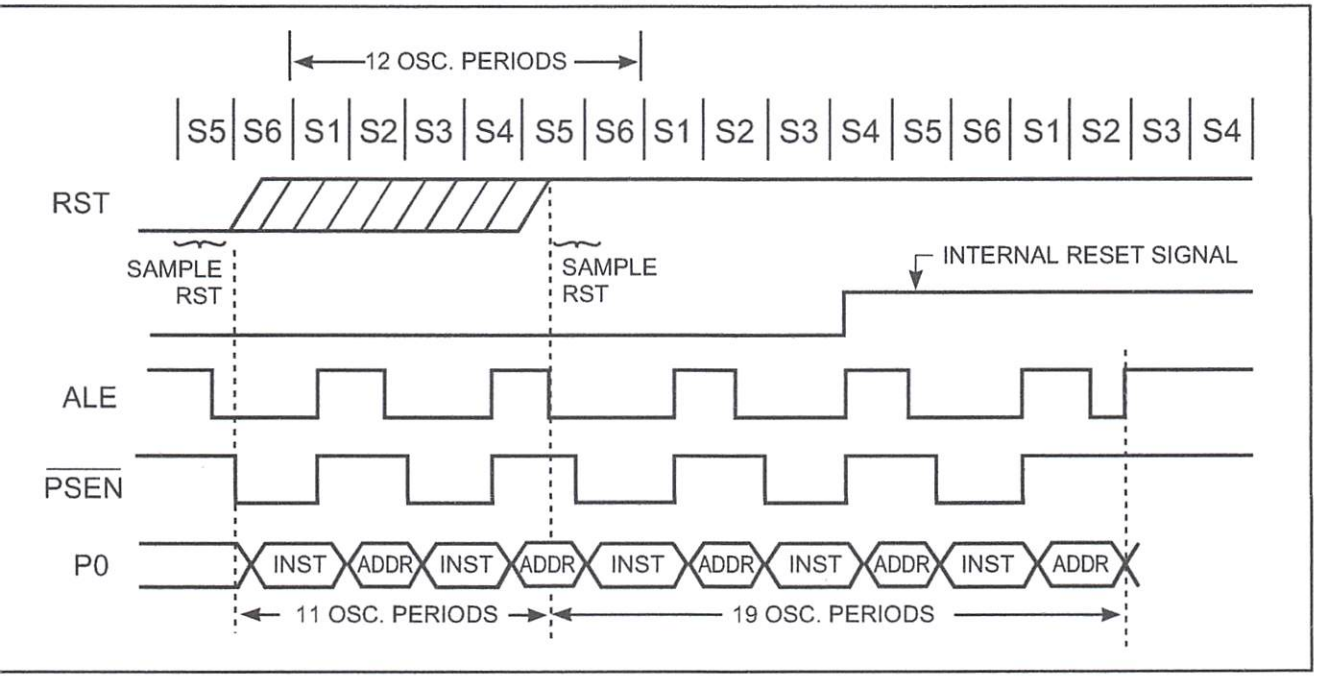


Figure 18. Reset Timing

ver-on Reset

the IS89C51, the external resistor can be removed ause the RST pin has an internal pulldown. capacitor value can then be reduced to 1 µF (see re 19).

n power is turned on, the circuit holds the RST pin high n amount of time that depends on the value of the citor and the rate at which it charges. To ensure a l reset, the RST pin must be high long enough to allow oscillator time to start-up (normally a few msec) plus machine cycles.

e that the port pins will be in a random state until the llator has start and the internal reset algorithm has en 1s to them.

this circuit, reducing Vcc quickly to 0 causes the RST voltage to momentarily fall below 0V. However, this ge is internally limited and will not harm the device.

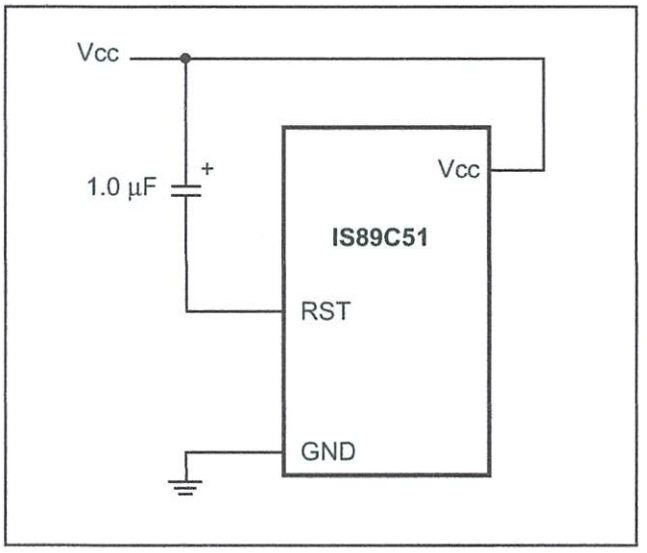


Figure 19. Power-On Reset Circuit

Power-Saving Modes of Operation

The IS89C51 has two power-reducing modes. Idle and Power-down. The input through which backup power is supplied during these operations is Vcc. Figure 20 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power-down (PD = 1), the oscillator is frozen. The Idle and Power-down modes are activated by setting bits in Special Function Register PCON.

Idle Mode

An instruction that sets PCON.0 is the last instruction executed before the Idle mode begins. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety; the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{PSEN}$ hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits GF0 and GF1 can be used to indicate whether an interrupt occurred during normal operation or during an Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset must be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time, the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 18, two or three machine cycles of program execution may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during his time, but access to the port pins is not inhibited. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not write to a port pin or to external data RAM.

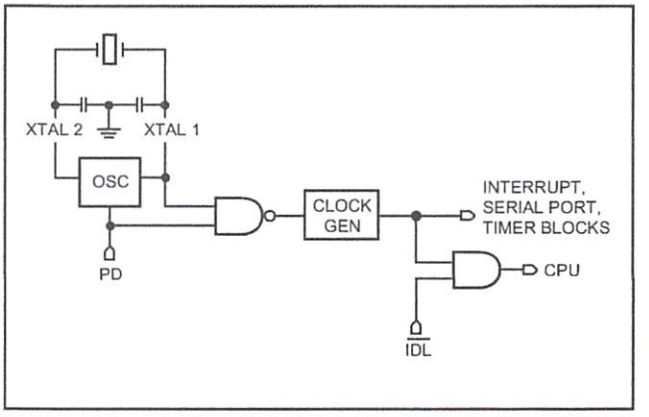


Figure 20. Idle and Power-Down Hardware

Power-down Mode

An instruction that sets PCON.1 is the last instruction executed before Power-down mode begins. In the Power-down mode, the on-chip oscillator stops. With the clock frozen, all functions are stopped, but the on-chip RAM and Special function Registers are held. The port pins output the values held by their respective SFRs. ALE and $\overline{PSEN}$ output lows.

In the Power-down mode of operation, Vcc can be reduced to as low as 2V. However, Vcc must not be reduced before the Power-down mode is invoked, and Vcc must be restored to its normal operating level before the Power-down mode is terminated. The reset that terminates Power-down also frees the oscillator. The reset should not be activated before Vcc is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10 msec).

The only exit from Power-down is a hardware reset. Reset redefines all the SFRs but does not change the on-chip RAM.

e 10. Status of the External Pins During Idle and Power-down Modes.

Mode	Memory	ALE	$\overline{\text{PSEN}}$	PORT 0	PORT 1	PORT 2	PORT 3
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

Chip Oscillators

on-chip oscillator circuitry of the IS89C51 is a single e inverter, intended for use as a crystal-controlled, tive reactance oscillator. In this application the crystal perated in its fundamental response mode as an ctive reactance in parallel resonance with capacitance rnal to the crystal (Figure 21). Examples of how to the clock with external oscillator are shown in re 22.

The crystal specifications and capacitance values (C1 and C2 in Figure 21) are not critical. 20 pF to 30 pF can be used in these positions at a 12 MHz to 24 MHz frequency with good quality crystals. (For ranges greater than 24 MHz refer to Figure 23.) A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values. The manufacturer of the ceramic resonator should be consulted for recommendation on the values of these capacitors.

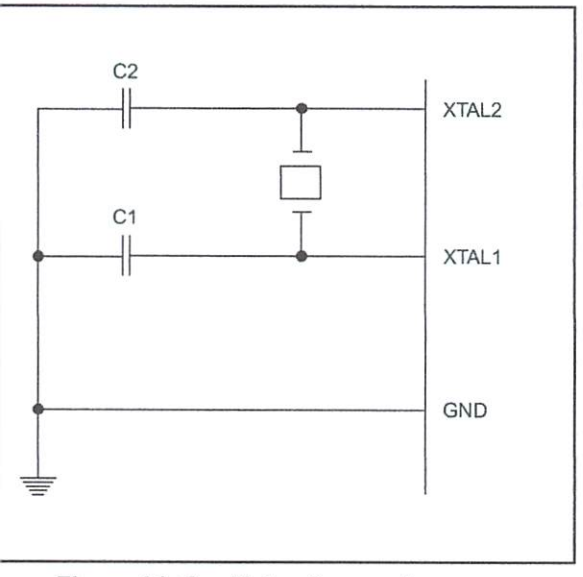


Figure 21. Oscillator Connections

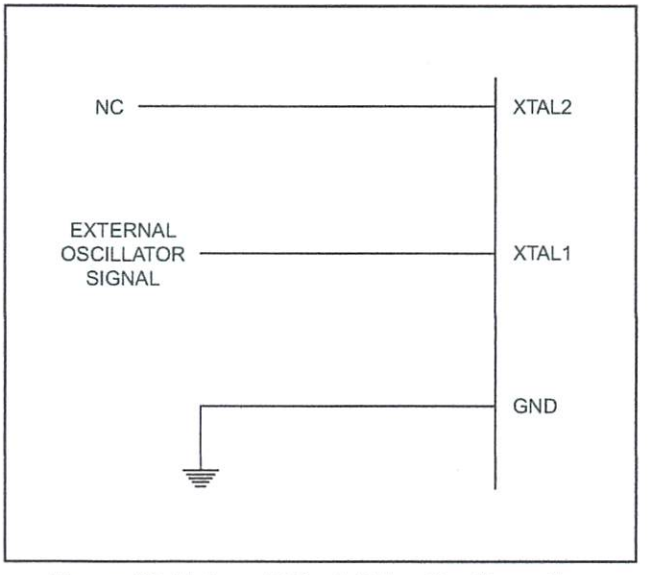


Figure 22. External Clock Drive Configuration

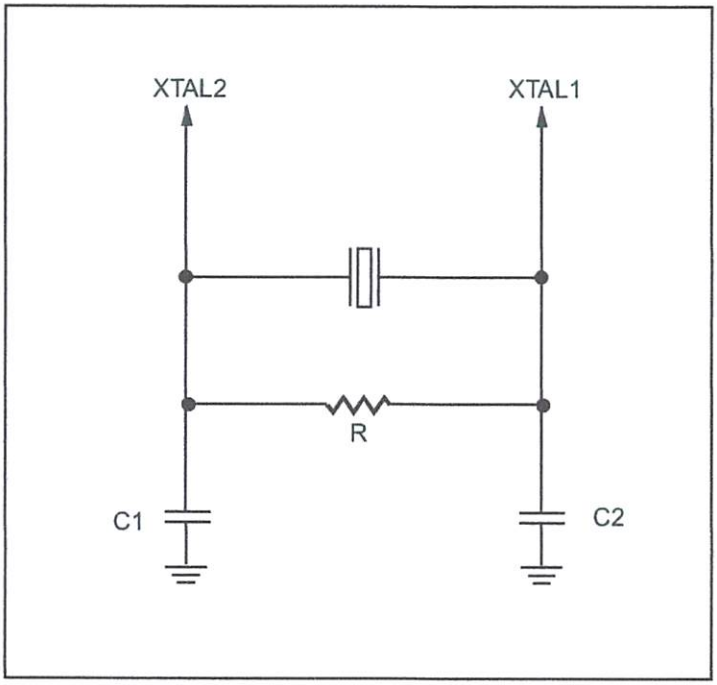


Figure 23. Oscillator Connections for High Speed (> 24 MHz)

Note:
When the frequency is higher than 24 MHz, please refer to Table 11 for recommended values of C1, C2, and R.

Table 11. Recommended Value for C1, C2, R

	Frequency Range	
	3.5 MHz - 24 MHz	**30 MHz - 40 MHz**
C1	20 pF-30 pF	3 pF-10 pF
C2	20 pF-30 pF	3 pF-10 pF
R	Not Apply	6.2K-10K

gram Memory Lock System

program lock system, when programmed, protects code against software piracy. The IS89C51 has a -level program lock system (see Table 12). The lock- re programmed in the same manner as the program ory.

detailed lock-bits features are listed in Table 12.

e 12. Program Lock Bits

	LB1	LB2	LB3	Protection Type
	U	U	U	No Program Lock Features enabled.
2	P	U	U	MOVC instructions executed from external program memory are diabled form fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on Reset and further programming of the Flash is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, also external execution is disabled.

SH MEMORY

gramming the IS89C51

S89C51 is normally shipped with the on-chip Flash ory array in the erased state (i.e., contents = FFH) eady to be programmed. The IS89C51 is programmed by-byte in programming mode. Before the on-chip code memory can be re-programmed, the entire ory array must be erased electrically.

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased using the appropriate combination of control signals. The write operation cycles is self-timed once initiated, will automatically time itself to completion. The programming interface is shown in Table 13 and Figures 24 and 25.

e 13. Flash Programming Mode

de	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
gram Code Data	H	L	⎍	12V	L	H	H	H
ify Code Data	H	L	H	H	L	L	H	H
gram Lock Bit 1	H	L	⎍	12V	H	H	H	H
gram Lock Bit 2	H	L	⎍	12V	H	H	L	L
gram Lock Bit 3	H	L	⎍	12V	H	L	H	L
d Signature Byte	H	L	H	H	L	L	L	L
o Erase	H	L	⎍	12V	H	L	L	L

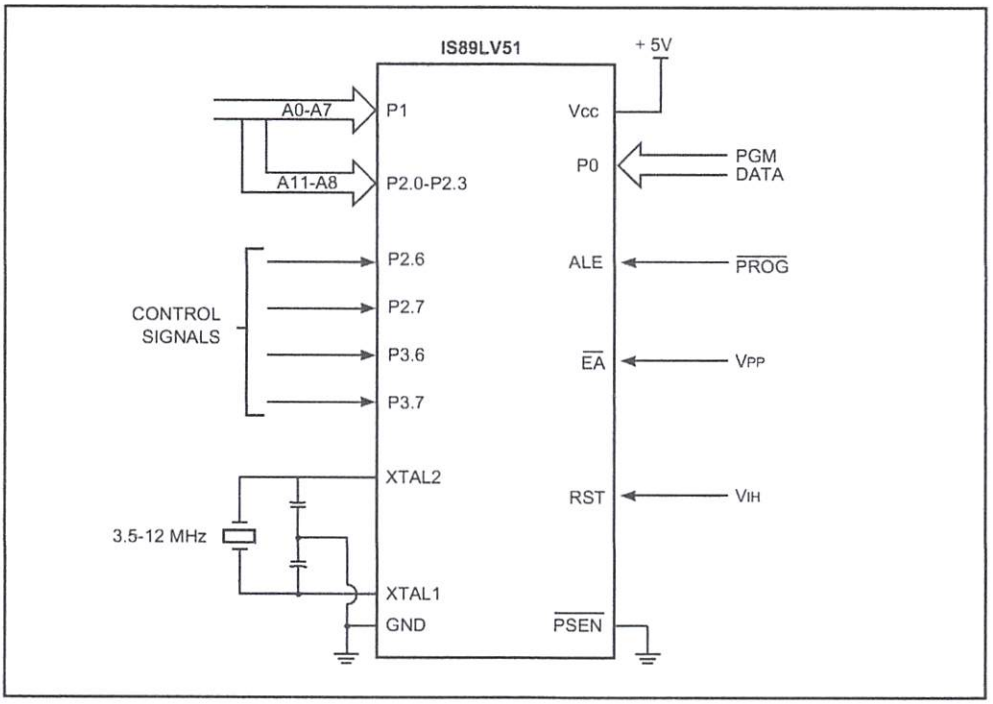


Figure 24. Programming the Flash Memory

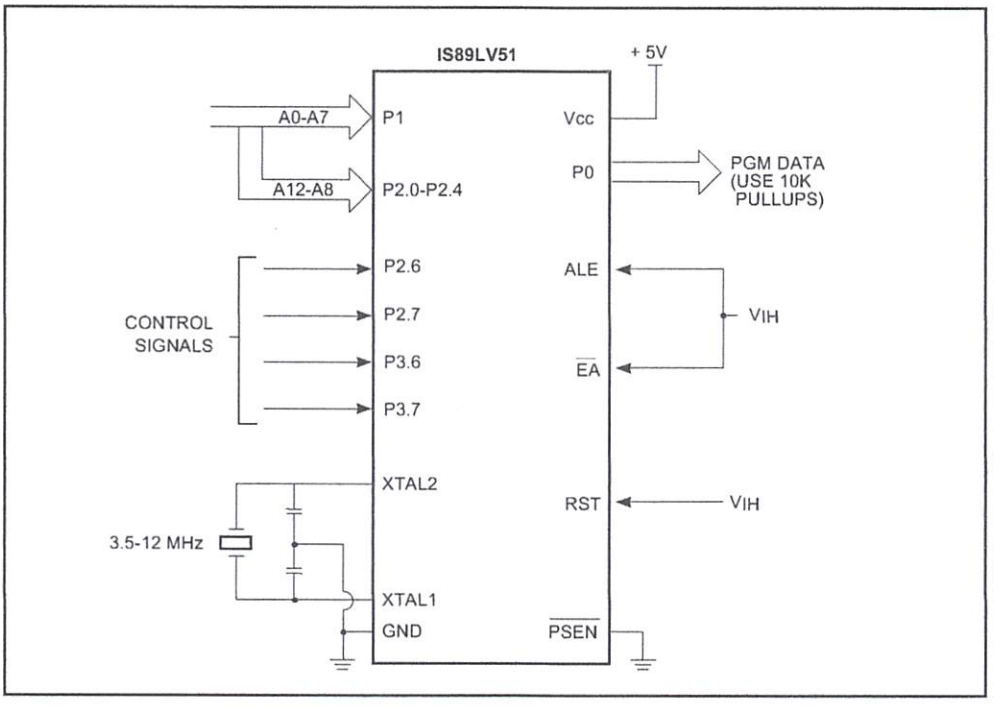


Figure 25. Verifying the Flash Memory

gramming Algorithm

re programming the IS89C51, the control signals, the ress, data should be setup according to the ramming mode table and programming interface. To ram the IS89C51, the following sequence should be wed:

nsert the desired memory location on the address bus.
nsert the appropriate data byte on the data bus.
Active the correct combination of control signals.
Raise $\overline{EA}$ / V_{PP} to 12V.
Pulse ALE / $\overline{PROG}$ once to program a byte in the Flash array, Encryption array or the lock bits.
Set $\overline{EA}$ / V_{PP} to 5 V and verify data. If the data is correct hen execute step 7, otherwise execute steps 1-6.
Repeat steps 1 through 6, changing the address and data for the entire array or until the end of the bject file is reached.

gram Verify

k bits LB1, LB2 and LB3 have not been programmed, programmed code data can be read back via the ess and data lines for verification. The lock bits cannot erified directly. Verification of lock bits is achieved by rving that their features are enabled.

erify the data after all addresses are programmed pletely, power down the IS89C51 and then reapply er. The programmed data can then be verified by ying the verify signals to the device.

Erasing Chip

All Flash memory cells must be programmed to '00' (include encryption array and lock bits) before the chip is erased. The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/ $\overline{PROG}$ low for t_{GLGHE} duration (See Table 14. Flash Programming and Verification Characteristics for t_{GLGHE} value.) After the chip is erased, the code array and lock bits are written with all "1"s. If any Flash memory cell is not '1' (including lock bits), repeat the chip erase again. The chip erase operation must be executed successfully before the code memory can be re-programmed.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H and 032H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

Signature Location	Value
(030H)	D5H indicates manufactured by ISSI
(031H)	52H indicates IS89C51
(032H)	FFH indicates programming voltage is 12V

FLASH PROGRAMMING AND VERIFICATION CHARACTERISTICS AND WAVEFORMS

Table 14. Flash Programming and Verification Characteristics[1]

Symbol	Parameter	Min	Max	Unit
V_{PP}	Programming Supply Voltage	11.5	12.5	V
$1/t_{CLCL}$	Oscillator Frequency	3.5	12	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	48	—	t_{CLCL}
t_{GHAX}	Address Hold after $\overline{PROG}$	48	—	t_{CLCL}
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	48	—	t_{CLCL}
t_{GHDX}	Data Hold after $\overline{PROG}$	48	—	t_{CLCL}
t_{EHSH}	COND ENABLE to V_{PPH}	48	—	t_{CLCL}
t_{SHGL}	V_{PPH} Setup to $\overline{PROG}$ Low	10	—	µs
t_{GHSL}	V_{PPH} Hold after $\overline{PROG}$	10	—	µs
t_{GLGH}	$\overline{PROG}$ Pulse Width	120	—	µs
t_{GHGL}	$\overline{PROG}$ High to $\overline{PROG}$ Low	10	—	µs
t_{AVQV}	Address to Data Valid	—	48	t_{CLCL}
t_{ELQV}	COND ENABLE to Data Valid	—	48	t_{CLCL}
t_{EHQZ}	Data Float after COND DISABLE	—	48	t_{CLCL}
t_{GLGHE}	Erase $\overline{PROG}$ Pulse Width	200	—	ms
t_{ELPL}	COND DISABLE to Power Low	0	—	ns
t_{PLPH}	Power Off Time	10	—	µs
t_{PHEH}	Power On to V_{PPL}	10	—	ms
t_{EHVH}	V_{PPL} to COND ENABLE	0	—	ns

Notes:
1. T_A = 21°C to 27°C, Vcc = 5.0V ± 10%.
2. COND ENABLE and COND DISABLE are generated and depend on the control signals on pins P2.6, P2.7, P3.6 and P3.7. The signals set the device in to and out of different processing conditions, such as programming condition, erasing condition, and verify condition.

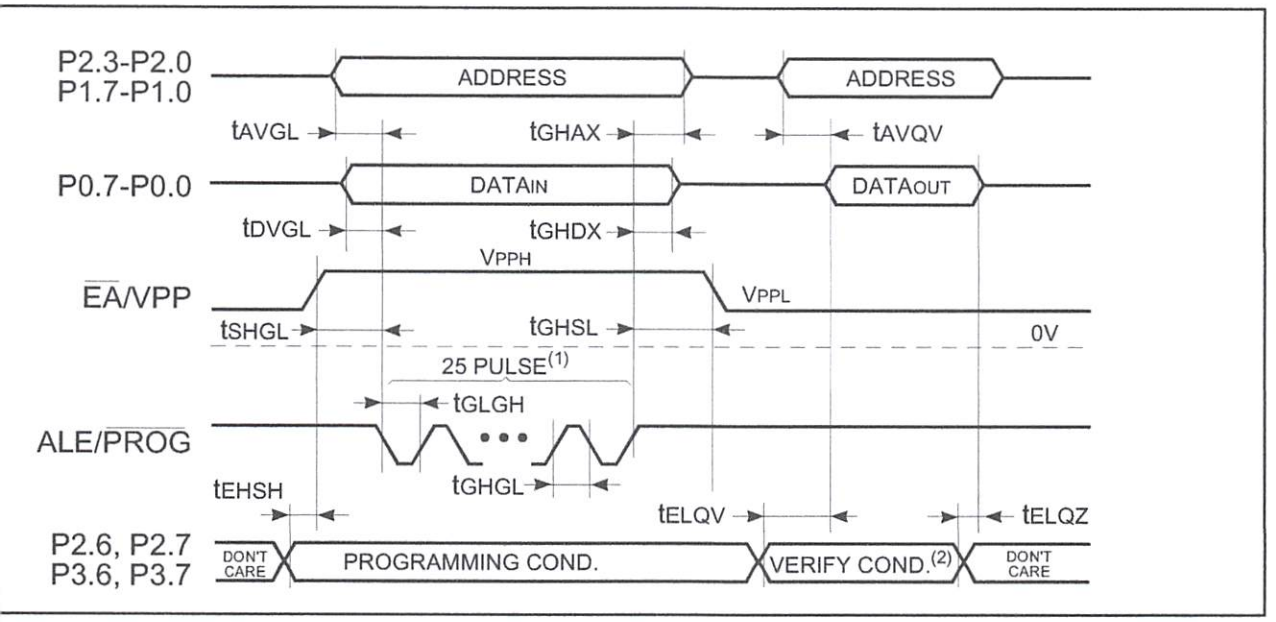


Figure 26. Flash Memory Programming and Verification Timing Waveform

otes:
One pulse for the main code array, 25 pulses for the encryption array and lock bits.
This verify condition is using at main code verification.
Power off waveform not shown.

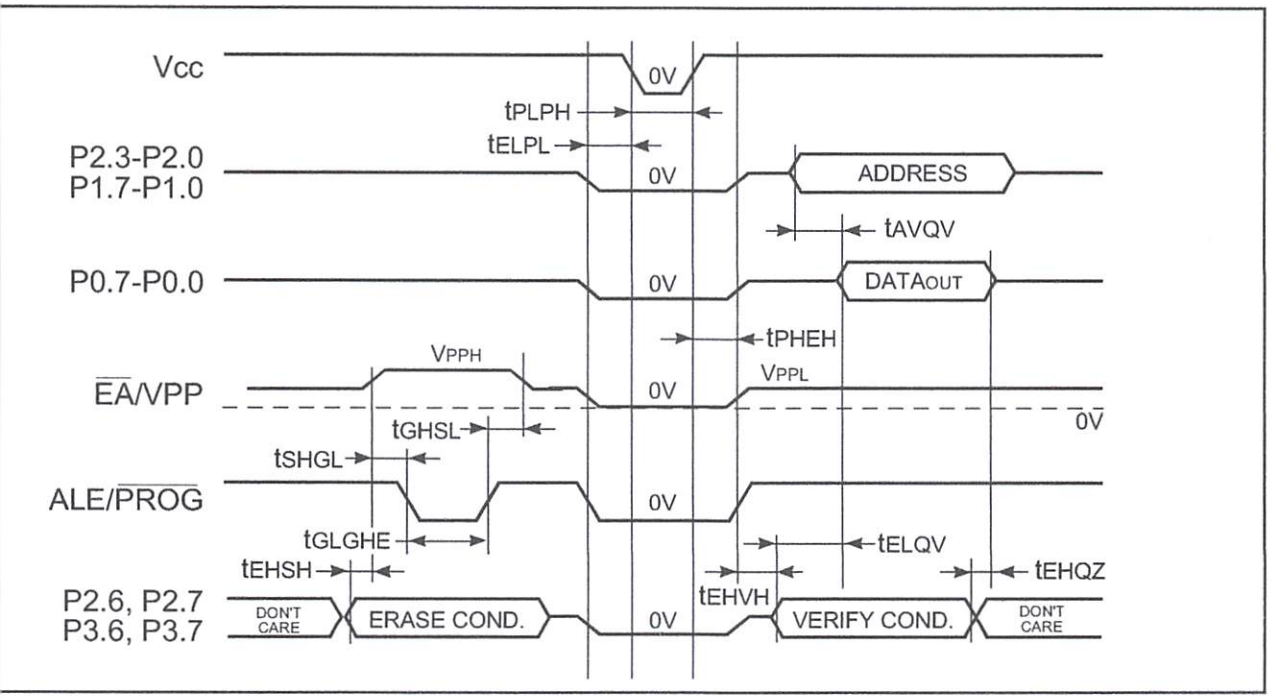


Figure 27. Flash Memory Erase Timing Waveform

ote:
The power off and power on waveform can be used in programming or erasing.

ABSOLUTE MAXIMUM RATINGS[1]

Symbol	Parameter	Value	Unit
V_{TERM}	Terminal Voltage with Respect to GND[2]	–2.0 to +7.0	V
T_{BIAS}	Temperature Under Bias[3]	–40 to +85	°C
T_{STG}	Storage Temperature	–65 to +125	°C
P_T	Power Dissipation	1.5	W

Note:
1. Stress greater than 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. Minimum DC input voltage is –0.5V. During transitions, inputs may undershoot to –2.0V for periods less than 20 ns. Maximum DC voltage on output pins is Vcc + 0.5V which may overshoot to Vcc + 2.0V for periods less than 20 ns.
3. Operating temperature is for commercial products only defined by this specification.

OPERATING RANGE[1]

Range	Ambient Temperature	Vcc	Oscillator Frequency
Commercial	0°C to +70°C	5V ± 10%	3.5 to 40 MHz
Industrial	–40°C to +85°C	5V ± 10%	3.5 to 40 MHz

Note:
1. Operating ranges define those limits between which the functionality of the device is guaranteed.

CHARACTERISTICS

r Operating Range; GND = 0V)

mbol	Parameter	Test conditions	Min	Max	Unit
V_{IL}	Input low voltage (All except $\overline{EA}$)		–0.5	0.2Vcc – 0.1	V
$I_{IL}1$	Input low voltage ($\overline{EA}$)		–0.5	0.2Vcc – 0.3	V
V_{IH}	Input high voltage (All except XTAL 1, RST)		0.2Vcc + 0.9	Vcc + 0.5	V
$I_{IH}1$	Input high voltage (XTAL 1)		0.7Vcc	Vcc + 0.5	V
$V_{CH}+$	RST positive schmitt-trigger threshold voltage		0.7Vcc	Vcc + 0.5	V
$V_{CH}-$	RST negative schmitt-trigger threshold voltage		0	0.2Vcc	V
V_{OL}[1]	Output low voltage	Iol = 100 µA	—	0.3	V
	(Ports 1, 2, 3)	I_{OL} = 1.6 mA	—	0.45	V
		I_{OL} = 3.5 mA	—	1.0	V
$V_{L}1$[1]	Output low voltage	I_{OL} = 200 µA	—	0.3	V
	(Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA	—	0.45	V
		I_{OL} = 7.0 mA	—	1.0	V
V_{OH}	Output high voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	I_{OH} = –10 µA Vcc = 4.5V-5.5V	0.9Vcc	—	V
		I_{OL} = –25 µA	0.75Vcc	—	V
		I_{OL} = –60 µA	2.4	—	V
$V_{OH}1$	Output high voltage (Port 0, ALE, $\overline{PSEN}$)	I_{OH} = –80 µA Vcc = 4.5V-5.5V	0.9Vcc	—	V
		I_{OH} = –300 µA	0.75Vcc	—	V
		I_{OH} = –800 µA	2.4	—	V
I_{IL}	Logical 0 input current (Ports 1, 2, 3)	V_{IN} = 0.45V	—	–80	µA
I_{LI}	Input leakage current (Port 0)	0.45V < V_{IN} < Vcc	–10	+10	µA
I_{TL}	Logical 1-to-0 transition current (Ports 1, 2, 3)	V_{IN} = 2.0V	—	–650	µA
R_{RST}	RST pulldown resister		50	300	KΩ

:
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
$_{OL}$ exceeds the test condition, V_{OL} may exceed the related specification.
ns are not guaranteed to sink greater than the listed test conditions.

POWER SUPPLY CHARACTERISTICS

Symbol	Parameter	Test conditions	Min	Max	Unit
Icc	Power supply current[1]	Vcc = 5.0V			
	Active mode	12 MHz	—	20	mA
		16 MHz	—	26	mA
		20 MHz	—	32	mA
		24 MHz	—	38	mA
		32 MHz	—	50	mA
		40 MHz	—	62	mA
	Idle mode	12 MHz	—	5	mA
		16 MHz	—	6	mA
		20 MHz	—	7.6	mA
		24 MHz	—	9	mA
		32 MHz	—	12	mA
		40 MHz	—	15	mA
	Power-down mode	Vcc = 5V	—	50	µA

Note:
1. See Figures 28, 29, 30, and 31 for Icc test conditiions.

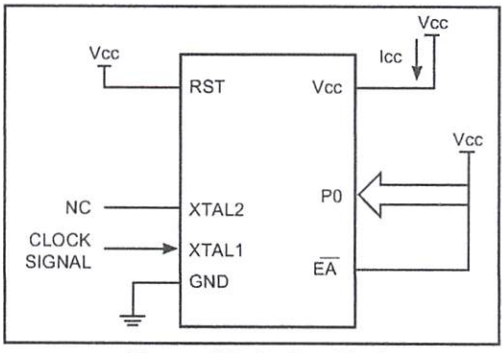


Figure 28. Active Mode

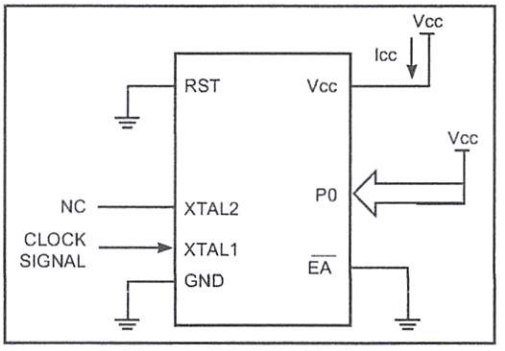


Figure 29. Idle Mode

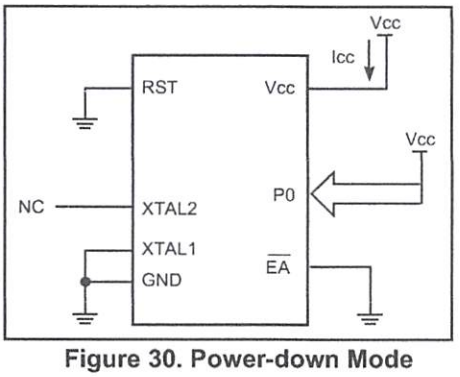


Figure 30. Power-down Mode

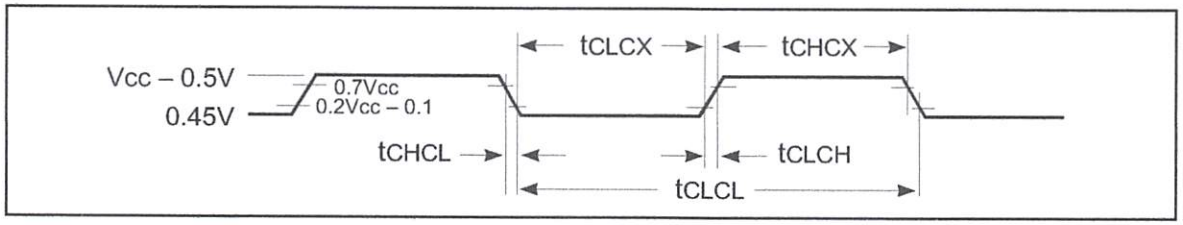


Figure 31. Icc Test Conditions

Note:
1. Clock signal waveform for Icc tests in active and idle mode (t_{CLCH} = t_{CHCL} = 5 ns)

CHARACTERISTICS

r Operating Range; GND = 0V; C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for Other Outputs = 80 pF)

TERNAL MEMORY CHARACTERISTICS

		24 MHz Clock		40 MHz Clock		Variable Oscillator (3.5 - 40 MHz)		
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
$1/t_{CLCL}$	Oscillator frequency	—	—	—	—	3.5	40	MHz
t_{LHLL}	ALE pulse width	68	—	35	—	$2t_{CLCL}$–15	—	ns
t_{AVLL}	Address valid to ALE low	26	—	10	—	t_{CLCL}–15	—	ns
t_{LLAX}	Address hold after ALE low	31	—	15	—	t_{CLCL}–10	—	ns
t_{LLIV}	ALE low to valid instr in	—	147	—	80	—	$4t_{CLCL}$–20	ns
t_{LLPL}	ALE low to $\overline{PSEN}$ low	31	—	15	—	t_{CLCL}–10	—	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	110	—	60	—	$3t_{CLCL}$–15	—	ns
t_{PLIV}	$\overline{PSEN}$ low to valid instr in	—	105	—	55	—	$3t_{CLCL}$–20	ns
t_{PXIX}	Input instr hold after $\overline{PSEN}$	0	—	0	—	0	—	ns
t_{PXIZ}	Input instr float after $\overline{PSEN}$	—	37	—	20	—	t_{CLCL}–5	ns
t_{AVIV}	Address to valid instr in	—	188	—	105	—	$5t_{CLCL}$–20	ns
t_{PLAZ}	$\overline{PSEN}$ low to address float	—	10	—	10	—	10	ns
t_{RLRH}	$\overline{RD}$ pulse width	230	—	130	—	$6t_{CLCL}$–20	—	ns
t_{WLWH}	$\overline{WR}$ pulse width	230	—	130	—	$6t_{CLCL}$–20	—	ns
t_{RLDV}	$\overline{RD}$ low to valid data in	—	157	—	90	—	$4t_{CLCL}$–10	ns
t_{RHDX}	Data hold after $\overline{RD}$	0	—	0	—	0	—	ns
t_{RHDZ}	Data float after $\overline{RD}$	—	78	—	45	—	$2t_{CLCL}$–5	ns
t_{LLDV}	ALE low to valid data in	—	282	—	165	—	$7t_{CLCL}$–10	ns
t_{AVDV}	Address to valid data in	—	323	—	190	—	$8t_{CLCL}$–10	ns
t_{LLWL}	ALE low to $\overline{RD}$ or $\overline{WR}$ low	105	145	55	95	$3t_{CLCL}$–20	$3t_{CLCL}$+20	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ low	146	—	80	—	$4t_{CLCL}$–20	—	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	26	—	10	—	t_{CLCL}–15	—	ns
t_{WHQX}	Data hold after $\overline{WR}$	31	—	15	—	t_{CLCL}–10	—	ns
t_{RLAZ}	$\overline{RD}$ low to address float	—	0	—	0	—	0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ high to ALE high	26	57	10	40	t_{CLCL}–15	t_{CLCL}+15	ns

SERIAL PORT TIMING: SHIFT REGISTER MODE

Symbol	Parameter	24 MHz Clock Min	24 MHz Clock Max	40 MHz Clock Min	40 MHz Clock Max	Variable Oscillator (3.5-40 MHz) Min	Variable Oscillator (3.5-40 MHz) Max	Unit
t_{XLXL}	Serial port clock cycle time	490	510	290	310	$12t_{CLCL}-10$	$12t_{CLCL}+10$	ns
t_{QVXH}	Output data setup to clock rising edge	406	—	240	—	$10t_{CLCL}-10$	—	ns
t_{XHQX}	Output data hold after clock rising edge	73	—	40	—	$2t_{CLCL}-10$	—	ns
t_{XHDX}	Input data hold after clock rising edge	0	—	0	—	0	—	ns
t_{XHDV}	Clock rising edge to input data valid	—	417	—	250	—	$10t_{CLCL}$	ns

EXTERNAL CLOCK DRIVE CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit
$1/t_{CLCL}$	Oscillator Frequency	3.5	40	MHz
t_{CHCX}	High time	10	—	ns
t_{CLCX}	Low time	10	—	ns
t_{CLCH}	Rise time	—	10	ns
t_{CHCL}	Fall time	—	10	ns

IING WAVEFORMS

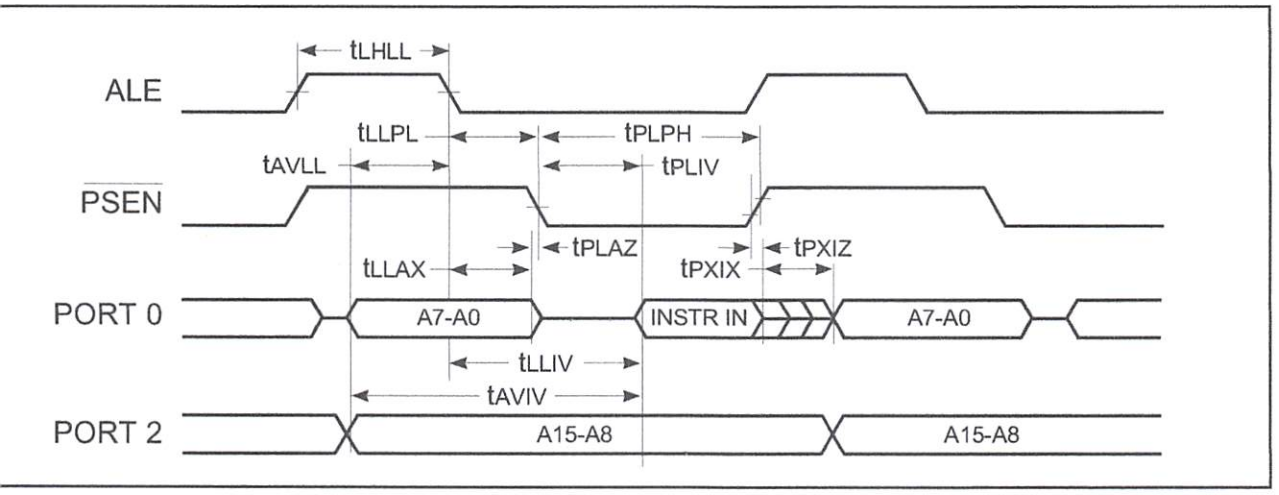


Figure 32. External Program Memory Read Cycle

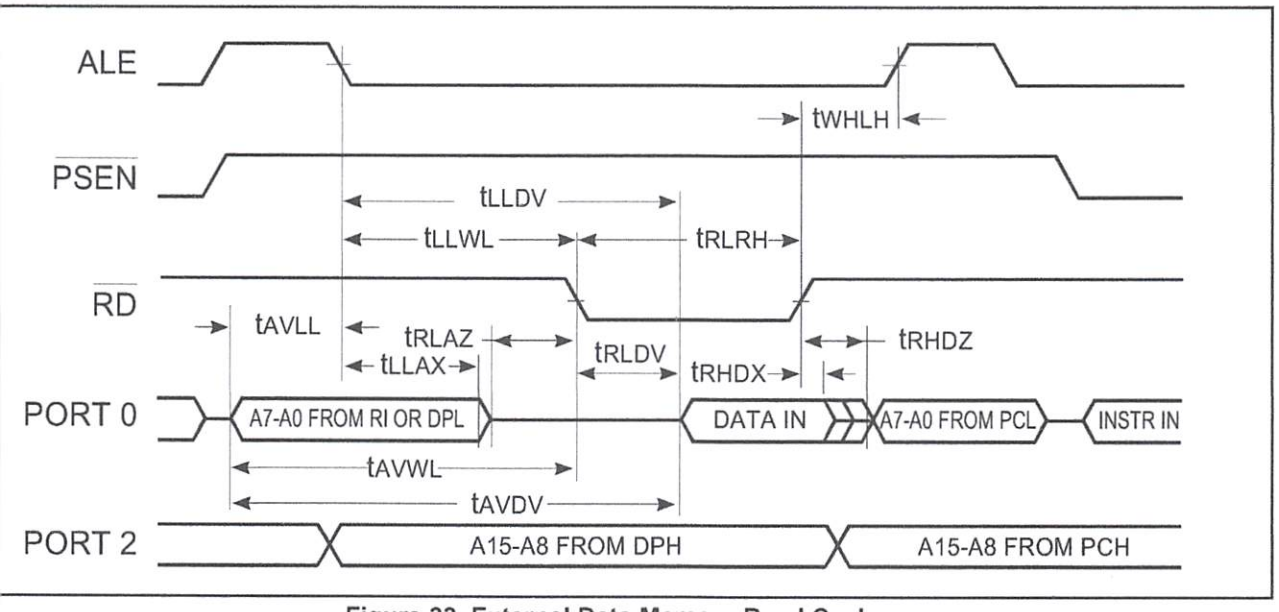


Figure 33. External Data Memory Read Cycle

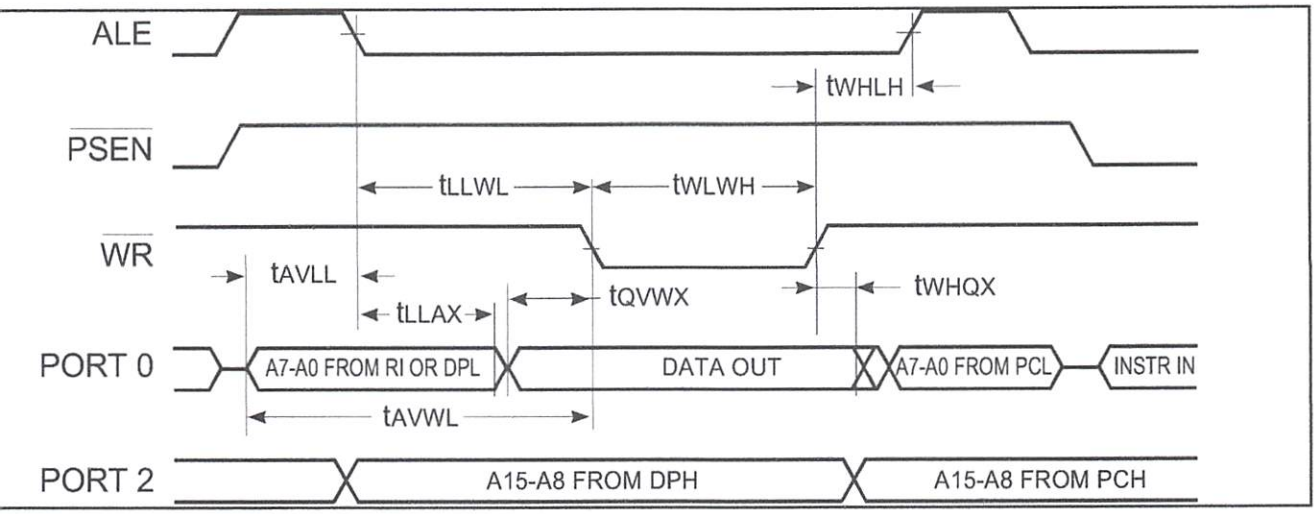


Figure 34. External Data Memory Write Cycle

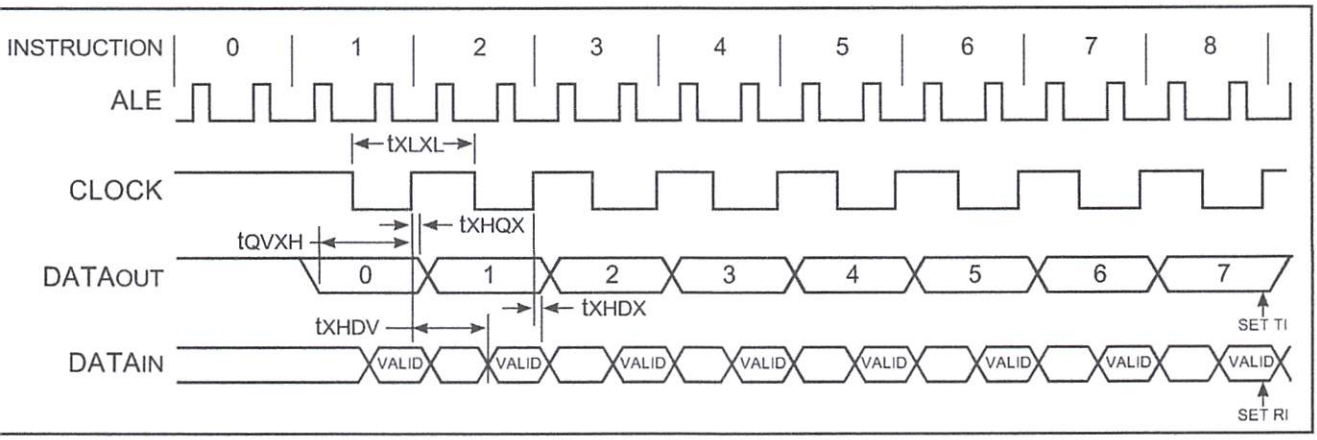


Figure 35. Shift Register Mode Timing Waveform

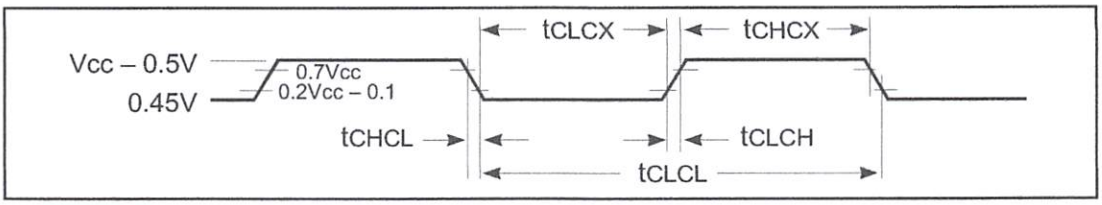


Figure 36. External Clock Drive Waveform

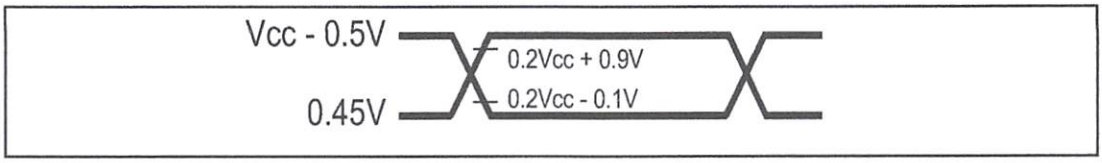


Figure 37. AC Test Point

Note:
1. AC inputs during testing are driven at VCC – 0.5V for logic "1" and 0.45V for logic "0". Timing measurements are made at V_{IH} min for logic "1" and max for logic "0".

ORDERING INFORMATION
Commercial Range: 0°C to +70°C

Speed	Order Part Number	Package
12 MHz	IS89C51-12PL	PLCC – Plastic Leaded Chip Carrier
	IS89C51-12W	600-mil Plastic DIP
	IS89C51-12PQ	PQFP
24 MHz	IS89C51-24PL	PLCC – Plastic Leaded Chip Carrier
	IS89C51-24W	600-mil Plastic DIP
	IS89C51-24PQ	PQFP
40 MHz	IS89C51-40PL	PLCC – Plastic Leaded Chip Carrier
	IS89C51-40W	600-mil Plastic DIP
	IS89C51-40PQ	PQFP

ORDERING INFORMATION
Industrial Range: –40°C to +85°C

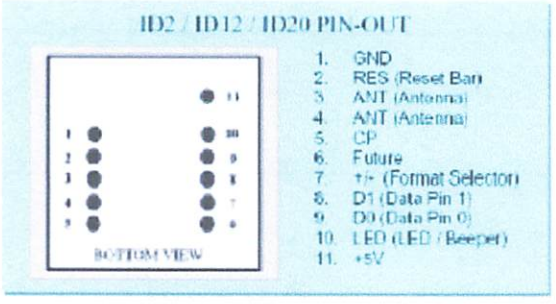
Speed	Order Part Number	Package
12 MHz	IS89C51-12PLI	PLCC – Plastic Leaded Chip Carrier
	IS89C51-12WI	600-mil Plastic DIP
	IS89C51-12PQI	PQFP
24 MHz	IS89C51-24PLI	PLCC – Plastic Leaded Chip Carrier
	IS89C51-24WI	600-mil Plastic DIP
	IS89C51-24PQI	PQFP
40 MHz	IS89C51-40PLI	PLCC – Plastic Leaded Chip Carrier
	IS89C51-40WI	600-mil Plastic DIP
	IS89C51-40PQI	PQFP

ISSI®

Integrated Silicon Solution, Inc.
2231 Lawson Lane
Santa Clara, CA 95054
Fax: (408) 588-0806
Toll Free: 1-800-379-4774
http://www.issi.com

The contactless R/O Readers (125 kHz) ID-2, ID12 and ID-20

The ID2, ID12 and ID20 are similar to the ID0,ID10 and ID15 NK(ii) series devices, but they have extra pins which allow Magnetic Emulation output to be included in the functionality. The ID-12 and ID-20 come with internal antennas, and have read ranges of 12cm and 16cm, respectively. With an external antenna, the ID-2 can deliver read ranges of up to 25 cm. All tree readers support ASCII, Wiegand 16 and Magnetic ABA Track2 data format.



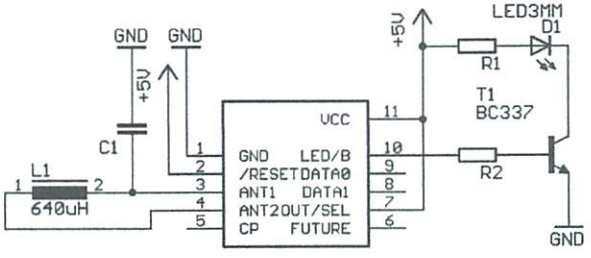
Operational and Physical Characteristics

Parameters	ID-2	ID-12	ID-20
React Range	NA (no internal antenna)	12+ cm	15+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125kHz	125kHz	125kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible'
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 V DC, 13 mA nominal	5 V DC, 30 mA nominal	5 V DC, 55 mA nominal
I/O Output Current	+/- 200 mA PK	-	-
Voltage Supply Range	+4,6V through +5,4 V	+4,6V through +5,4 V	+4,6V through +5,4 V
Operating Temperature	-10 to +50 °C	-10 to +50 °C	-10 to +50 °C

http://www.elateceurope.com, Email: support@elateceurope.com

Pin Description and Output Data Format

Pin No.	Description	ASCII	Magnetic Emulation	Wiegand 26
Pin 1	Zero Volts	GND	GND	GND
Pin 2	Strap to +5V	Reset bar	Reset Bar	Reset bar
Pin :3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present	No function
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to 5V
Pin8	Data 1	CMOS	Data	One Output
Pin 9	Data 0	TTL Data interval	Clock	Zero Output
Pin 10	3.1 kHz Logic	Beeper/LED	Beeper/LED	Beeper/LED
Pin 11	DC Voltage Supply	+5V	+5V	+5V



Data Format

Output Format - Serial ASCII 9600,8,1

02	10 ASCII Data Characters	Checksum	CR	LF	03

The checksum is the result of the 'exclusive' OR of the 5 Binary Data bytes (The 10 ASCII data characters)

Data Structure Wiegand 16 Bit1 mS/50uS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	P
Even Parity (E)													ODD Parity (O)												

P= Parity Start Bit and Stop Bit

http://www.elateceurope.com, Email: support@elateceurope.com

Coil Design Guide for ID-O

The recommended inductance is 1,08mH to be used with a tuning capacitor of 1n5.

In general the bigger the antenna the better, provided tie reader is generating enough field strength excite the tag. The ID-0 is fairly low power so a maximum coil size of 15x15 cm is recommended if it is intended to read ISO cards. If the reader is intended to read glass tags the maximum coil size should be smaller, say 10x10 cm. There is a science here to determine the exact size of the antenna but there are so many variables that an element of 'Try it and see' is unavoidable. If the reader is located in a position where there is a lot of heavy interference then less range cannot be avoided. In this situation the coil should be made smaller to increase the field strength and coupling.

The Tuning Capacitor

Do not forget that the choice of tiling capacitor can also substantially affect the quality of your system. The loss in an ID12 series antenna is required to be fairly high to limit the series current and this will hide a lot of the shortcomings of the capacitor, but for quality and reliability and repeatability the following capacitors are recommend.

Type: Polypropylene, COG/NPO, polyethelene sulfide, mica, polycarbonate.
Voltage Working. A capacitor capable of withstanding the RMS voltage at 125 KHz MUST be chosen.
The working voltage will depend on the coil design. I suggest the designer start with rugged 1n5 polypropylene 630v capacitor to do his experiments and the come down to a suitable size/value.
The capacitor manufacturer will supply information on her capacitors. Does not s simply go by the DC voltage? This means little. A tolerance of 2% is preferable. A tolerance of 5% is acceptable.

Step to Design an Antenna

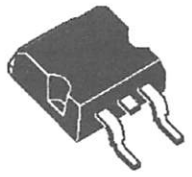
1. If the target coil is defined by the enclosure size and is less han 4x4cm then simply wind your coil to 1.08mH and skip steps 2-4.
2. If your target transducer is very small then do not wind a coil bigger than 10x10 cm.
3. If there is heavy interference then try a large coil first and then reduce the coil size and see if there is any improvement.
4. If the coil is placed near to a large expanse of metal his may cause the inductance to change did this must be allowed for. Normally the inductance drops so a higher initial inductance is required. Also the loss in the coil can be high so consider using a smaller coil to increase the field strength and at the same time lesson the loss due to the metal
5. The ID -O is a new device and is conservatively rated. At the moment the maximum peak current allowed is 7OmA. This corresponds to a peak-to-peak voltage at the antenna of 100vPKPK. If the antenna voltage is higher than this the designer has two options. One is to risk it, the other is to lower the antenna Q by adding a series resistor in the coil.

http://www.elateceurope.com, Email: support@elateceurope.com

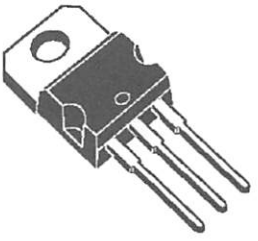
L7800AB/AC SERIES

PRECISION 1A REGULATORS

- OUTPUT CURRENT IN EXCESS OF 1 A
- OUTPUT VOLTAGES OF 5; 6; 8; 9; 12; 15; 18; 20; 24V
- THERMAL OVERLOAD PROTECTION
- OUTPUT TRANSITION SOA PROTECTION
- 2% OUTPUT VOLTAGE TOLERANCE
- GUARANTEED IN EXTENDED TEMPERATURE RANGE



D^2PAK

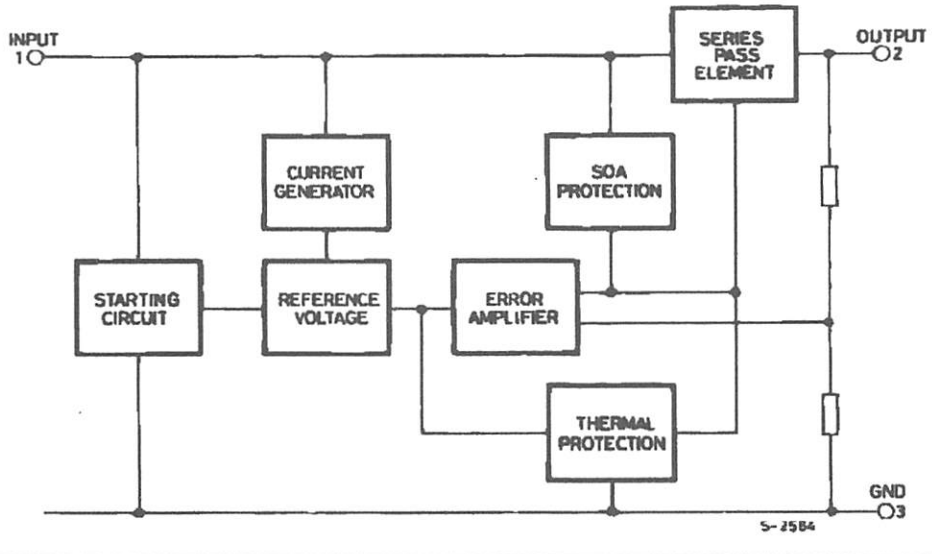


TO-220

DESCRIPTION

The L7800A series of three-terminal positive regulators is available in TO-220 and D^2PAK packages and several fixed output voltages, making it useful in a wide range of applications.These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

BLOCK DIAGRAM



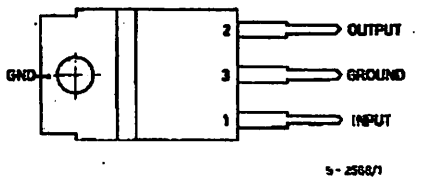
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_I	DC Input Voltage (for V_O = 5 to 18V) (for V_O = 20, 24V)	35 40	V V
I_o	Output Current	Internally limited	
P_{tot}	Power Dissipation	Internally limited	
T_{op}	Operating Junction Temperature Range (for **L7800AC**) (for **L7800AB**)	0 to 150 -40 to 125	°C °C
T_{stg}	Storage Temperature Range	- 65 to 150	°C

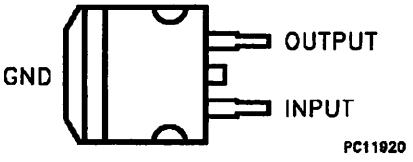
THERMAL DATA

Symbol	Parameter		D^2PAK	TO-220	Unit
$R_{thj\text{-}case}$	Thermal Resistance Junction-case	Max	3	3	°C/W
$R_{thj\text{-}amb}$	Thermal Resistance Junction-ambient	Max	62.5	50	°C/W

CONNECTION DIAGRAM AND ORDERING NUMBERS (top view)



TO-220



D^2PAK

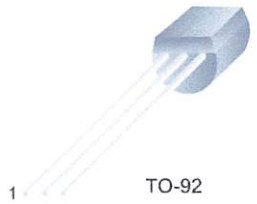
Type	TO-220	D^2PAK (*)	Output Voltage
L7805AB	L7805ABV	L7805ABD2T	5V
L7805AC	L7805ACV	L7805ACD2T	5V
L7806AB	L7806ABV	L7806ABD2T	6V
L7806AC	L7806ACV	L7806ACD2T	6V
L7808AB	L7808ABV	L7808ABD2T	8V
L7808AC	L7808ACV	L7808ACD2T	8V
L7809AB	L7809ABV	L7809ABD2T	9V
L7809AC	L7809ACV	L7809ACD2T	9V
L7812AB	L7812ABV	L7812ABD2T	12V
L7812AC	L7812ACV	L7812ACD2T	12V
L7815AB	L7815ABV	L7815ABD2T	15V
L7815AC	L7815ACV	L7815ACD2T	15V
L7818AB	L7818ABV		18V
L7818AC	L7818ACV		18V
L7820AB	L7820ABV		24V
L7820AC	L7820ACV		24V
L7824AB	L7824ABV		
L7824AC	L7824ACV		

(*) AVAILABLE IN TAPE AND REEL WITH "-TR" SUFFIX

FAIRCHILD
SEMICONDUCTOR®

KSP10

VHF/UHF transistor



TO-92

1. Base 2. Emitter 3. Collector

NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings T_a=25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage	30	V
V_{CEO}	Collector-Emitter Voltage	25	V
V_{EBO}	Emitter-Base Voltage	3.0	V
P_C	Collector Power Dissipation (T_a=25°C)	350	mW
	Derate above 25°C	2.8	mW/°C
P_C	Collector Power Dissipation (T_C=25°C)	1.0	W
	Derate above 25°C	8.0	W/°C
T_J	Junction Temperature	150	°C
T_{STG}	Storage Temperature	-55~150	°C
Rth(j-c)	Thermal Resistance, Junction to Case	125	°C/W
Rth(j-a)	Thermal Resistance, Junction to Ambient	357	°C/W

Electrical Characteristics T_a=25°C unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
BV_{CBO}	Collector-Base Breakdown Voltage	I_C=100μA, I_E=0	30		V
BV_{CEO}	Collector-Emitter Breakdown Voltage	I_C=1mA, I_B=0	25		V
BV_{EBO}	Emitter-Base Breakdown Voltage	I_E=10μA, I_C=0	3.0		V
I_{CBO}	Collector Cut-off Current	V_{CB}=25V, I_E=0		100	nA
I_{EBO}	Emitter Cut-off Current	V_{EB}=2V, I_C=0		100	nA
h_{FE}	DC Current Gain	V_{CE}=10V, I_C=4mA	60		
V_{CE} (sat)	Collector-Emitter Saturation Voltage	I_C=4mA, I_B=0.4mA		0.5	V
V_{BE} (on)	Base-Emitter On Voltage	V_{CE}=10V, I_C=4mA		0.95	V
f_T	Current Gain Bandwidth Product	V_{CE}=10V, I_C=4mA, f=100MHz	650		MHz
C_{ob}	Output Capacitance	V_{CB}=10V, I_E=0, f=1MHz		0.7	pF
C_{rb}	Collector Base Feedback Capacitance	V_{CB}=10V, I_E=0, f=1MHz	0.35	0.65	pF
$C_{c\cdot rbb'}$	Collector Base Time Constant	V_{CB}=10V, I_C=4mA, f=31.8MHz		9.0	ps

* Pulse Test: PW≤300μs, Duty Cycle≤2%

Typical Characteristics

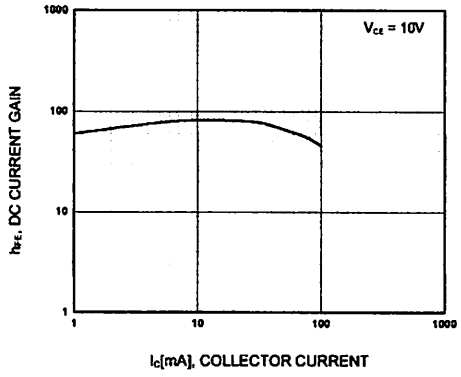


Figure 1. DC current Gain

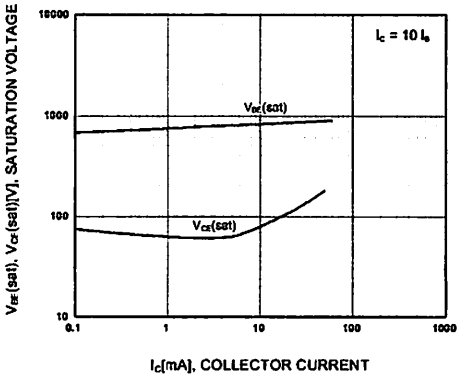


Figure 2. Base-Emitter Saturation Voltage Collector-Emitter Saturation Voltage

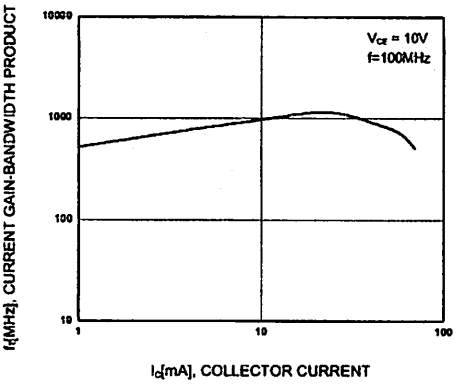


Figure 3. Current Gain Bandwidth Product

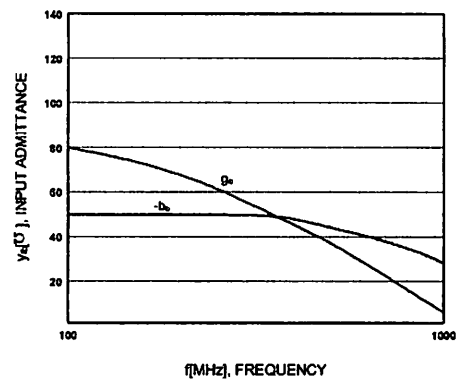


Figure 4. Rectangular Form

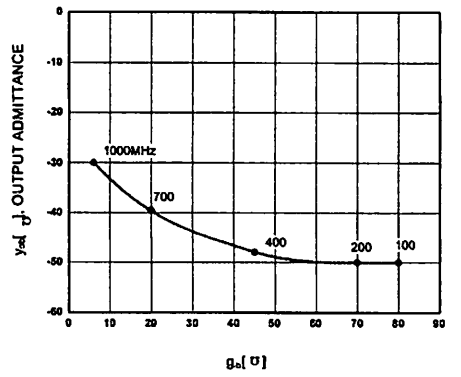


Figure 5. Polar Form

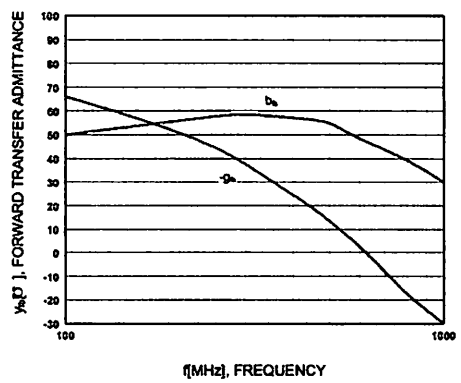


Figure 6. Rectangular Form

Typical Characteristics (Continued)

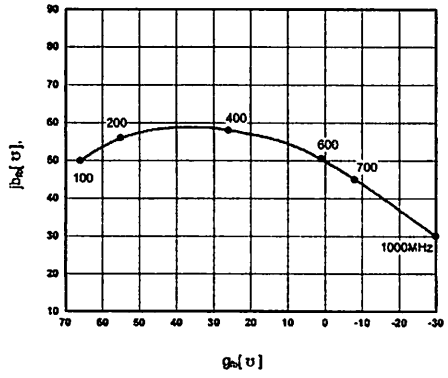


Figure 7. Polar Form

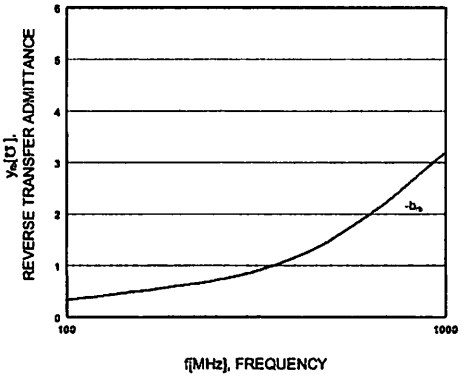


Figure 8. Rectangular Form

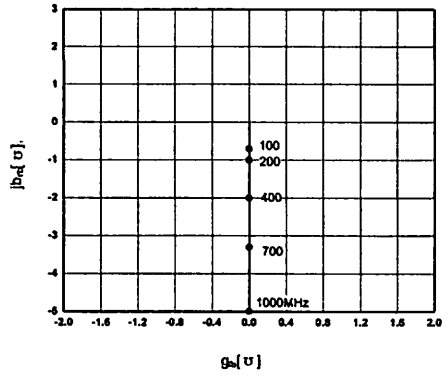


Figure 9. Polar Form

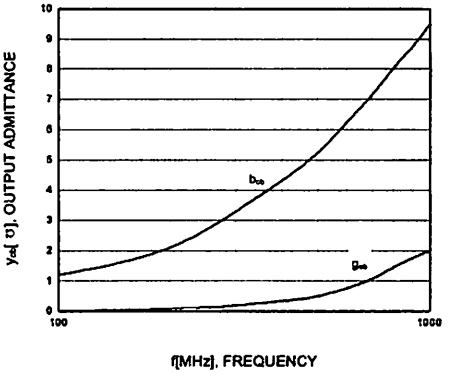


Figure 10. Rectangular Form

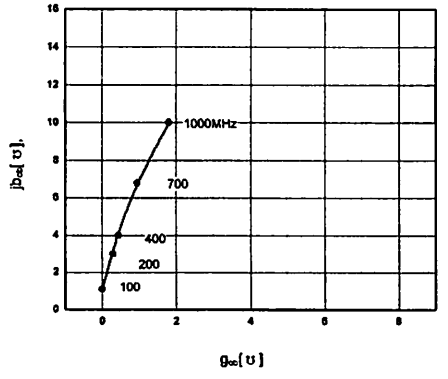
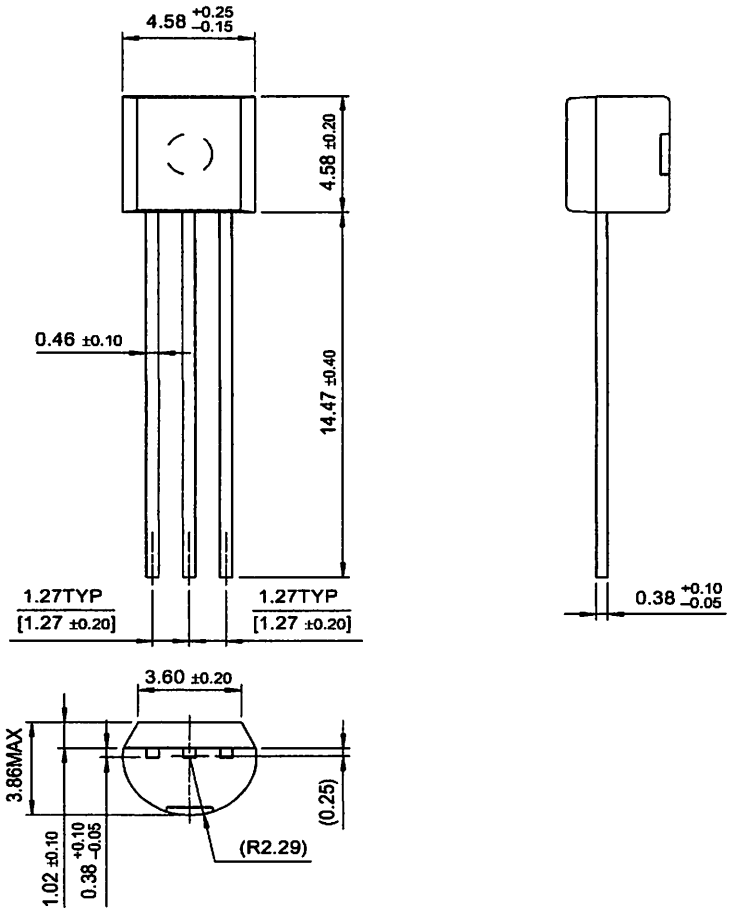


Figure 11. Polar Form

Package Dimensions

TO-92



Dimensions in Millimeters

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FACT™	ImpliedDisconnect™	PACMAN™	SPM™
ActiveArray™	FACT Quiet series™	ISOPLANAR™	POP™	Stealth™
Bottomless™	FAST®	LittleFET™	Power247™	SuperSOT™-3
CoolFET™	FASTr™	MicroFET™	PowerTrench®	SuperSOT™-6
CROSSVOLT™	FRFET™	MicroPak™	QFET™	SuperSOT™-8
DOME™	GlobalOptoisolator™	MICROWIRE™	QS™	SyncFET™
EcoSPARK™	GTO™	MSX™	QT Optoelectronics™	TinyLogic™
E^2CMOS™	HiSeC™	MSXPro™	Quiet Series™	TruTranslation™
EnSigna™	I^2C™	OCX™	RapidConfigure™	UHC™
Across the board. Around the world.™		OCXPro™	RapidConnect™	UltraFET®
The Power Franchise™		OPTOLOGIC®	SILENT SWITCHER®	VCX™
Programmable Active Droop™		OPTOPLANAR™	SMART START™	

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

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 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, or (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 significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

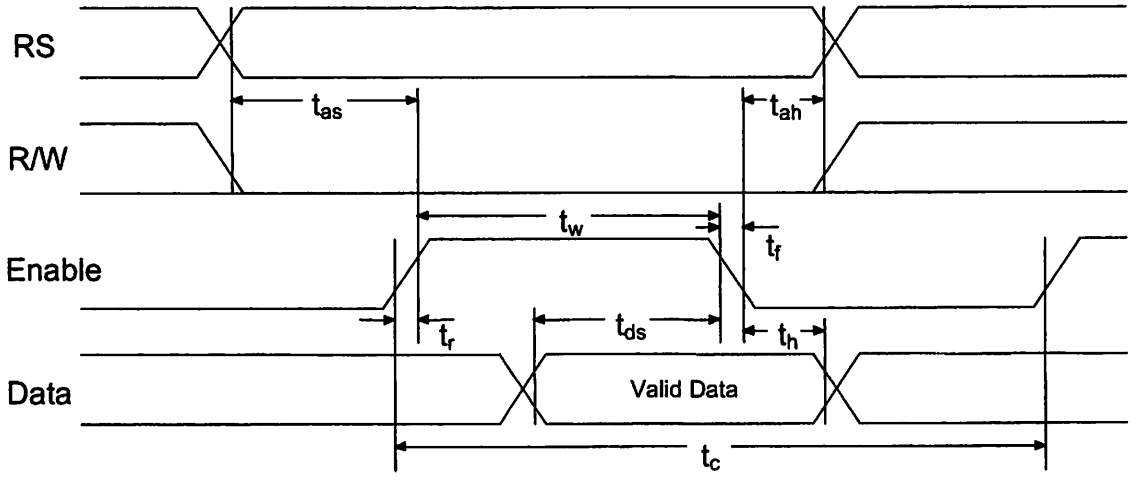
PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

struction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
	0	0	0	0	0	0	0	0	0	0	No Operation	0
Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero, returns shifted display to original position. DDRAM contents remains unchanged.	3
Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction, and specifies automatic shift.	3
y Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
on Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N,M) and voltage generator control (G).	3
GRAM Addr	0	0	0	1	Character Generator RAM						Sets CGRAM Address	3
DRAM Addr	0	0	1	Display Data RAM Address							Sets DDRAM Address	3
Flag & Addr	0	1	BF	Address Counter							Reads Busy Flag & Address Counter	0
Data	1	0	Read Data								Reads data from CGRAM or DDRAM	3
Data	1	1	Write Data								Writes data from CGRAM or DDRAM	3

Write Cycle

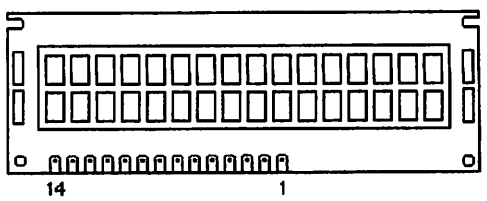


arameter	Symbol	Min [1]	Typ [1]	Max [1]	Unit
Cycle Time	t_c	500	-	-	ns
Pulse Width	t_w	230	-	-	ns
Rise/Fall Time	t_r, t_f	-	-	20	ns
s Setup Time	t_{as}	40	-	-	ns
s Hold Time	t_{ah}	10	-	-	ns
tup Time	t_{ds}	80	-		ns
old Time	t_h	10	-	-	ns

he above specifications are a indication only. Timing will vary from manufacturer ufacturer.

4 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x acter, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x acter, 2 lines x 40 character etc. modules compatible with the HD44780 LCD

(2)



Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable *(Strobe)*
7	D0	I/O	Data *LSB*
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data *MSB*