

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
PROGRAM STUDI ELEKTRONIKA**



SKRIPSI

**RANCANG BANGUN ALAT PERAJANG TEMPE SECARA
OTOMATIS BERBASIS MIKROKONTROLLER AT89C51**

Disusun Oleh :

WAHYU ARDIANSYAH

NIM : 99.17.107

MARET 2005

LEMBAR PERSETUJUAN

RANCANG BANGUN ALAT PERAJANG TEMPE SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89C51

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat Guna
Memperoleh Gelar Sarjana Teknik*

Disusun Oleh :

WAHYU ARDIANSYAH
NIM : 99.17.107

Diperiksa dan disetujui

Dosen Pembimbing I

Dosen Pembimbing II

(Ir. Usman Djuanda, MM)

(Ir. Mimien Mustikawati)

Mengetahui,
Ketua Jurusan Teknik Elektro

(Ir. F. Yudi Limpraptono, MT)
NIP. P. 1079500274

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

MILIK
PERPUSTAKAAN
ITN MALANG



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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Wahyu Ardiansyah
NIM : 9917107
Jurusan : Teknik Elektro S-1
Program Studi : Elektronika
Judul Skripsi : Rancang Bangun Alat Perajang Tempe
Secara Otomatis Berbasis Mikrokontroller
AT89C51
Dipertahankan dihadapan majelis penguji Skripsi jenjang Strata Satu (S-1) pada :
Hari : Selasa
Tanggal : 29 Maret 2005
Dengan nilai : 81 (Delapan Puluh Satu) *Sum*



Panitia Ujian Skripsi

Ketua

[Signature]
Ir. Mochtar Asroni, MSME
NIP. P. 1018100036

Sekretaris

[Signature]
Ir. F. Yudi Limpraptono, MT
NIP.P. 1039500274

Anggota Penguji

Penguji Pertama

[Signature]
Ir. Sidik Noerjahjono, MT

Penguji Kedua

[Signature]
Joseph Dedy Irawan, ST. MT

ABSTRAKSI

RANCANG BANGUN ALAT PERAJANG TEMPE SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89C51

(Wahyu Ardiansyah, NIM 9917107, Jurusan Teknik Elektro / Teknik Elektronika)
(Dosen Pembimbing : Ir. Usman Djuanda, MM; Ir. Mimien Mustikawati)

Perajangan tempe masih dilakukan secara manual oleh tenaga manusia. Hal ini akan menyebabkan untuk pembuatan dalam produksi skala yang lebih besar membutuhkan waktu yang lama dan ketebalan dari irisan berdasarkan dari pemikiran si perajang. Dari sisi ini pula keselamatan menjadi beresiko, sehingga dalam hal ini dibutuhkan suatu sistem perajang tempe yang cepat, tepat dan aman yang bekerja secara otomatis dengan memanfaatkan mikrokontroller AT89C51.

Alat yang dirancang terdiri atas mikrokontroller AT89C51, keypad, LCD, LED infra red dan photodiode, limit switch, PPI 8255, latch, dekoder, motor DC dan motor stepper. Mikrokontroller berfungsi pengendali bekerjanya seluruh sistem.

Adapun prinsip kerja dari alat ini yaitu dengan menekan tombol keypad untuk menentukan ketebalan perajangan tempe yang kemudian akan ditampilkan ke LCD. Tebal tipisnya perajangan diatur oleh mikrokontroller AT89C51 dengan mengontrol pergerakan motor stepper. Sedangkan LED infra red dan photodiode berfungsi sebagai pendeteksi ada atau tidak adanya tempe. Limit switch digunakan sebagai pembatas pergerakan motor DC (pisau perajang). Hasil yang dicapai dari pengujian menunjukkan alat yang dirancang dapat bekerja dengan baik sesuai dengan yang diharapkan.

KATA PENGANTAR

Segala puji dan syukur penulis ucapkan kehadiran Allah SWT karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan laporan skripsi ini. Laporan skripsi ini disusun sebagai salah satu persyaratan untuk mendapatkan gelar Sarjana Teknik

Laporan skripsi yang disusun oleh penulis ini mengambil judul :

“RANCANG BANGUN ALAT PERAJANG TEMPE SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89C51”, yang berisikan mulai latar belakang masalah, landasan teori, perancangan dan pengujian alat. Laporan skripsi ini merupakan tanggung jawab tertulis atas ilmu pengetahuan yang didapat selama penulis mengikuti kuliah di Institut Teknologi Nasional Malang.

Dalam penyusunan laporan skripsi ini ada banyak pihak yang telah membantu. Dan pada kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Mochtar Asroni, MSME, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro Institut Teknologi Nasional Malang,
4. Ir. Usman Djuanda, MM, selaku Dosen Pembimbing I.
5. Ir. Mimien Mustikawati, selaku Dosen Pembimbing II.

6. Kedua orang tua saya tercinta yang selalu mendoakan dan penuh kasih sayang memberi dorongan selama penulis menyelesaikan skripsi ini.
7. Warga Gajayana 28A dan seluruh teman-teman saya atas kekeluargaan dan kerjasamanya.

Penulis menyadari laporan skripsi ini masih banyak kekurangan dalam penyusunannya, maka penulis mengharapkan masukan dan kritik yang membangun agar dapat sempurna laporan skripsi ini.

Dan akhirnya, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penulis membuat kesalahan secara tidak sengaja dan semoga laporan skripsi ini dapat bermanfaat bagi yang memerlukannya dimasa-masa mendatang.

Malang, Maret 2005

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
BERITA ACARA UJIAN SKRIPSI	iii
ABSTRAKSI	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	2
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Metodologi	3
1.6. Sistematika Penulisan.....	3
BAB II DASAR TEORI	4
2.1. Minimum Sistem AT89C51	4
2.1.1. Mikrokontroller AT89C51	4
2.1.2. Memori	15

2.2. PPI 8255	17
2.3. LCD	23
2.4. Keypad.....	26
2.5. LED Infra Red	27
2.6. Photodiode.....	28
2.7. Limit Switch	29
2.8. Motor Stepper.....	30
2.9. Motor DC	32
 BAB III PERENCANAAN DAN PEMBUATAN ALAT.....	34
3.1. Pendahuluan	34
3.2. Blok Diagram Rangkaian	34
3.3. Perencanaan Perangkat Keras.....	35
3.3.1. Mikrokontroller AT89C51	36
3.3.2. Rangkaian PPI 8255	40
3.3.3. Latch.....	41
3.3.4. Dekoder dan Peta Alamat.....	42
3.3.5. Rangkaian LCD	45
3.3.6. Rangkaian Keypad	46
3.3.7. Rangkaian LED Infra Red dan Photodiode.....	47
3.3.8. Rangkaian Driver Motor DC.....	49
3.4. Perencanaan Perangkat Lunak.....	51

BAB IV	PENGUJIAN ALAT.....	53
4.1.	Umum.....	53
4.2.	Pengujian Sistem Mikrokontroller	53
4.2.1.	Tujuan.....	53
4.2.2.	Peralatan Yang Dibutuhkan	53
4.2.3.	Prosedur Pengujian.....	54
4.2.4.	Hasil Pengujian dan Analisis.....	55
4.3.	Pengujian Keypad.....	55
4.3.1.	Tujuan.....	55
4.3.2.	Peralatan Yang Dibutuhkan	56
4.3.3.	Prosedur Pengujian.....	56
4.3.4.	Hasil Pengujian dan Analisis.....	59
4.4.	Pengujian LCD	59
4.4.1.	Tujuan.....	59
4.4.2.	Peralatan Yang Dibutuhkan	60
4.4.3.	Prosedur Pengujian.....	60
4.4.4.	Hasil Pengujian dan Analisis.....	62
4.5.	Pengujian PPI 8255	62
4.5.1.	Tujuan.....	62
4.5.2.	Peralatan Yang Dibutuhkan	63
4.5.3.	Prosedur Pengujian.....	63
4.5.4.	Hasil Pengujian dan Analisis.....	64

4.6. Pengujian Relay Driver Motor DC.....	65
4.6.1. Tujuan.....	65
4.6.2. Peralatan Yang Dibutuhkan	65
4.6.3. Prosedur Pengujian.....	65
4.6.4. Hasil Pengujian dan Analisis.....	66
4.7. Pengujian Motor DC	67
4.7.1. Tujuan.....	67
4.7.2. Peralatan Yang Dibutuhkan	67
4.7.3. Prosedur Pengujian.....	67
4.7.4. Hasil Pengujian dan Analisis.....	68
4.8. Pengujian Keseluruhan Sistem	68
4.8.1. Tujuan.....	68
4.8.2. Peralatan Yang Dibutuhkan	68
4.8.3. Prosedur Pengujian.....	68
4.8.4. Hasil Pengujian dan Analisis.....	69
 BAB V PENUTUP	 71
5.1. Kesimpulan.....	71
5.2. Saran	72

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
2-1 Konfigurasi Pin AT89C51	5
2-2 Blok Disgram PPI 8255	17
2-3 Konfigurasi pin PPI 8255.....	18
2-4 Control Word PPI 8255	20
2-5 Diagram Blok Sistem LCD TM 1632	24
2-6 Penampang Dasar Keypad	26
2-7 LED Infra Red.....	28
2-8 Photodioda	29
2-9 Limit Switch.....	29
2-10 Bagan Dasar Motor Stepper.....	30
2-11 Proses Pemutaran Motor Stepper 90°.....	31
2-12 Konstruksi Motor DC	32
3-1 Blok Diagram Rangkaian.....	34
3-2 Rangkaian Pewaktuan	37
3-3 Rangkaian Power-On Reset	38
3-4 Rangkaian Mikrokontroller.....	39
3-5 Rangkaian PPI 8255.....	40
3-6 Rangkaian Latch	42
3-7 Rangkaian Dekoder Alamat.....	45
3-8 Rangkaian LCD	46

3-9	Rangkaian Keypad	47
3-10	Rangkaian Sensor LED Infra Red dan Photodiode.....	48
3-11	Rangkaian Driver Motor DC	49
4-1	Diagram Blok Pengujian Mikrokontroller	55
4-2	Pengujian Rangkaian Keypad	58
4-3	Blok Pengujian LCD	62
4-4	Hasil Pengujian LCD	62
4-5	Diagram Blok Pengujian PPI	64
4-6	Rangkaian Pengujian Relay	66
4-7	Rangkaian Pengujian Motor DC	67
4-8	Blok Diagram Rangkaian Pengujian Keseluruhan Sistem.....	69

DAFTAR TABEL

Tabel	Halaman
2-1 Fungsi Pengganti Dari Port 3	7
2-2 Keluarga MCS 51	9
2-3 Nama dan Alamat Register Pada Register Fungsi Khusus	11
2-4 Operasi Dasar PPI 8255	21
2-5 Pin-pin LCD dan Konfigurasinya	24
3-1 Tabel Fungsi Latch 74LS573	41
3-2 Pemetaan Alamat	43
3-3 Dekoder 74LS138	44
4-1 Hasil Pengujian Sistem Mikrokontroller.....	55
4-2 Hasil Pengujian Keypad.....	59
4-3 Hasil Pengujiana PPI.....	65
4-4 Hasil Pengujian Relay	66

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di Indonesia banyak sekali ditemukan beraneka ragam makanan khas daerah, salah satunya adalah kripik tempe yang berasal dari kota Malang. Di daerah kota Malang (di Kelurahan Sanan) banyak pengrajin makanan khas ini, baik itu pengrajin yang berskala besar ataupun yang berskala kecil.

Untuk pembuatan kripik tempe, perajangan (pengirisan) masih dilakukan secara manual dan dilakukan oleh tenaga manusia. Hal ini akan menyebabkan untuk pembuatan dalam jumlah yang besar dan dikirim keluar kota/pulau membutuhkan waktu lama dan ketebalan dari irisan berdasarkan dari pemikiran si pemotong. Dan dari sisi ini pula keselamatan menjadi beresiko, sehingga dibutuhkan suatu alat perajangan tempe yang cepat, tepat dan aman.

Perkembangan teknologi khususnya elektronika pada saat ini juga sangat berkembang dengan cepat. Salah satu pemanfaatannya adalah untuk proses perajangan tempe secara otomatis. Sistem dari perajangan ini memanfaatkan seperangkat minimum sistem AT89C51 ditunjang dengan beberapa komponen pendukung lainnya, sehingga dari perencanaan dan pembuatan alat ini dapat menghasilkan tebal tipisnya rajangan tempe yang sesuai kita kehendaki.

1.2. Rumusan Masalah

Rancang bangun alat perajang tempe secara otomatis berbasis mikrokontroller AT89C51 terdapat masalah, yaitu :

- Bagaimana cara mengatur kecepatan gerak motor stepper (belt conveyor) agar dihasilkan potongan keripik dengan ketebalan sesuai dengan yang diinginkan.

1.3. Batasan Masalah

Disebabkan karena luasnya masalah yang akan dipakai dalam perancangan dan pembuatan alat ini maka diberikan batasan-batasan sebagai berikut :

- a. Dalam laporan skripsi ini tidak membahas secara mendetail mengenai motor DC ataupun motor DC stepper.
- b. Aspek mekanis tidak dibahas.
- c. Catu daya tidak dibahas.
- d. Ketebalan yang diatur 1mm dan 2mm.
- e. Tempe sebagai bahan baku pengujian besarnya disesuaikan dengan mekanik dan dipilih tempe dengan kualitas yang bagus.

1.4. Tujuan

Tujuan rancang bangun alat ini adalah untuk membuat perajang tempe yang dapat menghasilkan ketebalan tertentu dengan mengatur kecepatan atau pergerakan motor stepper (belt conveyor) yang bergerak.

1.5. Metodologi

Metodologi penelitian yang dipakai dalam pembuatan laporan skripsi ini adalah :

- a. Studi literature.
- b. Perencanaan dan pembuatan alat.
- c. Pelaksanaan uji coba alat.
- d. Penyusunan laporan skripsi.

1.6. Sistematika Penulisan

- | | |
|---------|---|
| BAB I | Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi dan sistematika penulisan. |
| BAB II | Menjelaskan teori yang berkaitan dengan komponen penunjang yang digunakan dalam rancang bangun alat ini. |
| BAB III | Menjelaskan tentang perencanaan dan pembuatan alat. |
| BAB IV | Pengujian alat yang sudah terealisasi. |
| BAB V | Kesimpulan dan saran. |
-

BAB II

DASAR TEORI

Dalam merencanakan dan merealisasikan sistem ini dibutuhkan pemahaman mengenai pengetahuan yang berhubungan dengan aplikasi tersebut. Pemahaman tersebut akan sangat bermanfaat dalam perancangan perangkat keras maupun perangkat lunak sistem. Adapun pengetahuan yang mendukung perencanaan dan realisasi alat antara lain sistem mikrokontroller AT89C51, PPI, LCD, keypad, limit switch, motor DC, infra red dan photodiode.

2.1 Minimum Sistem AT89C51

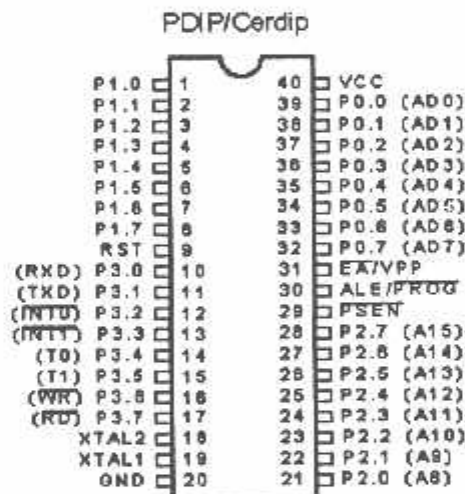
2.1.1. Mikrokontroller AT89C51

Mikrokontroller AT89C51 merupakan salah satu anggota keluarga MCS-51, yaitu suatu komponen produksi ATMEL yang berorientasi kontrol (*microcontroller*). Di dalam *chip* mikrokontroller AT89C51 ini sudah tersedia berbagai macam peralatan pendukung mikroprosesor seperti RAM, *serial port*, *bus-bus* data dan lainnya yang membuat pemakai *chip* ini dapat menekan penambahan komponen pendukung. Spesifikasi perangkat keras dari mikrokontroller AT89C51 adalah sebagai berikut :

- CPU (*Central processing unit*) dengan lebar data 8 *bit*.
- Prosesor *Boolean* untuk operasi logika 1 *bit*.
- Pembangkit *clock* internal.

- Dua buah *timer/counter* 16 bit.
- Dua buah saluran interupsi eksternal.
- Jalur I/O dua arah (*bidirectional*) 32 buah.
- Memori program terpisah dari memori data.
- Memori data internal 128 byte.
- Alamat memori program eksternal 64 Kilobyte.
- Alamat memori data eksternal 64 Kilobyte.

Gambar 2-1. Memperlihatkan konfigurasi kaki-kaki pin dari mikrokontroller AT89C51.



Gamabar 2-1. Konfigurasi pin AT89C51

Sumber : Data Sheet Atmel 89C51

Pada gambar diatas yaitu konfigurasi pin AT89C51 terdapat beberapa pin penting, yaitu:

1. *Vcc*

Mikrokontroller AT89C51 dioperasikan dengan tegangan supply +5V, yaitu pada pin 40.

2. *GND*

Hubungan dengan ground adalah pin 20.

3. *Port 0*

Port 0 (pin 32-39) merupakan saluran input/output yang bersifat dua arah. Dalam rancangan sederhana, port ini digunakan sebagai port I/O dengan fungsi untuk berbagai keperluan. Sedangkan untuk rancangan yang lebih besar dimana ditambahkan memori dari luar, port ini berfungsi untuk memultipleks alamat memori dengan data.

4. *Port 1*

Port 1 (pin 1-8) merupakan port I/O 8-bit yang bersifat dua arah. Port ini dapat digunakan untuk hubungan dengan peralatan luar bila diinginkan.

5. *Port 2*

Port 2 (pin 21-28) merupakan port yang dapat digunakan untuk beberapa fungsi yaitu keperluan I/O, juga sebagai byte alamat selama penerimaan suatu data dari program memori eksternal dan saat akses ke data memori eksternal yang menggunakan alamat 16-bit (yaitu instruksi `MOVX @ DPTR`). Selama akses ke memori data eksternal yang menggunakan alamat (`MOVX @ RI`), maka port 2 akan mengeluarkan isi dari *P2 Special Function Register (SPF)*.

6. Port 3

Port 3 (pin 10-17) merupakan port paralel 8 bit yang bersifat dua arah dan mempunyai fungsi ganda, dimana setiap pin mempunyai keistimewaan yang dimiliki AT89C51 seperti pada tabel berikut:

Tabel 2-1. Fungsi Pengganti Dari Port 3

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

Sumber : Data Sheet Atmel 89C51

7. Reset (RST)

Reset input berada pada pin nomor 9. Piranti akan mengalami reset saat terdapat logika high '1' pada pin 9 dan osilator bekerja.

8. $\text{ALE}/\overline{\text{PROG}}$

Pin $\text{ALE}/\overline{\text{PROG}}$ terdapat pada pin 30. ALE berfungsi untuk menahan alamat bit rendah selama berlangsungnya akses ke memori eksternal. Pin

ini juga merupakan input dari sinyal program ($\overline{\text{PROG}}$) selama program Flash bekerja.

9. $\overline{\text{PSEN}}$

Program Store Enable adalah sinyal kontrol dari memori program eksternal. Ketika AT89C51 mengeksekusi kode dari memori program eksternal, $\overline{\text{PSEN}}$ diaktifkan dua kali setiap mesin aktif, kecuali bila diloncati selama akses ke memori data eksternal.

10. $\overline{\text{EA}} / V_{\text{PP}}$

EA (*External Access Enable*) harus dihubungkan dengan GND untuk mengaktifkan program eksternal dan harus dihubungkan dengan V_{CC} saat terjadi eksekusi program secara internal. Pin ini juga menerima V_{PP} (programming enable voltage) selama pemrograman Flash, untuk komponen yang memerlukan 12 V_{PP} .

11. XTAL 1

Merupakan input dari penguat osilator dan clock internal untuk pengoperasian rangkaian.

12. XTAL 2

Merupakan output dari penguat pembalik osilator.

Keluarga MCS[®]51 yang diproduksi Intel mempunyai konfigurasi yang berbeda-beda sesuai dengan jenisnya. Masing-masing jenis saling kompatibel serta mempunyai kelebihan tersendiri. Misalnya mikrokontroler AT89C51

merupakan padanan dari mikrokontroller 8051, tetapi tidak memiliki ROM internal. Tabel 2-2 memperlihatkan sebagian dari keluarga MCS[®]51.

Tabel 2-2. Keluarga MCS[®]51

Tipe	Tipe tanpa EPROM	Tipe ber-EPROM	Kapasitas ROM	Kapasitas RAM	Port I/O	Pewaktu
8051	8031	-	4K	128	4	2
8051AH	8031AH	8751H	4K	128	4	2
8052AH	8032AH	8752BH	8K	256	4	3
80C51BH	80C31BH	87C51	4K	128	4	2
80C52	80C32	-	8K	256	4	3
83C51FA	80C51BH	8751FA	8K	256	4	3
83C51FB	80C51FB	87C51FB	16K	256	4	3
83C152	80C152	-	8K	256	5	3

Sumber : Bereksperimen Dengan Mikrokontroller 8031, 5.

a). Organisasi Memori

Organisasi memori pada mikrokontroller AT89C51 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program.

Memori program digunakan untuk menyimpan instruksi-instruksi yang akan dijalankan oleh mikrokontroller, sedangkan memori data digunakan sebagai tempat penyimpanan data yang sedang diolah mikrokontroller.

Program mikrokontroller disimpan dalam memori program berupa ROM. Mikrokontroller AT89C51 dilengkapi dengan ROM internal namun untuk program yang besar digunakan ROM eksternal yang terpisah dari mikrokontroller. Untuk dapat menggunakan memori program eksternal ini penyemat /EA dihubungkan dengan penyemat V_{SS} (logika 0).

Memori program mikrokontroller menggunakan alamat 16 *bit* mulai 0000_H - $FFFF_H$, sehingga kapasitas penyimpanan program maksimal adalah 2^{16} *byte* atau 64 Kb. Sinyal yang digunakan untuk membaca memori program eksternal adalah sinyal /PSEN (*program store enable*).

Selain memori program mikrokontroller AT89C51 juga memiliki memori data internal berkapasitas 128 *byte* dan mampu mengakses memori data eksternal sebesar 64 Kb. Semua memori data internal dapat dialamati dengan pengalamatan langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* berisi alamat data yang diolah. Sedangkan ciri dari pengalamatan tidak langsung adalah *operand* adalah alamat *register* yang berisi alamat data yang akan diolah. Sebagian memori tersebut dapat dialamati dengan pengalamatan register, dan sebagian lagi dapat dialamati dengan memori satu *bit*. Untuk membaca data digunakan sinyal /RD, sedangkan untuk menulis data digunakan sinyal /WR.

b). Register Fungsi Khusus

Register fungsi khusus (*Special Function Register*) terletak pada 128 *byte* bagian atas memori data internal dan berisi *register-register* untuk pelayanan *latch port*, *timer*, *program status words*, *control peripheral* dan sebagainya. Alamat register fungsi khusus ditunjukkan pada Tabel 2-3.

Register-register ini hanya dapat diakses dengan pengalamatan langsung. Enam belas alamat pada register fungsi khusus dapat dialamati *perbit* maupun *per-byte* dan terletak pada alamat 80_H-FF_H. Secara perangkat keras, register fungsi khusus ini dibedakan dengan memori data internal.

Tabel 2-3. Nama dan Alamat Register pada Register Fungsi Khusus

Simbol	Nama Register	Nilai pada saat reset	Alamat
ACC	<i>Accumulator</i>	0000 _H	0E0 _H
B	<i>Register B</i>	00 _H	0F0 _H
PSW	<i>Program Status Word</i>	00 _H	0D0 _H
SP	<i>Stack Pointer</i>	07 _H	81 _H
DPTR	<i>Data Pointer 2 bytes</i>		
DPL	<i>Low bytes</i>	0000 _H	82 _H
DPH	<i>High bytes</i>	0000 _H	83 _H

P0	<i>Port 0</i>	FF _H	80 _H
P1	<i>Port 1</i>	FF _H	90 _H
P2	<i>Port 2</i>	FF _H	0A0 _H
P3	<i>Port 3</i>	FF _H	0B0 _H
IP	<i>Interrupt priority control</i>	XXX00000 _B	0B8 _H
IE	<i>Interrupt enable control</i>	0XX00000 _B	0A8 _H
TMOD	<i>Timer/counter mode control</i>	00 _H	89 _H
TCON	<i>Timer/counter control</i>	00 _H	88 _H
TH0	<i>Timer/counter 0 high byte</i>	00 _H	8C _H
TL0	<i>Timer/counter 0 low byte</i>	00 _H	8A _H
TH1	<i>Timer/counter 1 high byte</i>	00 _H	8D _H
TL1	<i>Timer/counter 1 low byte</i>	00 _H	8B _H
SCON	<i>Serial control</i>	00 _H	98 _H
SBUF	<i>Serial data buffer</i>	Independen	99 _H

PCON	<i>Power control</i>	HMOS 0XXXXXXXX _B CHMOS 0XXX0000 _B	87 _H
------	----------------------	--	-----------------

Sumber : Bereksperimen Dengan Mikrokontroler 8031, 19-20.

Beberapa macam register fungsi khusus yang sering digunakan, dijelaskan sebagai berikut :

- *Accumulator (ACC)* merupakan *register* untuk penambahan dan pengurangan. Perintah *Mnemonic* untuk mengakses akumulator disederhanakan sebagai A.
- *Register B* merupakan *register* khusus yang berfungsi melayani operasi perkalian dan pembagian.
- *Program Status Word (PSW)* terdiri dari beberapa *bit* status yang menggambarkan kejadian di akumulator sebelumnya. Yaitu *carry bit*, *auxiliary carry*, dua *bit* pemilih bank, bendera *overflow*, *parity bit*, dan dua bendera yang dapat didefinisikan sendiri oleh pemakai.
- *Stack pointer (SP)* merupakan *register* 8 *bit* yang dapat diletakkan di alamat manapun pada RAM internal. Isi *register* ini ditambah sebelum data disimpan, selama instruksi PUSH dan CALL. Pada saat *reset*, *register* SP diinisialisasi pada alamat 07_H, sehingga *stack* akan dimulai pada lokasi 08_H.

- *Data pointer* (DPTR) terdiri dari dua *register*, yaitu untuk *byte* tinggi (*Data pointer high*, DPH) dan *byte* rendah (*Data pointer Low*, DPL) yang berfungsi untuk mengunci alamat 16 *bit*.
- *Port* 0 sampai *port* 3 merupakan *register* yang berfungsi untuk membaca dan mengeluarkan data pada *port* 0, 1, 2, 3. Masing-masing *register* ini dapat dialamati per-*byte* maupun per-*bit*.
- *Serial data buffer* (SBUF) merupakan dua *register* yang terpisah, *register buffer* pengirim dan sebuah *register buffer* penerima. Meletakkan data pada SBUF berarti meletakkan pada *buffer* pengirim yang akan mengirimkan data melalui transmisi serial. Membaca data SBUF berarti menerima data dari *buffer* penerima.
- *Control register* terdiri dari *register* yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua *register* khusus, yaitu *register* IP (*interrupt priority*) dan *register* IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat *register* khusus, yaitu *register* TCON (*timer/counter control*) serta untuk pelayanan *port* serial menggunakan *register* SCON (*serial port control*)

c). Port Masukan dan Keluaran

Mikrokontroller AT89C51 mempunyai 4 *port* dan masing-masing *port* terdiri dari 8 saluran *bit*. Ke empat *port* ini bersifat *bidirectional* yaitu dapat digunakan sebagai masukan atau keluaran. *Port* 0 digunakan sebagai saluran data yang dimultipleks dengan saluran alamat rendah untuk mengakses

memori eksternal, baik memori program maupun memori data. *Port 2* mengeluarkan bagian alamat tinggi untuk mode pengalamatan memori 16 *bit*. *Port 1* dan 3 berfungsi sebagai saluran masukan dan keluaran multi fungsi.

d). Sistem Interupsi

Mikrokontroller AT89C51 mempunyai dua sumber interupsi eksternal dan sumber interupsi internal yang dapat diprogram agar sensitif terhadap perubahan level atau transisi. Interupsi *timer* aktif saat *register timer* yang bersangkutan mengalami *rollover*, interupsi serial akan aktif pada saat mikrokontroller mengirim/menerima data. Setiap sumber interupsi dapat diaktifkan/dimatikan melalui perangkat lunak.

2.1.2 Memori

Memori dalam suatu sistem mikrokontroller merupakan suatu piranti yang berfungsi untuk menyimpan program dan data yang dibutuhkan oleh mikrokontroller. Memori secara garis besar dibagi menjadi dua macam yaitu memori yang hanya dapat dibaca (*Read Only Memory*) dan memori yang dapat dibaca maupun ditulisi (*Random Access Memory*).

a. Read Only Memory (ROM)

ROM adalah suatu bentuk memori yang hanya dapat dibaca isinya. Isi ROM tidak mudah dihapus atau tidak mudah hilang meskipun catu daya tidak

diberikan padanya. Karena sifatnya yang tidak mudah dihapus tersebut ROM disebut juga memori *non volatile* (tidak mudah menguap). Suatu program atau data statis yang diinginkan agar tidak mudah hilang dapat disimpan dalam ROM. Menurut sifatnya ROM dapat dibagi menjadi beberapa macam, yaitu :

- PROM (*Programmable Read Only Memory*) yaitu jenis ROM yang sekali ditulisi dan tidak dapat dihapus kembali.
- EPROM (*Erasable Programmable Read Only Memory*) yaitu jenis ROM yang dapat ditulisi maupun dihapus kembali. Menurut cara penghapusannya EPROM dapat dibagi menjadi dua yaitu UV-EPROM (*Ultra Violet EPROM*) dan EEPROM (*Electrically EPROM*).

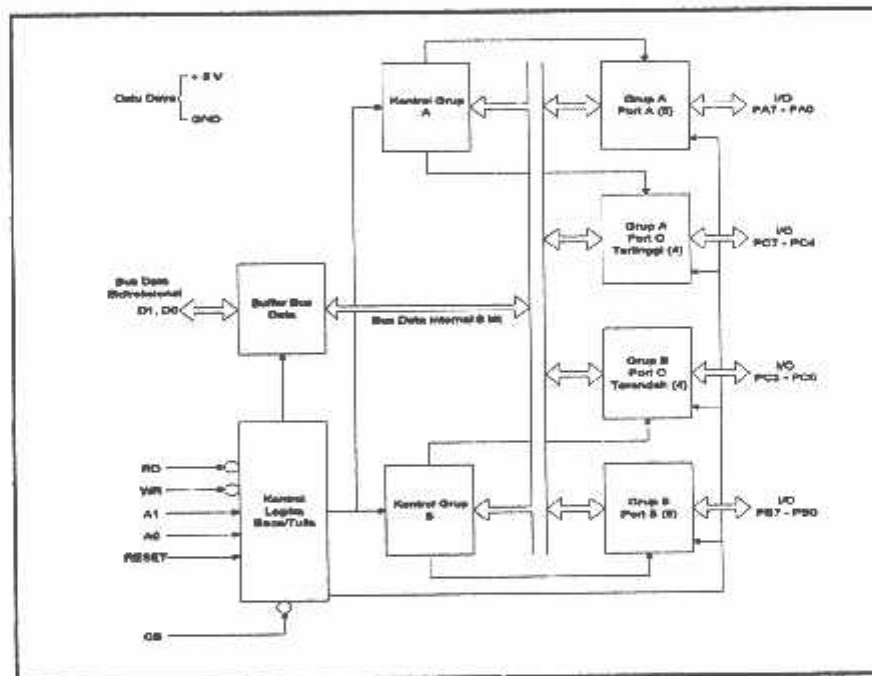
b. Random Access Memory (*RAM*)

RAM adalah memori yang dapat dibaca maupun ditulisi. Menurut sifatnya RAM biasa disebut sebagai memori yang mudah menguap (*volatile*), yaitu bila catu daya yang diberikan pada RAM dihilangkan, maka data pada RAM akan hilang. Ada dua macam RAM yaitu :

- RAM statik yaitu RAM yang tersusun atas flip-flop. Selama catu daya diberikan pada RAM, maka data akan tetap tersimpan.
 - RAM dinamik yaitu RAM yang menggunakan kapasitor sebagai penyimpan data. RAM ini memerlukan penyegaran data karena sifat kapasitor dapat menurun muatannya.
-

2.2. PPI (Programmable Peripheral Interface) 8255

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

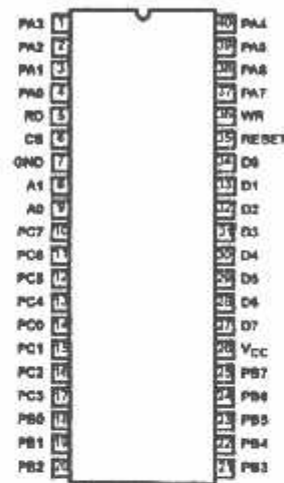


Gambar 2-2. Blok Diagram PPI 8255

Sumber : Harris Semiconductor 1996, 4-232.

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

- Port A, terdiri dari PA₀-PA₇
- Port B, terdiri dari PB₀-PB₇
- Port C, terdiri dari PC₀-PC₃ (*lower*) dan PC₄-PC₇ (*upper*)



Gambar 2-3. Konfigurasi pin PPI 8255

Pada gambar diatas yaitu konfigurasi pin PPI 8255 terdapat beberapa pin penting, yaitu:

1. *VCC*

PPI 8255 dioperasikan dengan tegangan supply +5V, yaitu pada pin 26.

2. *GND*

Hubungan dengan ground adalah pin 7.

3. *D0 – D7*

Jalur bus data dua arah (I/O) dengan tiga kondisi keadaan yang dihubungkan dengan bus data.

4. *Reset*

Reset input berada pada pin nomor 35.

5. $\overline{CS}$

Inputan Chip Select aktif low digunakan untuk menjalankan PPI 8255 pada data bus. Berada pada pin 6.

6. $\overline{RD}$

$\overline{RD}$ terdapat pada pin 5. Inputan aktif low digunakan untuk membaca status informasi dari data bus.

7. $\overline{WR}$

$\overline{WR}$ terdapat pada pin 36. Inputan aktif low digunakan untuk menulis control words pada PPI 8255.

8. *A0 – A1*

Sebagai bus alamat inputan $\overline{RD}$ dan $\overline{WR}$, yang berfungsi memilih register pada port A, B dan C.

9. *PA0 – PA7*

Port A0-A7 (pin 1-4, 37-40) merupakan saluran I/O 8-bit, bus yang dipakai pada port ini sebagai aktif high atau aktif low.

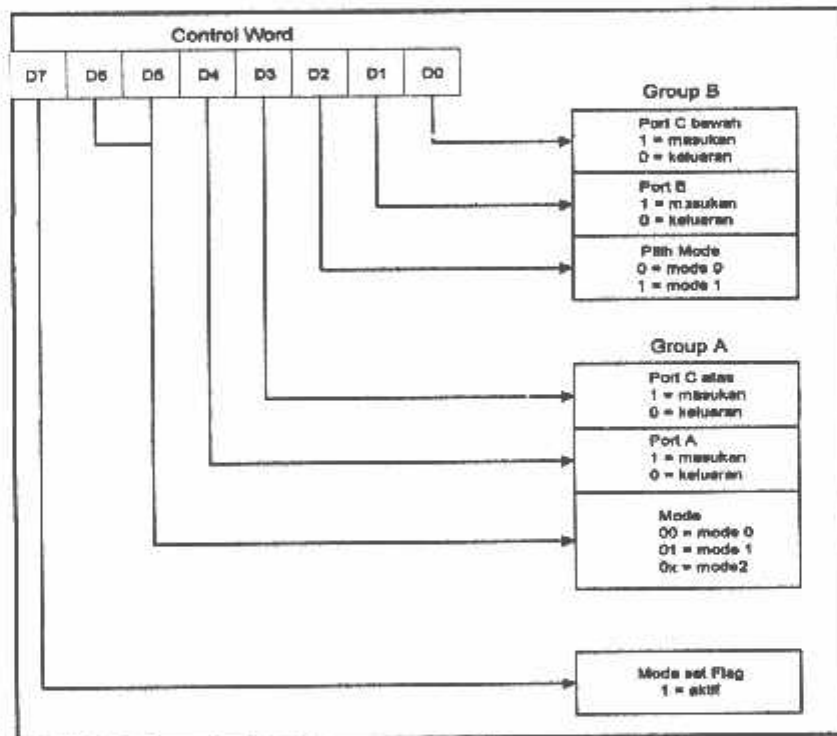
10. *PB0 – PB7*

Port B0-B7 (pin 18-25) merupakan saluran I/O 8-bit, bus yang dipakai pada port ini sebagai aktif high.

11. *PC0 – PC7*

Port C0-C7 (pin 10-17) merupakan saluran I/O 8-bit, bus yang dipakai pada port ini sebagai aktif high atau aktif low.

Konfigurasi fungsi setiap *port* pada PPI ditentukan oleh *control word* yang diberikan ke *bus* data PPI pada awal program sebelum PPI digunakan. Proses ini biasa disebut inisialisasi PPI dan merupakan syarat mutlak yang harus diberikan kepada PPI. *Control word* ini mengandung berbagai informasi yang berhubungan dengan pemanfaatan PPI, seperti mode operasi, fungsi masing-masing *port* dan aktivitas PPI. *Control word* ini akan disimpan pada *control register*. Format masing-masing *bit* dari *control word* ditunjukkan dalam Gambar 2-3.



Gambar 2-4. Control Word PPI 8255

Sumber : Harris Semiconductor 1996, 4-233.

Sebelum memakai peralatan PPI mikrokontroller harus menginisialisasinya dengan melaksanakan dua operasi dasar yaitu :

- Penentuan modus.
- Penunjukan arah saluran.

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

Tabel 2-4. Operasi Dasar PPI 8255

A0	A1	Operasi I/O
0	0	<i>Bus Data—Port A</i>
0	1	<i>Bus Data—Port B</i>
1	0	<i>Bus Data—Port B</i>
1	1	<i>Eus Data—Kontrol</i>

Sumber : Harris Semiconductor 1996, 4-232.

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

a. Mode Operasi 0 (*Basic Input Output*)

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

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

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

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

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

- Jalur kontrol 4-bit digunakan untuk mengendalikan dan mengetahui status dari jalur data 8-bit.

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

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

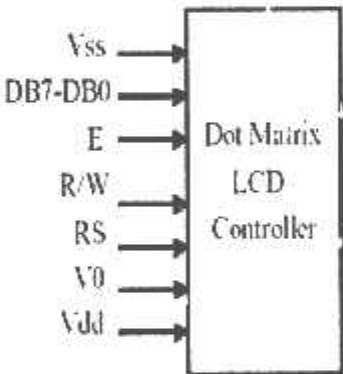
- Hanya digunakan pada grup A
- Terdapat sebuah 8-bit, *bidirectional bus* (*port A*) dan 5-bit jalur kontrol (*port C*).
- Masukan dan keluaran terkunci (*input and output latched*).
- 5-bit jalur kontrol pada port C digunakan untuk mengendalikan dan mengetahui status dari 8-bit *bidirectional bus*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

2.3. LCD (Liquid Cristal Display)

Liquid cristal display adalah modul tampilan yang mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroler CMOS didalamnya. Kontroler tersebut sebagai pembangkit ROM/RAM dan display data. Semua fungsi tampilan di kontrol oleh suatu instruksi modul LCD dapat dengan mudah diinterfacekan dengan MPU. Ciri-ciri dari LCD TM 1632 :

- Terdiri dari 32 karakter yang dibagi menjadi 2 baris dengan display dot matrik 5 X 7 ditambah cursor
 - Karakter generator ROM dengan 192 karakter
-

- Karakter generator RAM dengan 8 tipe karakter
- 80 X 8 bit display data RAM
- Dapat diinterfacekan dengan MPU 8 atau 4 bit
- Dilengkapi fungsi tambahan : Display clear,cursor home,display ON/OFF, cursor ON/ OFF, display character blink, cursor shift dan display shift
- Internal data
- Internal otomatis dan reset pada power ON
- +5 V power supply tunggal



Gambar 2-5. Diagram Blok Sistem LCD TM 1632

Sumber : Seiko Instrument, Inc

Tabel 2-5. Pin-pin LCD dan Konfigurasinya

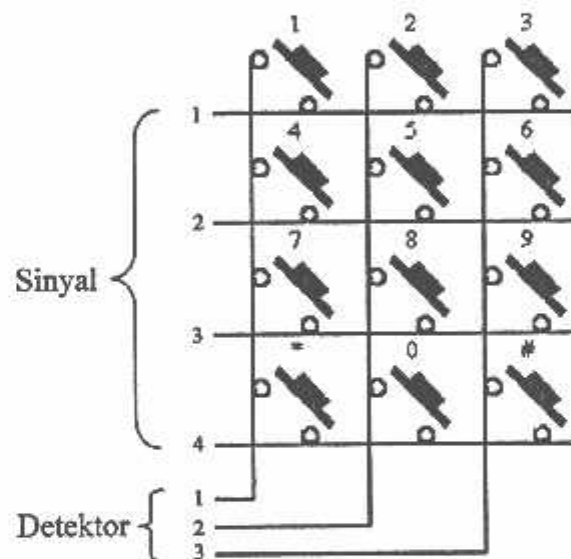
Nama pin	Jumlah	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Trisate bidirectional lower data bus: data dibaca dari modul ke

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

Sumber : *Selko Instrument, Inc*

2.4. Keypad

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port* pada PPI 8255. Papan tombol tersebut mempunyai matrik 4 baris dan 3 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port* pada PPI 8255 yang difungsikan sebagai masukan dan keluaran.



Gambar 2-6. Penampang Dasar Keypad

Prinsip kerja keypad ini menggunakan metode matrik. Sinyal atau tegangan (sesuai dengan gambar di atas) diberikan secara bergantian pada masing-masing pin. Bila pin 1 mendapat sinyal 1 (tegangan +5 V), maka pin lainnya mendapat sinyal 0 (tegangan 0 V). Metode pengiriman sinyal seperti ini dapat meng-

gunakan bilangan biner. Bila data yang dikirim 1000, berarti pin 1 mendapat sinyal 1, dan pin lainnya mendapat sinyal 0. Bila data yang dikirim 0010, berarti sinyal 1 diberikan pada pin 3.

Pada saat data yang dikirimkan 0100 (pin 2 aktif), dan tombol yang ada padanya (4, 5, dan 6) ditekan, maka sinyal tersebut akan dikirimkan ke bagian detektor sesuai dengan tombol yang ditekan. Jika tombol 4 ditekan, maka sinyal diteruskan pada detektor pin 1, jika tombol 4 dan 6 ditekan, maka sinyal akan diteruskan ke detektor pin 1 dan pin 3. Data pada detektor ini juga dapat dibaca secara biner, bila data yang diterima 010, berarti pin 2 pada detektor mendapat sinyal (tombol 5 ditekan). Bila data yang diterima 011, berarti tombol 5 dan tombol 6 ditekan.

2.5. LED Infra Red

LED infra red digunakan untuk menghasilkan radiasi infra red. Pada waktu LED infra red dibias forward, elektron dari pita konduksi melewati junction dan jatuh ke dalam hole pita valensi sehingga elektron-elektron tersebut memancarkan energi. Pada dioda penyearah biasa, energi ini dikeluarkan sebagai energi panas, tetapi pada LED infra red energi ini dipancarkan sebagai energi cahaya.

LED infra red merupakan PN junction yang memancarkan radiasi infra red yang tidak kelihatan oleh mata. Apabila pada anoda diberi tegangan positif (+) dari katoda maka LED akan menjadi ON dan arus akan mengalir dari anoda ke

katoda. Pada reaksi semi konduktor suatu dioda akan terjadi perpindahan elektron dari tipe N menjadi tipe P dan berpindahnya hole dari tipe P ke tipe N. proses kombinasi antara elektron dan hole melepaskan energi berupa pancaran cahaya.

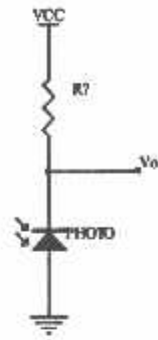
Efisiensi pancaran cahaya akan berkurang dengan berkurangnya arus input dan kenaikan suhu. Pada LED infra red cahaya yang dipancarkan mempunyai panjang gelombang yang sangat kecil ($10^{-3} - 10^{-6}$ m) sehingga pancaran tidak tampak.



Gambar 2-7. LED Infra Red

2.6. Photodiode

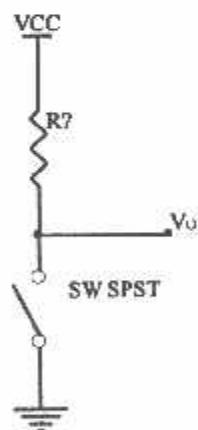
Bentuk dari photodiode mirip dengan LED infra red. Komponen ini peka terhadap cahaya. Bila photodiode diberi cahaya maka perlawanannya akan turun. Makin kuat cahayanya makin kecil harga perlawanannya. Namun bila tidak ada cahaya yang mengenainya dan dioda tersebut diberi tegangan muka terbalik maka perlawanannya akan sangat besar



Gambar 2-8. Photodiode

2.7. Limit Switch

Saklar pembatas (limit switch) digunakan untuk membatasi atau mengakhiri perintah yang telah diumpankan lewat program operasi. Saklar ini dipasang pada tempat yang dapat tersentuh oleh bodi motor DC (motor pisau perajang).

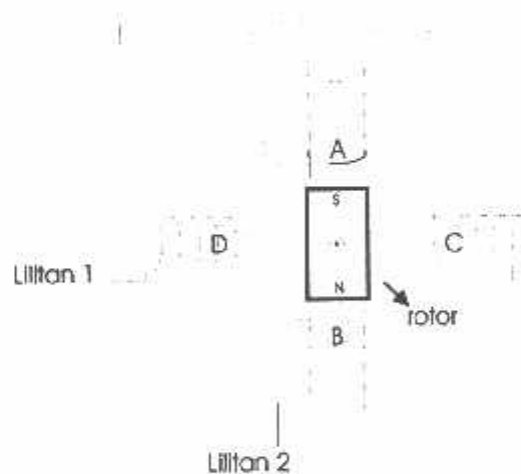


Gambar 2-9. Limit Switch

2.8. Motor Stepper

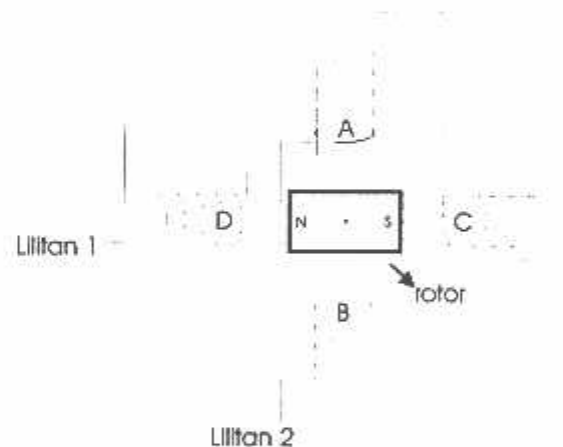
Motor stepper merupakan sebuah alat elektromagnetis yang dirancang untuk mengubah masukan berbentuk pulsa elektrik menjadi putaran tertentu berdasarkan derajat tertentu secara diskrit.

Ukuran derajat pergerakan dari motor stepper bervariasi antara lain : 1.8° , 2° , 2.5° dan 5° untuk tiap-tiap pulsa elektrik. . Dapat dilihat pada gambar 2-9 bahwa dalam hal ini merupakan magnet batang yang berputar pada pusatnya. Ditunjukkan pula disini, bahwa motor stepper terdiri atas dua lilitan yang setiap lilitannya membentuk medan elektromagnet sehingga menimbulkan polaritas yang berbeda diantara pasangan kutub tersebut.



Gambar 2-10. Bagan Dasar Motor Stepper

Jika pada kutub A dan B diberi tegangan, akan mengakibatkan kutub A akan berlaku sebagai kutub selatan dan kutub B sebagai kutub utara yang disebabkan karena lilitan kumparan sehingga timbul medan elektomagnet yang membuat rotor berputar. Hal ini merupakan prinsip dasar kerja motor stepper. Pada gambar 2-9 diatas dapat dilihat bahwa rotor akan tetap pada posisinya, selama tegangan diberikan pada lilitan kedua. Apabila tegangan pada lilitan kedua dihilangkan dan kemudian diberikan pada lilitan pertama, maka yang terjadi adalah kutub A dan B tidak mempunyai gaya elektromagnet, dan sebaliknya antara kutub C dan D akan timbul gaya elektromagnet. Ditunjukkan seperti pada gambar 2-10.



Gamabar2-11. Proses Pemutaran Motor Stepper 90°

2.9. Motor DC

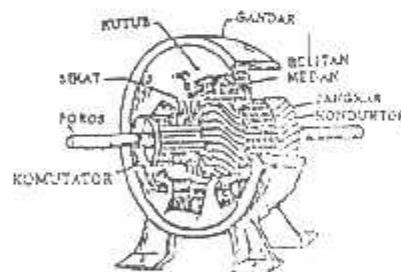
Motor DC merupakan suatu alat yang berfungsi mengubah arus listrik yang diterima menjadi energi mekanik berupa kecepatan putar poros yang dapat digunakan untuk memutar peralatan-peralatan produksi dipabrik atau industri.

- **Konstruksi motor DC**

Konstruksi motor DC terbagi atas :

1. Stator adalah bagian dari motor DC yang diam.
2. Rotor atau angker atau jangkar atau armature adalah bagian motor yang berputar.
3. Celah udara adalah ruangan yang ada diantara stator dan rotor.

Gambar konstruksi dari motor DC dapat dilihat pada Gambar 2-5.



a. Stator dan Rotor Motor DC



b. Penampang Motor DC

Gambar 2-12. Konstruksi Motor DC

Sumber : Wasito, S, *Elektronika Dalam Industri*, 1994

- Prinsip kerja

Jika pada penghantar yang membawa arus ditempatkan suatu medan magnet maka penghantar tersebut akan mengalami gaya. Gaya menimbulkan torsi yang akan menghasilkan rotasi mekanik, sehingga motor akan berputar. Jadi motor DC ini menerima sumber arus searah dari jala-jala kemudian diubah menjadi energi mekanik berupa putaran yang nantinya dipakai oleh peralatan lain. Ringkasnya prinsip kerja motor DC membutuhkan :

1. Garis gaya medan magnet (fluks), antara kutub yang berada di stator.
2. Penghantar yang dialiri arus ditempatkan pada jangkar yang berada pada medan magnet.
3. Pada penghantar timbul gaya yang menghasilkan torsi.

Fungsi dari masing-masing blok rangkaian sistem adalah sebagai berikut :

- Keypad berfungsi sebagai data masukan untuk memasukkan ketebalan tempe.
- LCD untuk menampilkan menu dan menampilkan ketebalan tempe yang sudah diatur melalui keypad.
- Mikrokontroller digunakan untuk memproses inputan serta sebagai unit pengendali dari sistem yang dibuat.
- Driver motor berfungsi pengendali pergerakan motor yang akan menggerakkan pisau perajang dan motor stepper pada belt konveyor.
- Belt konveyor berfungsi sebagai tempat untuk menggerakkan tempe dimana belt konveyor ini digerakkan oleh motor stepper.
- Motor pisau digunakan sebagai pemotong dari tempe.

Sistem kerja dari alat ini yaitu apabila mikrokontroller mendapat masukan data (ketebalan tempe) dari keypad maka mikrokontroller akan memproses inputan tersebut, kemudian ketebalan yang sudah diatur tadi ditampilkan melalui LCD dan mikrokontroller ini juga mengendalikan pergerakan motor stepper sebagai penggerak belt konveyor dan motor perajang. Untuk pengaturan ketebalan tempe diatur dari mikrokontroller ke pergerakan motor stepper.

3.3. Perencanaan Perangkat Keras

Perencanaan perangkat keras merupakan penjabaran dari masing-masing blok diagram sistem. Perencanaan perangkat keras terdiri dari perencanaan

rangkaian kontrol AT89C51, PPI 8255, latch, decoder, driver motor, unit tampilan dan keypad.

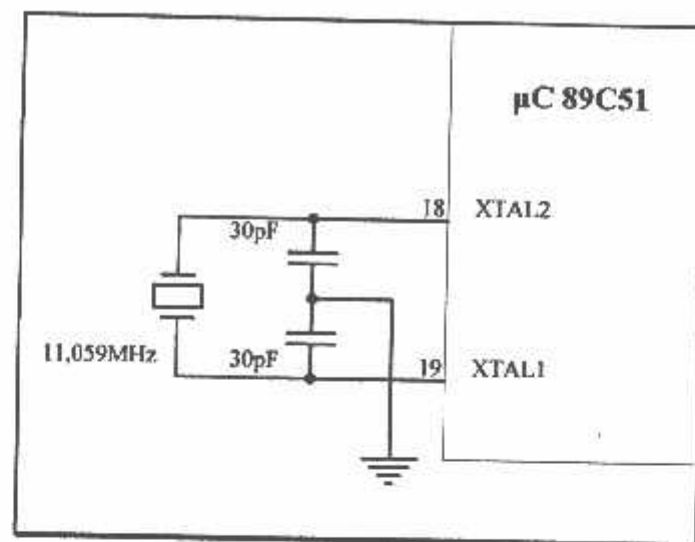
3.3.1. Mikrokontroller AT89C51

Mikrokontroller AT89C51 harus didukung oleh beberapa rangkaian lain agar dapat melakukan prosesnya, yaitu berupa rangkaian clock dan reset. Selain itu juga harus ditentukan penggunaan port-portnya dan sinyal-sinyal yang digunakan untuk mendukung proses yang akan dilakukan.

Hal hal itu adalah sebagai berikut:

- *Clock*

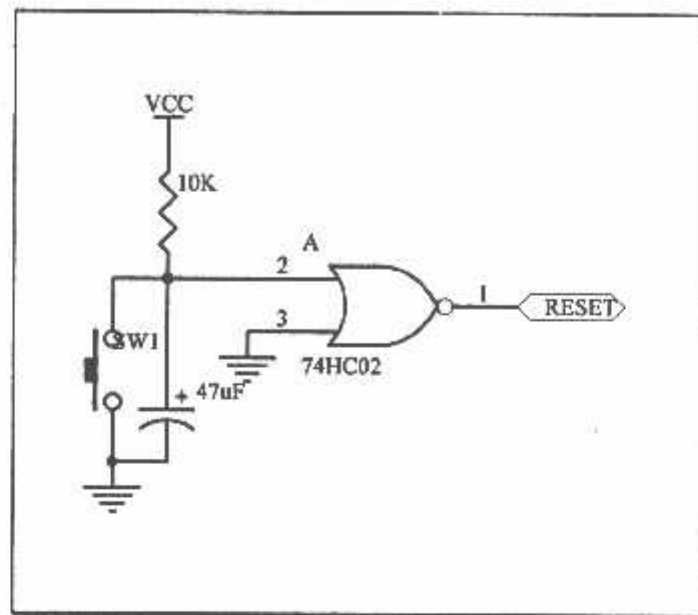
Kecepatan proses yang dilakukan oleh mikrokontroller ditentukan oleh sumber *clock* (pewaktuan) yang mengendalikan mikrokontroller tersebut. Sistem yang dirancang ini akan menggunakan osilator internal yang sudah tersedia di dalam chip AT89C51. Untuk menentukan frekuensi osilatornya cukup dengan cara menghubungkan kristal pada pin XTAL1 dan XTAL2 serta dua buah kapasitor ke ground. Besar kapasitansinya disesuaikan dengan spesifikasi pada lembar data AT89C51 yaitu 30 pF. Pemilihan besar frekuensi kristal disesuaikan dengan pemilihan kecepatan yang diharapkan untuk transfer data melalui pin *serial interface* AT89C51 tersebut. Sistem ini dirancang untuk memiliki kemampuan *baud rate* sebesar 9600 bps, sehingga dipilih kristal dengan nilai 11,059 MHz sesuai dengan spesifikasi pada lembar data AT89C51. Gambar 3-2. memperlihatkan rangkaian pewaktu yang digunakan.



Gambar 3-2. Rangkaian Pewaktuan

- **Reset**

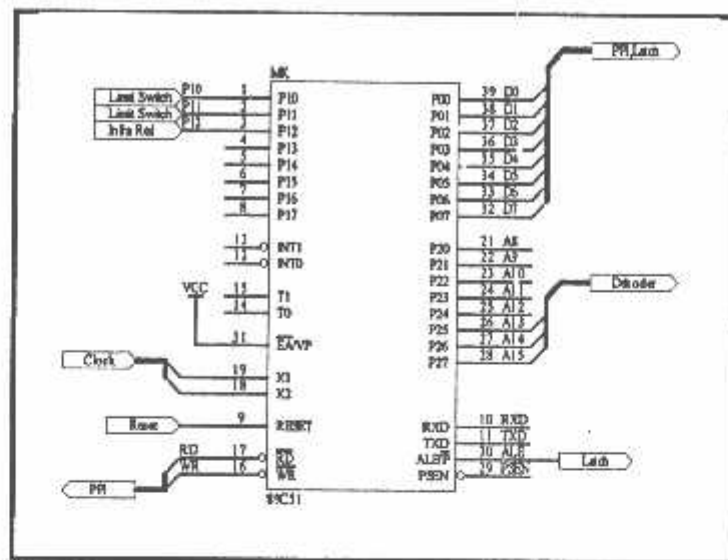
Untuk mereset mikrokontroller, pin RST harus diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset pada saat awal catu daya dihidupkan, suatu reset otomatis dapat dilakukan dengan menghubungkan pin RST ke rangkaian *Power-On Reset*, seperti dalam Gambar 3-3. Ketika catu daya dinyalakan, rangkaian akan menahan pin RST dalam kondisi logika tinggi selama selang beberapa saat tergantung nilai kapasitor dan kecepatan pengisian muatannya. Pemberian catu daya pada mikrokontroller tanpa suatu sinyal reset dapat menyebabkan CPU memulai eksekusi instruksinya dari lokasi yang tak tertentu. Ini disebabkan karena *Program Counter* tidak terinisialisasi.



Gambar 3-3. Rangkaian *Power-On Reset*

- **Pembagian Port**

Mikrokontroller AT89C51 mempunyai 4 buah port yaitu port 0 (P0), port 1 (P1), port 2 (P2) dan port 3 (P3). Mikrokontroller AT89C51 tidak memiliki EPROM internal, maka port 0 digunakan untuk saluran data dan alamat bagian rendah yang dimultiplek (AD0-AD7) dan alamat bagian tinggi dikeluarkan pada port 2 (A8-A15). Karena menggunakan komponen PPI pada sistem mikrokontrollernya maka beberapa bit port 3 digunakan sebagai pembangkit sinyal kontrol (Tx, Rx, RD dan WR). Dalam perancangan port 1 digunakan sebagai masukan dari *limit switch* dan *LED infra red*.



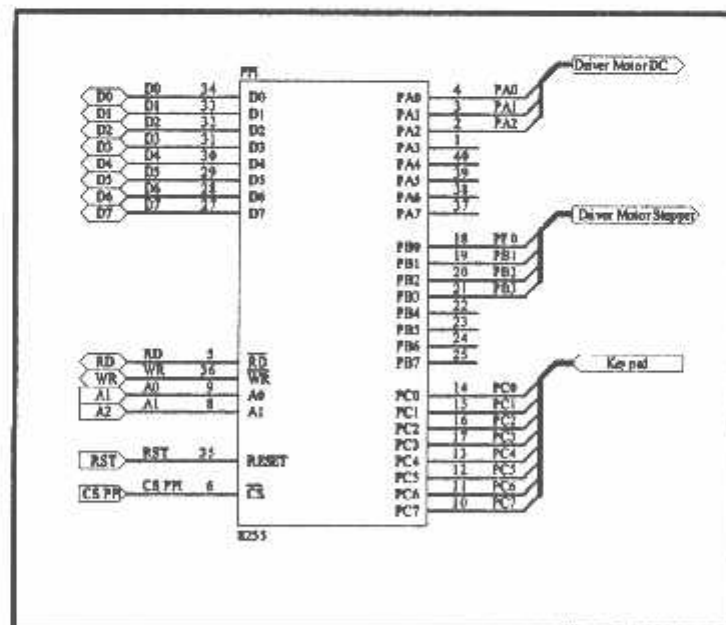
Gambar 3-4. Rangkaian Mikrokontroller

- Port 1.0 – Port 1.1 digunakan sebagai inputan dari limit switch.
- Port 1.2 digunakan sebagai inputan dari infra red.
- Pin 9 (reset), reset aktif tinggi yang terhubung dengan rangkain power on reset dan jika diaktifkan akan mereset mikrokontroler.
- Pin 19 (X1) digunakan sebagai input dari rangkaian pewaktu luar.
- Pin 18 (X2) digunakan sebagai output dari rangkaian pewaktu luar,
- Pin 17 dan Pin 16 (/RD, /RW) yaitu external data memori write, read yang dihubungkan ke PPI.
- Pin 30 (ALE/*PROG*) untuk menahan alamat bit rendah yang dihubungkan ke latch.
- Port D0 – D7 output dari mikrokontroller yang dihubungkan ke PPI dan Latch.

3.3.2. Rangkaian PPI 8255

Mikrokontroler AT89C51 memiliki 4 buah port, yaitu port 0, port 1, port 2 dan port 3. Tetapi karena tidak mempunyai EPROM internal, maka untuk mengakses EPROM eksternal, port 0 dan port 2 bisa difungsikan apabila menggunakan memori eksternal dan menjadi bus alamat dan bus data. Sementara itu beberapa bit dari port 3 juga memiliki fungsi lain yaitu sinyal RD, WR, TxD, RxD, INT0, INT1, T0 dan T1. Jadi port bebas yang tersisa adalah port 1. Untuk menambah jumlah port digunakanlah PPI 8255.

Dalam perencanaan, PPI 8255 ditempatkan pada alamat E000H-E003H. Pin A0 dan A1 PPI yang dipergunakan untuk memilih register yang diakses dihubungkan ke A0 dan A1 sistem mikrokontroler. Sementara pin CS IC 8255 dihubungkan ke CS7 (keluaran dari *address decoder* 74LS138). Rangkaian PPI 8255 seperti terlihat dalam Gambar 3-5.



Gambar 3-5. Rangkaian PPI 8255

3.3.3. Latch

Latch digunakan untuk memisahkan antara bit-bit data dengan bit-bit alamat rendah yang dikeluarkan pada port 0 oleh mikrokontroller AT89C51. Tipe yang akan dipakai adalah tipe 74LS573. Latch ini merupakan latch berdelapan tipe D transparan yang berarti bahwa saat enable dan C dalam keadaan logika tinggi maka keluaran Q akan mengikuti masukan data D. Bila enable diberi logika rendah maka keluaran akan dilatch pada level data yang telah diset sebelumnya (Q0). Tabel fungsi 74LS573 dapat dilihat dalam Tabel 3-1.

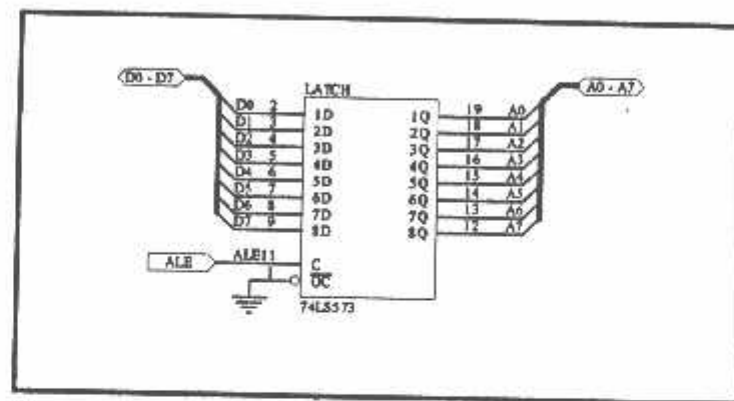
Tabel 3-1. Tabel Fungsi Latch 74LS573

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

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

Sinyal ALE berlogika tinggi maka bit-bit yang dikeluarkan melalui port 0 adalah bit-bit alamat, dan 74LS573 akan meneruskan bit-bit ini ke keluarannya

yang terhubung dengan saluran alamat komponen yang dituju. Apabila sinyal ALE dalam keadaan logika rendah maka 74LS573 menahan bit-bit tersebut sehingga bit-bit ini akan dibaca sebagai data oleh komponen yang terhubung.



Gamabar 3-6. Rangkaian Latch

3.3.4. Dekoder dan Peta Alamat

Proses aliran data yang diperlukan oleh mikrokontroller tergantung dari pengalamatan yang dipilih mikrokontroller tersebut. Untuk memilih komponen mana yang diakses maka diperlukan suatu rangkaian dekoder. Dekoder ini akan dihubungkan dengan alamat tinggi yang dikeluarkan mikrokontroller. Sistem ini mempergunakan pengalamatan 16 bit untuk mengamati PPI.

Peta alamat pada sistem ini direncanakan dibagi per-delapan kilo byte, yang berarti bahwa alamat mikrokontroller yang tersedia sebesar 64 kilo byte dibagi menjadi 8 blok. Masing-masing blok dipilih dengan menggunakan 3 bit

alamat tinggi yang dikeluarkan melalui port 2. Tabel 3-2 memperlihatkan pemetaan alamat pada sistem ini.

Tabel 3-2. Pemetaan Alamat

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

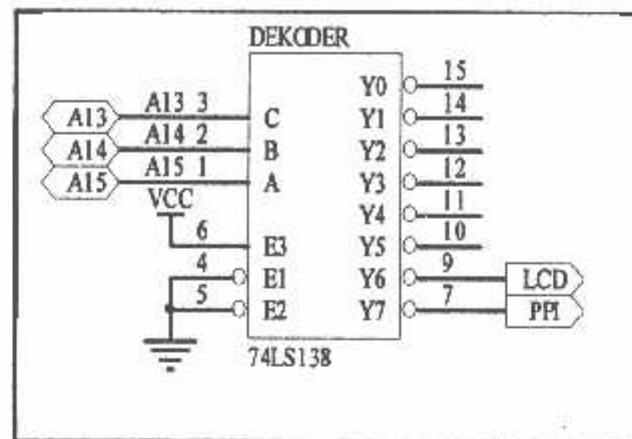
Dekoder yang dipergunakan adalah dekode 74LS138, suatu dekode 3 ke

8. Tabel fungsinya seperti terlihat dalam Tabel 3-3

Tabel 3-3. Tabel Dekoder 74LS138

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

Dekoder ini dikondisikan selalu membaca data masukannya setiap saat, sehingga pin G1 diberi logika tinggi dan pin G2A, G2B dihubungkan dengan pentanahan. Saluran alamat yang digunakan untuk memilih alamat diambil dari mikrokontroler pada bit A13, A14 dan A15, dihubungkan dengan pin A, B dan C pada dekoder. Rangkaian dekoder ditunjukkan dalam Gambar 3-7.



Gambar 3-7. Rangkaian Dekoder Alamat

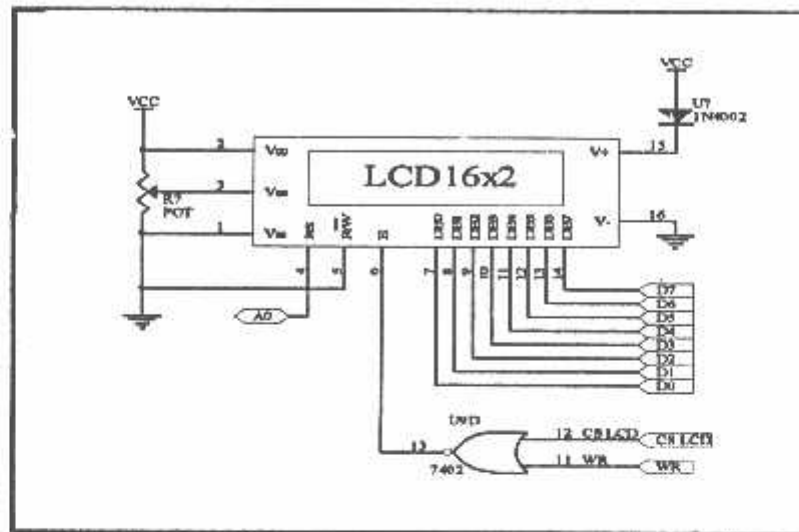
3.3.5. Rangkaian LCD (*Liquid Cristal Display*)

Pada sistem yang direncanakan digunakan LCD (*Liquid Cristal Display*) sebagai tampilan. LCD yang digunakan adalah jenis TM 1632 yang merupakan LCD dua baris dengan tiap barisnya terdiri dari 16 karakter.

LCD ini membutuhkan 3 sinyal control, $R/\overline{W}$ (*Read/Write*) untuk menentukan apakah data akan dibaca atau ditulis, E (*Enable*) yang merupakan sinyal untuk meng-eneble-kan dan RS (*Register Select*) untuk memilih register yang diakses. LCD TM 1632 memiliki dua register yaitu register data dan register instruksi.

Dalam sistem ini, LCD menempati ruang alamat A000H-A001H. Pin $R/\overline{W}$ dihubungkan ke ground atau selalu berlogika 0 dalam perancangan. LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin A0 sistem mikrokontroller. Pengaktifkan LCD ini selanjutnya tergantung pada pin E.

Dimana pin E ini tergantung dari CS5 dari *address* decoder dan perintah write mikrokontroller. Rangkaian LCD terlihat pada Gambar 3-8.



Gambar 3-8. Rangkaian LCD

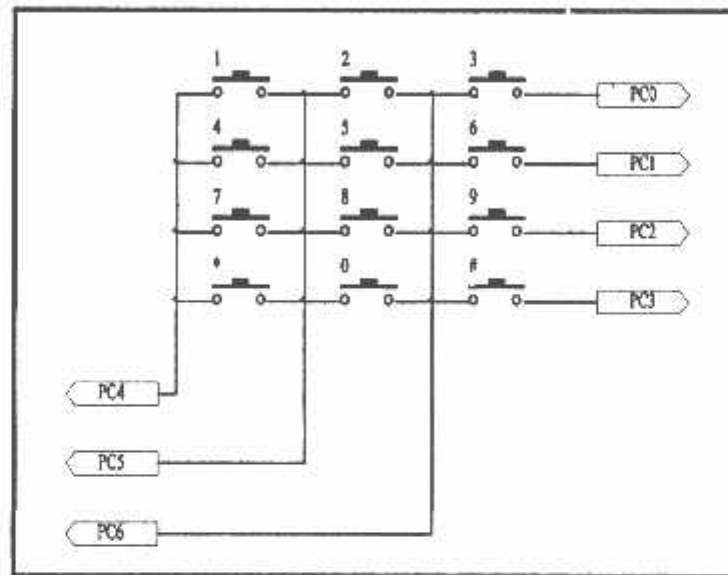
3.3.6. Rangkaian Keypad

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

Papan tombol tersebut mempunyai metrik 3 kolom dan 4 baris. Deretan kolom dan baris dari papan tombol dihubungkan dengan *port C* PPI 8255 yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan *ground* (berlogika 0) dan *port C* (PC5-PC7) yang difungsikan sebagai input mikrokontroller. Sedangkan deretan baris dihubungkan ke *port C* (PC0-PC3) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri.

Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. Port ini difungsikan sebagai output dari

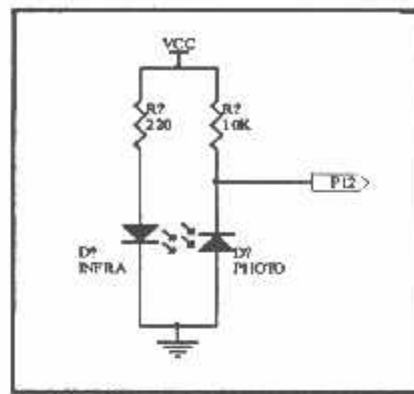
mikrokontroller. Dengan demikian kalau tombol tidak ditekan maka masukan *port* C (PC5-PC7) di pin yang terhubung tombol tersebut berlogika 0 dan bila tombol ditekan akan berlogika 1. Rangkaian papan tombol tersebut dapat terlihat pada Gambar 3-9.



Gambar 3-9. Rangkaian Keypad

3.3.7. Rangkaian LED Infra Red dan Photodiode

Sensor pendeteksi adanya benda menggunakan LED infra red dan photodiode, dimana rangkaian sensor ini untuk mengetahui ada atau tidak adanya benda/tempe yang melewati konveyor. Rangkaian tersebut dapat terlihat pada Gambar 3-10.



Gambar 3-10. Rangkaian Sensor LED Infra Red dan Photodiode

Agar LED infra red dapat memancarkan sinar infra merah diperlukan arus (I_{IR}) sebesar 20 mA dan tegangan (V_{IR}) sebesar 1,2 V. sehingga dapat dicari besarnya R.

$$R_1 = \frac{V_{CC} - V_{IR}}{I_{PD}}$$

$$R_1 = \frac{5 - 1,2}{20 \times 10^{-3}}$$

$$R_1 = 190 \Omega$$

Karena untuk $R_1 = 190 \Omega$ tidak ada dipasaran, maka diambil yang mendekati yaitu $R_1 = 220 \Omega$.

Dan untuk menentukan resistor pada receiver infra merah adalah sebagai berikut :

$$R_2 = \frac{V_{CC}}{I_{Pd}}$$

$$R_2 = \frac{5}{0,52 \times 10^{-3}}$$

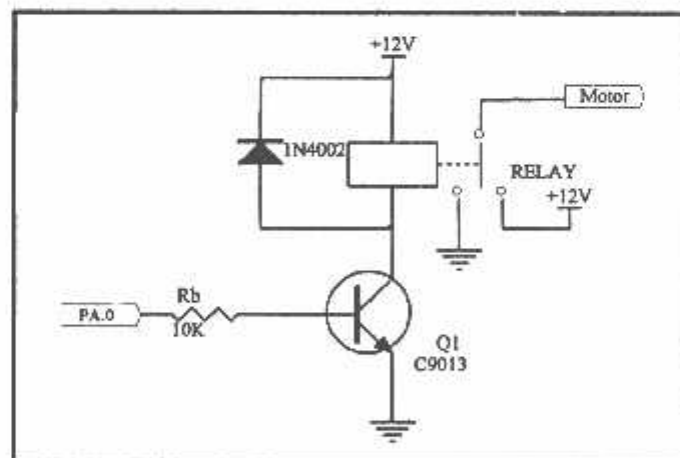
$$R_2 = 9,43 \text{ k}\Omega$$

Karena untuk $R_2 = 9,43 \text{ k}\Omega$ tidak ada dipasaran, maka diambil yang mendekati yaitu $R_2 = 10 \text{ k}\Omega$.

3.3.8. Rangkaian Driver Motor DC

Driver motor terdiri dari komponen resistor, transistor, dan relay. Prinsip dari transistor, jika terdapat arus bias dari *Basis* maka arus akan mengalir dari *Colektor* ke *Emitor* sehingga relay aktif. Bila tidak ada arus bias dari *Basis* maka tidak ada arus yang mengalir dari *Colektor* ke *Emitor* sehingga relay tidak aktif.

Relay yang digunakan adalah relay 12V. Untuk membatasi arus yang melewati pada transistor maka diperlukan sebuah hambatan R . Rangkaian tersebut dapat terlihat pada Gambar 3-11.



Gambar 3-11. Rangkaian Driver Motor DC

Pada gambar diatas dioda dipasang reverse berfungsi untuk melindungi transistor dari tegangan induksi yang dikeluarkan oleh kumparan relay pada saat transfer dari off ke on atau sebaliknya. Untuk mencari harga dari R yaitu :

Diketahui data transistor :

- C9013

$$H_{fe} = 75 \quad \text{dan} \quad V_{BE} = 0,7V$$

$$R_{Relay} = 350 \, \Omega$$

$$I_C = \frac{V_{CC}}{R_{Relay}}$$

$$= \frac{12}{350} = 0,0343 \, A$$

$$I_B = \frac{I_C}{h_{fe}}$$

$$= \frac{0,0343}{75} = 0,457 \, mA$$

$$R_b = \frac{5 - V_{BE}}{I_B}$$

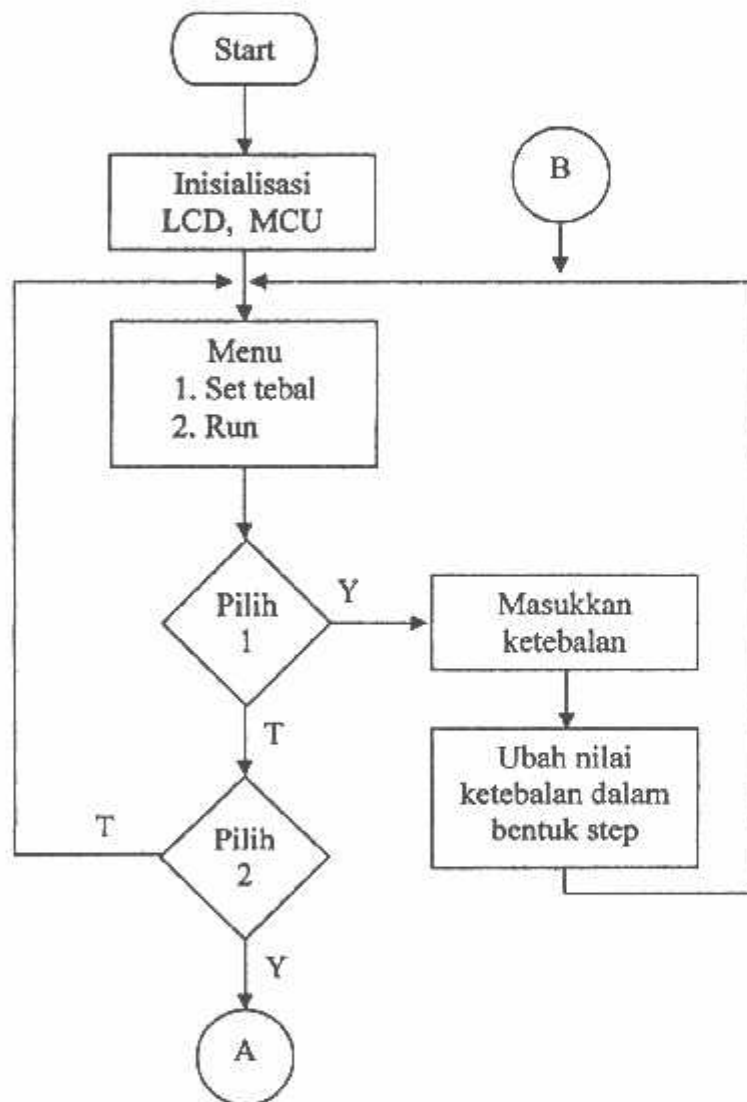
$$= \frac{5 - 0,7}{0,457 \times 10^{-3}}$$

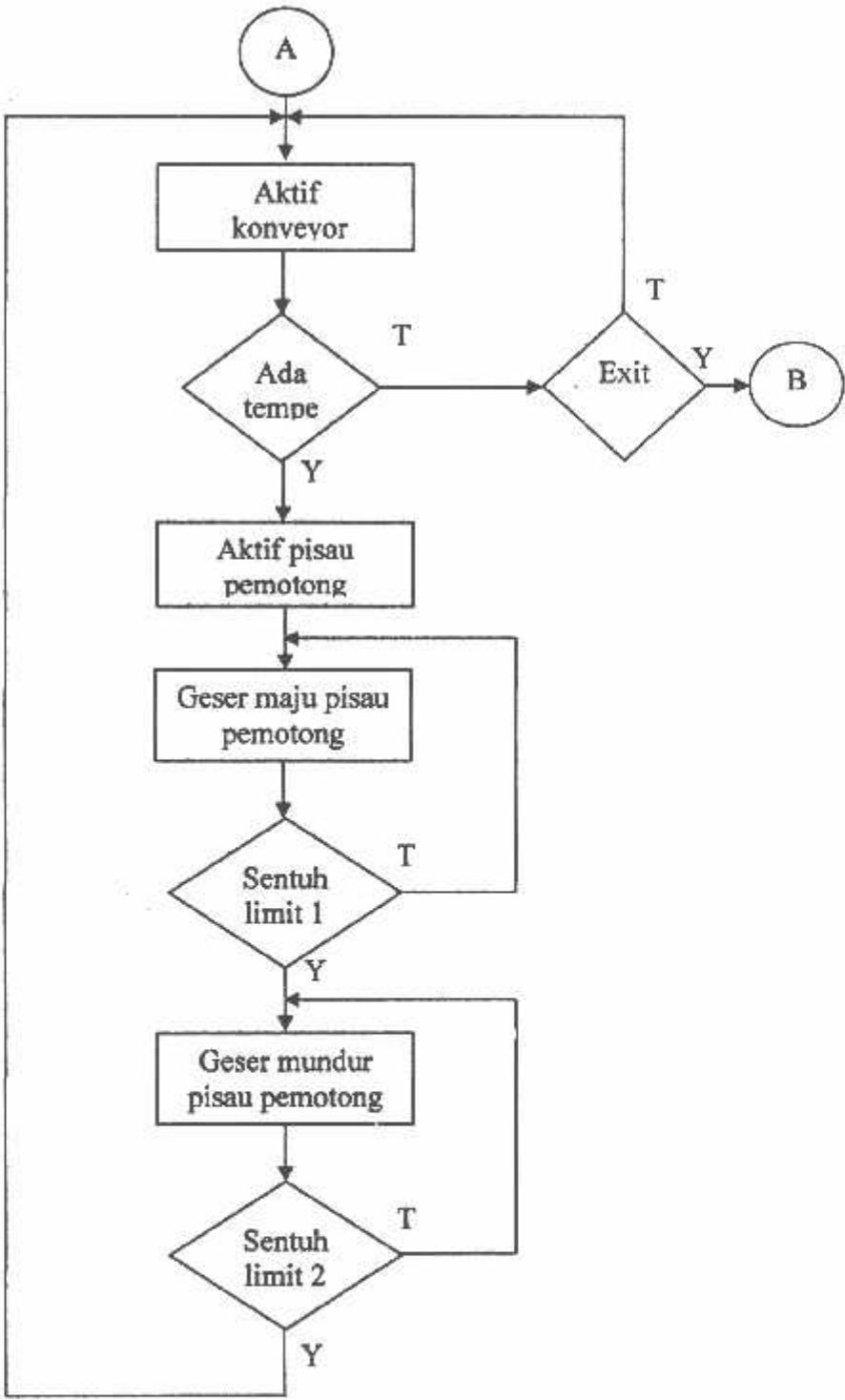
$$= 9402 \, \Omega \quad \text{atau} \quad 9K4\Omega$$

Karena nilai resistansi diatas tidak ada dipasaran, maka diambil yang mendekati yaitu 10K Ω .

3.4. Perencanaan Perangkat Lunak

Perangkat lunak untuk mengendalikan sistem ini didasarkan pada perancangan perangkat keras yang telah dibuat sebelumnya, guna mendapatkan sistem kerja yang diharapkan. Berikut diagram alir dari perancangan perangkat lunak :





BAB IV

PENGUJIAN ALAT

4.1. Umum

Bab ini membahas tentang pengujian dan analisis sistem yang telah dibuat. Secara umum, pengujian ini bertujuan untuk mengetahui apakah alat dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pertama-tama akan dilakukan pengujian unjuk kerja perangkat keras. Pengujian dilakukan pada masing-masing blok rangkaian penyusun sistem, antara lain sistem mikrokontroler, keypad, LCD, PPI dan relay pada driver motor. Setelah itu dilakukan pengujian terhadap perangkat lunak sistem. Pengujian terakhir dilakukan pada sistem secara keseluruhan untuk mengetahui unjuk kerja sistem secara keseluruhan.

4.2. Pengujian Sistem Mikrokontroler

4.2.1. Tujuan

Untuk mengetahui kondisi awal dari mikrokontroler apakah sudah sesuai dengan yang direncanakan.

4.2.2. Peralatan Yang Dibutuhkan

1. Komputer.
2. Sistem mikrokontroler.
3. Led Display.

4.2.3. Prosedur Pengujian

1. Membuat program yang digunakan dalam pengujian mikrokontroller.
2. Program yang digunakan dalam pengujian mikrokontroller ini merupakan program sederhana yang meletakkan $0F_H$ dan $F0_H$ pada ACC secara bergantian kemudian memindahkannya pada *Port 1* AT89C51. Program yang dibuat adalah sebagai berikut :

```

3.          ORG 0000H

                        JMP  START

START:  MOV  A, #0FH

                        MOV  P1, A

                        CALL  TUNDA

                        MOV  A, #F0H

                        MOV  P1, A

                        JMP  START

TUNDA:  MOV  R3, #0FFH

TUNDA1: MOV  R2, #0FFH

                        DJNZ  R2, $

                        MOV  R1, #0FH

                        DJNZ  R1, $

                        DJNZ  R3, TUNDA1

                        RET

                        END

```

4. Rangkaian dibuat seperti Gambar 4-1.
 5. Memasang catu rangkaian sebesar 5 volt
 6. Mendownload program diatas .
 7. Mengamati keluaran pada LED Display .
-



Gambar 4-1. Diagram blok Pengujian Mikrokontroller

4.2.4. Hasil Pengujian dan Analisis

Hasil pengujian ditunjukkan dalam Tabel 4-1.

Tabel 4-1. Hasil Pengujian Sistem Mikrokontroler

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

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

4.3. Pengujian Keypad

4.3.1. Tujuan

Untuk mengetahui konfigurasi logika keluaran dari keypad saat tombol angka pada keypad ditekan.

4.3.2. Peralatan Yang Digunakan

1. Catu daya 5V DC.
2. Komputer.
3. Keypad.
4. Sistem mikrokontroller AT89C51.
5. Led display.

4.3.3. Prosedur Pengujian

1. Menyiapkan catu daya dengan keluaran 5 V DC.
2. Menyusun rangkaian pengujian keypad seperti terlihat dalam Gambar 4-2 .
3. Menjalankan program pengujian keypad. Kemudian menekan sembarang tombol angka pada keypad dan mengamati keluaran pada Led display.

```

kolom1      Bit    P3.4    ; kiri (1,4,7,redial)
kolom2      Bit    P3.5    ;      (2,5,8,0)
kolom3      Bit    P3.6    ;      (3,3,9,#
kolom4      Bit    P3.7

baris1      Bit    P3.0    ; atas (1,2,3)
baris2      Bit    P3.1    ;      (4,5,6)
baris3      Bit    P3.2    ;      (7,8,9)
baris4      Bit    P3.3    ;      (*,0,#)
;
keyport     Equ    P3
keydata     Equ    50h
keybounc    Equ    51h

        Mov    SP, #40h
        Mov    P1, #0

start:

        Call   KeyIn
        Mov    P1, A
        Jump   start

```

```

; =====
;
Subroutine KeyIn:
    Push    B                                ; amankan register B
KeyInGet1:
    Call    Keypad3x4                        ; scan keypad
    Mov     A, keydata                        ; isi keydata = data di
    rutin keypad
    Cjne    A, #0FFh, KeyInGet0              ; jk isi a # 0ffh ==>lompat
    Jump    KeyInGet1
KeyInGet0:
    Mov     B, A                             ; simpan isi a to b
KeyInGet:
    Call    Keypad3x4
    Mov     A, keydata
    Cjne    A, B, KeyInOut                  ; jk isi A # B lompat
    Jump    KeyInGet
KeyInOut:
    Mov     A, B
    Pop     B                               ; idem
EndSub
; =====
; routine u/ baca keypad 3x4
; output pd keydata (0-9)
; =====
Subroutine Keypad3x4:
    Mov     keybounc, #100
    Mov     keyport, #0FFh
    Clr     kolom1
; -----
ull:
    Jb      baris1, key1
    Mov     keydata, #'1'
    Ret
key1:
    Jb      baris2, key2
    Mov     keydata, #'4'
    Ret
key2:
    Jb      baris3, key3
    Mov     keydata, #'7'
    Ret
key3:
    Jb      baris4, key4
    Mov     keydata, #'+'
    Ret
; -----
key4:
    Setb    kolom1
    Clr     kolom2
    Jb      baris1, key5
    Mov     keydata, #'2'
    Ret

```

```

key5:      Jb      baris2, key6
           Mov     keydata, #'5'
           Ret

key6:      Jb      baris3, key7
           Mov     keydata, #'8'
           Ret

key7:      Jb      baris4, key8
           Mov     keydata, #'0'
           Ret

;-----
key8:      Setb    kolom2
           Clr     kolom3
           ;-----
           Jb      baris1, key9
           Mov     keydata, #'3'
           Ret

key9:      Jb      baris2, key10
           Mov     keydata, #'6'
           Ret

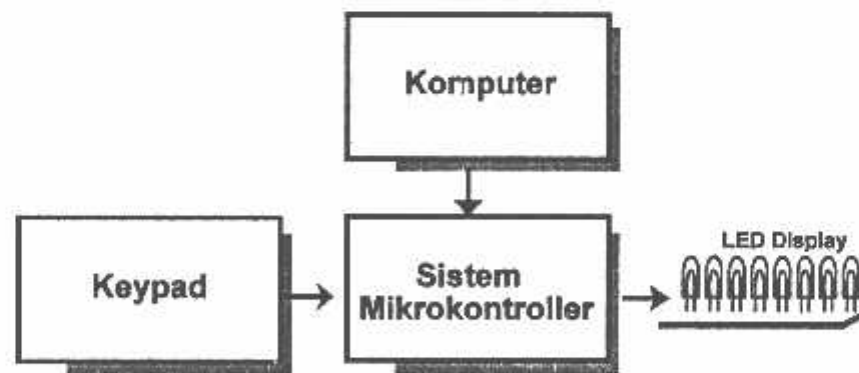
key10:     Jb      baris3, key11
           Mov     keydata, #'9'
           Ret

key11:     Jb      baris4, key12
           Mov     keydata, #'#'
           Ret

key12:     Mov     keydata, #0FFh

EndSub

```



Gambar 4-2. Pengujian Rangkaian Keypad

4.3.4. Hasil Pengujian dan Analisis

Hasil pengujian ditunjukkan dalam Tabel 4-2.

Tabel 4-2. Tabel Pengujian Keypad

Tombol Yang Ditekan	Karakter Yang Tampil Pada LED
1	0001B
2	0010B
3	0011B
4	0100B
5	0101B
6	0110B
7	0111B
8	1000B
9	1001B
0	0000B

Dari hasil pengujian, didapatkan data seperti dalam Tabel 4-2, maka dapat diketahui bahwa saat tombol ditekan maka keluaran port 1 mikrokontroller AT89C51 akan ditampilkan di LED sesuai dengan tombol yang ditekan. Hasil pengujian dalam Tabel 4-2 terlihat bahwa rangkaian keypad yang telah direalisasikan sesuai dengan unjuk kerja perencanaan.

4.4. Pengujian LCD

4.4.1. Tujuan

Mengetahui apakah rangkaian LCD dapat menampilkan data karakter yang sesuai.

4.4.2. Peralatan Yang Dibutuhkan

1. Komputer.
2. LCD
3. Minimum sistem mikrokontroller AT89C51.
4. Catu daya 5V DC.

4.4.3. Prosedur Pengujian

1. Merangkai peralatan seperti dalam Gambar 4-3.
2. Membuat perangkat lunak pengujian rangkaian LCD. Program ini berisi inisialisasi mikrokontroller, dan LCD.

```

        Mov     SP,#40h
        Call    LCD_Initialization

        Call    LCD_Line1
        Mov     DPTR,#txt_lcd_ok
        Call    LCD_String
        Sjmp    $

txt_lcd_ok:  Db      ;0123456789abcdef
              '      LCD OK',0

Subroutine LCD_Data:
    Push     DPL
    Push     DPH
    Push     ACC
    Mov      DPTR,#1cd1
    Movx     @DPTR,A
    Call     LCD_Delay
    Lcall    LCD_Delay
    Lcall    LCD_Delay
    Lcall    LCD_Delay
    Pop      ACC
    Pop      DPH
    Pop      DPL
EndSub
Subroutine LCD_Line1:
    Push     ACC
    Mov      A,#80h
    Lcall    LCD_Command
    Pop      ACC

```

EndSub

```
lcd0    Equ    0a000h    ;lcd control operation
lcd1    Equ    lcd0+1    ;lcd data operation
```

Subroutine LCD_Command:

```
    Push    DPL
    Push    DPH
    Mov     DPTR,#lcd0
    Movx    @DPTR,A
    Lcall   LCD_Delay
    Lcall   LCD_Delay
    Pop     DPH
    Pop     DPL
    Ret
```

EndSub

Subroutine LCD_Delay:

```
    Push    01h
    Push    02h
    Push    03h
    Mov     3,#5
tundalqqa:
    Mov     1,#30
    Djnz    1,$
    Mov     2,#40
    Djnz    2,$
    Mov     2,#1
    Djnz    2,$
    Djnz    3,tundalqqa
    Pop     03h
    Pop     02h
    Pop     01h
```

EndSub

Subroutine LCD_Initialization:

```
    Mov     A,#38h    ;function set 8 bit
    Lcall   LCD_Command
    Mov     A,#0Ch    ;display on, cursor off, blink off
    Lcall   LCD_Command
    Mov     A,#06h    ;increment, no display shift
    Lcall   LCD_Command
    Mov     A,#01h    ;clear display
    Lcall   LCD_Command
```

EndSub

;-----

Subroutine LCD_String:

```
    Push    ACC
    Push    DPL
    Push    DPH
getcarl:
    Clr     A        ; mengambil data dari eprom
    Movc    A,@A+DPTR
    Cjne    A,#0,tampill ; tes apakah data habis?
    Ljmp    mettul
tampill:
    ; keluarkan data ke lcd

    Call    LCD_Data
    Inc     DPTR      ; naikan dptr
```

```

      Ljmp    getcar1
mettul:
      Pop     DPH
      Pop     DPL
      Pop     ACC
EndSub

```

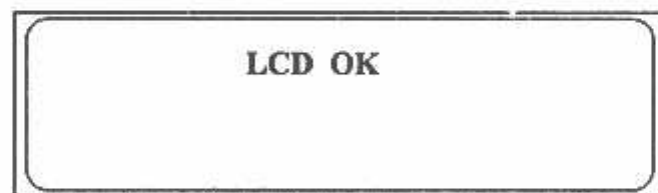
3. Mengaktifkan catu daya
4. Mengoperasikan program. Hasil keluaran akan ditunjukkan pada layar penampil LCD.



Gambar 4-3. Blok Pengujian LCD

4.4.4. Hasil Pengujian dan Analisis

Dari hasil pengujian didapatkan bahwa rangkaian LCD dapat menampilkan karakter-karakter, sesuai dengan data yang dikirimkan.



Gambar 4-4. Hasil Pengujian LCD

4.5. Pengujian PPI 8255

4.5.1. Tujuan

Untuk mengetahui unjuk kerja dari PPI 8255.

4.5.2. Peralatan Yang Dibutuhkan

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

4.5.3. Prosedur Pengujian

1. Untuk menguji PPI, sistem mikrokontroller yang sudah dibuat perangkat lunaknya dihubungkan dengan PPI dan keluaran dari masing-masing port PPI dihubungkan dengan LED *display*.

```

ppi_porta      Equ      0e000h
ppi_portb      Equ      0e001h
ppi_portc      Equ      0e002h
ppi_portcw     Equ      0e003h

    Mov        SP, #40h
    Mov        A, #80h
    Call       PPI_Initialization

    Mov        A, #0ffh
    Call       PPI_Send_Port_A
    Mov        A, #0f0h
    Call       PPI_Send_Port_B
    Mov        A, #84h
    Call       PPI_Send_Port_C
    Sjmp       $

Subroutine PPI_Initialization:
    Mov        DPTR, #ppi_portcw
    Movx       @DPTR, A          ; kirim ke port control ppi 8255

EndSub
Subroutine PPI_Send_Port_A:
    Mov        DPTR, #ppi_porta
    Movx       @DPTR, A

EndSub
Subroutine PPI_Send_Port_B:
    Mov        DPTR, #ppi_portb
    Movx       @DPTR, A

EndSub

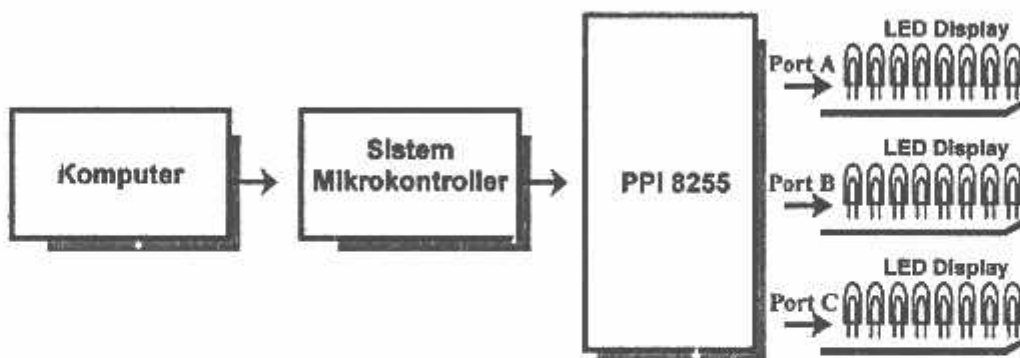
```

```

Subroutine PPI_Send_Port_C:
    Mov     DPTR, #ppi_portc
    Movx    @DPTR, A
EndSub

```

2. Merangkai rangkaian seperti pada diagram blok untuk pengujian PPI diperlihatkan dalam Gambar 4-5.
3. Bila PPI dapat bekerja dengan baik, maka LED *display* akan menyala menunjukkan kode biner dari masukan.



Gambar 4-5. Diagram Blok Pengujian PPI

4.5.4. Hasil Pengujian dan Analisis

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

Tabel 4-3. Hasil Pengujian PPI

Port	Masukan	Keluaran pada LED display							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	FFh	1	1	1	1	1	1	1	1
B	F0h	1	1	1	1	0	0	0	0
C	84h	1	0	0	0	0	1	0	0

4.6. Pengujian Relay Driver Motor DC

4.6.1. Tujuan

Untuk mengetahui unjuk kerja relay pada saat dioperasikan. Relay yang digunakan adalah relai DPDT (*dual pole dual totem*) sehingga terdapat dua keluaran yaitu NC (*normally close*) dan NO (*normally open*) yang perlu diuji.

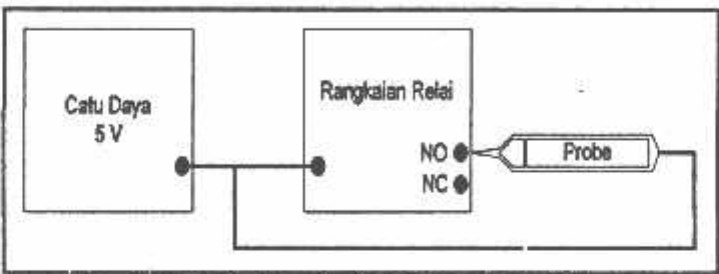
4.6.2. Peralatan Yang Dibutuhkan

- 1. Logic probe model LP-3500.
- 2. Catu daya 5V DC.
- 3. Relay

4.6.3. Prosedur Pengujian

- 1. Menghubungkan catu daya pada rangkaian relay.
 - 2. Menghubungkan *common* relay.
 - 3. Mengaktifkan catu daya dan memicu rangkaian relay.
-

- 4. Mengamati perubahan indikator *logic probe* saat *logic probe* dihubungkan pada keluaran *normally close* dan *normally open* rangkaian sebelum dan sesudah pemicuan.



Gambar 4-6. Rangkaian Pengujian Relay

4.6.4. Hasil Pengujian dan Analisis

Hasil pengujian ditunjukkan pada Tabel 4-4.

Tabel 4-4. Hasil Pengujian Relay

Keluaran	Indikator <i>Logic Probe</i>	
	Sebelum Pemicuan	Sesudah Pemicuan
<i>NormallyOpen</i> (NO)	Hijau (low '0')	Merah (high '1')
<i>NormallyClose</i> (NC)	Merah (high '1')	Hijau (low '0')

Saat terjadinya pemicuan, arus kolektor (I_C) pada transistor akan mengalir menuju *ground*. Pada keadaan *normally open* sebelum terjadinya pemicuan *common* tidak terhubung dengan keluaran, sedangkan pada keadaan *normally close common* terhubung dengan keluaran sebelum terjadinya pemicuan.

4.7. Pengujian Motor DC

4.7.1. Tujuan

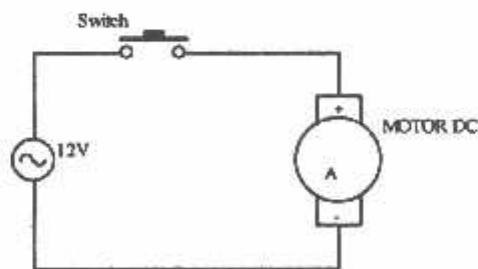
Untuk mengetahui unjuk kerja dari motor DC apakah bisa berjalan atau tidak.

4.7.2. Peralatan Yang Dibutuhkan

1. Motor DC 12 V.
2. Saklar on-off.
3. Catu daya 12V DC.

4.7.3. Prosedur Pengujian

1. Merangkai peralatan seperti pada Gambar 4-7.
2. Menghidupkan catu daya, menekan saklar on-off untuk mengaktifkan kerja dari motor DC.
3. Mengamati kerja motor DC, pada waktu posisi saklar off dan pada waktu posisi saklar on.



Gambar 4-7. Rangkaian Pengujian Motor DC

4.7.4. Hasil Pengujian dan Analisis

Setelah catu daya diaktifkan, posisi saklar pada posisi off, motor DC tidak berputar (tidak aktif). Pada posisi saklar on, motor DC berputar (aktif). Apabila saklar di offkan lagi, maka motor DC tidak berputar. Hasil pengujian terlihat bahwa rangkaian pengujian motor DC yang telah direalisasikan sesuai dengan unjuk kerja perencanaan.

4.8. Pengujian Keseluruhan Sistem

4.8.1. Tujuan

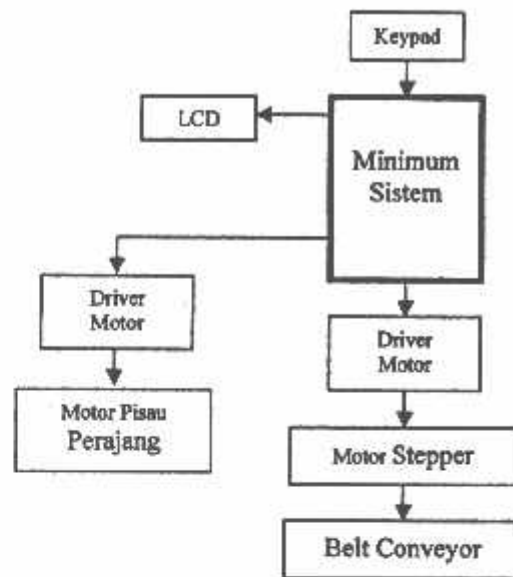
Untuk mengetahui kerja sistem secara keseluruhan, apakah telah berhasil sesuai dengan yang diharapkan.

4.8.2. Peralatan Yang Dibutuhkan

1. Modul rangkaian sistem mikrokontroller secara keseluruhan.
2. Catu daya 12V DC.
3. Tempe.

4.8.3. Prosedur Pengujian

1. Membuat program untuk menjalankan sistem secara keseluruhan sesuai dengan spesifikasi yang telah dibuat.
 2. Merangkai keseluruhan modul seperti pada gambar 4-8.
-



Gambar 4-8. Blok Diagram Rangkaian Pengujian Keseluruhan Sistem

3. Menghubungkannya dengan catu daya.
4. Menjalankan alat dengan langkah-langkah sebagai berikut :
 - a. Menaruh tempe pada konveyor.
 - b. Menekan tombol 1 sebagai pengesetan ketebalan tempe serta melihat tampilannya pada LCD.
 - c. Menekan tombol 2 untuk menjalankan semua sistem serta melihat hasil dari perajangan.

4.8.4. Hasil Pengujian dan Analisis.

Pada bagian ini semua komponen pendukung seperti keypad, LCD, PPI, driver motor dan mikrokontroller dihubungkan sesuai dengan perancangan sistem.

Dalam pengujian ini untuk melakukan pengesetan yaitu dengan menekan tombol pada keypad dan sebelumnya disitu akan tampil pada unit tampilan inisialisasi LCD dan mikrokontroller yaitu menu utama yang terdiri dari set ketebalan (pada angka 1) dan set untuk menjalankan sistem (pada angka 2). Set ketebalan (menekan angka 1) yaitu untuk pengesetan ketebalan tempe yang diinginkan. Pengesetan ketebalan ini untuk mengontrol pergerakan dari motor stepper dimana ketebalan tempe yang diinginkan diatur melalui pergerakan motor stepper. Setelah proses pengesetan selesai maka selanjutnya melakukan set untuk menjalankan sistem (menekan angka 2). Apabila semua proses ingin dihentikan, maka menekan reset dan akan kembali pada posisi awal inisialisasi. Pada proses ini semua kerja sistem berjalan sesuai dengan sistem yang telah direncanakan.

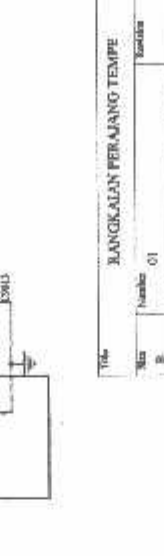
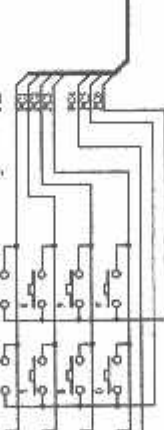
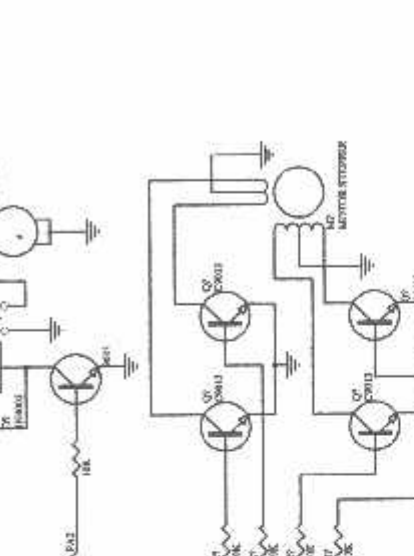
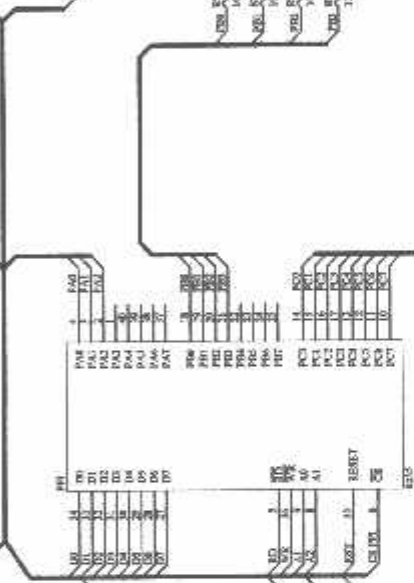
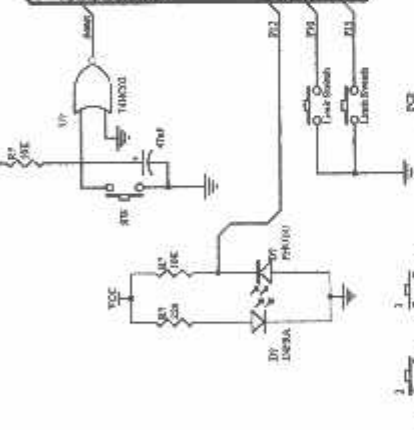
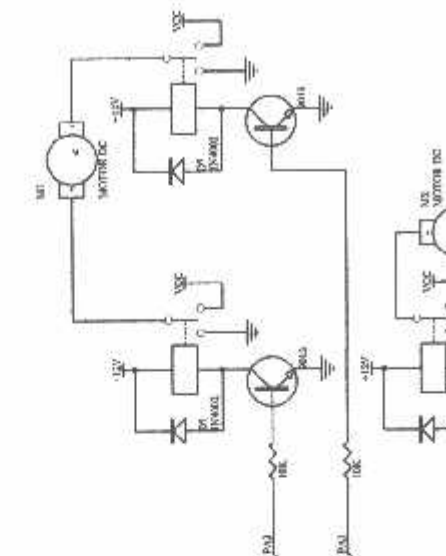
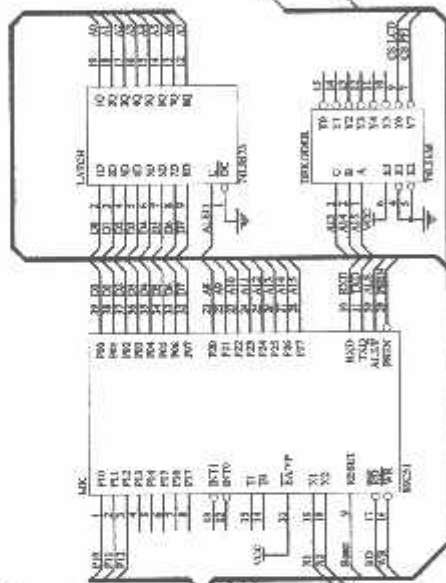
BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil pengujian dan analisa dari alat yang telah dibuat maka dapat disimpulkan :

1. Pada pengujian mikrokontroller AT89C51, mikrokontroller dapat berjalan dengan baik sebagai pengendali sistem.
2. Pada pengujian tampilan kristal cair, LCD dapat menampilkan informasi yang sedang berlangsung dengan baik sesuai dengan program.
3. Pada pengujian rangkaian keypad yang berfungsi sebagai masukan data referensi dapat berjalan dengan baik.
4. Relay sebagai pengendali yang akan memutus dan menghubungkan arus ke motor DC dengan baik sesuai dengan yang diharapkan.
5. Untuk limit switch yang digunakan sebagai saklar pembatas dari pergerakan motor DC (pisau perajang) menggunakan limit switch *normaly open*.
6. Untuk mendeteksi adanya tempe pada konveyor menggunakan LED infra red dan photodiode sehingga semua sistem bisa mulai bekerja.
7. Pengaturan ketebalan tempe diatur oleh motor stepper dengan pergerakan stepnya.
8. Penekanan tombol reset, maka semua sistem akan kembali pada kondisi awal (proses inisialisai LCD dan Mikrokontroller).



BANGKALAN PERAJANG TEMPE

No.	Nama	Nomor	Nilai
1	1	01	1

Disusun oleh:
Nama: ...
NIM: ...

Disetujui oleh:
Nama: ...
NIM: ...



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

LEMBAR BIMBINGAN SKRIPSI

Nama : Wahyu Ardiansyah
NIM : 9917107
Jurusan : Teknik Elektro S-1
Program Studi : Elektronika
Judul Skripsi : Rancang Bangun Alat Perajang Tempe
Secara Otomatis Berbasis Mikrokontroller
AT89C51
Tanggal Pengajuan Skripsi : 25 Oktober 2004
Selesai Penulisan Skripsi : 26 Maret 2005
Dosen Pembimbing : Ir. Usman Djuanda, MM
Ir. Mimien Musikawati.
Telah Dievaluasi Dengan Nilai : 90 (Sembilan Puluh) *Sm*

Diperiksa dan disetujui

Dosen Pembimbing I

Ir. Usman Djuanda, MM

Dosen Pembimbing II

Ir. Mimien Mustikawati

Mengetahui
Ketua Jurusan Elektro S-1

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



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

FORMULIR PERBAIKAN SKRIPSI

Dari hasil ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro
Konsentrasi Elektronika yang diselenggarakan pada :

Hari : Selasa

Tanggal : 29 Maret 2005

Telah dilakukan perbaikan Skripsi oleh :

Nama : Wahyu Ardiansyah

NIM : 9917107

Jurusan : Teknik Elektro S-1

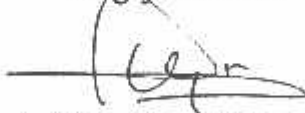
Program Studi : Elektronika

Judul Skripsi : Rancang Bangun Alat Perajang Tempe Secara Otomatis
Berdasarkan Mikrokontroler AT89C51

No.	Materi Perbaikan	Keterangan
1.	Deskripsi PPI 8255	
2.	Pengujian • Keypad • LCD dst	

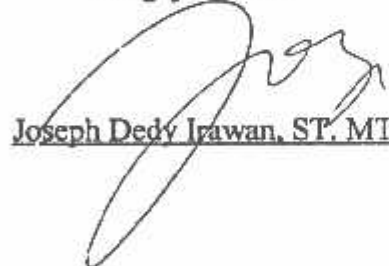
Disetujui

Penguji Pertama



Ir. Sidik Noertjahjono, MT

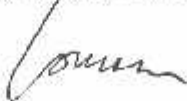
Penguji Kedua



Joseph Dedy Irawan, ST. MT

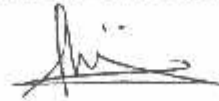
Mengetahui

Dosen Pembimbing I



Ir. Usman Djuanda, MM

Dosen Pembimbing II



Ir. Mimien Mustikawati



Formulir Perbaikan Ujian Skripsi

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

NAMA : WATHYU ARDIANSYAH
NIM : 9917107
Perbaikan meliputi :

Deskripsi tes PPI 8285.

Malang,

200

(Wathyu Ardiansyah)
29/3/20



Formulir Perbaikan Ujian Skripsi

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

NAMA

Wahyu Ardiansyah

NIM

0917107

Perbaikan meliputi

*1) Penugasan

→ KLPD

→ LCP

→ RPI

dan

Malang, 24 - 3 - 2005

(Joseph D.)



FORMULIR BIMBINGAN SKRIPSI

Nama : Wahyu Ardiansyah
Nim : 9917107
Masa Bimbingan : 7-Feb-2005 s/d 10-Aug-2005
Judul Skripsi : Rancang bangun alat perajang tempe secara otomatis berbasis Mikrokontroller AT89C51

NO	Tanggal	Uraian	Paraf Pembimbing
1.	10 Februari 2005	Ace Bab I	
2.	13 Februari 2005	Bab II Revisi no halaman	
3.	18 Februari 2005	Ace Bab II	
4.	22 Februari 2005	Bab III revisi jarak qbr dgn ket	
5.	28 Februari 2005	Bab III revisi qbr blok diagram	
6.	8 Maret 2005	Ace Bab III	
7.		Ace Bab IV Umpre Demo alat	
8.			
9.			
10.			

Malang, 2005
Dosen Pembimbing

Ir. Usman Djuanda, MM

Form. S-4a

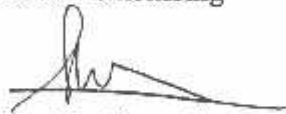


FORMULIR BIMBINGAN SKRIPSI

Nama : Wahyu Ardiansyah
Nim : 9917107
Masa Bimbingan : 7-Feb-2005 s/d 10-Aug-2005
Judul Skripsi : Rancang bangun alat perajang tempe secara otomatis berbasis Mikrokontroller AT89C51

NO	Tanggal	Uraian	Paraf Pembimbing
1.	17/05/05	Bab III revisi	NS
2.		Bab II revisi	NS
3.		Buat Bab IV + V buat makalah seminar	NS
4.		Demo alat Hiy segera dikumpulkan	NS
5.		revisi Bab IV + V	NS
6.		Revisi Demo	NS
7.		Revisi Kumpulan	NS
8.			
9.			
10.			

Malang, 2005
Dosen Pembimbing


Ir. Mimien Mustikawati

Form. S-4a

SensorTempe	Bit	P1.2
LimitKanan	Bit	P1.0
LimitKiri	Bit	P1.1

Tebal Data 20h

```

Mov     SP,#40h
Call    LCD_Initialization
Mov     R0,#1
Call    Keypad_Initialization
; aktifkan motor iris
Mov     A,#00
Call    PPI_Send_Port_A

Mov     A,#00
Call    PPI_Send_Port_B

Mov     A,Tebal
Cjne    A,#1,CekTebal2
Jump    Start
CekTebal2:
Cjne    A,#2,CekTebal3
Jump    Start
CekTebal3:
Mov     Tebal,#1
Start:
Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR,#txt_menu1
Call    LCD_String
Call    LCD_Line2
Mov     DPTR,#txt_menu2
Call    LCD_String
Pilih1:
Mov     R0,#2
Call    Keypad_Get
Cjne    A,'#1',Pilih2
Jump    Menu_Pilih_Tebal
Pilih2:
Cjne    A,'#2',Pilih1
Jump    Menu_Jalan
;-----
Menu_Pilih_Tebal:
Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR,#txt_menu11
Call    LCD_String
PilihTebal1:
Mov     R0,#2
Call    Keypad_Get
Cjne    A,'#1',PilihTebal2
Mov     Tebal,#1
Jump    Start
PilihTebal2:
Cjne    A,'#2',PilihTebal1
Mov     Tebal,#2

```

```

      Jmp      Start
;-----
Menu_Jalan:
      Call    LCD_Clrscr
      Call    LCD_Line1
      Mov     DPTR,#txt_menu21
      Call    LCD_String
      Call    LCD_Line2
      Mov     DPTR,#txt_menu22
      Call    LCD_String
      Mov     A,Tebal
      Orl     A,#30h
      Call    LCD_Data
      Mov     A,#01
Menu_Jalan1:
      Call    PPI_Send_Port_B
      Mov     R0,#10
      Call    Delay_Var_10ms
      Jb      SensorTempe,AdaTempe
      Rl      A
      Cjne    A,#10h,Menu_Jalan1
      Mov     A,#01
      Jmp     Menu_Jalan1

      ; ekstra panjang
AdaTempe:
      Mov     A,#00
      Call    PPI_Send_Port_B
      Mov     R2,#2
AdaTempe1:
      Mov     A,#01
AdaTempe2:
      Call    PPI_Send_Port_B
      Mov     R0,#10
      Call    Delay_Var_10ms
      Rl      A
      Cjne    A,#10h,AdaTempe2
      Jmp     MajukanTempe
AdaTempe3:
      Jmp     AdaTempe2

MajukanTempe:
      Call    LCD_Clrscr
      Call    LCD_Line1
      Mov     DPTR,#txt_menu21
      Call    LCD_String
      Call    LCD_Line2
      Mov     DPTR,#txt_menu22
      Call    LCD_String
      Mov     A,Tebal
      Orl     A,#30h
      Call    LCD_Data
      Mov     A,Tebal
      Cjne    A,#1,MajukanTempe1
      Call    Conveyer_Majul
      Jmp     Proses_Iris

```

```

MajukanTempel:
    Call    Conveyer_Maju2
    Jmp     Proses_Iris

Proses_Iris:
    Call    LCD_Clrscr
    Call    LCD_Line1
    Mov     DPTR,#txt_Iris
    Call    LCD_String
    Call    LCD_Line2
    Mov     DPTR,#txt_menu22
    Call    LCD_String
    Mov     A,Tebal
    Orl     A,#30h
    Call    LCD_Data

Proses_Iris1:
    Mov     A,#01
    Call    PPI_Get_Port_A
    Call    PPI_Send_Port_A
    Call    Delay_Fix_100ms
    ; aktifkan motor kiri
    Call    PPI_Get_Port_A
    Anl     A,#11111110b
    Orl     A,#00000010b
    Call    PPI_Send_Port_A
    ;tunggu sampai menyentuh limit switch
    Jb      LimitKiri,$
    ; aktifkan motor kanan
    Call    PPI_Get_Port_A
    Anl     A,#11111001b
    Orl     A,#00000001b
    Call    PPI_Send_Port_A
    Jb      LimitKanan,$
    Jmp     MajukanTempe

Subroutine Conveyer_Maju1:
    Mov     A,#01
    Call    PPI_Send_Port_B
    Mov     R0,#10
    Call    Delay_Var_10ms

    Rl     A
    Call    PPI_Send_Port_B

EndSub
Subroutine Conveyer_Maju2:
    Mov     A,#01
    Rl     A
    Call    PPI_Send_Port_B
    Mov     R0,#10
    Call    Delay_Var_10ms

    Rl     A
    Call    PPI_Send_Port_B
    Mov     R0,#10
    Call    Delay_Var_10ms

```

```

        Rl      A
        Call    PPI_Send_Port_B
        Mov     R0,#10
        Call    Delay_Var_10ms

        Mov     A,#00
        Call    PPI_Send_Port_B
EndSub

;*****
; subrutin clear lcd1
;*****
Subroutine LCD_Clrscr:
    Push     ACC
    Mov      A,#01h
    Call     LCD_Command
    Pop      ACC
EndSub

;*****
; subrutin menempatkan kursor
;*****
Subroutine LCD_Cursor_Position:
    Push     ACC
    Mov      A,R0
    Anl      A,#0F0h
    Cjne     A,#10h,lcd_cursor_position1
    Mov      A,R0
    Anl      A,#0Fh
    Orl      A,#80h
    Call     LCD_Command
    Jmp      lcd_cursor_position_end

lcd_cursor_position1:
    Cjne     A,#20h,lcd_cursor_position_end
    Mov      A,R0
    Anl      A,#0Fh
    Orl      A,#0C0h
    Call     LCD_Command
lcd_cursor_position_end:
    Pop      ACC
EndSub

;*****
; kirim data ke lcd1
;*****
Subroutine LCD_Data:
    Push     DPL
    Push     DPH
    Push     ACC
    Mov      DPTR,#lcd1
    Movx     @DPTR,A
    Call     LCD_Delay
    Lcall    LCD_Delay
    Lcall    LCD_Delay
    Lcall    LCD_Delay
    Pop      ACC
    Pop      DPH
    Pop      DPL

```

```

EndSub

;*****
; kirim instruksi ke lcd1
;*****
lcd0 Equ 0a000h ;lcd control operation
lcd1 Equ lcd0+1 ;lcd data operation
Subroutine LCD_Command:
    Push DPL
    Push DPH
    Mov DPTR,#lcd0
    Movx @DPTR,A
    Lcall LCD_Delay
    Lcall LCD_Delay
    Pop DPH
    Pop DPL
    Ret
EndSub
Subroutine LCD_Delay:
    Push 01h
    Push 02h
    Push 03h
    Mov 3,#5
tundalqqqa:
    Mov 1,#30
    Djnz 1,$
    Mov 2,#40
    Djnz 2,$
    Mov 2,#1
    Djnz 2,$
    Djnz 3,tundalqqqa
    Pop 03h
    Pop 02h
    Pop 01h
EndSub
Subroutine LCD_Delete_Line1:
    Push ACC
    Push DPH
    Push DPL
    Call LCD_Line1
    Mov DPTR,#txt_blank
    Call LCD_String
    Call LCD_Line1
    Pop DPL
    Pop DPH
    Pop ACC
EndSub
Subroutine LCD_Delete_Line2:
    Push ACC
    Push DPH
    Push DPL
    Call LCD_Line2
    Mov DPTR,#txt_blank
    Call LCD_String
    Call LCD_Line2
    Pop DPL

```

```

        Pop    DPH
        Pop    ACC
        Ret
txt_blank:    Db    ' ',0
EndSub
Subroutine LCD_Initialization:
    Mov    A,#38h    ;function set 8 bit
    Lcall  LCD_Command
    Mov    A,#0Ch    ;display on, cursor off, blink off
    Lcall  LCD_Command
    Mov    A,#06h    ;increment, no display shift
    Lcall  LCD_Command
    Mov    A,#01h    ;clear display
    Lcall  LCD_Command
EndSub
Subroutine LCD_Line1:
    Push   ACC
    Mov    A,#50h
    Lcall  LCD_Command
    Pop    ACC
EndSub
Subroutine LCD_Line2:
    Push   ACC
    Mov    A,#0C0h
    Lcall  LCD_Command
    Pop    ACC
EndSub
Subroutine LCD_Off:
    Push   ACC
    Mov    A,#00001000b
    Lcall  LCD_Command
    Pop    ACC
EndSub
Subroutine LCD_On:
    Push   ACC
    Mov    A,#00001100b
    Lcall  LCD_Command
    Pop    ACC
EndSub
;-----
Subroutine LCD_String:
    Push   ACC
    Push   DPL
    Push   DPH
getcarl:
    Clr    A        ; mengambil data dari eprom
    Movc   A,@A+DPTR
    Cjne   A,#0,tammpill    ; tes apakah data habis?
    Ljmp   mettul
tammpill:
    ; keluarkan data ke lcd
    Call   LCD_Data
    Inc    DPTR        ; naikkan dptr
    Ljmp   getcarl
mettul:
    Pop    DPH

```

```

        Pop    DPL
        Pop    ACC
EndSub

;*****
; subrutin tampil text pada lcd1
; dari data di ram
;*****
Subroutine LCD_String_Ram:
    Push    ACC
    Push    DPL
    Push    DPH
    Push    00h
getcarlz:
    Movx    A,@DPTR
    Cjne    A,#0,tammpillz    ; tes apakah data habis?
    Ljmp    mettulz
tammpillz:    ; keluarkan data ke lcd
    Call    LCD_Data
    Inc     DPTR    ; naikan dptr
    Ljmp    getcarlz

mettulz:
    Pop     00h
    Pop     DPH
    Pop     DPL
    Pop     ACC
EndSub
Subroutine LCD_Blink_Off:
    Push    ACC
    Mov     A,#00001100b    ;
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Blink_On:
    Push    ACC
    Mov     A,#00001101b    ;
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Cursor_Off:
    Push    ACC
    Mov     A,#00001100b    ; cursor on
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Cursor_On:
    Push    ACC
    Mov     A,#00001110b    ; cursor on
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Cursor_Left:
    Push    ACC
    Mov     A,#00010000b    ; cursor left
    Lcall   LCD_Command

```

```

        Pop      ACC
EndSub
Subroutine LCD_Cursor_Right:
    Push      ACC
    Mov       A,#00010100b    ; cursor left
    Lcall     LCD_Command
    Pop       ACC
EndSub
Subroutine LCD_Display_Left:
    Push      ACC
    Mov       A,#00011000b    ; display left
    Lcall     LCD_Command
    Pop       ACC
EndSub
Subroutine LCD_Display_Right:
    Push      ACC
    Mov       A,#00011100b    ; display right
    Lcall     LCD_Command
    Pop       ACC
EndSub
Subroutine LCD_Cursor_Home:
    Push      ACC
    Mov       A,#00000010b    ; display right
    Lcall     LCD_Command
    Pop       ACC
EndSub
Subroutine LCD_Hexa:
    Push      07h
    Push      ACC

    Mov       7,A
    Anl       A,#070h
    Swap      A
    Orl       A,#30h
    Lcall     tes_huruf_
    Call      LCD_Data
    Mov       A,7
    Anl       A,#0Fh
    Orl       A,#30h
    Lcall     tes_huruf_
    Lcall     LCD_Data

    Pop       ACC
    Pop       07h
    Ret

tes_huruf_:
    Cjne      A,#3Ah,tes_huruf_1
    Mov       A,'#a'
    Ret
tes_huruf_1:
    Cjne      A,#3Bh,tes_huruf_2
    Mov       A,'#b'
    Ret
tes_huruf_2:
    Cjne      A,#3Ch,tes_huruf_3

```

```

        Mov     A, #'c'
        Ret
tes_huruf_3:
        Cjne   A, #3Dh, tes_huruf_4
        Mov     A, #'d'
        Ret
tes_huruf_4:
        Cjne   A, #3Eh, tes_huruf_5
        Mov     A, #'e'
        Ret
tes_huruf_5:
        Cjne   A, #3Fh, tes_huruf_6
        Mov     A, #'f'
        Ret
tes_huruf_6:
        Ret

EndSub

ppi_porta      Equ     0e000h
ppi_portb      Equ     0e001h
ppi_portc      Equ     0e002h
ppi_portcw     Equ     0e003h

Subroutine PPI_Get_Port_A:
        Push   DPH
        Push   DPL
        Mov    DPTR, #ppi_porta
        Movx   A, @DPTR
        Pop    DPL
        Pop    DPH
EndSub

Subroutine PPI_Get_Port_B:
        Push   DPH
        Push   DPL
        Mov    DPTR, #ppi_portb
        Movx   A, @DPTR
        Pop    DPL
        Pop    DPH
EndSub

Subroutine PPI_Get_Port_C:
        Push   DPH
        Push   DPL
        Mov    DPTR, #ppi_portc
        Movx   A, @DPTR
        Pop    DPL
        Pop    DPH
EndSub

Subroutine PPI_Initialization:
        Push   DPH
        Push   DPL
        Mov    DPTR, #ppi_portcw
        Movx   @DPTR, A ; kirim ke port control ppi 8255
        Pop    DPL
        Pop    DPH
EndSub

Subroutine PPI_Send_Port_A:
        Push   DPH

```

```

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

        Mov     7,A
        Anl     A,#0F0h
        Swap    A
        Orl     A,#30h
        Call    tes_huruf_
        Call    Serial_Transmit
        Mov     A,7
        Anl     A,#0Fh
        Orl     A,#30h

```

```

        Lcall    tes_huruf_
        Lcall    Serial_Transmit

        Pop      ACC
        Pop      07h
        Ret

EndSub

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

xtampill:          ; keluarkan data ke lcd
        Lcall    Serial_Transmit
        Inc      DPTR        ; naikkan dptr
        Ljmp     xgetcarl
xmettul:
        Pop      DPH
        Pop      DPL
        Pop      ACC

EndSub
Subroutine Serial_Transmit_String_Ram:
        Push     ACC
        Push     DPL
        Push     DPH
axgetcarl:
        Movx     A,@DPTR

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

axtampill:          ; keluarkan data ke lcd

        Call     Serial_Transmit
        Inc      DPTR        ; naikkan dptr
        Ljmp     axgetcarl
axmettul:
        Pop      DPH
        Pop      DPL
        Pop      ACC

EndSub
;=====
;===== MODUL KEYPAD =====
keypad_id_keypad      Equ      02h      ;register r2
keypad_cek_key_88      Equ      03h      ;register r3
keypad_temp_baris      Equ      04h      ;register r4
keypad_temp_data       Equ      05h      ;register r5
keylx                  Equ      06h
keydelay               Equ      07h

```

```

keydat1      Equ      08h
keydat2      Equ      09h
keypad_curpos Equ      10h
keypad_port_a Equ      0e000h
keypad_port_b Equ      0e001h
keypad_port_c Equ      0e002h
keypad_port_cw Equ     0e003h
keypad_shift Equ      06h
Keypad_cons_1: Db      '1^',0
Keypad_cons_2: Db      'ABC2',0
Keypad_cons_3: Db      'DEF3',0
Keypad_cons_4: Db      'GHI4',0
Keypad_cons_5: Db      'JKL5',0
Keypad_cons_6: Db      'MNO6',0
Keypad_cons_7: Db      'PQRS7',0
Keypad_cons_8: Db      'TUV8',0
Keypad_cons_9: Db      'WXYZ9',0
Keypad_cons_0: Db      '0 ',0
Keypad_cons_10: Db     '.,?!()''',0
Keypad_dphh Equ      0Ah
Keypad_dp11 Equ      0Bh
Keypad_dphh1 Equ      0Ch
Keypad_dp111 Equ     0Dh
Keypad_dphh2 Equ     0Eh
Keypad_dp112 Equ     0Fh
Subroutine Keypad_Initialization:
    Push    DPL
    Push    DPH
    Mov     A,R0
    Jmp     keypad_end_init
keypad_88_init:
    Cjne    A,#2,keypad_end_initx
    Mov     A,#82h ;pa:out pb:in pc:out
    Mov     DPTR,#keypad_port_cw
    Movx    @DPTR,A
keypad_end_init:
    Pop     DPH
    Pop     DPL
EndSub
Subroutine Keypad_Alpha:
    Mov     keypad_shift,#0
keypad_alpa1:
    Mov     R0,#2
    Call    Keypad_Get
    Jmp     proses_keypad_alpha
proses_key_2:
    Cjne    A,'#2',proses_key_3
    Mov     DPTR,#Keypad_cons_2
    Jmp     proses_keypad_alpha
proses_key_3:
    Cjne    A,'#3',proses_key_4
    Mov     DPTR,#Keypad_cons_3
    Jmp     proses_keypad_alpha
proses_key_4:
    Cjne    A,'#4',proses_key_5
    Mov     DPTR,#Keypad_cons_4

```

```

        Jmp      proses_keypad_alpha
proses_key_5:
        Cjne    A,#'5',proses_key_6
        Mov     DPTR,#Keypad_cons_5
        Jmp     proses_keypad_alpha
proses_key_6:
        Cjne    A,#'6',proses_key_7
        Mov     DPTR,#Keypad_cons_6
        Jmp     proses_keypad_alpha
proses_key_7:
        Cjne    A,#'7',proses_key_8
        Mov     DPTR,#Keypad_cons_7
        Jmp     proses_keypad_alpha
proses_key_8:
        Cjne    A,#'8',proses_key_9
        Mov     DPTR,#Keypad_cons_8
        Jmp     proses_keypad_alpha
proses_key_9:
        Cjne    A,#'9',proses_key_0
        Mov     DPTR,#Keypad_cons_9
        Jmp     proses_keypad_alpha
proses_key_0:
        Cjne    A,#'0',proses_key_a
        Mov     DPTR,#Keypad_cons_0
        Jmp     proses_keypad_alpha
proses_key_a:
        Mov     B,A
        Call    LCD_Data
        Mov     A,B
        Ret
proses_keypad_alpha:
        Mov     A,keypad_shift
        Movc    A,@A+DPTR
        Cjne    A,#0,proses_keypad_alphaxx
        Ret
proses_keypad_alphaxx:
        Call    Delay_Fix_100ms
        Mov     B,A
        Call    LCD_Data
        Mov     A,B
        Ret
EndSub
Subroutine Keypad_Get:
        Call    keypad_get_44
        Mov     B,A
        Mov     A,R2
        Cjne    A,#01,Keypad_Get
        Call    Delay_Fix_100ms
        Call    Delay_Fix_100ms
        Call    Delay_Fix_100ms
        Mov     A,B
EndSub
Subroutine Keypad_Get_Matrix_44:
;-----
keypad_get_matrik_44:
keypad_loop_awal_44:

```

```

keypad_get_44:
    Mov     A,#8Ch                ;kolom satu
    Mov     keypad_temp_baris,A   ;kolom
keypad_loopstart_44:
    Call    keypad_get_port_cl
    Anl     A,#00001111b
    Rr      A                    ;naikan kolom
    Cjne    A,#08h,keypad_loop_44 ;cek jika > 4 baris
    Ret

;-----
;          BARIS SATU
;-----
keypad_baris1_44:
    Mov     keypad_id_keypad,#1   ;simpan id keypad
(<>0:ditekan)
    Cjne    A,#00000001b,keypad_baris2_44 ;baris 1
    Mov     A,keypad_temp_baris
    Cjne    A,#10000000b,keypad_kolom21_44 ;kolom 1
    Mov     A,#'1'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_kolom21_44:
    Cjne    A,#01000000b,keypad_kolom31_44 ;kolom 2
    Mov     A,#'2'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_kolom31_44:
    Cjne    A,#00100000b,keypad_kolom41_44 ;kolom 3
    Mov     A,#'3'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_kolom41_44:
    Cjne    A,#00010000b,keypad_end_baris1_44 ;kolom 5
    Mov     A,#'A'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_end_baris1_44:
    Jump    keypad_end_matrik1_44

;-----
;          BARIS DUA
;-----
keypad_baris2_44:
    Cjne    A,#00000010b,keypad_baris3_44 ;baris 2
    Mov     A,keypad_temp_baris
    Cjne    A,#10000000b,keypad_kolom22_44 ;kolom 1
    Mov     A,#'4'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_kolom22_44:
    Cjne    A,#01000000b,keypad_kolom32_44 ;kolom 2
    Mov     A,#'5'
    Mov     keypad_temp_data,A
    Jump    keypad_end_matrik_44
keypad_kolom32_44:
    Cjne    A,#00100000b,keypad_kolom42_44 ;kolom 3

```

```

Mov     A,#'E'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom42_44:
Cjne    A,#00010000b,keypad_end_baris2_44 ;kolom 5
Mov     A,#'B'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_end_baris2_44:
Jmp     keypad_end_matrik1_44

```

```

;-----
;          BARIS TIGA
;-----

```

```

keypad_baris3_44:
Cjne    A,#00000100b,keypad_baris4_44 ;baris 3
Mov     A,keypad_temp_baris
Cjne    A,#10000000b,keypad_kolom23_44 ;kolom 1
Mov     A,#'7'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom23_44:
Cjne    A,#01000000b,keypad_kolom33_44 ;kolom 2
Mov     A,#'8'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom33_44:
Cjne    A,#00100000b,keypad_kolom43_44 ;kolom 3
Mov     A,#'9'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom43_44:
Cjne    A,#00010000b,keypad_end_baris3_44 ;kolom 5
Mov     A,#'C'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_end_baris3_44:
Jmp     keypad_end_matrik1_44

```

```

;-----
;          BARIS EMPAT
;-----

```

```

keypad_baris4_44:
Cjne    A,#00001000b,keypad_baris4_44 ;baris 4
Mov     A,keypad_temp_baris
Cjne    A,#10000000b,keypad_kolom24_44 ;kolom 1
Mov     A,#'+'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom24_44:
Cjne    A,#01000000b,keypad_kolom34_44 ;kolom 2
Mov     A,#'0'
Mov     keypad_temp_data,A
Jmp     keypad_end_matrik_44
keypad_kolom34_44:
Cjne    A,#00100000b,keypad_kolom44_44 ;kolom 3

```

```

        Mov     A,###'
        Mov     keypad_temp_data,A
        Jmp     keypad_end_matrik_44
keypad_kolom44_44:
        Cjne    A,#00010000b,keypad_end_baris4_44    ;kolom 4
        Mov     A, #'D'
        Mov     keypad_temp_data,A
        Jmp     keypad_end_matrik_44
keypad_end_baris4_44:
        Jmp     keypad_end_matrik1_44
;----- end deteksi keypad -----

keypad_end_matrik_44:
        Call    keypad_syarat_key_88
        Jmp     keypad_end_matrik2_44
keypad_end_matrik1_44:
        Jmp     keypad_loop_awal_44
keypad_end_matrik2_44:
        Mov     A,keypad_temp_data
        Ret

;-----
; send port c high
;-----
keypad_send_port_ch:
        Mov     DPTR,#keypad_port_c
        Movx    @DPTR,A
        Ret
;---- end of send port ch ----

;-----
; get port c low
;-----
keypad_get_port_cl:
        Mov     DPTR,#keypad_port_c
        Movx    A,@DPTR
        Ret
;--- end of get port cl
;-----
; Syarat tampilkan data
;-----
keypad_syarat_key_88:
        Mov     A,R0    ;syarat tampilkan karakter
        Cjne    A,#0,keypad_sandi_data_88
        Mov     A,keypad_temp_data
        Jmp     keypad_tampil_data_88
keypad_sandi_data_88:
        Cjne    A,#1,keypad_hide_data_88
        Mov     A,R1    ;sandi
        Jmp     keypad_tampil_data_88
keypad_hide_data_88:
        Jmp     keypad_ora_tampil_88
keypad_tampil_data_88:
        Lcall   LCD_Data
keypad_ora_tampil_88:
        Ret
key1:

```

```

        Push    DPH
        Push    DPL
        Mov     keylx, #2
        Mov     keydelay, #50
keyin1:
        Mov     R0, keylx
        Call    keypad_get_44
        Mov     keydat1, A
        Mov     A, R2
        Cjne    A, #01, keyin1
        Mov     A, keydat1
        Jump    keyin12
keyin12:
        Cjne    A, #'1', keyin13
        Mov     DPTR, #Keypad_cons_1
        Jump    keyin_proses
keyin13:
        Cjne    A, #'2', keyin14
        Mov     DPTR, #Keypad_cons_2
        Jump    keyin_proses
keyin14:
        Cjne    A, #'3', keyin15
        Mov     DPTR, #Keypad_cons_3
        Jump    keyin_proses
keyin15:
        Cjne    A, #'4', keyin16
        Mov     DPTR, #Keypad_cons_4
        Jump    keyin_proses
keyin16:
        Cjne    A, #'5', keyin17
        Mov     DPTR, #Keypad_cons_5
        Jump    keyin_proses
keyin17:
        Cjne    A, #'6', keyin18
        Mov     DPTR, #Keypad_cons_6
        Jump    keyin_proses
keyin18:
        Cjne    A, #'7', keyin19
        Mov     DPTR, #Keypad_cons_7
        Jump    keyin_proses
keyin19:
        Cjne    A, #'8', keyin20
        Mov     DPTR, #Keypad_cons_8
        Jump    keyin_proses
keyin20:
        Cjne    A, #'9', keyin21
        Mov     DPTR, #Keypad_cons_9
        Jump    keyin_proses
keyin21:
        Cjne    A, #'0', keyin22
        Mov     DPTR, #Keypad_cons_0
        Jump    keyin_proses
keyin22:
        Cjne    A, #'*', keyin23
        Mov     DPTR, #Keypad_cons_10
        Jump    keyin_proses

```

```

keyin23:
    Jmp      keyin_end
;-----
keyin_proses:
    Mov      Keypad_dphh,DPH
    Mov      Keypad_dp11,DPL
    Mov      Keypad_dphh1,DPH
    Mov      Keypad_dp111,DPL
    Mov      DPH,Keypad_dphh
    Mov      DPL,Keypad_dp11
    Clr      A
    Movc     A,@A+DPTR
    Mov      B,A
    Mov      R0,keypad_curpos
    Call     LCD_Cursor_Position
    Mov      A,B
    Call     LCD_Data
    Mov      A,#0
    Call     PPI_Send_Port_C
    Mov      R0,keydelay
    Call     Delay_Var_10ms
keyin100:
    Mov      R0,#2
    Call     keypad_get_44
    Mov      keydat2,A
    Mov      A,R2
    Cjne     A,#01,keyin101a
    Jmp      keyin_proses1
keyin101a:
    Mov      A,#0
    Call     PPI_Send_Port_C
    Mov      DPH,Keypad_dphh
    Mov      DPL,Keypad_dp11
    Clr      A
    Movc     A,@A+DPTR
    Cjne     A,#C,keyin_end
    Mov      DPH,Keypad_dphh1
    Mov      DPL,Keypad_dp111
    Clr      A
    Movc     A,@A+DPTR
    Jmp      keyin_end
keyin_proses1:
    Mov      A,#0
    Call     PPI_Send_Port_C
    Mov      A,keydat1
    Cjne     A,keydat2,keyin101a
    Mov      DPH,Keypad_dphh
    Mov      DPL,Keypad_dp11
    Inc      DPTR
    Clr      A
    Movc     A,@A+DPTR
    Cjne     A,#C,keyin_200
    Mov      Keypad_dphh,Keypad_dphh1
    Mov      Keypad_dp11,Keypad_dp111
    Mov      R0,keypad_curpos
    Call     LCD_Cursor_Position

```

```

    Mov     DPH,Keypad_dphh
    Mov     DPL,Keypad_dp11
    Clr     A
    Movc    A,@A+DPTR
    Call    LCD_Data
    Call    LCD_Blink_On
    Mov     R0,keydelay
    Call    Delay_Var_10ms
    Jmp     keyin100
keyin_200:
    Mov     Keypad_dphh,DPH
    Mov     Keypad_dp11,DPL
    Mov     B,A
    Mov     R0,keypad_curpos
    Call    LCD_Cursor_Position
    Mov     A,B
    Call    LCD_Data
    Call    LCD_Blink_On
    Mov     R0,keydelay
    Call    Delay_Var_10ms
    Jmp     keyin100

;-----
keyin_end:

    Call    LCD_Blink_On
    Pop     DPL
    Pop     DPH
    Ret

EndSub
Subroutine EEPROM_Write:
    Push    01
    Mov     01,A
save_03:
    Movx    @DPTR,A
save_01:
    Movx    A,@DPTR
    Cjne    A,01h,save_01
    Pop     01
EndSub
Subroutine EEPROM_Read:
    Movx    A,@DPTR
EndSub

;----- DELAY -----
Subroutine Delay_Var_1ms:
    Call    Delay_Fix_1ms
    Djnz    R0,Delay_Var_1ms
EndSub
Subroutine Delay_Var_10ms:
    Call    Delay_Fix_10ms
    Djnz    R0,Delay_Var_10ms
EndSub
Subroutine Delay_Var_100ms:
    Call    Delay_Fix_100ms

```

```

        Djnz    R0,Delay_Var_100ms
EndSub
Subroutine Delay_Var_1s:
    Call    Delay_Fix_1s
    Djnz    R0,Delay_Var_1s
EndSub
Subroutine Delay_Var_10s:
    Call    Delay_Fix_10s
    Djnz    R0,Delay_Var_10s
EndSub
Subroutine Delay_Var_10us:
    Call    Delay_Fix_10us
    Djnz    R0,Delay_Var_10us
EndSub
Subroutine Delay_Fix_10us:
    Push    1
    Mov     1,#2J
    Djnz    1,$
    Pop     1
EndSub
Subroutine Delay_Fix_10s:
    Push    1
    Mov     1,#100
delay_fix_10s_1:
    Call    Delay_Fix_100ms
    Djnz    1,delay_fix_10s_1
    Pop     1
EndSub
Subroutine Delay_Fix_1s:
    Push    1
    Mov     1,#100
delay_fix_1000ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_1000ms_1
    Pop     1
EndSub
Subroutine Delay_Fix_100ms:
    Push    1
    Mov     1,#10
delay_fix_100ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_100ms_1
    Pop     1
EndSub
Subroutine Delay_Fix_10ms:
    Mov     TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
    Mov     TLO,#3Dh    ; siapkan waktu tunda 50 mili-detik
    Mov     TH0,#0B0h
    Clr     TF0    ; me-nol-kan bit limpahan
    Setb    TR0    ; timer mulai bekerja
    Jnb     TF0,$    ; tunggu di sini sampai melimpah
    Clr     TR0    ; timer berhenti kerja
    Ret
EndSub
Subroutine Delay_Fix_1ms:
    Mov     TMOD,#00000001b    ; Timer 1 bekerja pada mode 1

```

```

Mov     TLO,#0EDh    ; siapkan waktu tunda 50 mili-detik
Mov     TH0,#78h
Clr     TFO           ; me-nol-kan bit limpahan
Setb    TRO           ; timer mulai bekerja
Jnb     TFO,$         ; tunggu di sini sampai melimpah
Clr     TRO           ; timer berhenti kerja
Ret

EndSub

;
;0123456789ABCDEF
txt_menu1: Db '1.Set Tebal',0
txt_menu2: Db '2.Jalankan',0
txt_menu11: Db '1.1mm 2.2mm',0
txt_menu21: Db 'Conveyer Jalan',0
txt_menu22: Db 'Ketebalan:',0
txt_Iris: Db 'Piringan Jalan',0

```

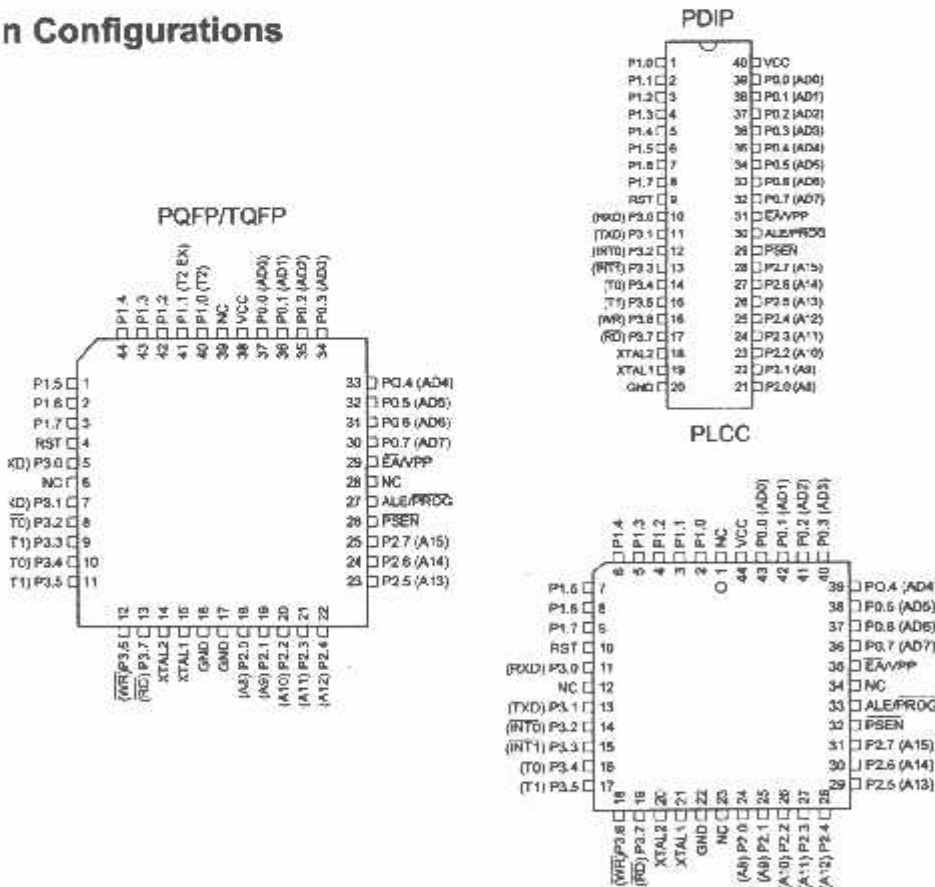
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations



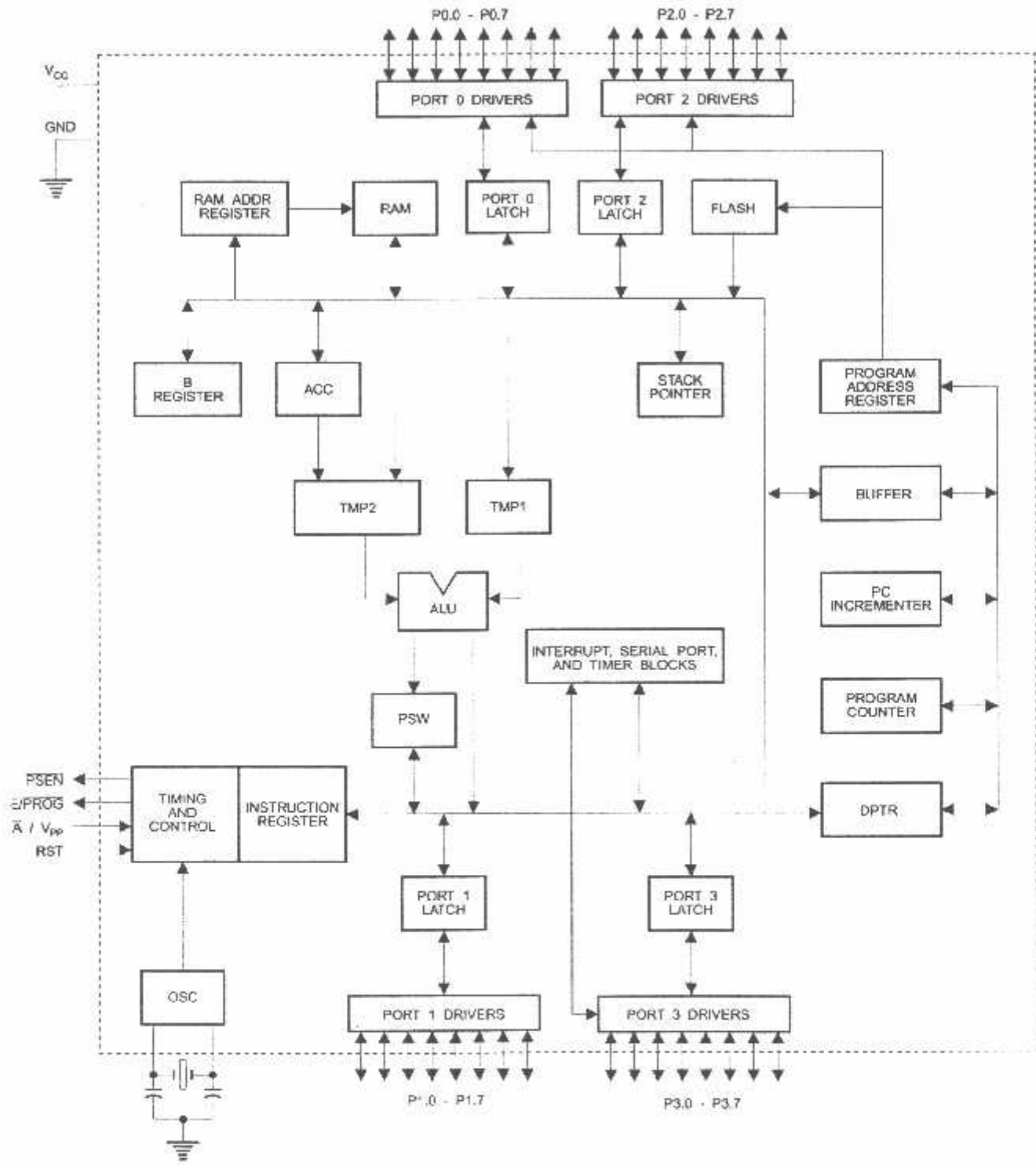
8-bit
Microcontroller
with 4K Bytes
Flash

AT89C51

Not Recommended
for New Designs.
Use AT89S51.



Block Diagram



The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

P0
P0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to P0 pins, the pins can be used as high-impedance inputs.

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

P0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

P1
P1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by internal pullups and can be used as inputs. As inputs, P1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

P1 also receives the low-order address bytes during Flash programming and verification.

P2
P2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by internal pullups and can be used as inputs. As inputs,

Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

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

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

Port 3

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

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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

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

RST

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

ALE/PROG

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

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

else is skipped during each access to external Data memory.

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

PSEN

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

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP

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

EA should be strapped to VCC for internal program executions.

This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming, for parts that require -volt VPP.

AL1

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

AL2

Input from the inverting oscillator amplifier.

Oscillator Characteristics

AL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left

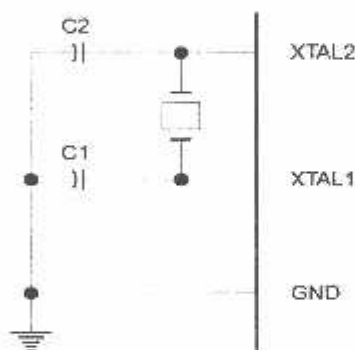
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

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

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections

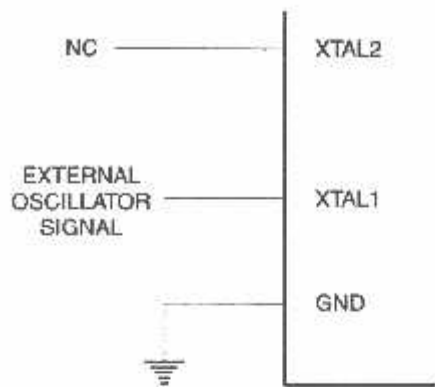


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

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

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers

retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

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



Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{PP}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective p-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) = FFH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-bank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

Input the desired memory location on the address lines.

Input the appropriate data byte on the data lines.

Activate the correct combination of control signals.

Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.

Pulse $\overline{ALE}/\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address

and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on P0.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/ $\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding $\overline{ALE}/\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed 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 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel

(031H) = 51H indicates 89C51

(032H) = FFH indicates 12V programming

(032H) = 05H indicates 5V programming

Programming Interface

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

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

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	 (1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

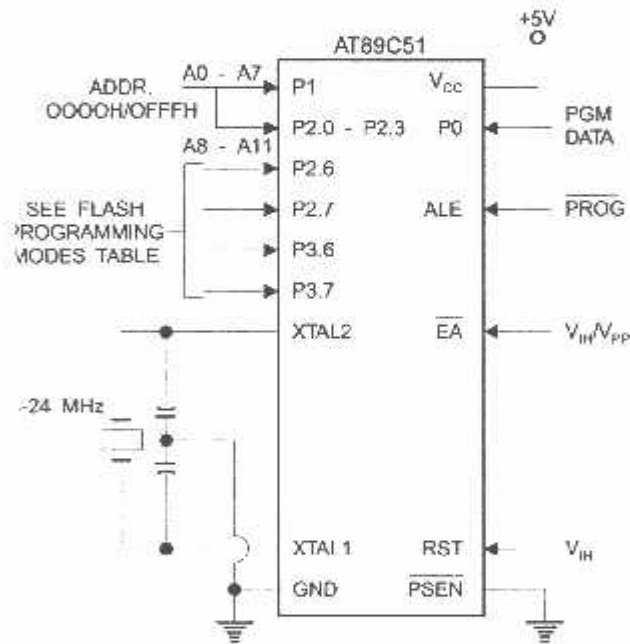
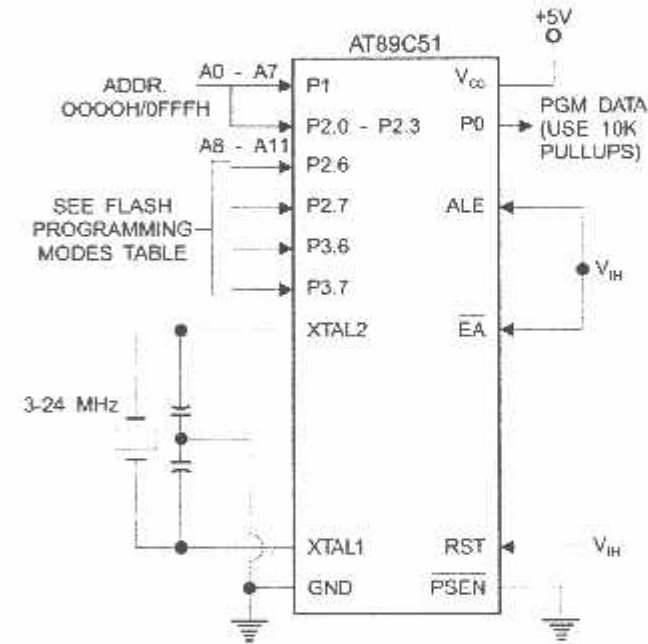
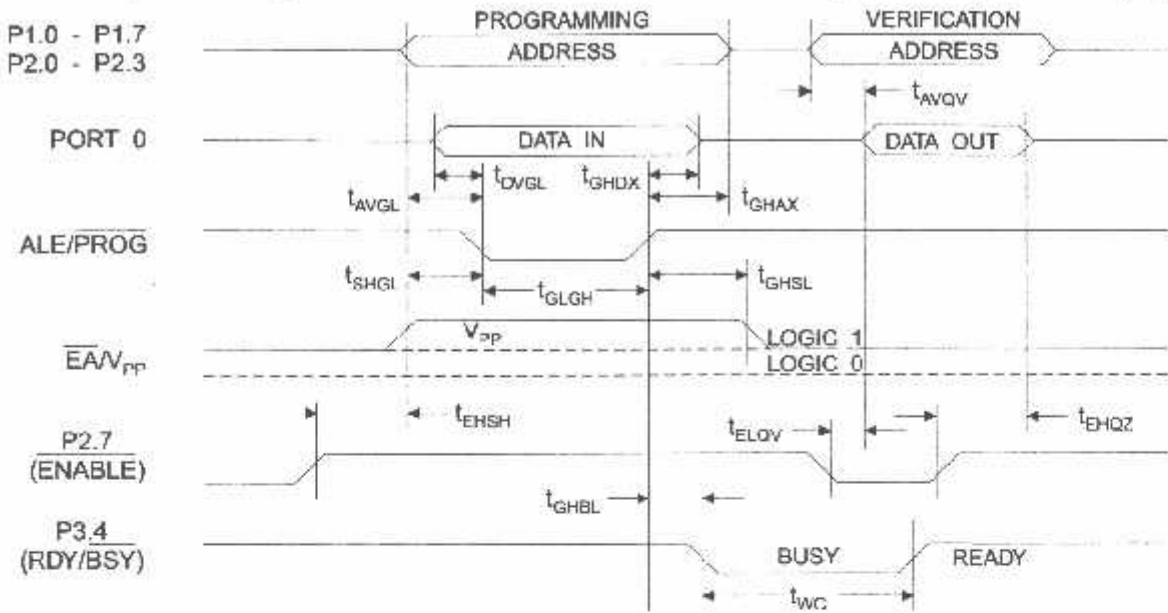


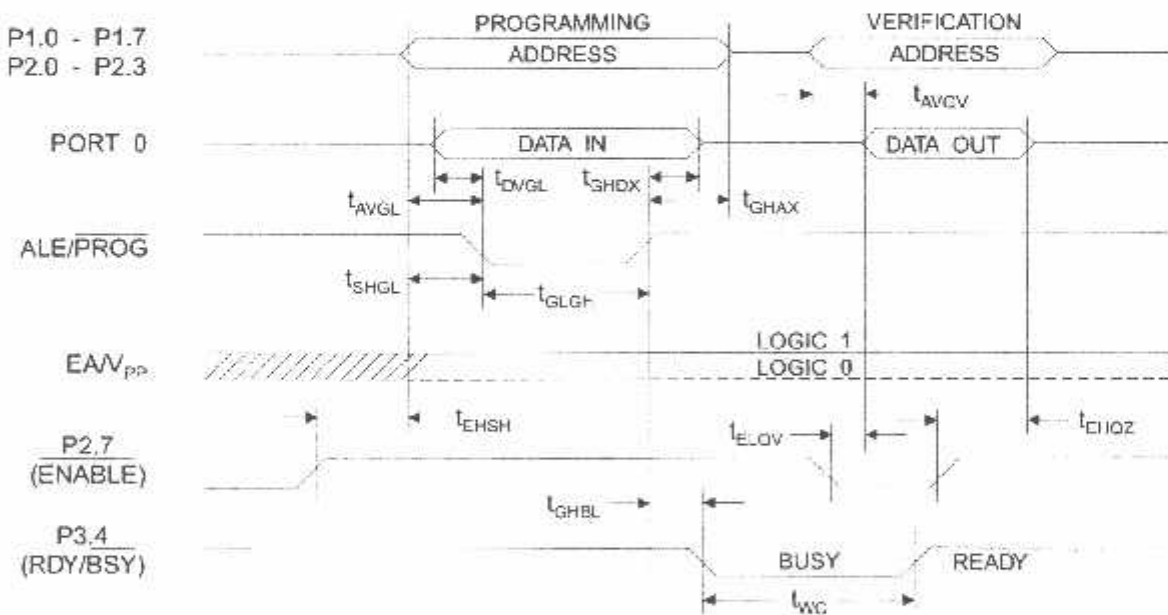
Figure 4. Verifying the Flash



Flash Programming and Verification Waveforms - High-voltage Mode ($V_{pp} = 12V$)



Flash Programming and Verification Waveforms - Low-voltage Mode ($V_{pp} = 5V$)



Flash Programming and Verification Characteristics

$T = 0^{\circ}\text{C to } 70^{\circ}\text{C}, V_{\text{CC}} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{\text{PP}}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{\text{PP}}^{(1)}$	Programming Enable Current		1.0	mA
f_{CLCL}	Oscillator Frequency	3	24	MHz
t_{WGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{\text{CLCL}}$		
t_{HAX}	Address Hold after $\overline{\text{PROG}}$	$48t_{\text{CLCL}}$		
t_{WGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{\text{CLCL}}$		
t_{HDX}	Data Hold after $\overline{\text{PROG}}$	$48t_{\text{CLCL}}$		
t_{HSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{\text{CLCL}}$		
t_{HGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
$t_{\text{HSL}}^{(1)}$	V_{PP} Hold after $\overline{\text{PROG}}$	10		μs
t_{LGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{QV}	Address to Data Valid		$48t_{\text{CLCL}}$	
t_{QV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{\text{CLCL}}$	
t_{QZ}	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{\text{CLCL}}$	
t_{HBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{C}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.



Absolute Maximum Ratings*

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

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

DC Characteristics

T = -40°C to 85°C, VCC = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
IL	Input Low-voltage	(Except EA)	-0.5	0.2 VCC - 0.1	V
LI	Input Low-voltage (EA)		-0.5	0.2 VCC - 0.3	V
HI	Input High-voltage	(Except XTAL1, RST)	0.2 VCC + 0.9	VCC + 0.5	V
HI1	Input High-voltage	(XTAL1, RST)	0.7 VCC	VCC + 0.5	V
OL	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	IOL = 1.8 mA		0.45	V
OL1	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	IOL = 3.2 mA		0.45	V
OH	Output High-voltage (Ports 1,2,3, ALE, PSEN)	I _{OH} = -60 µA, VCC = 5V ± 10%	2.4		V
		I _{OH} = -25 µA	0.75 VCC		V
		I _{OH} = -10 µA	0.9 VCC		V
OH1	Output High-voltage (Port 0 in External Bus Mode)	I _{OH} = -800 µA, VCC = 5V ± 10%	2.4		V
		I _{OH} = -300 µA	0.75 VCC		V
		I _{OH} = -80 µA	0.9 VCC		V
	Logical 0 Input Current (Ports 1,2,3)	VIN = 0.45V		-50	µA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	VIN = 2V, VCC = 5V ± 10%		-850	µA
	Input Leakage Current (Port 0, EA)	0.45 < VIN < VCC		±10	µA
RST	Reset Pull-down Resistor		50	300	KΩ
CP	Pin Capacitance	Test Freq. = 1 MHz, TA = 25°C		10	pF
PS	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode ⁽²⁾	VCC = 6V		100	µA
		VCC = 3V		40	µA

- Notes: 1. Under steady state (non-transient) conditions, IOL must be externally limited as follows:
Maximum IOL per port pin: 10 mA
Maximum IOL per 8-bit port: Port 0: 26 mA
Ports 1, 2, 3: 15 mA
Maximum total IOL for all output pins: 71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power-down is 2V.

C Characteristics

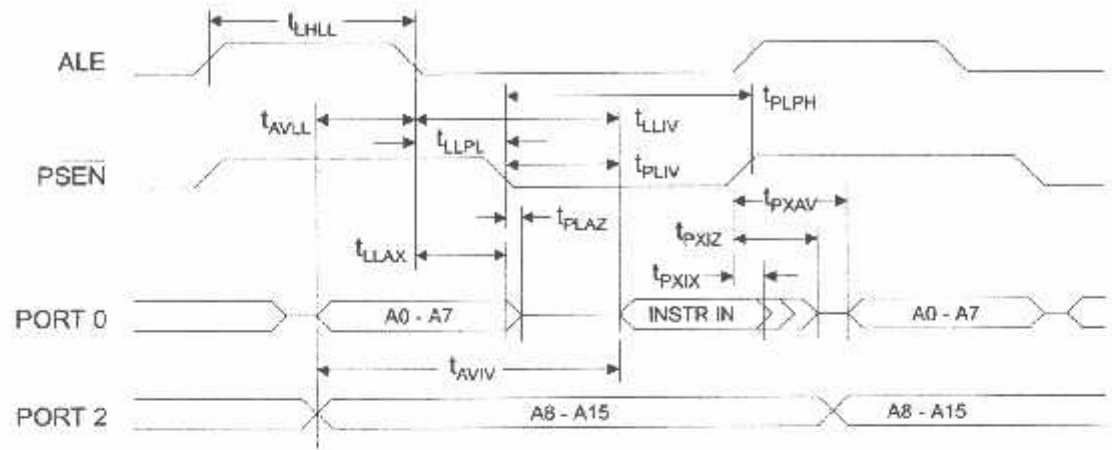
Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other inputs = 80 pF.

External Program and Data Memory Characteristics

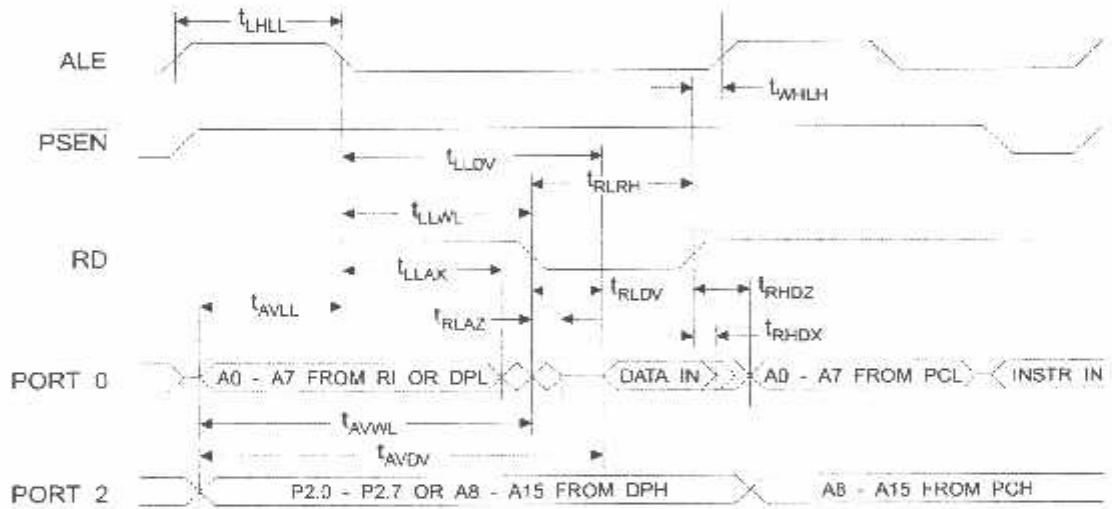
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
f_{CLCL}	Oscillator Frequency			0	24	MHz
t_{HLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{VLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LAX}	Address Hold after ALE Low	48		$t_{CLCL}-20$		ns
t_{LV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LPL}	ALE Low to PSEN Low	43		$t_{CLCL}-13$		ns
t_{LPH}	PSEN Pulse Width	205		$3t_{CLCL}-20$		ns
t_{LV}	PSEN Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{IDX}	Input Instruction Hold after PSEN	0		0		ns
t_{IZ}	Input Instruction Float after PSEN		59		$t_{CLCL}-10$	ns
t_{XAV}	PSEN to Address Valid	75		$t_{CLCL}-8$		ns
t_{VIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{LAZ}	PSEN Low to Address Float		10		10	ns
t_{LRH}	RD Pulse Width	400		$6t_{CLCL}-100$		ns
t_{LWH}	WR Pulse Width	400		$6t_{CLCL}-100$		ns
t_{LDV}	RD Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{HDX}	Data Hold after RD	0		0		ns
t_{DZ}	Data Float after RD		97		$2t_{CLCL}-28$	ns
t_{DV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
$t_{/DV}$	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{WL}	ALE Low to RD or WR Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{/WL}$	Address to RD or WR Low	203		$4t_{CLCL}-75$		ns
$t_{/WX}$	Data Valid to WR Transition	23		$t_{CLCL}-20$		ns
$t_{/VH}$	Data Valid to WR High	433		$7t_{CLCL}-120$		ns
t_{HQX}	Data Hold after WR	33		$t_{CLCL}-20$		ns
t_{AZ}	RD Low to Address Float		0		0	ns
t_{LH}	RD or WR High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns



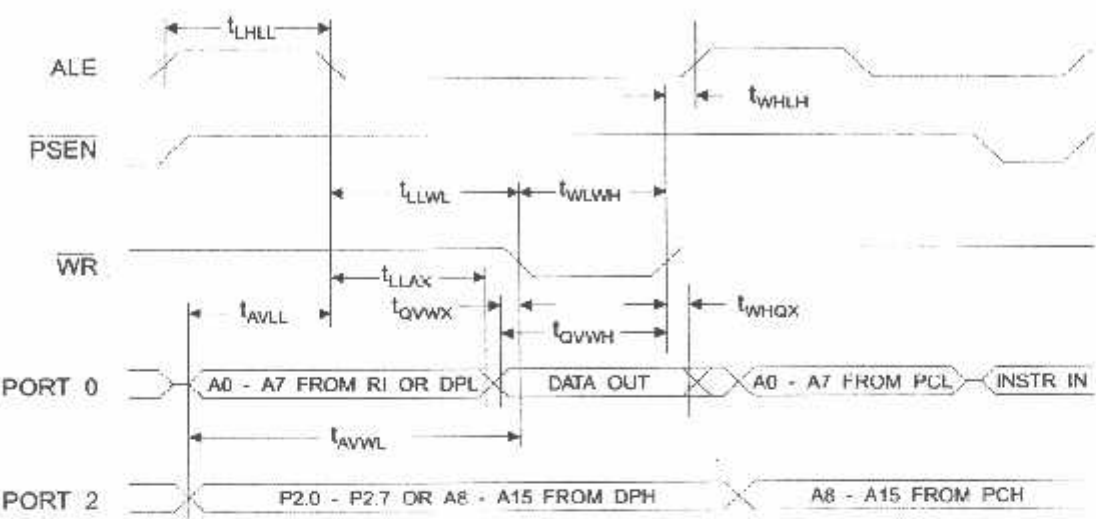
External Program Memory Read Cycle



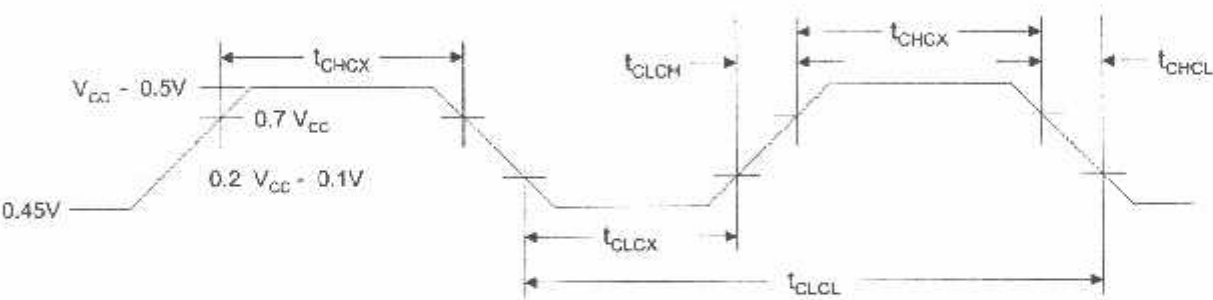
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
f_{CLCL}	Oscillator Frequency	0	24	MHz
T_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

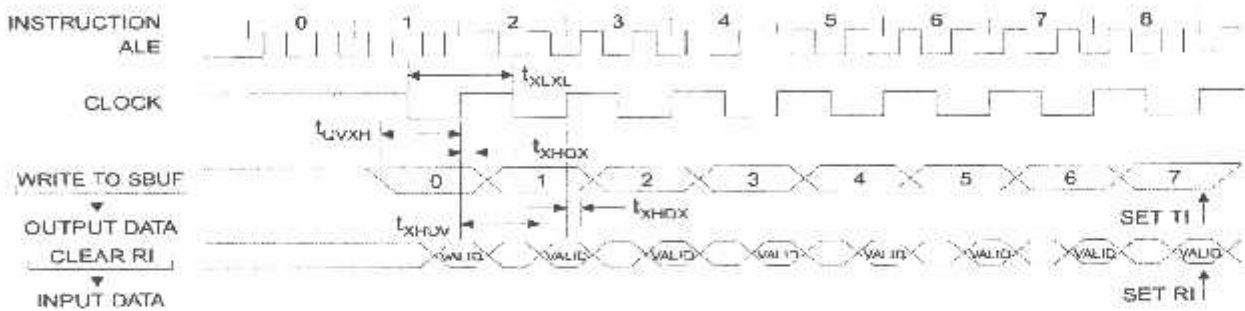


Serial Port Timing: Shift Register Mode Test Conditions

V_{CC} = 5.0 V ±20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t _{XLXL}	Serial Port Clock Cycle Time	1.0		12t _{CLCL}		μs
t _{UVXH}	Output Data Setup to Clock Rising Edge	700		10t _{CLCL} -133		ns
t _{XHQX}	Output Data Hold after Clock Rising Edge	50		2t _{CLCL} -117		ns
t _{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t _{XHDV}	Clock Rising Edge to Input Data Valid		700		10t _{CLCL} -133	ns

Shift Register Mode Timing Waveforms



C Testing Input/Output Waveforms⁽¹⁾



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

Float Waveforms⁽¹⁾



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

Ordering Information

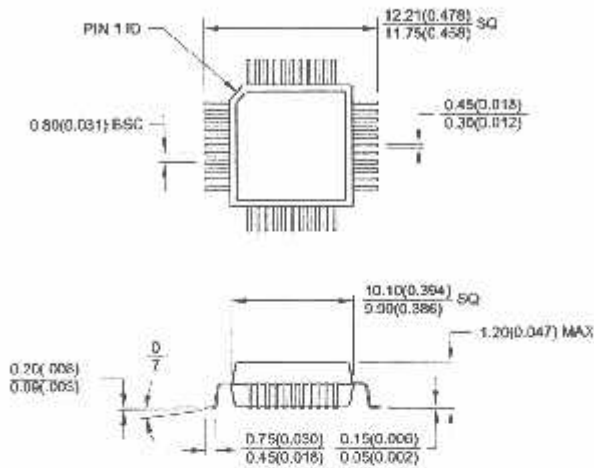
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0° C to 70° C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40° C to 85° C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0° C to 70° C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40° C to 85° C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0° C to 70° C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40° C to 85° C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0° C to 70° C)
		AT89C51-24JC	44J	
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40° C to 85° C)
		AT89C51-24JI	44J	
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

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



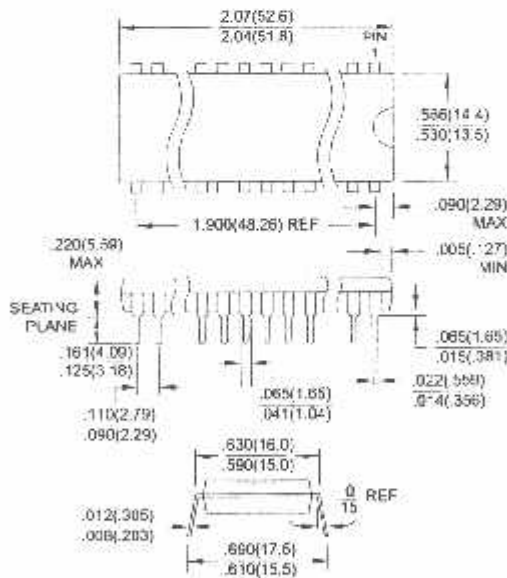
ackaging Information

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

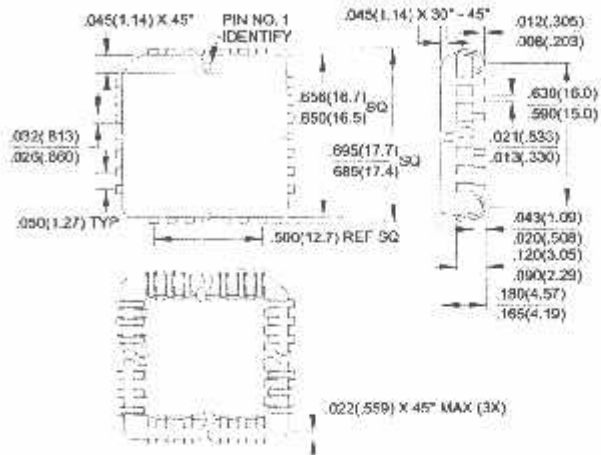


Controlling dimension: millimeters

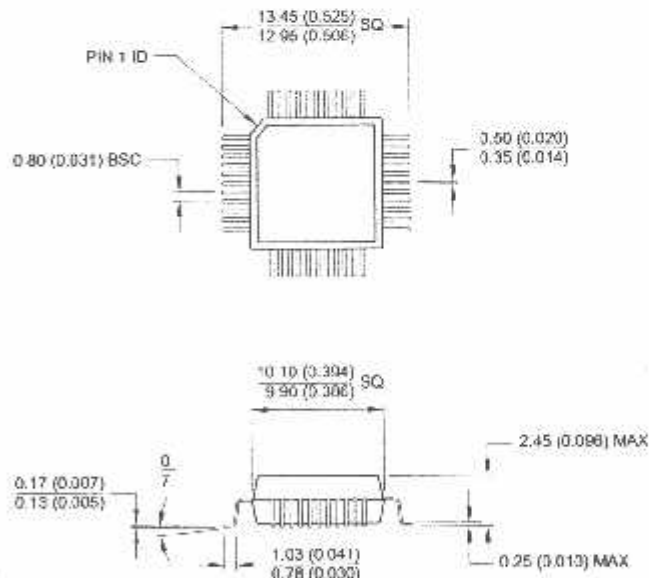
40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



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



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



Controlling dimension: millimeters



Atmel Headquarters

Corporate Headquarters

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

Europe

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

Asia

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

Japan

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

Atmel Operations

Atmel Colorado Springs

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

Atmel Rousset

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

Fax-on-Demand

North America:

1-(800) 292-8635

International:

1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are authorized for use as critical components in life support devices or systems.

Logos bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Names and product names in this document may be trademarks of others.



Printed on recycled paper.

0265G-02/00/xM

August 1996

Features

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

Ordering Information

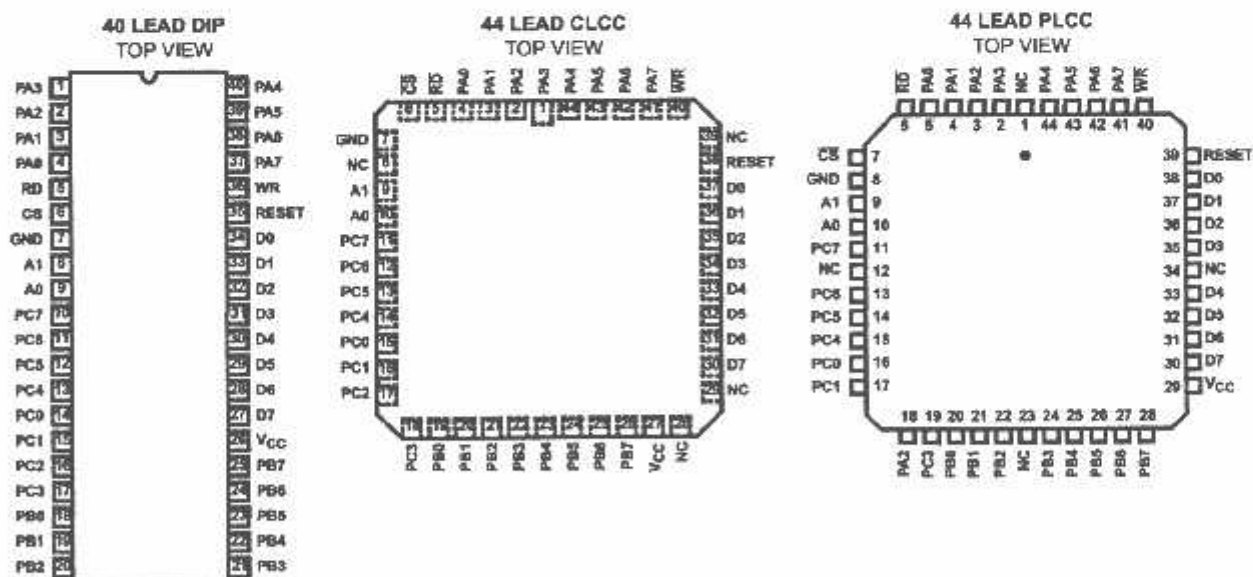
PACKAGE	TEMPERATURE RANGE	5MHz	8MHz	PKG. NO.
Plastic DIP	0°C to +70°C	CP82C55A-5	CP82C55A	E40.8
	-40°C to +85°C	IP82C55A-5	IP82C55A	E40.6
PLCC	0°C to +70°C	CS82C55A-5	CS82C55A	N44.65
	-40°C to +85°C	IS82C55A-5	IS82C55A	N44.65
CERDIP	0°C to +70°C	CD82C55A-5	CD82C55A	F40.6
	-40°C to +85°C	ID82C55A-5	ID82C55A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

Description

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

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

Pinouts



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

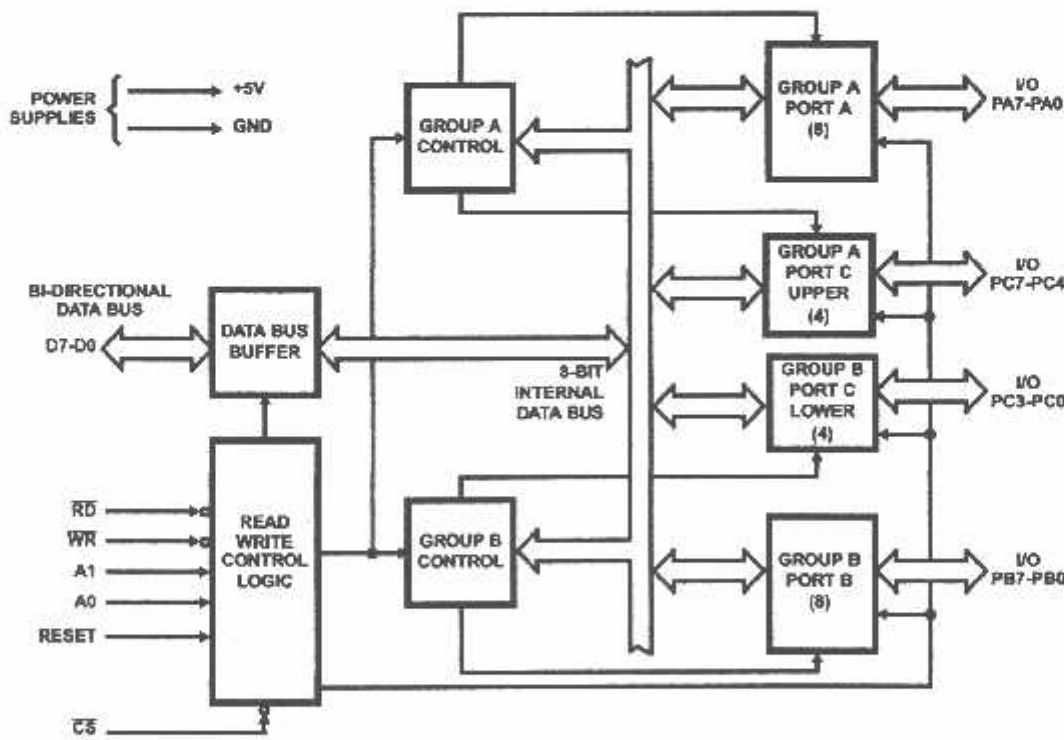
Copyright © Harris Corporation 1996

File Number: 2969.1

Pin Description

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

Functional Description



Functional Description

Data Bus Buffer

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

Read/Write and Control Logic

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

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

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

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

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

82C55A BASIC OPERATION

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

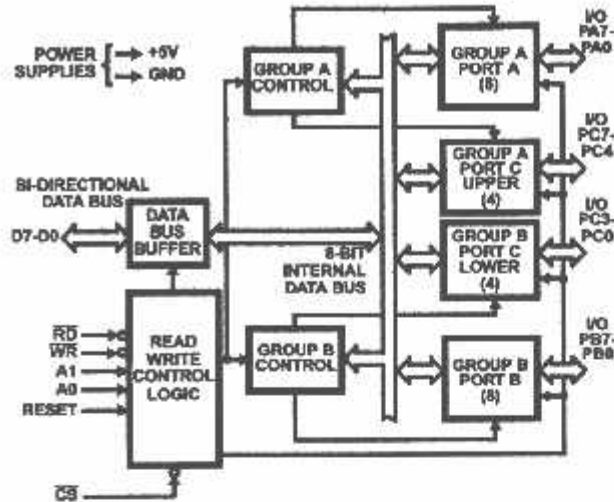


FIGURE 1. 82C55A BLOCK DIAGRAM. DATA BUS BUFFER, READ/WRITE, GROUP A & B CONTROL LOGIC FUNCTIONS

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

Group A and Group B Controls

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

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

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

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

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

Ports A, B, and C

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

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

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

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

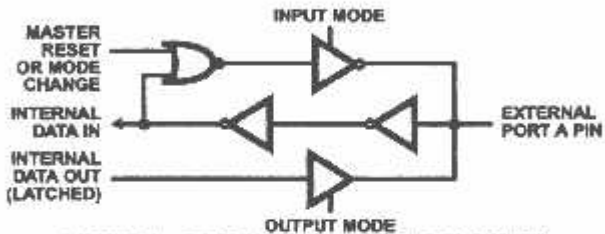


FIGURE 2A. PORT A BUS-HOLD CONFIGURATION

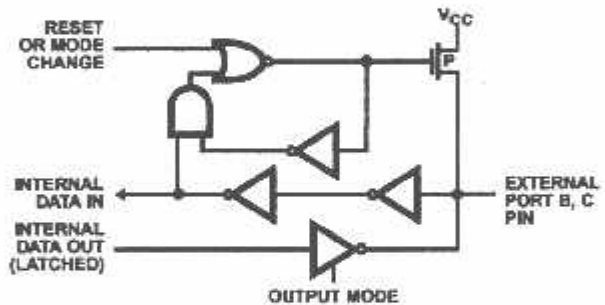


FIGURE 2B. PORT B AND C BUS-HOLD CONFIGURATION
FIGURE 2. BUS-HOLD CONFIGURATION

Operational Description

Mode Selection

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

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

When the reset input goes "high", all ports will be set to the input mode with all 24 port lines held at a logic "one" level by internal bus hold devices. After the reset is removed, the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need to pullup or pull-down resistors in all-CMOS designs. The control word register will contain 9Bh. During the execution of the system

program, any of the other modes may be selected using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine. Any port programmed as an output port is initialized to all zeros when the control word is written.

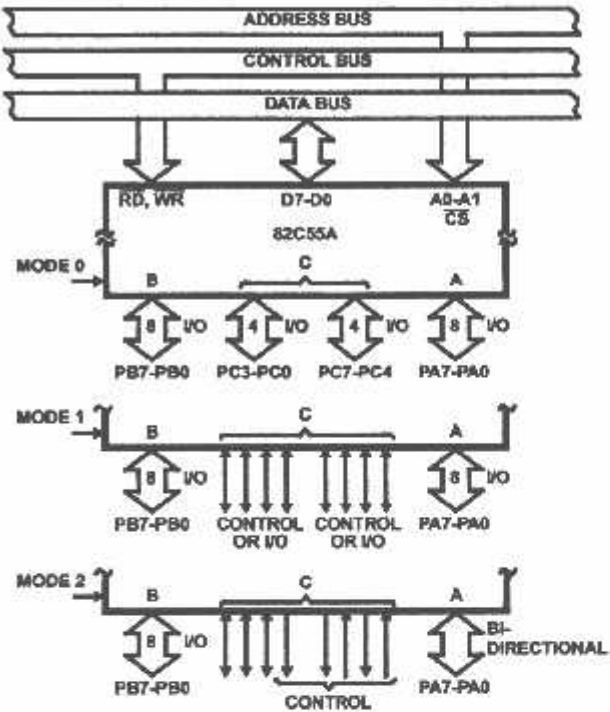


FIGURE 3. BASIC MODE DEFINITIONS AND BUS INTERFACE

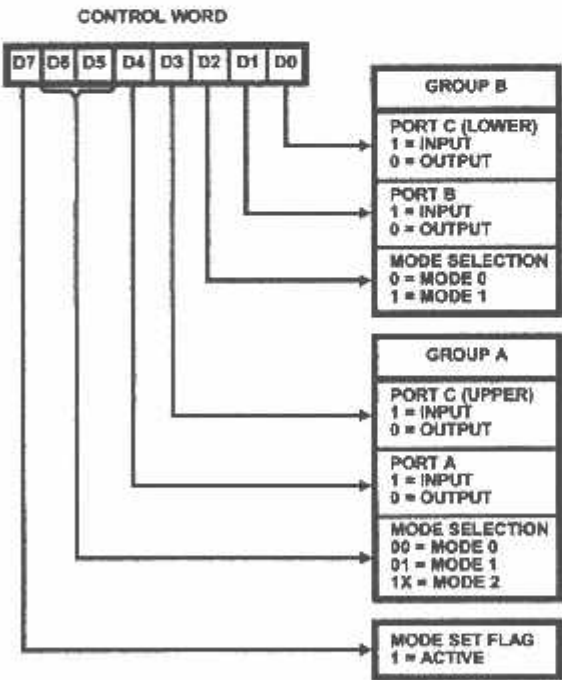


FIGURE 4. MODE DEFINITION FORMAT

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

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

Single Bit Set/Reset Feature (Figure 5)

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

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

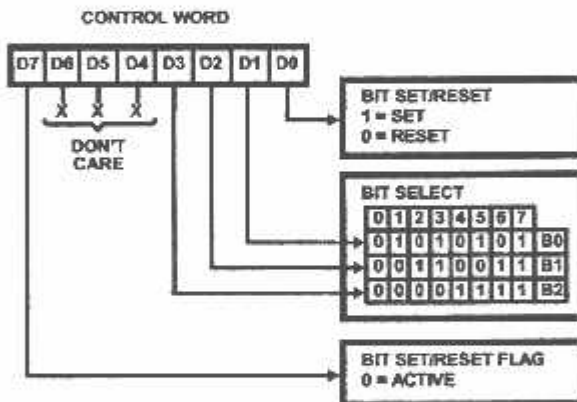


FIGURE 5. BIT SET/RESET FORMAT

Interrupt Control Functions

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

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

INTE Flip-Flop Definition

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

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

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

Operating Modes

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

Mode 0 Basic Functional Definitions:

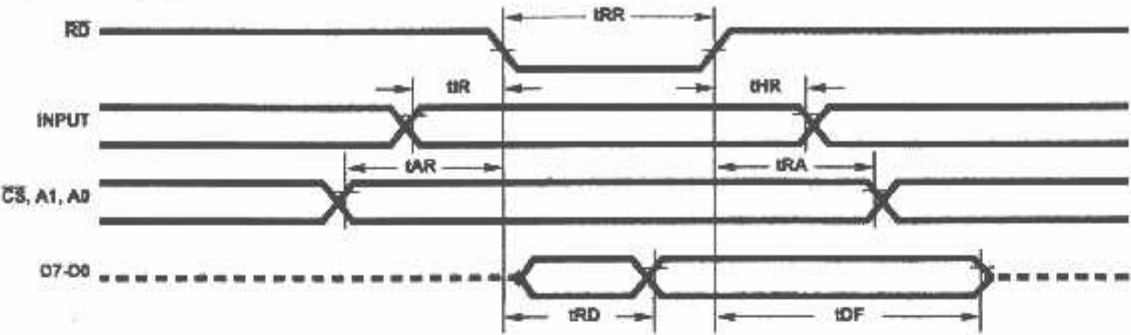
- Two 8-bit ports and two 4-bit ports
- Any Port can be input or output
- Outputs are latched
- Input are not latched
- 16 different Input/Output configurations possible

MODE 0 PORT DEFINITION

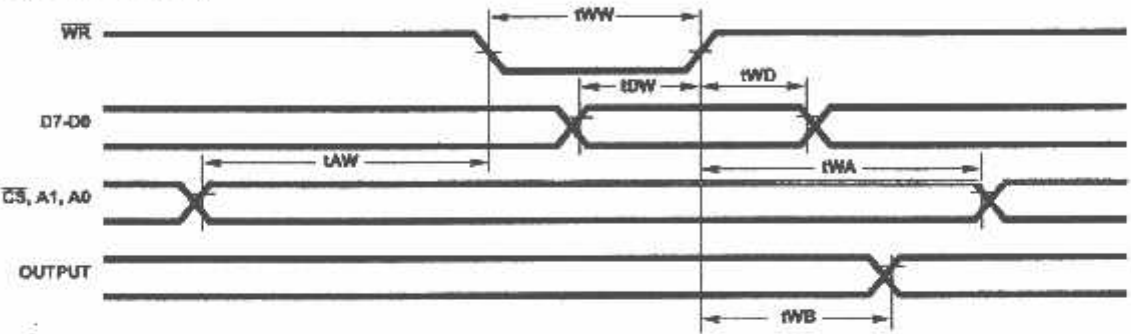
A		B		GROUP A		#	GROUP B	
D4	D3	D1	D0	PORT A	PORT C (Upper)		PORT B (Lower)	PORT C (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output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

82C55A

Mode 0 (Basic Input)



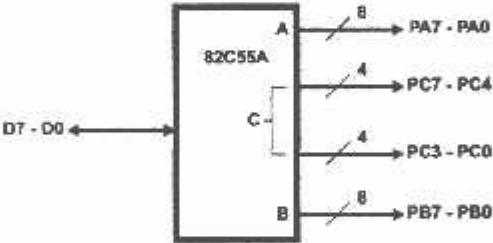
Mode 0 (Basic Output)



Mode 0 Configurations

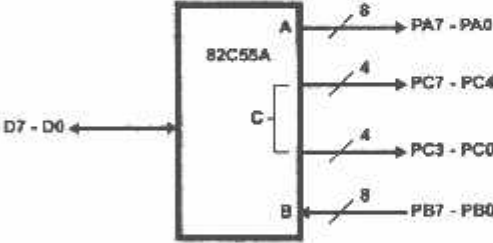
CONTROL WORD #0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0



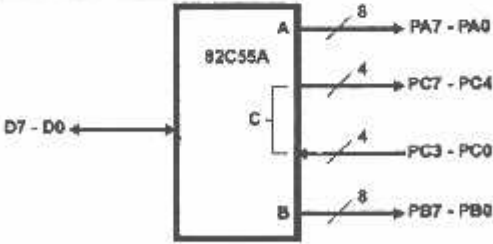
CONTROL WORD #2

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0



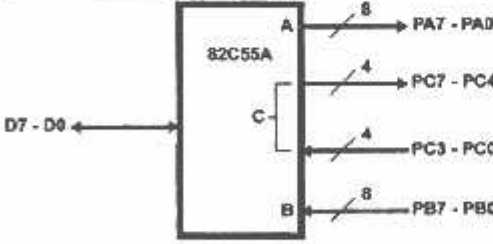
CONTROL WORD #1

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	1



CONTROL WORD #3

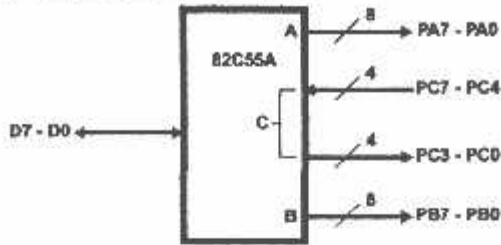
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	1



Mode 0 Configurations (Continued)

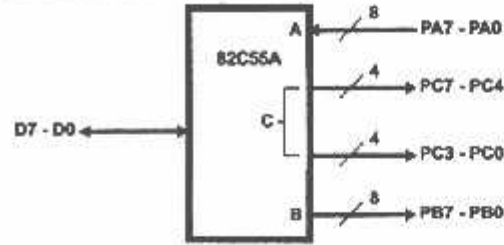
CONTROL WORD #4

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	0



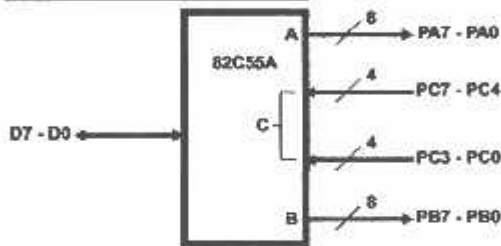
CONTROL WORD #8

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	0



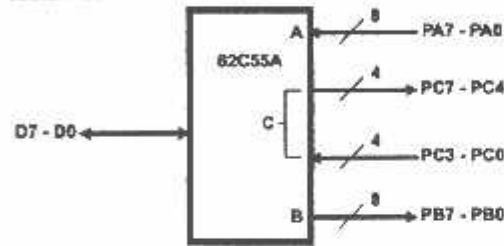
CONTROL WORD #5

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	1



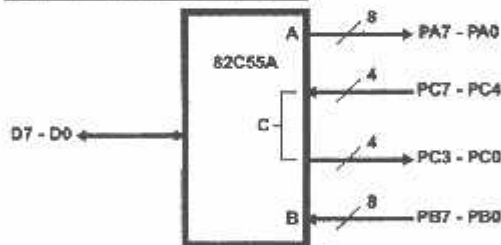
CONTROL WORD #9

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	0	1



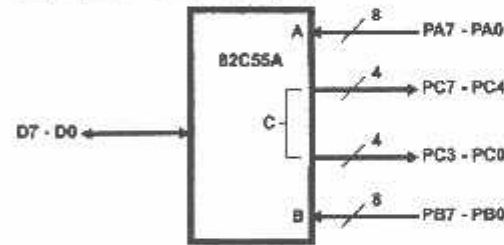
CONTROL WORD #6

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0



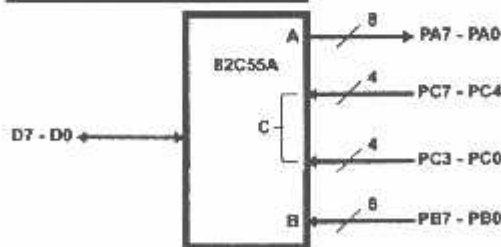
CONTROL WORD #10

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0



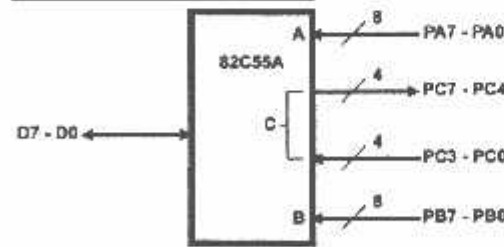
CONTROL WORD #7

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	1



CONTROL WORD #11

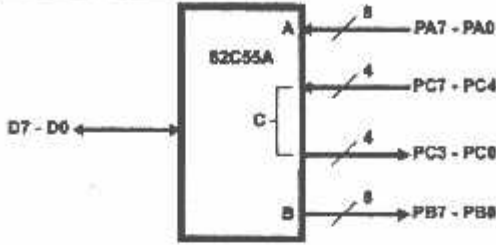
D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1



Mode 0 Configurations (Continued)

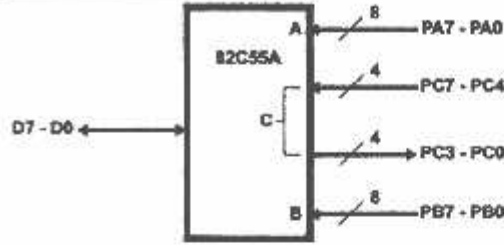
CONTROL WORD #12

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0



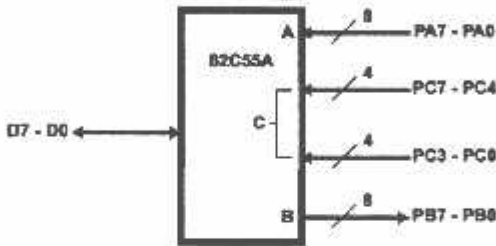
CONTROL WORD #14

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	0



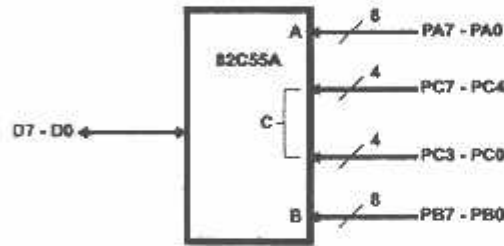
CONTROL WORD #13

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	1



CONTROL WORD #15

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	1



Operating Modes

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

Mode 1 Basic Function Definitions:

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

Input Control Signal Definition

(Figures 6 and 7)

STB (Strobe Input)

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

IBF (Input Buffer Full F/F)

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

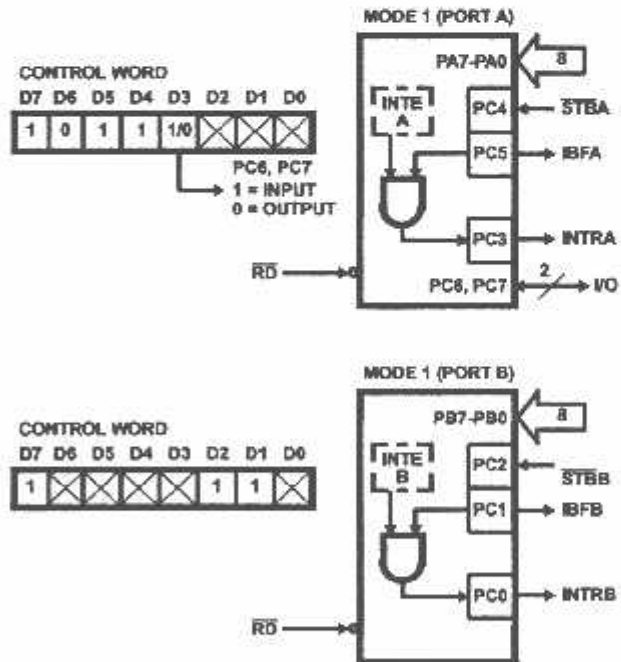


FIGURE 6. MODE 1 INPUT

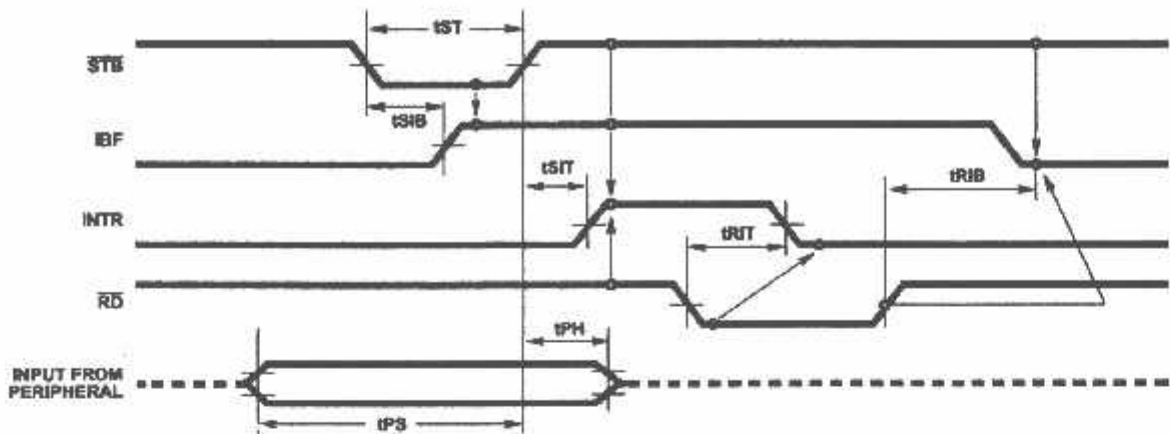


FIGURE 7. MODE 1 (STROBED INPUT)

INTR (Interrupt Request)

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

INTE A

Controlled by bit set/reset of PC4.

INTE B

Controlled by bit set/reset of PC2.

Output Control Signal Definition

(Figure 8 and 9)

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

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

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

INTE A

Controlled by Bit Set/Reset of PC6.

INTE B

Controlled by Bit Set/Reset of PC2.

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

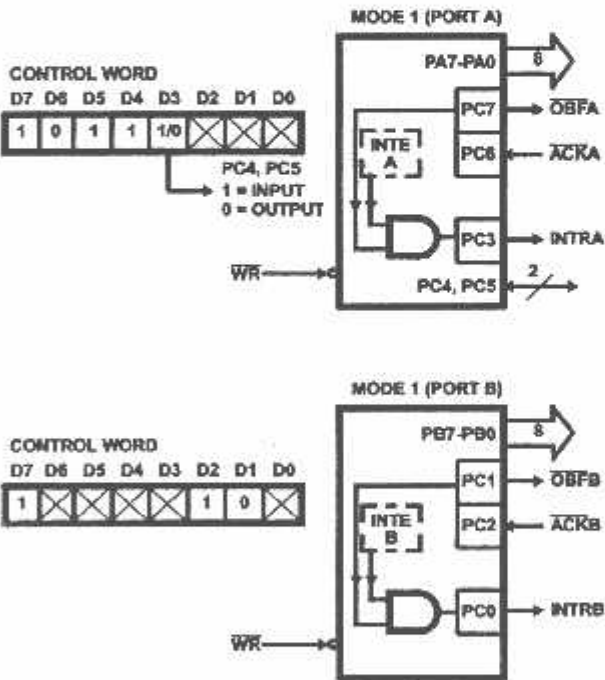


FIGURE 8. MODE 1 OUTPUT

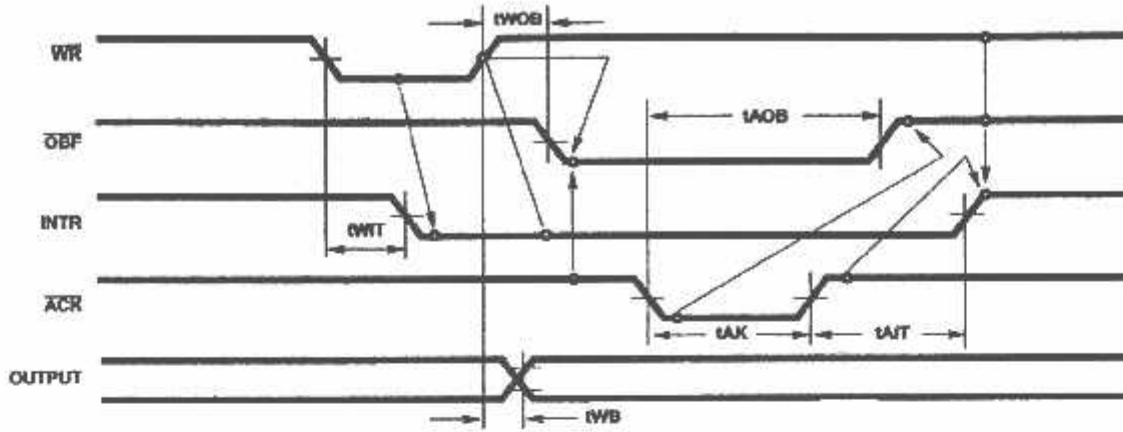
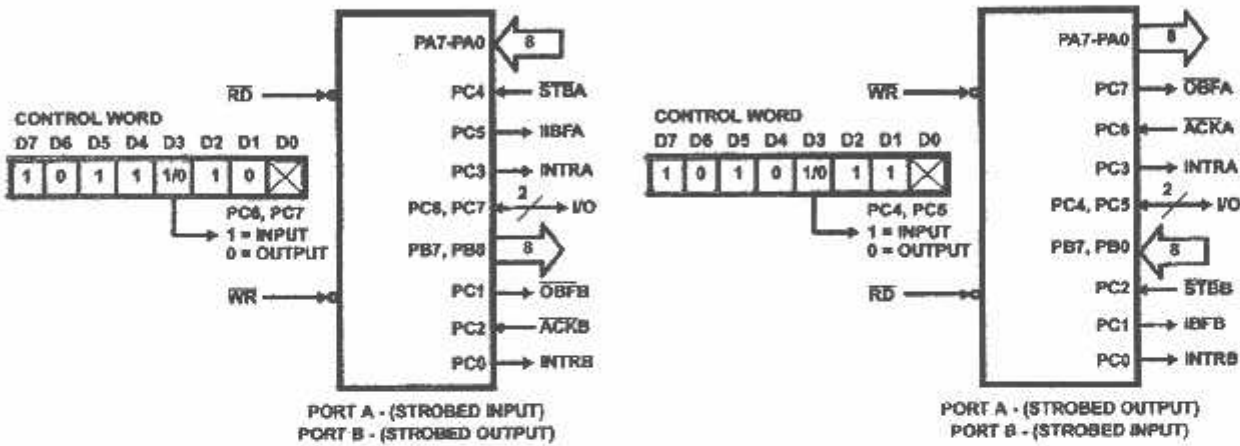


FIGURE 9. MODE 1 (STROBED OUTPUT)



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

FIGURE 10. COMBINATIONS OF MODE 1

Operating Modes

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

The functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both input/output and receiving data (bi-directional bus I/O). "Level-shedding" signals are provided to maintain proper bus flow discipline similar to Mode 1. Interrupt generation and control are similar to Mode 1, and are also available.

Mode 2 Basic Functional Definitions:

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

Bi-Directional Bus I/O Control Signal Definition

(Figures 11, 12, 13, 14)

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

Output Operations

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

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

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

Input Operations

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

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

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

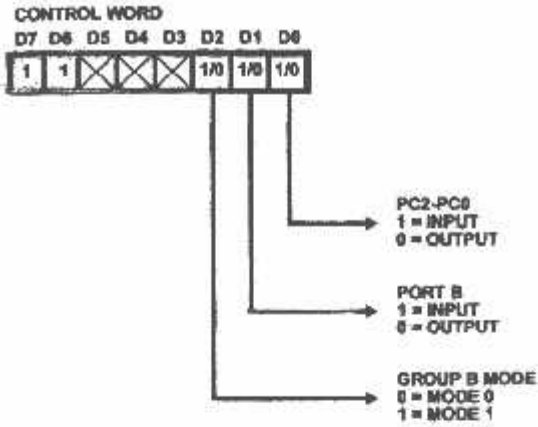


FIGURE 11. MODE CONTROL WORD

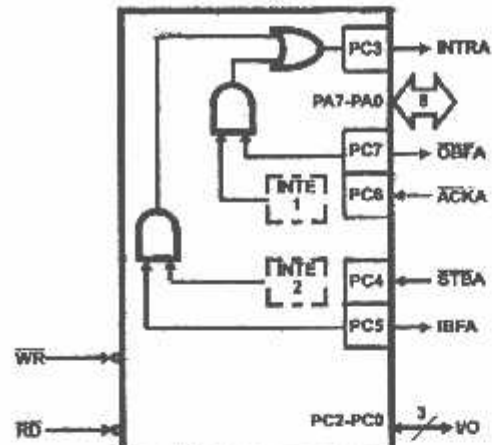
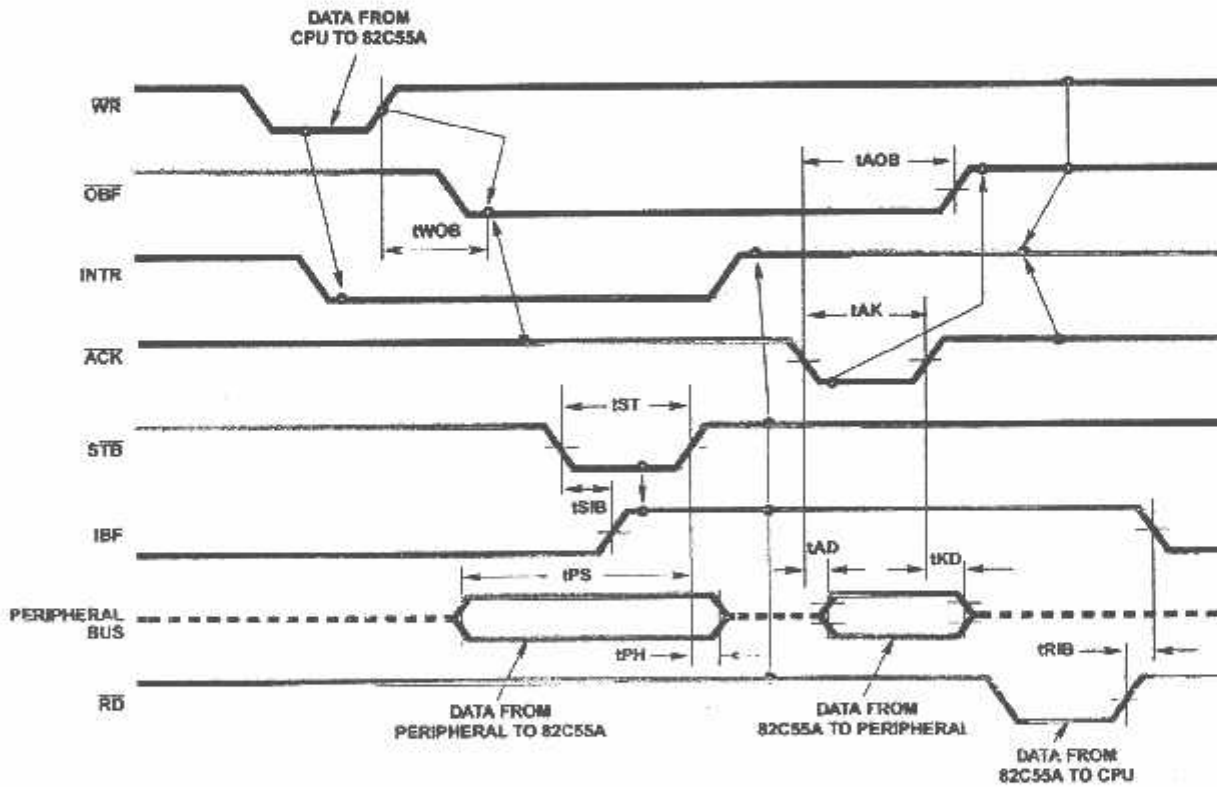


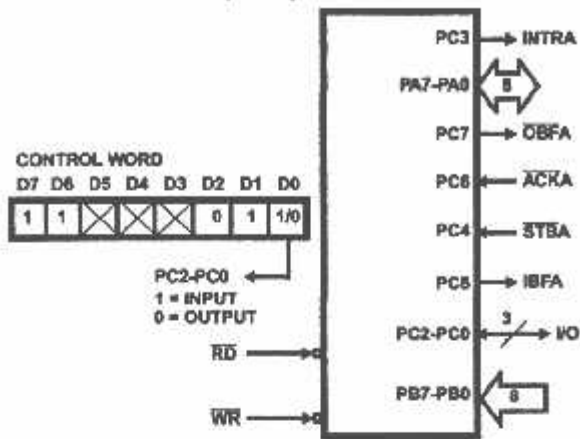
FIGURE 12. MODE 2



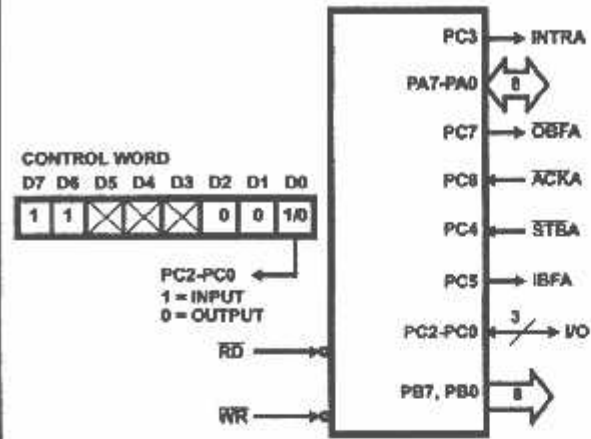
NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible. ($\text{INTR} = \text{IBF} \cdot \text{MASK} \cdot \overline{\text{STB}} \cdot \overline{\text{RD}} + \overline{\text{OBF}} \cdot \text{MASK} \cdot \overline{\text{ACK}}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

FIGURE 13. MODE 2 (BI-DIRECTIONAL)

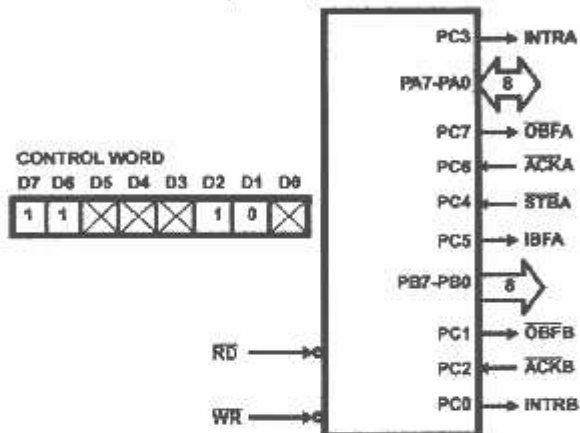
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

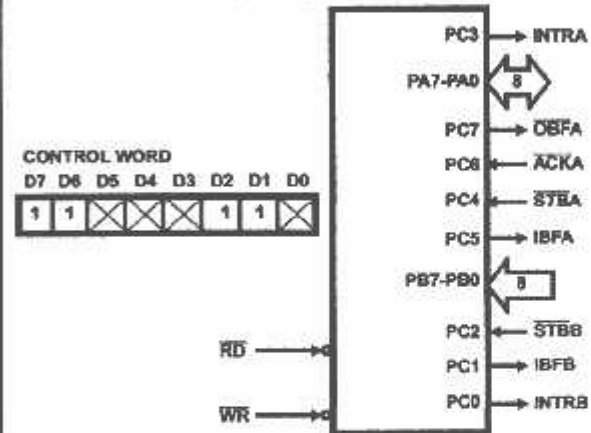


FIGURE 14. MODE 2 COMBINATIONS

MODE DEFINITION SUMMARY

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

Special Mode Combination Considerations

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

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

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

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

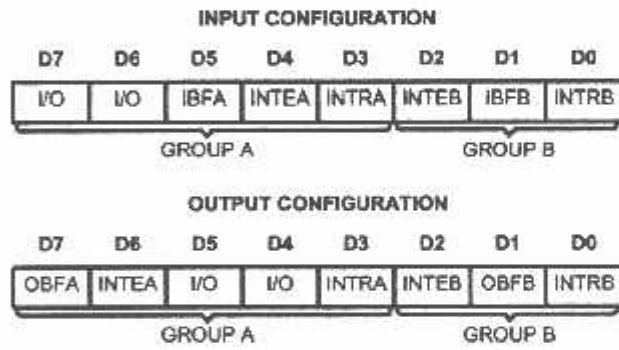


FIGURE 15. MODE 1 STATUS WORD FORMAT

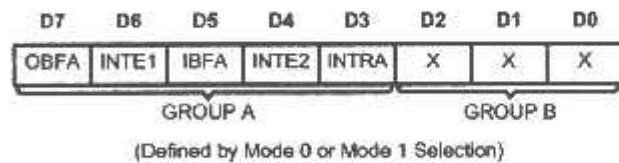


FIGURE 16. MODE 2 STATUS WORD FORMAT

Current Drive Capability

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

Reading Port C Status (Figures 15 and 16)

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

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

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

FIGURE 17. INTERRUPT ENABLE FLAGS IN MODES 1 AND 2

Applications of the 82C55A

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

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

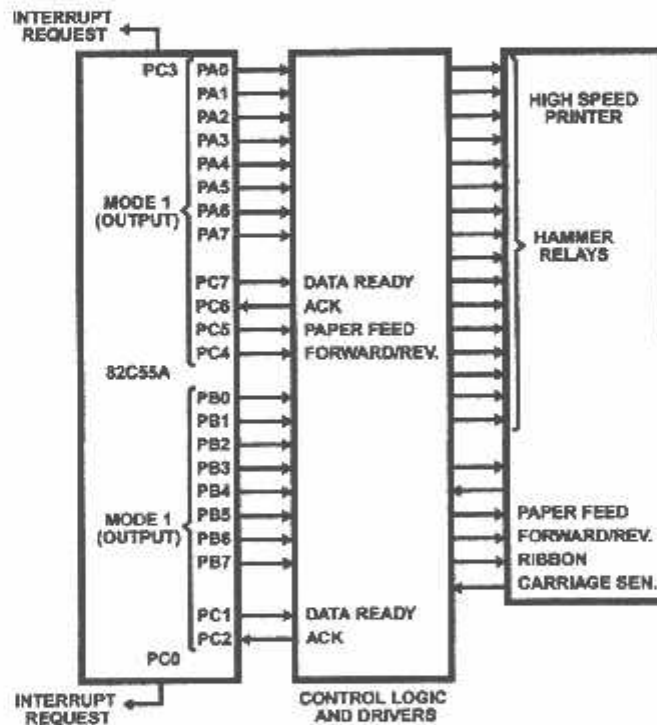


FIGURE 18. PRINTER INTERFACE

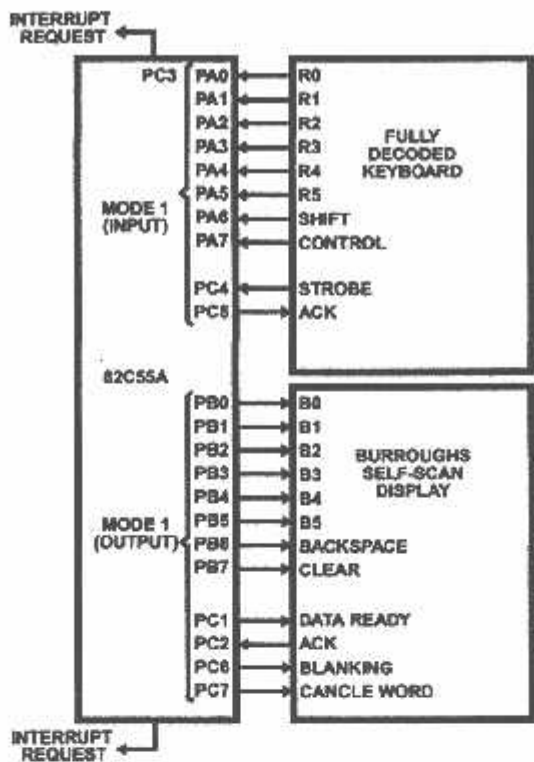


FIGURE 19. KEYBOARD AND DISPLAY INTERFACE

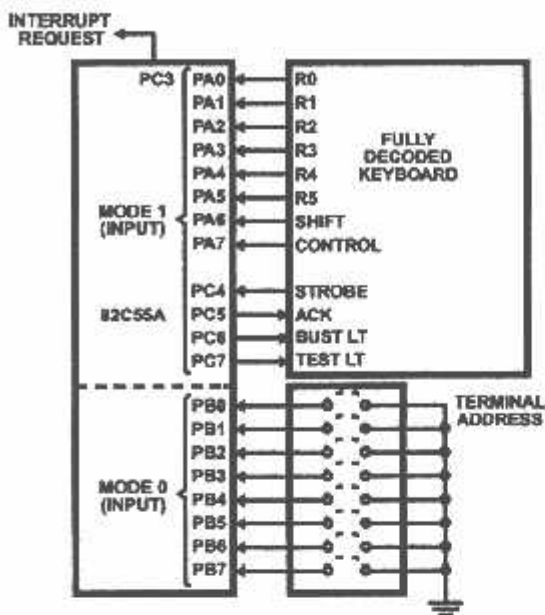


FIGURE 20. KEYBOARD AND TERMINAL ADDRESS INTERFACE

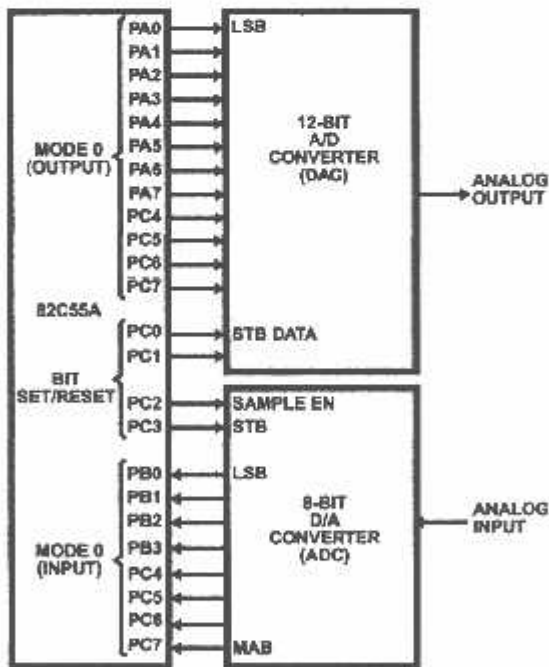


FIGURE 21. DIGITAL TO ANALOG, ANALOG TO DIGITAL

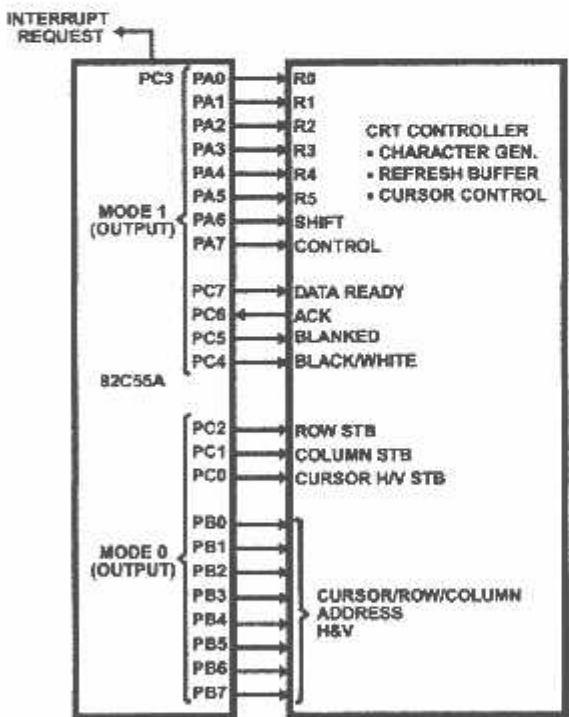


FIGURE 22. BASIC CRT CONTROLLER INTERFACE

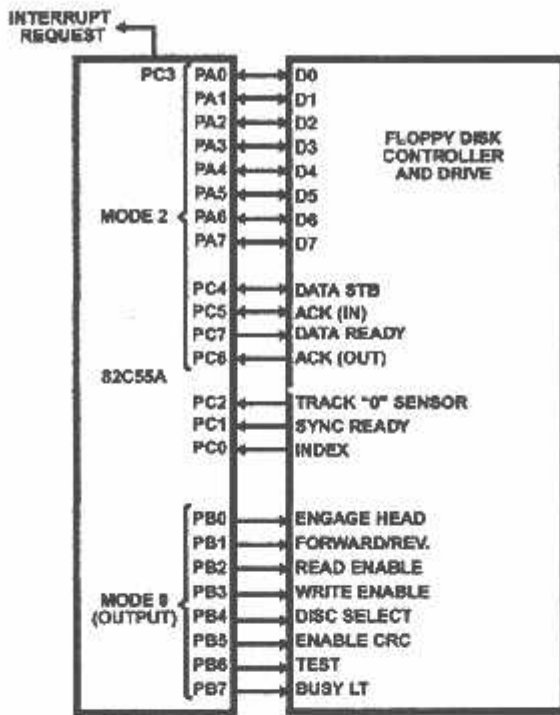


FIGURE 23. BASIC FLOPPY DISC INTERFACE

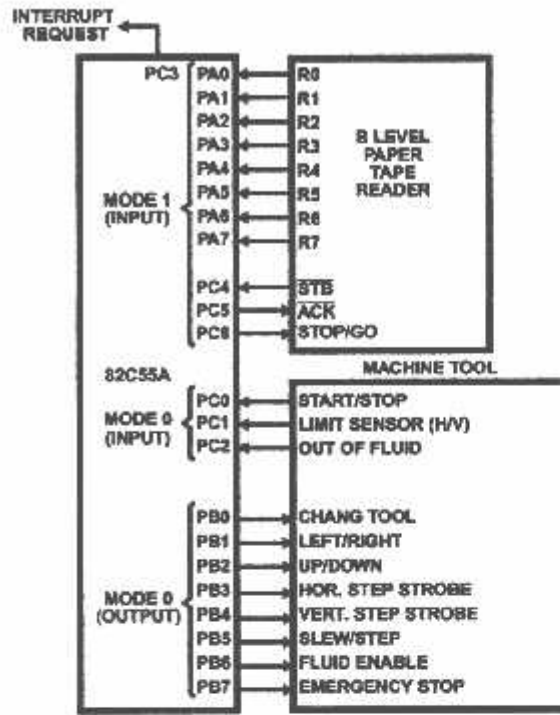


FIGURE 24. MACHINE TOOL CONTROLLER INTERFACE

82C55A

Absolute Maximum Ratings

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

Operating Conditions

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

Thermal Information

Thermal Resistance (Typical) Note 1	θ_{JA}	θ_{JC}
CERDIP Package	55°C/W	20°C/W
Ceramic LCC Package	70°C/W	16°C/W
Plastic DIP Package	50°C/W	-
PLCC	46°C/W	-
Storage Temperature Range	-85°C to +150°C	
Maximum Junction Temperature		
Ceramic Package	+175°C	
Plastic Package	+150°C	
Maximum Lead Temperature (Soldering 10s)	+300°C	
(Lead Tips Only for Surface Mount Packages)		

Die Characteristics

Gate Count	1000 Gates
------------	------------

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

DC Electrical Specifications $V_{CC} = 5.0V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$ (C82C55A);

$T_A = -40^\circ C$ to $+85^\circ C$ (I82C55A);

$T_A = -55^\circ C$ to $+125^\circ C$ (M82C55A)

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

NOTES:

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

Capacitance $T_A = 25^\circ C$

SYMBOL	PARAMETER	TYPICAL	UNITS	TEST CONDITIONS
C_{IN}	Input Capacitance	10	pF	FREQ = 1MHz, All Measurements are referenced to device GND
$C_{I/O}$	I/O Capacitance	20	pF	

Specifications 82C55A

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

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

NOTE:

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

Timing Waveforms

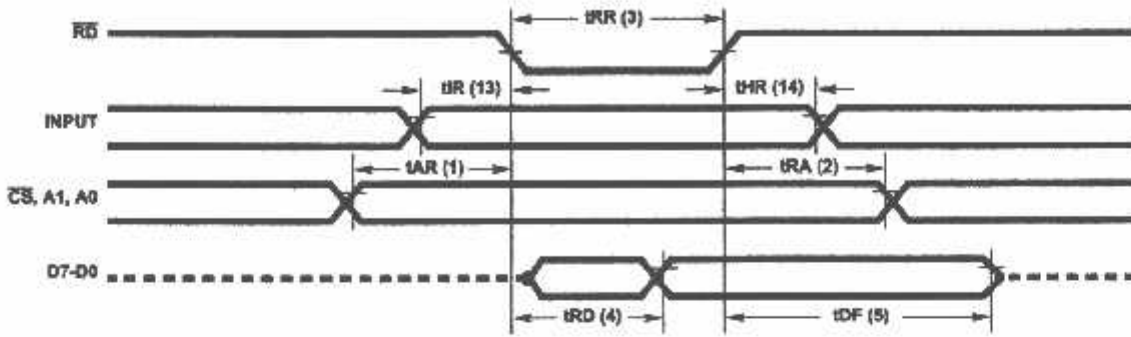


FIGURE 25. MODE 0 (BASIC INPUT)

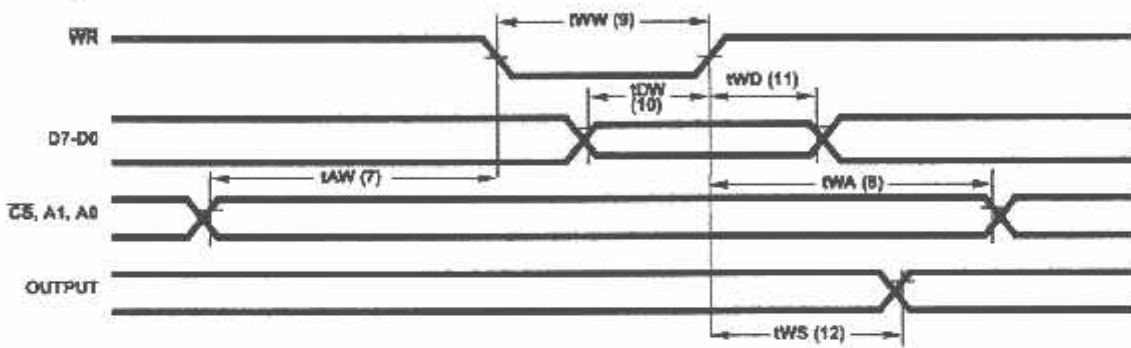


FIGURE 26. MODE 0 (BASIC OUTPUT)

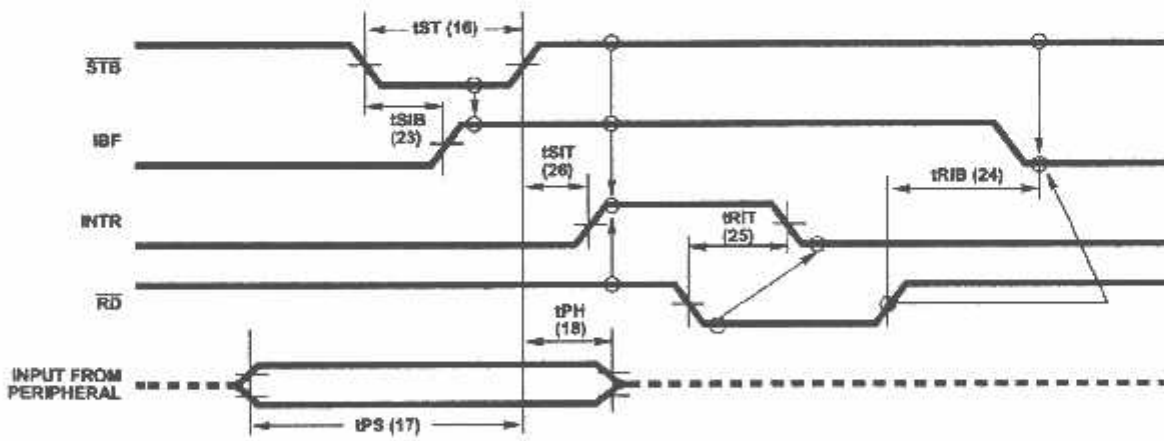


FIGURE 27. MODE1 (STROBED INPUT)

Timing Waveforms (Continued)

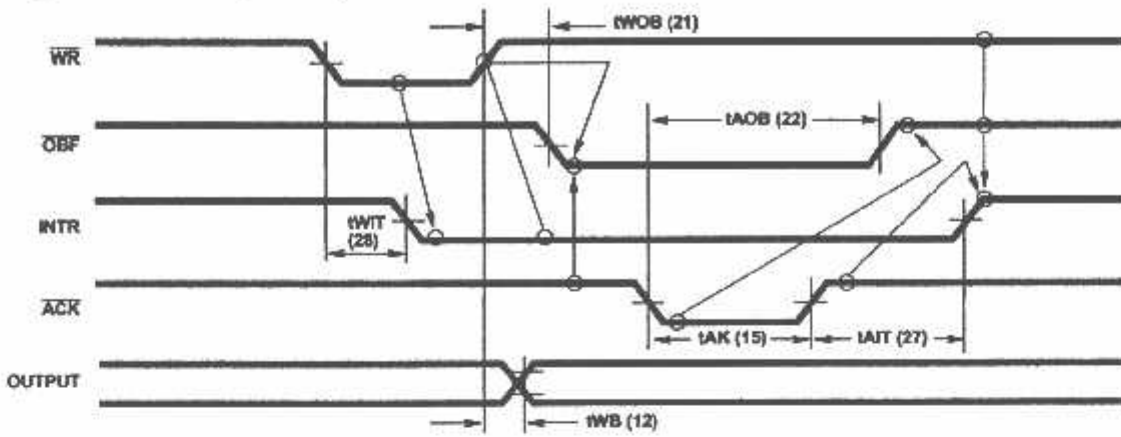


FIGURE 28. MODE 1 (STROBED OUTPUT)

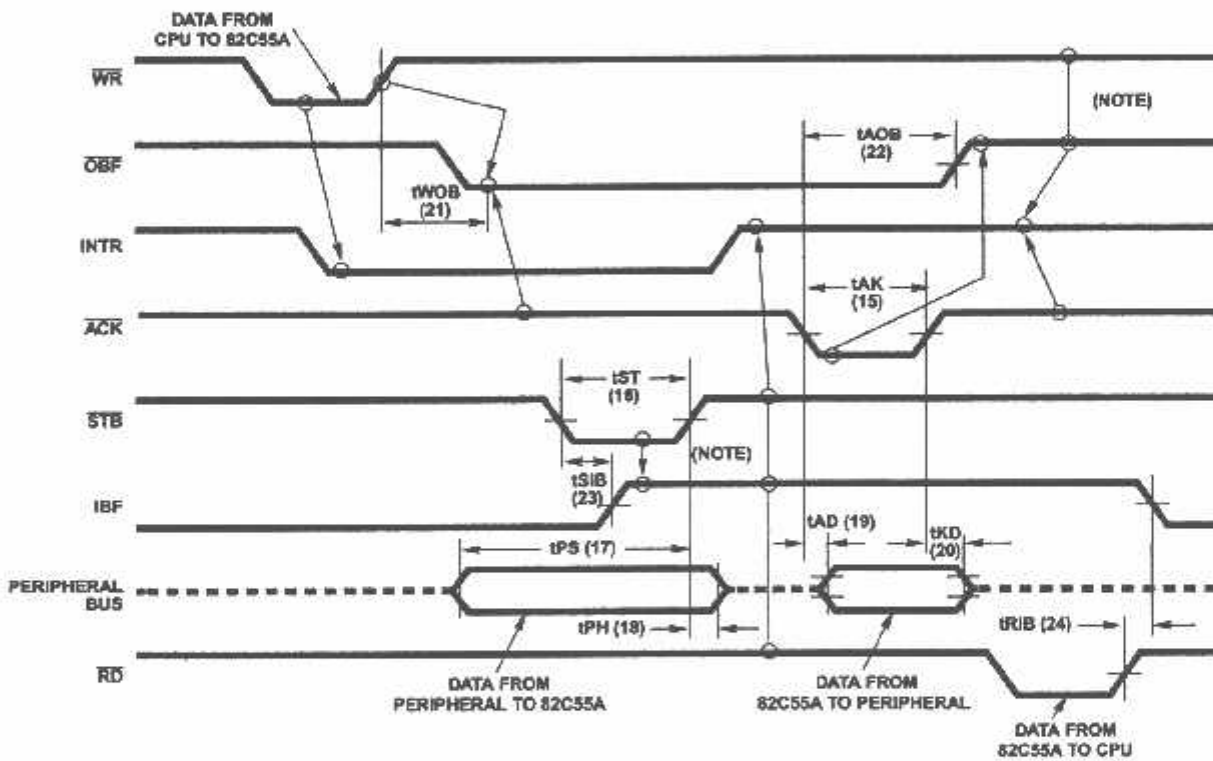


FIGURE 29. MODE 2 (BI-DIRECTIONAL)

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

Timing Waveforms (Continued)

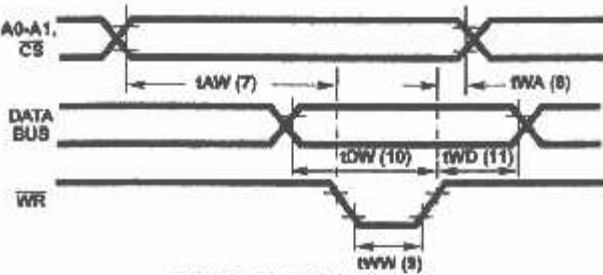


FIGURE 30. WRITE TIMING

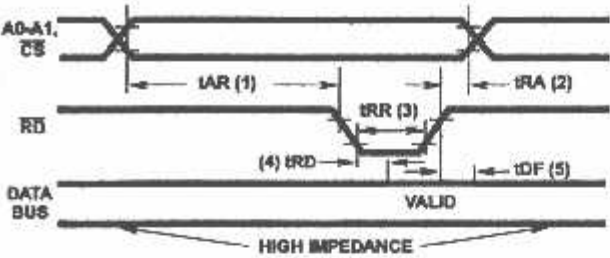
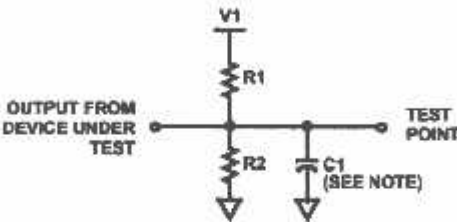


FIGURE 31. READ TIMING

AC Test Circuit



NOTE: Includes STRAY and JIG Capacitance

AC Testing Input, Output Waveforms



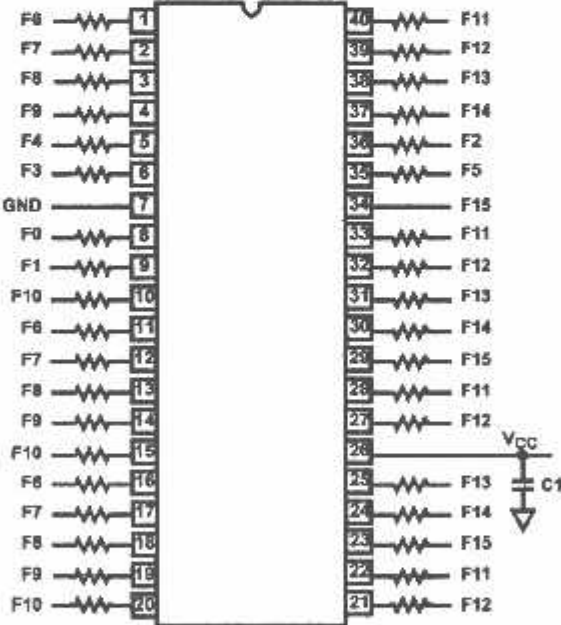
AC Testing: All AC Parameters tested as per test circuits. Input RiSE and FALL times are driven at 1ns/V.

TEST CONDITION DEFINITION TABLE

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

Burn-In Circuits

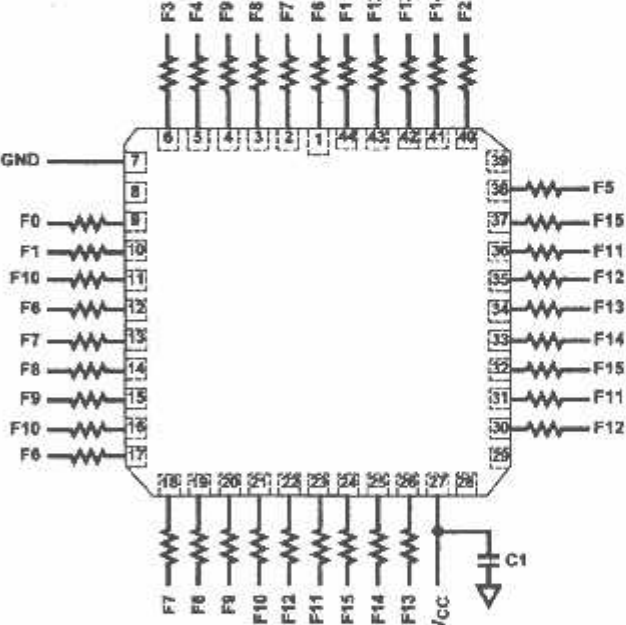
MD82C55A Cerdip



NOTES:

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

MR82C55A CERAMIC LCC



NOTES:

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

82C55A

Die Characteristics

DIE DIMENSIONS:

95 x 100 x 19 ±1mils

METALLIZATION:

Type: Silicon - Aluminum

Thickness: 11kÅ ±1kÅ

GLASSIVATION:

Type: SiO₂

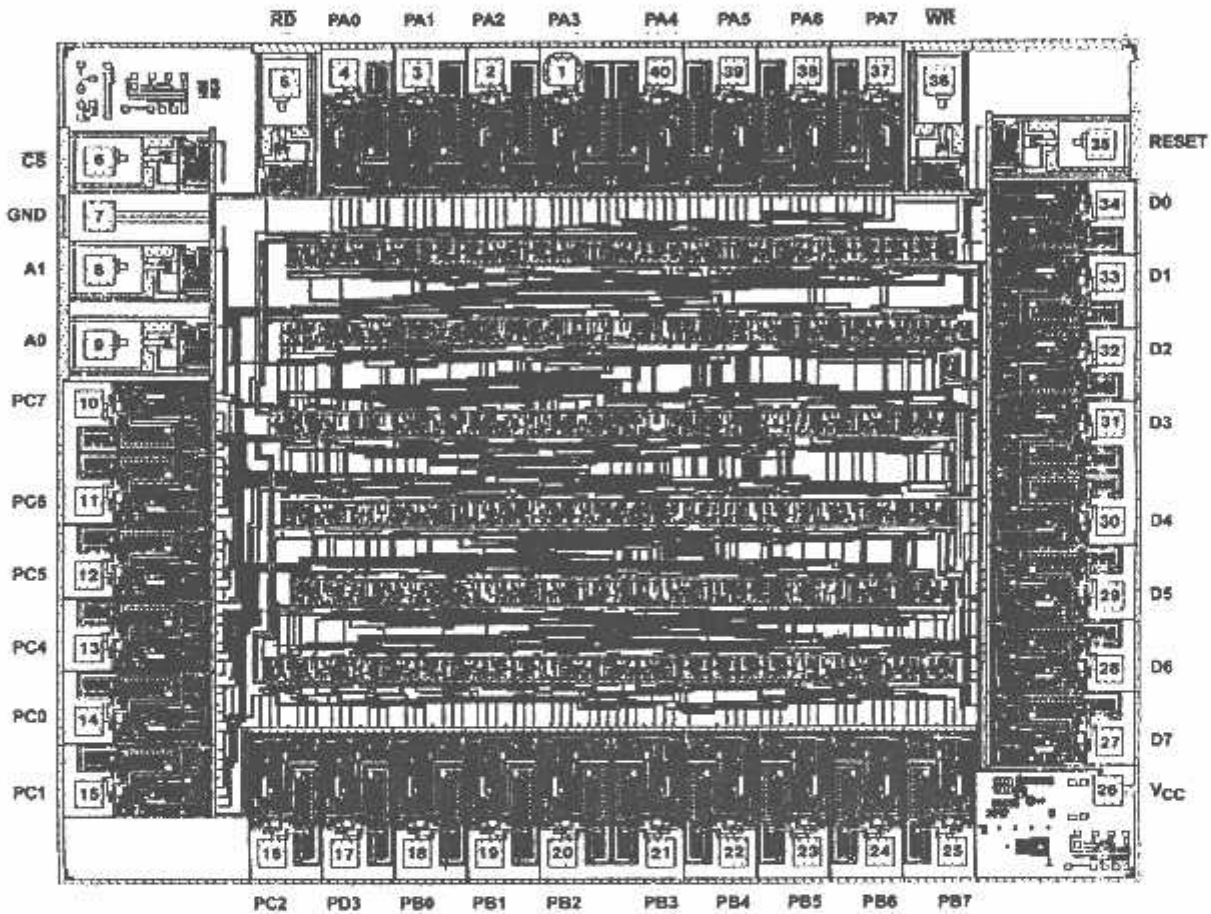
Thickness: 8kÅ ±1kÅ

WORST CASE CURRENT DENSITY:

$0.78 \times 10^5 \text{ A/cm}^2$

Metallization Mask Layout

82C55A



DM74LS573 Octal D Latch with 3-STATE Outputs

General Description

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

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

Features

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

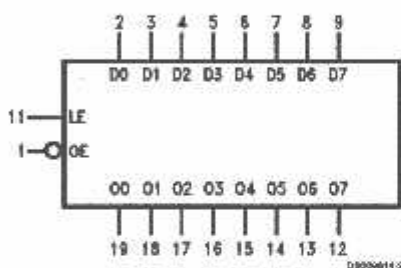
Connection Diagram

Dual-In-Line Package



Order Number DM74LS573WM or DM74LS573N
See Package Number M20B or N20A

Logic Symbol



V_{CC} = Pin 20
GND = Pin 10

Pin Names	Description
D0-D7	Data Inputs
LE	Latch Enable Input (Active HIGH)
OE	3-STATE Output Enable Input (Active LOW)
Q0-Q7	3-STATE Latch Outputs

Function Table

OUTPUT Enable	Latch Enable	D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q _O
H	X	X	Z

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

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V

Operating Free Air Temperature Range

DM74LS	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Recommended Operating Conditions

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

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

Electrical Characteristics

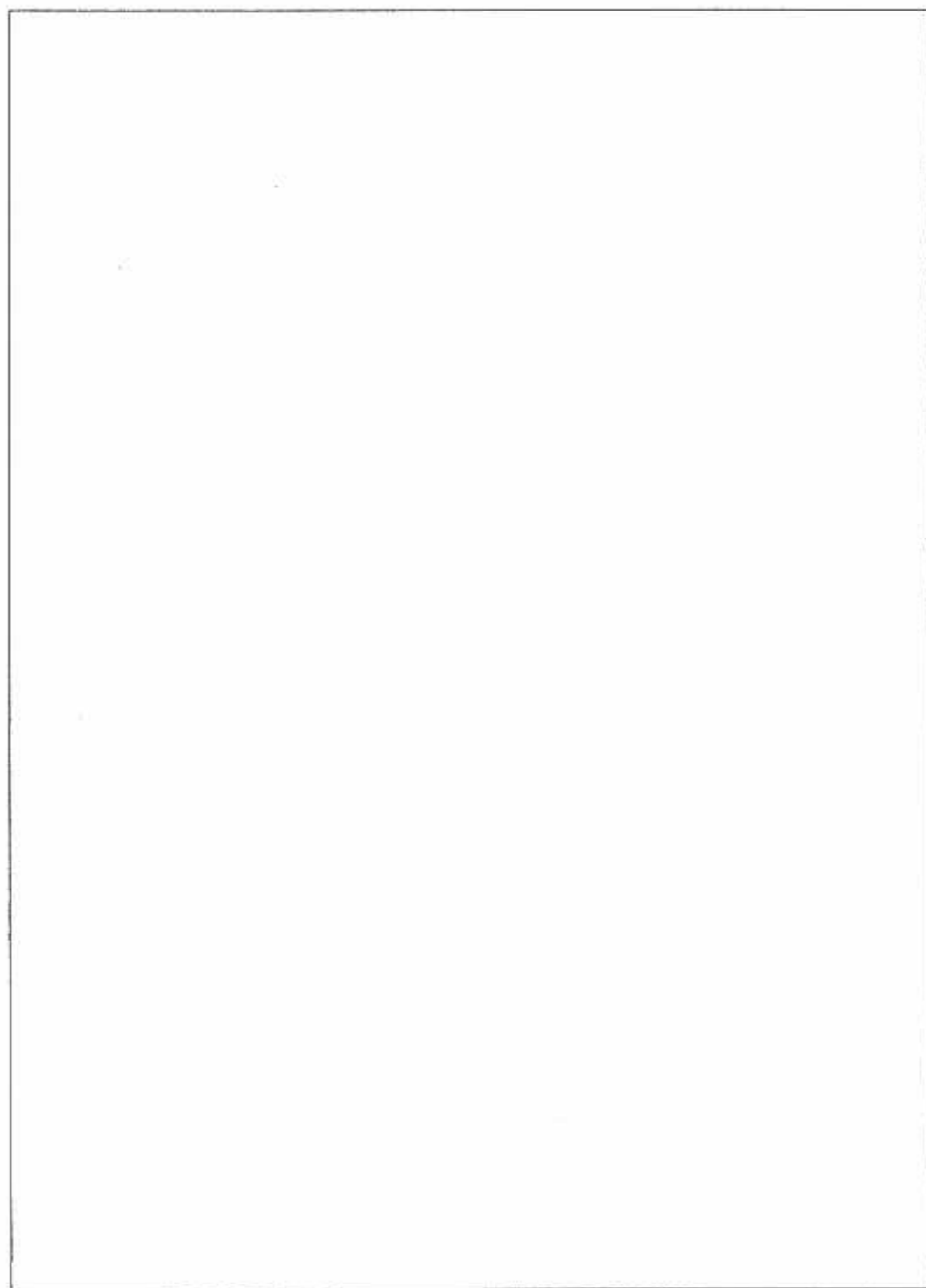
Over recommended operating free air temperature range (unless otherwise noted)

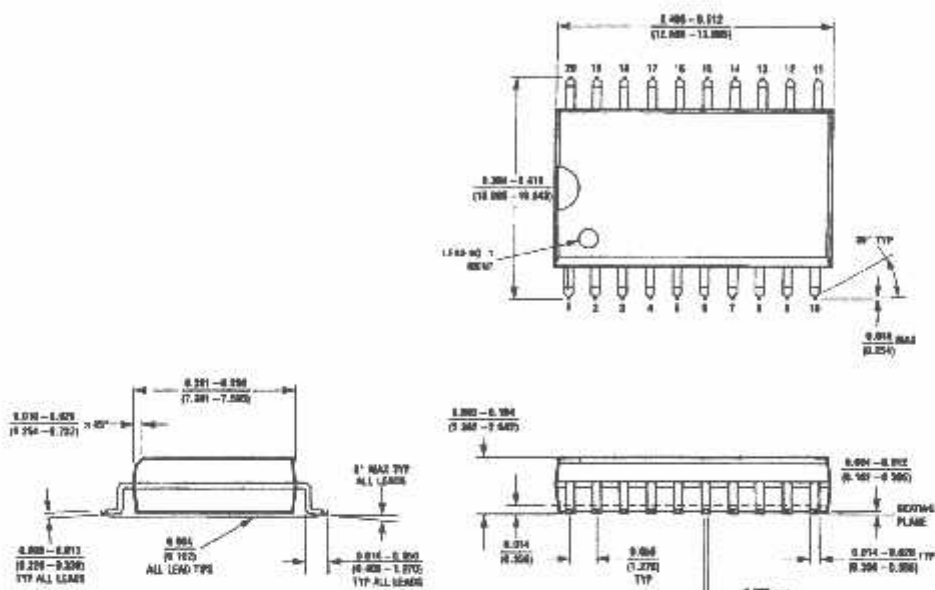
Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$			-1.5	V
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, I_{OH} = \text{Max}, V_{IL} = \text{Max}$	2.7	3.4		V
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, I_{OL}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

Note 2: All typicals are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$.

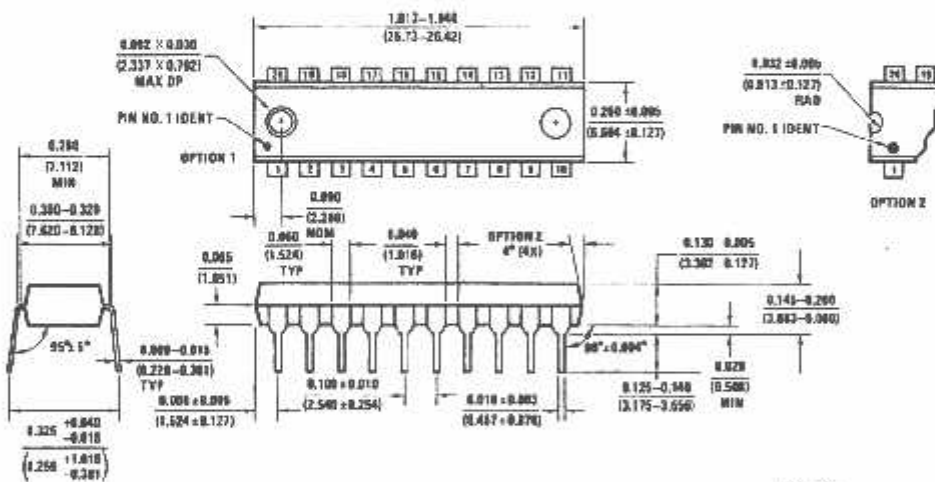
Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics				
at V_{CC} = 5V and T_A = 25°C (see Section 1 for Test Waveforms and output loading)				
Symbol	Parameter	R_L = 2 k Ω , C_L = 50 pF		Units
		Min	Max	
t_{PLH}	Propagation Delay		27	ns
t_{PHL}	Data to Q		18	
t_{PLH}	Propagation Delay		36	ns
t_{PHL}	LE to Q		25	
t_{PZH}	3-STATE Enable Time		20	ns
t_{PZL}	$\overline{OE}$ to Q		25	
t_{PHZ}	3-STATE Enable Time		20	ns
t_{PLZ}	$\overline{OE}$ to Q		25	
$t_s(H)$	Setup Time (High/Low)	3		ns
$t_s(L)$	Data to LE	7		
$t_h(H)$	Hold Time (High/Low)	10		ns
$t_h(L)$	Data to LE	10		
$t_w(H)$	Pulse Width (High)	15		ns
	Data to LE			



Physical Dimensions inches (millimeters) unless otherwise noted

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



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

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Fairchild Semiconductor
Corporation
Americas
Customer Response Center
Tel: 1-800-522-4372

www.fairchildsemi.com

Fairchild Semiconductor
Europe
Fax: +40 (0) 1 80-530 85 88
E-mail: europe.support@fsc.com
Deutsch: Tel: +49 (0) 8 141-35-0
English: Tel: +44 (0) 1 793-85-65-66
Italy: Tel: +39 (0) 2 57 5831

Fairchild Semiconductor
Hong Kong Ltd.
13th Floor, Straits Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: +852 2737-7200
Fax: +852 2514-0081

Fairchild Semiconductor
Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6176

Fairchild does not assume any responsibility for use of any circuitry described; no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change cell circuitry and specifications.

DM74LS138 • DM74LS139 Decoder/Demultiplexer

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The DM74LS138 decodes one-of-eight lines, based upon the conditions of the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The DM74LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

Features

- Designed specifically for high speed:
 - Memory decoders
 - Data transmission systems
- DM74LS138 3-to-8-line decoders incorporate 3 enable inputs to simplify cascading and/or data reception
- DM74LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)

DM74LS138	21 ns
DM74LS139	21 ns
- Typical power dissipation

DM74LS138	32 mW
DM74LS139	34 mW

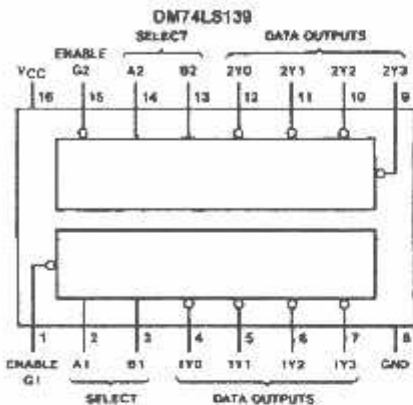
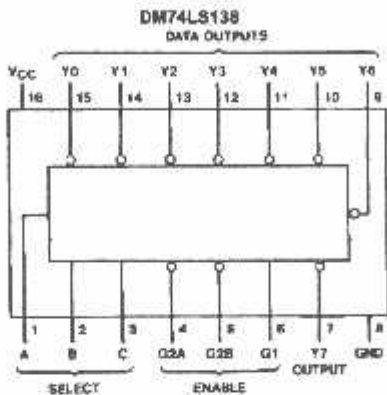
Ordering Code:

Order Number	Package Number	Package Description
DM74LS138M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS138SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS138N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS139M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS139SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS139N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

DM74LS138 • DM74LS139 Decoder/Demultiplexer

Connection Diagrams



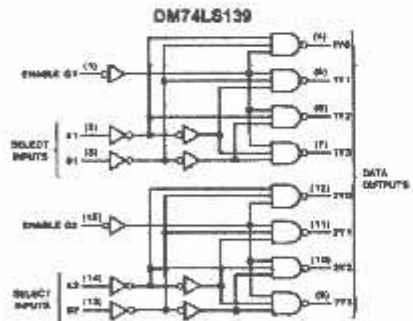
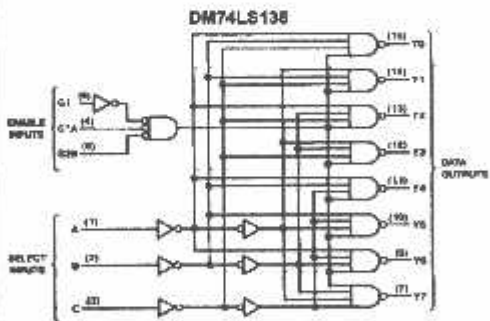
Function Tables

DM74LS138												
Inputs			Outputs									
Enable	Select											
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	H	L	H	L	H	H	H	H	H	H
H	L	L	H	H	L	H	L	H	H	H	H	H
H	L	L	H	L	L	H	L	H	H	H	H	H
H	L	L	H	H	L	L	H	L	H	H	H	H
H	L	L	H	H	H	L	L	H	L	H	H	H
H	L	L	H	H	H	H	L	L	H	L	H	H
H	L	L	H	H	H	H	H	L	L	H	L	H
H	L	L	H	H	H	H	H	H	L	L	H	L

DM74LS139						
Inputs		Outputs				
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care
Note 1: G2 = G2A = G2B

Logic Diagrams



Absolute Maximum Ratings (Note 2)

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

Note 2: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

DM74LS138 Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _{HI}	HIGH Level Input Voltage	2			V
V _{IL}	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			8	mA
T _A	Free Air Operating Temperature	0		70	°C

DM74LS138 Electrical Characteristics

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

Symbol	Parameter	Conditions	Min	Typ (Note 3)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -18 mA			-1.5	V
V _{OH}	HIGH Level Output Voltage	V _{CC} = Min, I _{OH} = Max, V _{IL} = Max, V _{HI} = Min	2.7	3.4		V
V _{OL}	LOW Level Output Voltage	V _{CC} = Min, I _{OL} = Max, V _{IL} = Max, V _{HI} = Min I _{OL} = 4 mA, V _{CC} = Min		0.35 0.25	0.5 0.4	V
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V			0.1	mA
I _{HH}	HIGH Level Input Current	V _{CC} = Max, V _I = 2.7V			20	µA
I _{IL}	LOW Level Input Current	V _{CC} = Max, V _I = 0.4V			-0.36	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 4)	-20		-100	mA
I _{CC}	Supply Current	V _{CC} = Max (Note 5)		6.3	10	mA

Note 3: All typicats are at V_{CC} = 5V, T_A = 25°C.

Note 4: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 5: I_{CC} is measured with all outputs enabled and OPEN.

DM74LS138 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	From (Input) To (Output)	Levels of Delay	R _L = 2 kΩ				Units
				C _L = 15 pF		C _L = 50 pF		
				Min	Max	Min	Max	
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output	2		18		27	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output	2		27		40	ns
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output	3		18		27	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output	3		27		40	ns
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output	2		18		27	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output	2		24		40	ns
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output	3		18		27	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output	3		28		40	ns

DM74LS139 Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage	2			V
V_{IL}	LOW Level Input Voltage			0.8	V
I_{OH}	HIGH Level Output Current			-0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

DM74LS139 Electrical Characteristics

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

Symbol	Parameter	Conditions	Min	Typ (Note 6)	Max	Units
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}$, $I_I = -18 \text{ mA}$			-1.5	V
V_{OH}	HIGH Level Output Voltage	$V_{CC} = \text{Min}$, $I_{OH} = \text{Max}$, $V_{IL} = \text{Max}$, $V_{IH} = \text{Min}$	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	$V_{CC} = \text{Min}$, $I_{OL} = \text{Max}$, $V_{IL} = \text{Max}$, $V_{IH} = \text{Min}$ $I_{OL} = 4 \text{ mA}$, $V_{CC} = \text{Min}$		0.35 0.25	0.5 0.4	V
I_I	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}$, $V_I = 7 \text{ V}$			0.1	mA
I_{IH}	HIGH Level Input Current	$V_{CC} = \text{Max}$, $V_I = 2.7 \text{ V}$			20	μA
I_{IL}	LOW Level Input Current	$V_{CC} = \text{Max}$, $V_I = 0.4 \text{ V}$			-0.35	mA
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 7)	-20		-100	mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}$ (Note 8)		5.5	11	mA

Note 6: All typicals are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

Note 7: Not more than one output should be shorted at a time, and the duration should not exceed one second.

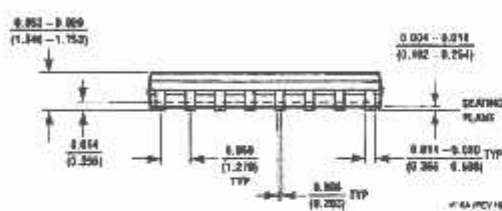
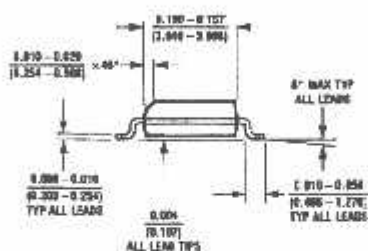
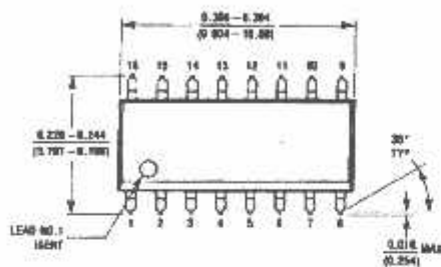
Note 8: I_{CC} is measured with all outputs enabled and OPEN.

DM74LS139 Switching Characteristics

at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$

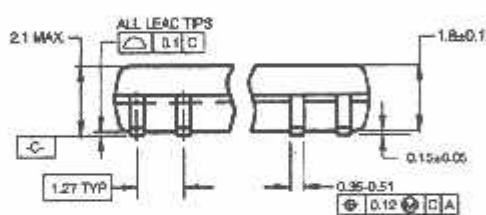
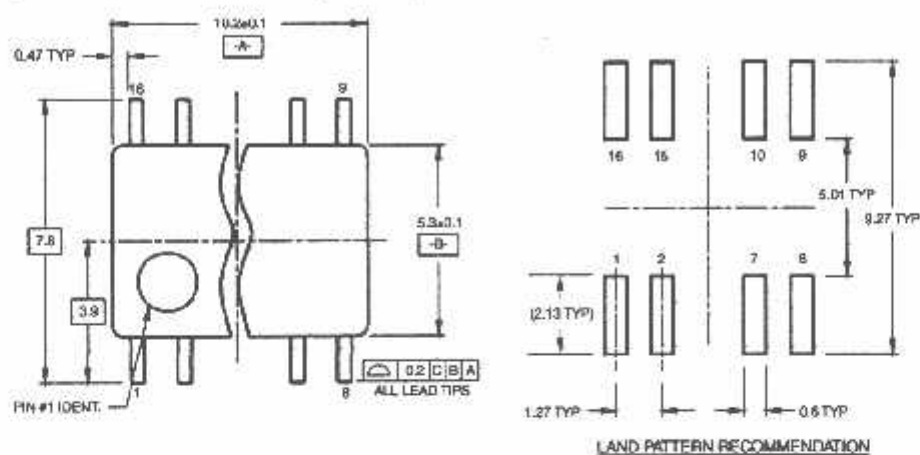
Symbol	Parameter	From (Input) To (Output)	$R_L = 2 \text{ k}\Omega$				Units
			$C_L = 15 \text{ pF}$		$C_L = 50 \text{ pF}$		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Select to Output		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Select to Output		27		40	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Output		18		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Output		24		40	ns

Physical Dimensions inches (millimeters) unless otherwise noted

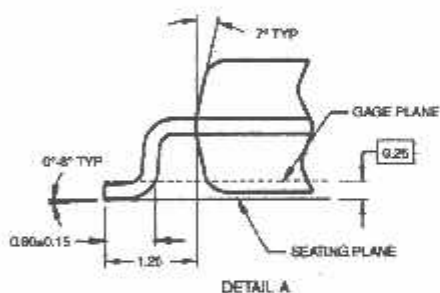
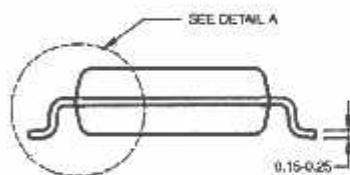


16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
Package Number M15A

DM74LS138 • DM74LS139

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


DIMENSIONS ARE IN MILLIMETERS

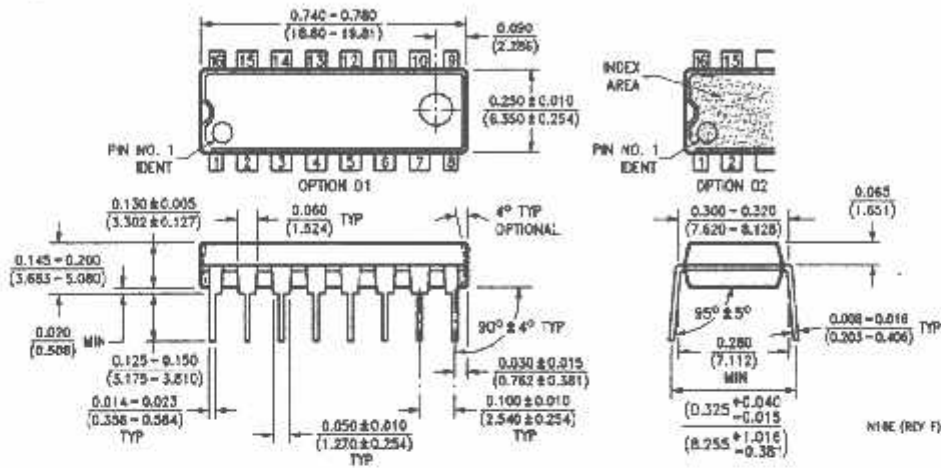

NOTES:

- CONFORMS TO EIAJ EDR-7320 REGISTRATION, ESTABLISHED IN DECEMBER, 1968.
- DIMENSIONS ARE IN MILLIMETERS.
- DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.

M160RevB1

16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
Package Number M160

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



Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com