TA/SEKJUR.TE/2007/009

TUGAS AKHIR

ALAT KONTROL KUALITAS PRODUKSI BERBASIS MIKROKONTROLER PADA UNIT PRODUKSI

Diajukan Sebagai Salah Satu Syarat
Memperoleh Gelar Sarjana Pada Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Disusun Oleh :

Nanang Hidayat
00524115

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

ALAT KONTROL KUALITAS PRODUKSI BERBASIS MIKROKONTROLER PADA UNIT PRODUKSI

TUGAS AKHIR

oleh:

Nama : Nanang Hidayat

No. Mahasiswa : 00 524 115

Yogyakarta, Januari 207

Pembimbing I,

Ir. H.Suyamto

Pembimbing II,

Tito Yuwono, ST, MSc

LEMBAR PENGESAHAN PENGUJI

ALAT KONTROL KUALITAS PRODUKSI BERBASIS MIKROKONTROLER PADA UNTI PRODUKSI

TUGAS AKHIR

oleh:

Nama : Nanang Hidayat

No. Mahasiswa : 00 524 115

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Januari 2007

Tim Penguji

Ir. H. Suyamto

Ketua

Tito Yuwono, ST., MSc.

Anggota I

Medilla K, ST

Anggota II

Mengetahui,
Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia

(Tito Yuwono, ST, MSc.)

PERSEMBAHAN

Sebagai ucapan terimakasih yang tulus

dari lubuk hati yang paling dalam

Aku Persembahkan Karya Ini Untuk :

Ayahanda (NAZRI IDRIS) & Ibunda (BASTYN IDRIS) tercinta
yang telah mencurahkan segala cinta, kasih sayang,
pengorbanan yang tak ada henti-hentinya.
Kakak2ku tersayang (Mas Nanang, Mb' Nitha, Mas Niko, Mb' Siska)
makasih atas dukungannya
Keponakanku (Dhifa, Naufal, Nashyita)
kalian selalu bikin om nank tersenyum dech...
Inna Setyowati yang selalu mengisi hari-hari ku...
Kel. Kajadi beserta Ibuk makasih atas dukungan dan doa nya...

MOTTO

Kami perintahkan kepada manusia (Berbuat Baik) kepada dua orang ibu-bapaknya,
ibunya telah mengandung dalam keadaan lemah dan bertambah-tambah,
dan menyapihnya dalam dua tahun. Bersyukurlah kepadaKu dan kedua orang ibu-bapakmu,
hanya kepada-Kulah kembalimu

(Q S Lukman : 14)

" .... Allah meninggikan orang yang beriman diantara kamu dan orang yang diberi
Ilmu Pengetahuan beberapa derajat ..."

(Q S Mujadillah : 11)

" Pengetahuan adalah satu-satunya kekayaan yang tidak dapat dirampas.

Hanya kematian yang bisa memadamkan

lampu pengetahuan yang ada dalam dirimu "

(Kahlil Gibran)

Really people only will reeach science if owning six matter : Intellegency, spirit, manfull, stock, tuition, learn and procces non stopped by the no desisting

(Friend Idyl Ali R.A)

KATA PENGANTAR

Assalamualaikum Wr.Wb.

Syukur alhamdulillah penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan Rahmat dan Hidayah serta Shalawat dan salam penulis sampaikan kepada junjungan kita Rasullulah S.A.W, sahabat serta pengikutnya sampai akhir zaman, sehingga penulis dapat menyelesaikan tugas akhir ini untuk memenuhi salah satu persyaratan akedemis pada jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Judul tugas akhir yang penulis ajukan adalah "ALAT KONTROL KUALITAS PRODUKSI BERBASIS MIKROKONTROLER PADA UNIT PRODUKSI"

Tidak sedikit halangan dan rintangan yang penulis alami selama penulisan tugas akhir ini namun atas bantuan dan kerjasama yang baik dari berbagai pihak, keseluruhan tahap penyusunan dapat diselesaikan sesuai rencana.

Dengan terselesaikannya tugas akhir ini, maka sangat banyak hutang budi yang tampak didepan mata penyusun yang tidak dapat dibayar dengan apapun sampai kapanpun, kecuali dengan mengucapkan terima kasih yang tulus kepada:

1. Allah SWT atas kehendak dan kasih sayangMu sehingga laporan ini dapat diselesaikan.
2. Bapak Fathul Wahid, ST, MSc. Selaku dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Tito Yuwono, ST, MSc selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia.
4. Bapak Ir. H Suyamto dan Bapak Tito Yuwono, ST, MSc selaku dosen pembimbing dalam tugas akhir ini.
5. Bapak Medilla K, ST selaku dosen penguji.
6. Kedua orang tuaku atas doa dan dukungan moril dan materil yang diberikan serta kakak dan adikku tersayang atas doa tulusnya
7. Ina Setyowati atas dukungan, semangat yang tidak kenal lelah dan kasih sayang selama ini.
8. Teman-teman Jurusan Teknik Elektro '00 Hadi, Eko, Ruli, Mas Yudi, atas bantuan dan kerjasamanya, semoga sukses selalu.
9. Semua pihak yang telah memberikan jalan terang dan memungkinkan penulis menyelesaikan tugas akhir ini.

Semoga bantuan dan perhatian yang telah diberikan pada penyusun menjadi amal sholeh di haribaan Allah SWT.

Akhir kata penyusun menyadari bila tulisan ini masih banyak terdapat kekurangan, karena manusia juga diciptakan tidak ada yang sempurna dan memiliki kemampuan yang sangat terbatas.

Semoga laporan tugas akhir ini dapat bermanfaat bagi pembaca pada umumnya dan bagi penulis pada khususnya.

Wassalamualaikum Wr. Wb.

Yogyakarta, Januari 2007

Penulis

ABSTRAKSI

Dalam dunia industri ada beberapa kendala yang dihadapi untuk meningkatkan hasil produksi. Mengingat hasil produksi yang dibutuhkan oleh perusahaan adalah produksi dengan kualitas yang tinggi dan efisiens, maka diperlukan penyelesaian-penyelesaian atas kendala-kendala yang ada. Diharapkan kendala yang ada bisa ditekan seminimal mungkin. Permasalahan yang mungkin terjadi antara lain perhitungan hasil produksi yang masih menggunakan cara manual, sehingga dapat terjadi kesalahan perhitungan yang telah dilakukan, lambatnya mengakses informasi hasil produksi secara cepat, tidak adanya rekaman hasil produsi yang dihasilkan pada setiap produksi belangsung dan lain-lain. Oleh sebab itu diperlukan suatu alat yang diharapkan mampu mengatasi kendala-kendala yang mungkin terjadi. Salah satu solusi yang bisa dilakukan adalah dengan membuat piranti keras atau *hardware* yang dilengkapi dengan piranti lunak atau *software.*

Dalam tugas akhir ini, penulis menggunakan rangkaian *slave* mikrokontroler, rangkaian *master* mikrokontroler dan *visual basic* sebagai *software. Slave* mikrokontroler adalah rangkaian mikrokontroler yang dilengkapi tombol *good* dan tombol *not good* sebagai masukan dan LCD *(liquid cristal display)* sebagai keluaran. Sedangkan *master* mikrokontroler adalah rangkaian mikrokontroler dengan tambahan IC MAX-232 sebagai komunikasi *serial* yang berfungsi sebagai komunikasi antara rangkaian mikrokontroler dengan komputer. Adapun *software* yang digunakan adalah *Microsoft Visual Basic 6.0* dengan *database* yang dibuat dengan menggunakan *Microsoft Access.*

Hasil pengujian yang telah dilakukan, peralatan dapat berfungsi dengan baik. Namun masih ada kekurangan dalam tampilan dalam *visual basic* yang disebabkan terjadinya *delay* antara komunikasi mikrokontroler dengan komputer.

DAFTAR ISI

Halaman

BAB II LANDASAN TEORI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah.

Untuk dapat bersaing dan bertahan di era globalisasi yang semakin kompetitif, dunia industri dituntut untuk mempunyai daya saing tinggi. Dua faktor utama yang dapat meningkatkan daya saing industri adalah kualitas dari sumber daya manusia (SMD) dan inovasi teknologi secara terus menerus.

Dalam dunia industri dikenal adanya salah satu bagian unit prdoksi yaitu *Departement Unit Kontol Produksi.* Dalam departement ini tingkat produksi yang dihasilkan dimonitoring dan dicatat secara *manual.* Maka dengan perkembangan teknologi yang ada pada saat ini, terutama perkembangan dunia elektronika khususnya mikrokontroler. Mengingat kemampuan dan dimensi mikrokontroler, perangkat ini juga disebut komputer. Untuk itu melihat kondisi kontrol produksi pada unit produksi yang masih *manual* tersebut, maka dibuat suatu alat elektronika berbasis mikrokontroler yang dihubungkan dengan komputer, yang data-data tentang produksi bisa langsung diperoleh melalui mesin-mesin unit produski yang beroperasi. Software yang digunakan dalam perancangan alat kontrol produksi ini yaitu *Visual Basic,* dengan alasan bahwa program ini sudah sangat populer, cocok dengan alat yang dirancang, serta mempermudah jalannya kerja alat.

1.2. Rumusan Masalah.

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat di ambil suatu rumusan masalah sebagai berikut "Bagaimana membuat alat kontrol kualitas produksi berbasis mikrokontroler pada unit produksi"

1.3. Batasan Masalah.

Agar permasalahan yang dibahas dalam tugas akhir ini tidak menyimpang dan melebar dari judul yang telah ditetapkan maka perlu disusun pokok-pokok permasalahan yang akan dibahas. Penulis membatasi penulisan pada masalah :

1. Pembuatan rangkaian kontrol produksi yang menggunakan mikrokontroler AT89S52.
2. Program *assembler* digunakan sebagai bahasa pemorgraman pada mikrokontroler.
3. Rangkaian LCD digunakan untuk menampilkan data yang diperoleh.
4. *Software Visual Basic 6.0* digunakan sebagai aplikasi pada unit komputer.

1.4. Sistematika Penulisan.

Sistematika penulisan tugas akhir ini terdiri dari V bab, dengan masing-masing bab adalah sebagai berikut :

BAB I PENDAHULUAN : Membahas latar belakang, permasalahan dan batasan, metodalogi, sistematika pembahasan.

BAB II LANDASAN TEORI : Penjelasan tentang komponen-komponen yang akan digunakan serta bagian-bagian yang membangun sistem ini.

BAB III PERANCANGAN ALAT : Penjelasan mengenai keseluruhan sistem yang akan dibangun yaitu pembahasan mengenai perangkat keras dan perangkat lunak yang digunakan.

BAB IV UJI ALAT, ANALISA DAN PEMBAHASAN : Pembahasan mengenai analisa hasil pengujian sistem yang dibuat dibandingkan dengan kriteria hasil pengujian yang telah dilakukan.

BAB V PENUTUP : Pada bab ini merupakan kesimpulan akhir dari hasil perancangan dan saran untuk memajukan dan pengembangan alat hasil perancangan untuk pengembangan lebih lanjut.

1.5. Tujuan Penelitian.

Tujuan akhir dari penulisan tugas akhir ini adalah untuk mengontrol kualitas produksi dan menghitung hasil produksi yang dihasilkan secara komputerisasi.

1.6. Manfaat Penelitian.

Penulis mengharapkan dari penelitian ini akan menghasilkan suatu alat yang dapat memperbaiki sistem perhitungan pada unit produksi yang masih menggunakan cara konvensional.

BAB II

LANDASAN TEORI

2.1. Mikrokontroler AT89S52

Lahirnya mikrokontroler merupakan perkembangan dari sistem elektronika digital khususnya mikroprosesor. Seperti halnya sebuah komputer, baik itu komputer desktop (PC) maupun *mainframe,* mikrokontroler memiliki beberapa persamaan, yaitu :

a. Setiap komputer memiliki *CPU (central processing unit)* yang bertugas menjalankan program.

b. Program yang dijalankan tersebut disimpan ke dalam storage media (misalnya pada komputer berbentuk hardisk). Dalam mikrokontroler, program tersebut disimpan dalam *ROM (read only memory).*

c. Komputer dan mikrokontroler, keduanya memeliki *RAM (random access memory)* yang berfungsi untuk menyimpan data.

d. Untuk dapat berkomunikasi dengan pengguna, maka komputer dan mikrokontroler memiliki piranti masukan *(input device)* dan piranti keluaran *(output device).* Piranti masukan pada mikrokontroler misalnya berupa saklar, tombol dan sensor. Piranti keluaran misalnya berupa lampu *LED,* LCD dan lain-lain.

Sebuah mikrokontroler dapat diangap sebagai komputer mini yang memiliki kemampuan khusus. Dan program yang berada di dalam ROM berisi untuk aplikasi yang tidak berubah. Salah satu keunggulan mikrokontroler yaitu lebih lengkap jika dibadingkan dengan mikroprosesor yang harus menambah komponen diluar seperti

ROM, RAM dan *interface* sedangkan dalam sebuah mikrokontroler sudah memiliki ROM, RAM dan *interface.*

Mikrokontroler merupakan alat yang bekerja pada daya rendah, sekitar 50 mWatt dan tegangan kerja sebesar 5 volt DC. Dengan konsumsi daya rendah ini, mikrokontroler biasa dipakai untuk mengendalikan berbagai jenis sistem dengan mudah dan murah. Untuk mengontrol peralatan dengan daya lebih tinggi digunakanlah *driver* yang sesuai dengan taraf tegangan *output.* Contoh aplikasi mikrokontroler yang dapat dilihat sehari-hari antara lain : sistem kendali mesin cuci, pemograman pada televisi, sistem elektrik pada mobil, pengunaan pada telpon genggam dan alat-alat elektronik lainnya. Sedangkan penggunaan di dunia industri antara lain : *temperature control, speed control, production display* dan lain-lain.

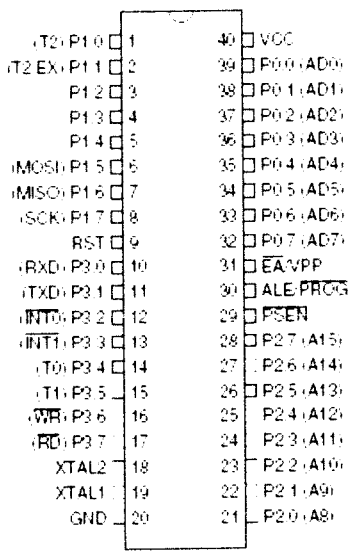
Ada berbagai macam mikrokontroler yang telah diproduksi dan digunakan di dunia. Perusahaan besar yang berkecimpung dengan semikonduktor yang memproduksi mikrokontroler dan mikroprosesor antara lain Intel Co. Ltd, Microchip, Atmel. Contoh mikroprosesor yang diproduksi oleh Intel adalah Z-80, sedangkan yang diproduksi oleh Microchip adalah MCU-PIC *(Peripheral Interface Controller)*, dan yang diproduksi oleh Atmel antara lain AT89C51, AT89S52 dan masih banyak lagi. Pada pembuatan tugas akhir ini digunakan mikroprosesor buatan Atmel dengan seri AT89S52.

AT89S52 merupakan mikrokomputer dengan CMOS 8 bit dengan 8 Kbyte *Flash Programmable and Erasable Read Only Memory* (PEROM). Mikrokontroler ini kompatibel dengan seri keluarga M-51 seperti AT89C51/52. Perbedaan yang mendasar yang membedakan antara mikrokontroler AT89C51/AT89S51 dengan mikrokontroler AT89C52/AT89S52 adalah kapasitas Flash memori yang diberikan. Adapun perbandingannya dapat dilihat dalam Tabel 2.1. berikut :

Tabel 2.1. Perbandingan mikrokontroler keluarga AT89XXX

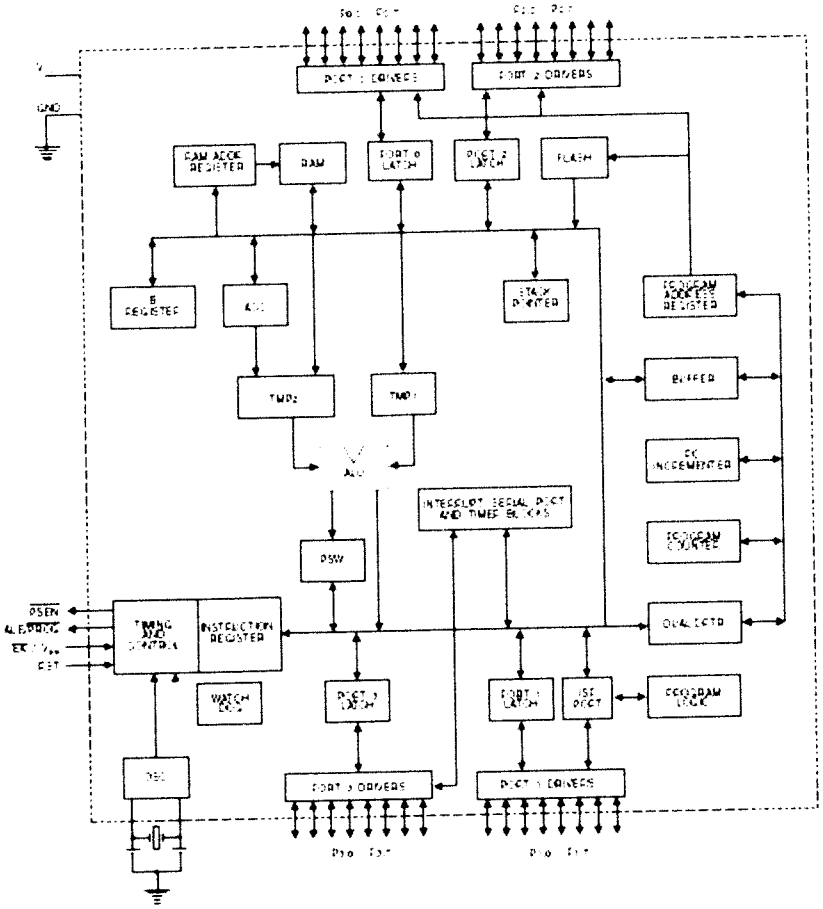
Tipe	RAM	Flash Memori	EEPROM
AT89C51/AT89S51	8 X 128 byte	4 K byte	Tidak ada
AT89C52/AT89S52	8 X 256 byte	8 K byte	Tidak ada
AT89C55	8 X 256 byte	20 K byte	Tidak ada
AT89S53	8 X 256 byte	12 K byte	Tidak ada
AT89S8252	8 X 256 byte	8 K byte	2 K byte

Adapun IC AT89S52 terdiri dari 40 pin dan ditunjukan pada Gambar 2.1. dibawah ini :



Gambar 2.1. Susunan pin AT89S52

Sedangkan bila dilihat dari blok diagram AT89S51 pada Gambar 2.2., maka terlihat jelas bahwa mikrokontroler AT89S52 mempunyai kesempurnaan fasilitas yang diberikan.



Gambar 2.2. Diagram blok AT89S52

Dari diagram blok AT89S52 pada Gambar 2.2 terlihat bahwa mikrokontroler AT89S52 mempunyai empat buah port untuk I/O data tersedia pula Akumulator, *register, RAM, stack pointer, Aritmatic Logic Unit (ALU), timer/counter, pengunci (latch),* komunikasi *serial* atau *pararel* dan rangkaian osilasi yang membuat AT89S52 dapat bekerja hanya dengan sekeping IC. Adapun penjelasan untuk masing-masing pin dijabarkan dalam fungsi masing-masing port.

2.2. Fungsi Masing-masing Port.

A. Vcc.

Berfungsi sebagai sumber tegangan +5 Volt. Berada pada pin 40.

B. Gnd

Bergungsi sebagai pentanahan *(ground)*. Berada pada pin 20.

C. Port 0

Port 0 adalah masukan/keluaran 8 bit dengan nama P0.0-P0.7. Jenisnya terbuka dengan masukan dua arah (*open drain bi directional I/O port*). Jika port 0 berlogika 1 maka dapat digunakan sebagai masukan yang mempunyai impendansi tinggi.

Selain berfungsi sebagai masukan/ keluaran, Port 0 juga berfungsi sebagai :

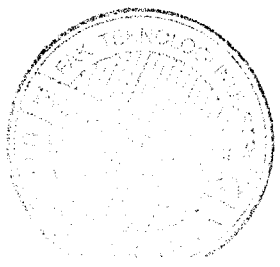
a. Multipleks antara alamat rendah (A0 s/d A7) dan data (D0-D7) pada saat mengakses memori program eksternal atau memori data eksternal. Pada fungsi ini, port 0 membutuhkan resistor *pullup*.
b. Masukan byte kode program selama pemograman flash memori (memori program internal atau *onchip*) dan keluaran saat verivikasi. Resistor *pullup* dibutuhkan selama verivikasi.

D. Port 1

Port 1 adalah masukan /keluaran 8 bit dengan nama masing-masing pin P1.0 s/d pin P1.7 yang bersifat dua arah. Port 1 sudah dipasang resistor *pullup* secara internal. Jika logika satu dituliskan pada port 1 maka keluaran akan berlogika satu dan dapat digunakan sebagai masukan.

Selain itu port 1 dapat berfungsi juga sebagai :

a. Port 1 sebagai masukan alamat rendah pada saat pemograman memori flash internal dan verivikasi.
b. Pada AT89S52 port P1.0 juga mempunyai fungsi lain yaitu sebagai T2 (masukan pencacah eksternal ke Timer/Counter 2) dan P1.1 mempunyai fungsi lain yaitu T2EX (Timer/Counter 2 reload trigger dan kendali arah).



E. Port 2

Port 2 adalah masukan /keluaran 8 bit dengan nama masing-masing pin P2.0 s/d pin P2.7 yang bersifat dua arah. Port 2 sudah dipasang resistor *pullup* secara internal. Jika logika satu dituliskan pada port 2 maka keluaran akan berlogika satu dan dapat digunakan sebagai masukan.

F. Port 3

Port 3 adalah masukan /keluaran 8 bit dengan nama masing-masing pin P3.0 s/d pin P3.7 yang bersifat dua arah. Port 3 sudah dipasang resistor *pullup* secara internal. Jika logika satu dituliskan pada port 3 maka keluaran akan berlogika satu dan dapat digunakan sebagai masukan.

Selain itu port 3 juga memiliki fungsi khusus seperti pada Tabel 2.2. :

Tabel 2.2 Fungsi khusus pada port 3

Bit	Nama	Funsi Alternatif
P3.0	RDX	Untuk menerima data port serial
P3.1	TDX	Untuk menerima data port serial
P3.2	INT0	Intrupsi eksternal 0
P3.3	INT1	Intrupsi eksternal 1
P3.4	T0	Input eksternal waktu / pencacah 0
P3.5	T1	Input eksternal waktu / pencacah 1
P3.6	WR	Jalur menulis memori data eksternal
P3.7	RD	Jalur membaca memori data eksternal

G. Rst.

Berfungsi sebagai masukan reset. Jika reset diberi logika tinggi dalam waktu 2 siklus mesin maka mikrokontroler akan reset.

H. ALE/PROG

Signal adress Latch Enable (ALE) digunakan untuk mengaktifkan IC *latch* agar data alamat rendah disimpan. ALE aktif ketika mengakses program eksternal. Pin ini juga memberikan pulsa pemograman memori flash eksternal.

I. PSEN *(Program Store Enable)*

PSEN adalah keluaran signal *stobe* untuk membaca kode program (code memori). Ketika AT89S52 mengeksekusi memori program eskternal, signal PSEN diaktifkan dua kali setiap siklus mesinnya.

J. EA/Vpp *(Eksternal Access Enable)*

EA harus dihubungkan ke ground (GND) jika semua program diakses dari memori program eksternal *(ekstrnal code memory)* yang dimulai dari alamat 0x0000 sampai dengan 0xFFFF. Jika program yang akan dieksekusi berasal dari memori program eskternal dan internal maka EA dihubungkan ke VCC

Pin EA juga digunakan sebagai masukan tegangan pemograman ketika akan memprogram memori flash internal.

K. XTAL-1

Masukan penguat osilator membalik dan masukan rangkaian *clock internal.*

L. XTAL-2

Keluaran dari penguat osilator membalik.

2.3. Organisasi Memori.

Mikrokontroler ATMEL-51 mempunyai organisasi memori yang terdiri atas :

1. Memori program (CODE).

2. Memori data (DATA).
3. Memori data *indirect* (IDATA)
4. Memori data pengalamatan Bit (BIT).
5. Memori data *ekternal* (XDATA).
6. Memori data halaman *eksternal* (PDATA).
7. Spesial *function register* (SFR).

2.3.1. Memori program (CODE).

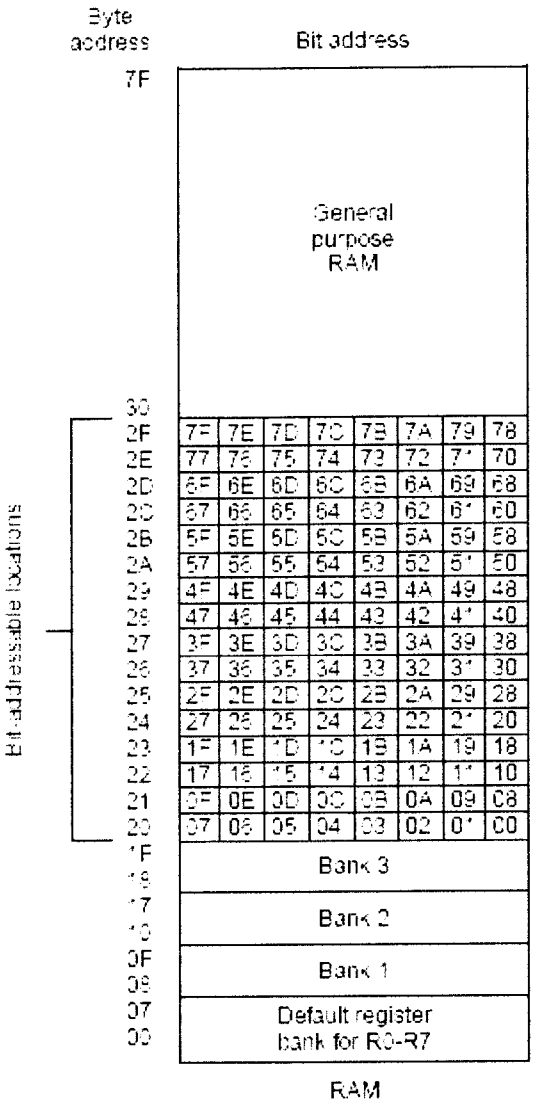
Memori program merupakan ruang memori yang digunakan untuk menyimpan kode program dan konstanta yang sifatnya tetap. Memori program bersifat hanya baca saja (*read only memory*) yang artinya ketika sedang eksekusi program memori ini hanya bersifat dibaca saja tidak dapat diubah isinya.

Memori program sebagian terdapat di dalam chip mikrokonroler (*on-chip*). Jika mikrokontroler mempunyai memori program *on-chip* dan diaktifkan maka akan menempati pada alamat awal yang kemudian dilanjutkan oleh memori program *off-chip*. Kapasitas memori program *on-chip* untuk AT89S52 sebesar 8 kB.

2.3.2. Memori data (DATA).

Yang dimaksud memori data adalah RAM *internal* (*on-chip)*. Untuk versi AT89S52 mempunyai memori data internal sebesar 256 byte. Pada segment data ini dibagi menjadi tiga bagian mulai alamat 0x00 s/d 0xFh dikenal sebagai *register* R0 s/d R7 yang diorgainisasikan menjadi 4 *bank*. Pemilihan *bank* yang aktif dilakukan dengan memberikan kombinasi logika pada *register program status word* (PSW). Bagian berikutnya adalah mulai alamat 0x20 s/d 0x2f sebanyak 128 bit merupakan lokasi memori yang dapat dimanipulasi perbit (*bit addressable*) juga dikenal dengan segment bit (BDATA). Bagian berikutnya adalah *general purpose* RAM mulai alamat

0x30 s/d 0x7f. Adapun memori data AT89S52 diperlihatkan pada Gambar 2.3. berikut:



Gambar 2.3. Memori data AT89S52

2.3.3. Memori data *indirect* (IDATA).

Memori data *indirect* (IDATA) merupakan segment data seluruh ruang memori data internal yaitu mulai alamat 0x00h s/d 0xFFh, 128 byte awal yaitu 00h s/d 7Fh secara fisik sama dengan segment DATA dan 128 byte di atasnya yaitu mulai alamat 0x80 s/d 0xFF *overlap* dengan *special function register.* Untuk mengatasi

permasalahan ini dilakukan dengan teknik pengalamatan yaitu *indirect* untuk IDATA dan *direct* untuk SFR. AT89S52 mempunyai ruang memori data *indirect* pada alamat 0x80 s/d 0xFF.

2.3.4. Memori data pengalamatan bit (BIT).

Secara fisik memori data pengalamatan BIT ini berada pada memori data (DATA) yang dimulai pada alamat 0x20 s/d 0x30 sebesar 128 bit. Pada jangkauan ini alamat ini masing-masing bit dapat dimanipulasi sendiri-sendiri (*bit addressable).*

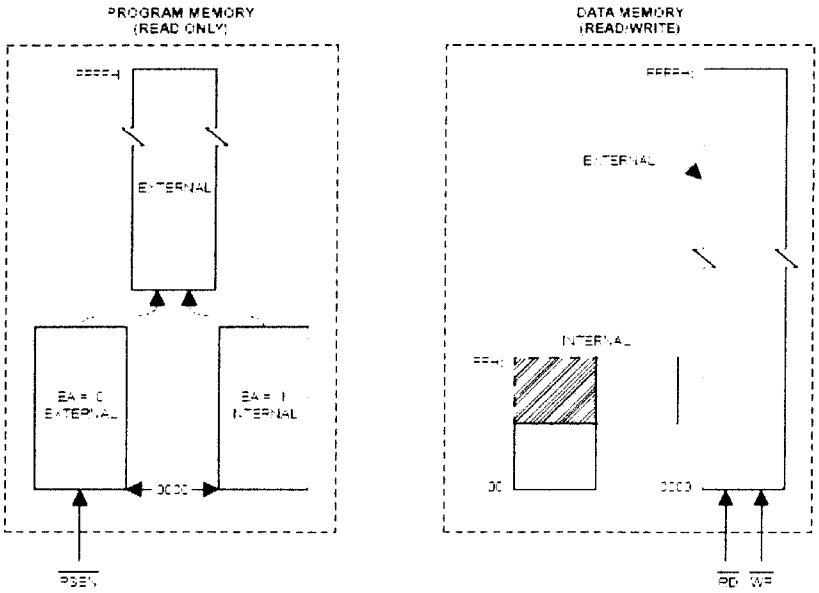
2.3.5. Memori data *eksternal* (XDATA).

Memori data *eksternal* (XDATA) adalah ruang memori data *off-chip* atau tidak terdapat didalam *chip* mikrokontroler. Ruang alamat ini diakses melaui port 0 (P0) dan port 2(P2). Port 0 sebagai bus alamat rendah yang termultipleks dengan bus data sedangkan port 2 sebagai bus alamat saja. Seluruh jalur alamat sebesar 16 bit sehingga mikrokontroler AT89S52 mampu mengakses memori data *eksternal* sebesar 64 kbyte.

Memori data eksternal bersifat dapat dibaca dan ditulis (*read/write memory).* Alamat memori data *eksternal* dengan memori program *eksternal* terjadi *overlap* karena itu penyelesaiannya adalah dengan menggunakan signal kendali yang berbeda. Pengaksesan memori program *eksternal* dikendalikan oleh signal PSEN (*program store enable)* sedangkan pengaksesan memori data *eksternal* dikendalikan oleh signal RD untuk baca dan RW untuk tulis. Adapun perbedaan signal kendali pada memori program dan memori data eksternal ditunjukan pada Gambar 2.4. :

2.3.6. Memori data halaman *eksternal* (PDATA).

Secara fisik PDATA sama dengan XDATA perbedaannya pada PDATA hanya menggunakan P0 untuk alamat rendah dan P2 tetap terhubung dengan register P2 di SFR.



Gambar 2.4. Perbedaan signal kendali pada memori program dan memori data eksternal.

2.3.7. *Special function register.*

Special function register merupakan register khusus yang digunakan sebagai kendali, *buffer* atau fungsi khusus lainnya. *Special function register* dipetakan mulai alamat 0x80h s/d 0xFF. Tidak seluruh ruang alamat diimplementasikan dengan suatu register. Beberapa register dapat dialamati per-bit. Adapun *special function register* (SFR) pada AT89S52 diperlihatkan pada Tabel 2.3. :

2.3.8. Penjelasan singkat fungsi *special function register* (SFR).

A. P0 (Port 0).

Port 0 berada pada alamat 0x80 dengan mode pengalamatan bit atau *bit addressable.* Merupakan register penyangga (*buffer*) port 0. Masing-masing BIT SFR dihubungkan dengan satu pin port 0 mikrokontroler. Sebagai contoh bit 0 port 0 adalah P0.0 dan bit 7 adalah P0.7. Menuliskan nilai 1 pada SFR ini akan menyebabkan pin yang bersesuaian menjadi logika tinggi dan sebaliknya jika ditulis 0 maka akan berlogika rendah.

Tabel 2.3. Alamat *special function register :*

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON * 00000000	T2MOD * XXXXXX00	RCAP2L * 00000000	RCAP2H * 00000000	TL2 * 00000000	TH2 * 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPH 00000000					PCON 0XXX0000 87H

B. SP (*Stack Pointer*).

Stack pointer beralamat pada 0x81. Ini merupakan pointer tumpukan (*stack pointer)* pada mikrokontroler. SFR ini menunjukan lokasi yang akan ditulis atau dibaca dari tumpukan (*stack*) pada RAM *internal.* Jika melakukan perintah *push* maka data akan disimpan pada alamat SP+1, ini berarti jika nilai SP=07h maka instruksi *push* akan menyimpan nilai pada tumpukan dialamat 0x80. Nilai SP akan dimodifikasi oleh perintah yang semisal *push, pop lcall, ret reti* dan intrupsi.

C. DPL/DPH (*Data Pointer Low/High*).

SFR DPL berada pada alamat 0x82 sedangkan SFR DPH berada pada alamat 0x83. SFR DPH dan DPL bekerjasama membentuk 16 bit yang dikenal dengan *pointer data* (DPTR). *Pointer data* (DPTR) digunakan dalam operasi yang melibatkan RAM *eksternal* maupun beberapa instruksi yang melibatkan memori program. Karena DPTR mempunyai lebar 16 bit maka dapat menyatakan/ menunjuk mulai 0x0000 s/d 0xFFFF (0 s/d 65.535).

D. PCON (*Power Control*).

SFR PCON berada pada alamat 0x87. SFR ini digunakan untuk mengendalikan mode *power control.* Contohnya adalah pada mode "*sleep*" yang membutuhkan daya yang rendah. Satu bit didalam PCON juga digunakan untuk menggandalkan kecepatan *baud rate* pada komunikasi *serial.*

E. TCON (*Timer/Counter Control*).

SFR TCON berada pada alamat 0x88. SFR ini juga dapat dialamati per-bit atau *bit addressable.* TCON digunakan sebagai untuk mengkonfigurasi dan memodifikasi operasi *timer/counter* 0 dan 1 pada ATMEL-51. SFR ini mengendalikan masing-masing *timer/counter* apakah berjalan atau berhenti dan juga terdapat *flag* yang menandakan masing-masing *timer/counter* sudah melimpah (*overflow*). Pada SFR ini juga terdapat bit yang tidak berhubungan dengan operasi *timer/counter.* Bit ini digunakan untuk mengaktifkan intrupsi *eksternal* dan juga terdapat *flag* intrupsi yang akan diset jika terjadi intrupsi *eksternal.*

F. TMOD (*Timer/Counter Mode*).

SFR TMOD berada pada alamat 0x89. SFR ini digunakan untuk mengkonfigurasi mode operasi setiap *timer/counter.* Dengan SFR ini program

dapat memodifikasi timer sebagai *timer* 16 bit, *timer* 8 bit *autoreload, timer/counter* selalu menghitung naik (*up counter*).

G. TL0/TH0 (*Timer/Counter 0 Low/High*).

SFR TL0 berada pada alamat 0x8A sedangkan SFR TH0 berada pada alamat 0x8C. TL0 dan TH0 besama-sama membentuk *timer/counter* 0. Kelakuan *timer/counter* ini dikendalikan oleh TMOD.

H. TL1/TH1 (*Timer/Counter 1 Low/High*).

SFR TL1 berada pada alamat 0x8B sedangkan SFR TH1 berada pada alamat 0x8D. TL1 dan TH1 besama-sama membentuk *timer/counter* 1. Kelakuan *timer/counter* ini dikendalikan oleh TMOD.

I. P1 (Port1)

P1 berada pada alamat 0x90. SFR ini dapat dialamati per bit atau (*bit addressable*). SFR ini merupakan masukan/keluaran. Masing-masing bit SFR dihubungkan dengan satu pin mikrokontroler. Sebagai contoh bit 0 port 1 adalah P1.0 dan bit 7 adalah P1.7. Menuliskan nilai 1 pada SFR ini akan menyebabkan pin yang bersesuaian menjadi berlogika tinggi dan sebaliknya jika ditulis 0 maka akan berlogika rendah.

J. SCON (*Serial Control*).

SFR SCON berada pada alamat 0x98 dengan mode pengalamatan bit. SFR ini digunakan untuk mengkonfigurasi kelakuan port *serial* ATMEL-51, kecepatan *baud rate,* apakah diaktifkan untuk menerima data dan juga terdapat *flag* yang menandakan pengiriman dan penerimaan data sukses.

K. SBUF(*Serial Buffer*).

SFR SBUF berada pada alamat 0x99. *Serial buffer* digunakan untuk mengirim maupun menerima data melaui masukan/keluaran port *serial.* Data yang

dituliskan pada SBUF dikirimkan melalui pin TXD. Data yang diterima melauli RXD akan disampaikan ke SBUF. Dengan kata lain SBUF melayani port keluaran jika ditulis dan melayani masukan jika dibaca.

L. P2 (Port 2).

P2 berada pada alamat 0xA0. SFR ini dapat dialamati per bit atau (*bit addressable*). SFR ini merupakan masukan/keluaran. Masing-masing bit SFR dihubungkan dengan satu pin mikrokontroler. Sebagai contoh bit 0 port 2 adalah P2.0 dan bit 7 adalah P2.7. Menuliskan nilai 1 pada SFR ini akan menyebabkan pin yang bersesuaian menjadi berlogika tinggi dan sebaliknya jika ditulis 0 maka akan berlogika rendah.

M. IE (*Intrupt Enable*).

SFR ini berada pada alamat 0xA8. SFR ini juga digunakan untuk mengaktifkan atau menonaktifkan instrupsi tertentu. Bit ke-7 digunakan untuk mengaktifkan atau menonaktifkan semua instrupsi. Jika bit ini 0 maka semua instrupsi tidak aktif dan jika 1 maka aktif. Kondisi ini menyebabkan suatu instrupsi ditentukan oleh masing-masing bit.

N. P3 (Port 3).

P3 berada pada alamat 0xB0. SFR ini dapat dialamati per bit atau (*bit addressable*). SFR ini merupakan masukan/keluaran. Masing-masing bit SFR dihubungkan dengan satu pin mikrokontroler. Sebagai contoh bit 0 port 3 adalah P3.0 dan bit 7 adalah P3.7. Menuliskan nilai 1 pada SFR ini akan menyebabkan pin yang bersesuaian menjadi berlogika tinggi dan sebaliknya jika ditulis 0 maka akan berlogika rendah.

O. IP (*Intrupt Priority*).

SFR ini berada pada alamat 0xB8 dengan mode pengalamatan per bit.SFR ini juga digunakan untuk mengatur prioritas intrupsi satu terhadap yang lainnya. Prioritas rendah dinyatakan dengan logika rendah (0) dan prioritas tinggi dinyatakan dengan logika tinggi (1). Sebagai contoh jika semua intrupsi prioritasnya rendah kecuali intrupsi *serial* maka intrupsi *serial* akan selalu dieksekusi jika aktif bahkan jika intrupsi yang lain sedang dieksekusi. Tetapi jika intrupsi *serial* sedang dieksekusi maka tidak ada intrupsi yang lain yang dapat menyela.

P. T2CON (*Timer/Counter 2 Control*).

SFR T2CON berada pada alamat 0xC8 dengan mode pengalamatan per-bit. T2CON digunakan sebagai pengendali *timer/counter mode 2* dan status dari *timer/counter 2,* seperti menjalankan *timer/counter,* memilih fungsi *timer/counter.*

Q. T2MOD (*Timer/Counter 2 Mode*).

T2MOD berada pada alamat 0xC9. T2MOD adalah register yang digunakan untuk mengaktifkan keluaran *timer/counter* 2 atau tidak mengaktifkan keluaran dan digunakan untuk pemilihan *upconter* atau *downcounter.*

R. RCAP2L (*Register Capture Low).*

SFR RCAP2L berada pada alamat 0xCA. SFR ini adalah register *capture* rendah yang digunakan untuk menyimpan data TL2 pada saat kejadian tertentu.

S. RCAP2H (*Register Capture High).*

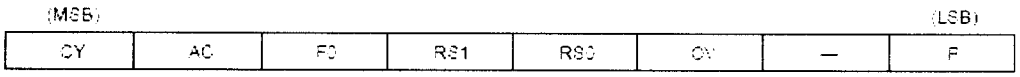
SFR RCAP2H berada pada alamat 0xCB. SFR ini adalah register *capture* tinggi yang digunakan untuk menyimpan data TL2 pada saat kejadian tertentu.

T. ACC (*Accumulator*).

ACC adalah SFR yang banyak digunakan dalam instruksi dianataranya adalah operasi aritmatika, pengaksesan *eksternal* RAM dan lain-lain.

U. PSW (*Program Status Word*).

SFR ini berada pada alamat 0xD0 dan dapat dilakukan pengalamatan per bit atau *bit addressable*. SFR ini digunakan untuk menyimpan bit-bit yang penting yang akan di-*set* atau akan di-*clear*. PSW berisi *carry flag, auxiliary carry flag, overflow flag,* dan *parity flag*. Adapun susunan bit PSW ditunjukan pada Gambar 2.5 :



Gambar 2.5. Susunan bit PSW.

2.4. Komunikasi Serial RS 232

RS 232 adalah (*Recommended Standart* No.232) dari *Electronic Industries Association* (EIA). RS 232 merupakan antar muka (*interface*) antara *Data Terminal Equipment* (DTE) atau komputer data dengan *Data Communication Equipment* (DCE) biasanya berupa modem.

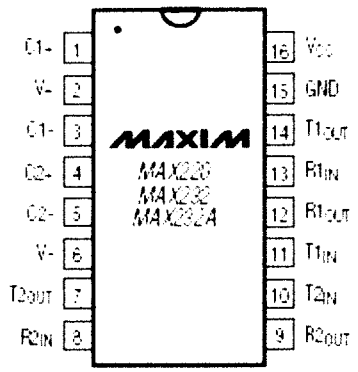
Spesifikasi RS 232 meliputi aspek :

1. Mekanis, bentuk koneksi yang digunakan DB 9.
2. Karakteristik RS 232.

Tabel 2.4. Fungsi register PSW :

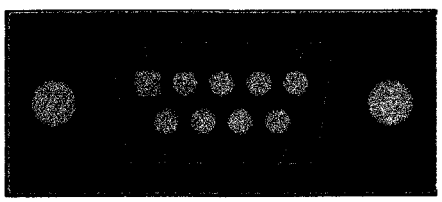
Simbol	Posisi	Fungsi
CY	PSW.7	*Carry flag* diset pada saat operasi aritmatika yang menghasilkan sisa.
AC	PSW.6	*Auxiliary carry flag* digunakan pada operasi BCD
F0	PSW.5	*Flag* 0 digunakan sebagai masukan umum
RS1	PSW.4	Bit 1 pemilihan *bank register*
RS0	PSW.3	Bit 0 pemilihan *bank register*
OV	PSW.2	*Over flow*
-	PSW.1	cadangan
P	PSW.0	Bit *parity*

Bentuk IC komunikasi serial RS-232 terlihat pada Gambar 2.6. berikut :



Gambar 2.6. IC max-232

Bentuk koneksi yang digunakan ditunjukan pada Gambar 2.7. berikut :



Gambar 2.7 Bentuk koneksi *serial.*

Karakteristik RS 232 antara lain :

1. Kecepatan maksimal 19200 bps.
2. Panjang kabel maksimal 50 *feet* (±15 meter)
3. Arus tidak lebih dari 0,5 A.

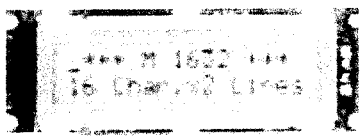
2.5. LCD(*Liquid Crystal Display*)

Tampilan yang digunakan dalam tugas akhir ini menggunakan modul tampilan kristal cair (LCD). LCD yang digunakan dalam tugas akhir ini adalah LCD M1632. Modul ini merupakan modul tampilan kristal cair matriks titik dengan pengendali LCD di dalamnya. Pengendali ini mempunyai sebuah ROM/RAM pembangkit karakter di dalamnya dan RAM data tampilan. Semua fungsi tampilan dikendalikan oleh perintah-perintah. Modul tampilan kristal cair (LCD) mempunyai karateristik sebagai berikut :

- 16 karakter, 2 baris tampilan.
- ROM pembangkit karakter untuk 192 tipe karakter (bentuk karakter :5 x 7 matriks titik)
- RAM pembangkit karakter untuk 8 tipe karakter (program tulis) dan bentuk karakter :5 x 7 matriks titik
- RAM data tampilan 80 x 8 (maksimum 80 karakter)
- Antar muka dengan 4 bit atau 8 bit untuk mikroprosesor
- RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit mikroprosesor
- Beberapa fungsi perintah anatar lain : penghapusan tampilan (*clear display*), posisi awal *cursor* (*cursor home*), tampilan karakter kedip (*display character blink*), penggeseran *cursor* (*cursor shift*)
- Rangkaian pembangkit detak yang sudah didalam

- Sumber daya tungal +5V
- Rangkaian otomatis reset saat dinyalakan
- Karena piranti ini didukung oleh mikrokontroler HD44780U maka piranti ini *support* dengan mikrokontroler HD44780S.

LCD M1632 memiliki 3 memori, yaitu DDRAM (*Display Data Random Access Memory*), CGRAM (*Character Generator Random Access Memory*), dan CGROM (*Character Generator Read Only Memory*). Bentuk fisik LCD M1632 ditunjukkan pada Gambar 2.8.



Gambar 2.8. Bentuk fisik LCD

2.5.1. DDRAM (*Display Data Random Access Memory*)

DDRAM merupakan memori tempat karakter yang ditampilkan berbeda. Contoh, untuk karater 'A' ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karater tersebut ditulis pada alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD. Adapun tata letak pengalamatan DDRAM dapat dilihat pada Gambar 2.9.

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

Gambar 2.9. Alamat DDRAM M1632

2.5.2. CGRAM (*Character Generator Random Access Memory*)

CGRAM merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai keinginan, tetapi memori ini akan hilang saat catu daya dimatikan/*power supply* tidak aktif, sehingga pola karakter akan hilang.

2.5.3. CGROM (*Character Generator Read Only Memory*)

CGROM merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut sudah ditentukan secara permanen dari HD44780U sehingga pengguna tidak dapat mengubah lagi. Tetapi karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun *power supply* tidak aktif.

Sebagai contoh pada saat HD44780U akan menampilkan data 41H yang tersimpan pada DDRAM, maka HD44780U akan mengambil data dialamat 41H (01000001b) yang ada pada CGROM yaitu pola karakter 'A'.

2.5.4. Konfigurasi pin

LCD M1632 memiliki 14 pin dengan fungsi yang berbeda-beda. Terdapat 3 pin sebagai bit kontrol yaitu E sebagai input *clock*, R/W sebagai input untuk memilih *read* atau *write* dar RS sebagai *register select*, 8 PIN sebagai bit data yaitu DB0 sampai DB7 dan 2 pin lainnya adalah VCC dan *Ground*. Konfigurasi pin LCD M1632 ditunjukkan pada Tabel 2.5.

2.5.5. Register

HD44780U mempunyai dua buah register yang aksesnya diatur dengan menggunakan kaki RS. Pada saat RS berlogika '0', maka register yang akan diakses

adalah register perintah dan pada saat RS berlogika 1, maka register yang diakses adalah Register data.

Tabel 2.5. Konfigurasi pin LCD M1632

No	Nama PIN	Keterangan
1	VCC	+5 V
2	GND	0 V
3	VEE	Tegangan kontras LCD
4	RS	Register Select, 0 = Register Perintah, 1 = Register Data
5	R/W	1 = Read, 0 = Write
6	E	Enable Clock LCD, Logika 1 setiap kali pengiriman atau pembacaan data
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7

2.5.5.1 Register perintah

Register ini adalah register dimana perintah dari mikrokontroler ke HD44780U pada saat proses penulisan data atau tempat status dari HD44780U dapat dibaca pada saat pembacaan data.

2.5.5.2 Register data

Register ini adalah register dimana perintah mikrokontroler dapat menuliskan atau membaca data ke atau dari DDRAM. Penulisan data pada register ini akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

2.5.6. Penulisan data ke register perintah dan register data

Penulisan data ke register perintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur *Address Counter* ataupun *Address Data.* Kondisi RS berlogika 0 menunjukkan akses data ke register perintah. RW berlogika 0 yang menunjukkan proses penulisan data. Sedangkan penulisan data pada register data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke register data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Pengiriman data dari data bus DB0 – DB7 diawali dengan pemberian pulsa logika 1 pada E clock. Dan akhiri dengan pulsa logika 0 pada E *Clock.*

BAB III

PERANCANGAN ALAT

3.1. Konsep Perancangan

Dalam perancangan peralatan ada beberapa perangkat pendukung antara lain :

a. Unit komputer.

b. Rangkaian komuniksai serial MAX-232

c. Rangakaian *master* mikrokontroler.

d. Rangkaian *slave* mikrokontroler 3 buah.

e. Rangkaian input berupa tombol *push button.*

f. Perangkat lunak "*Visual Basic*"

Secara keseluruhan diagram blok dapat ditunjukan pada Gambar 3.1.

3.2. Cara Kerja Alat.

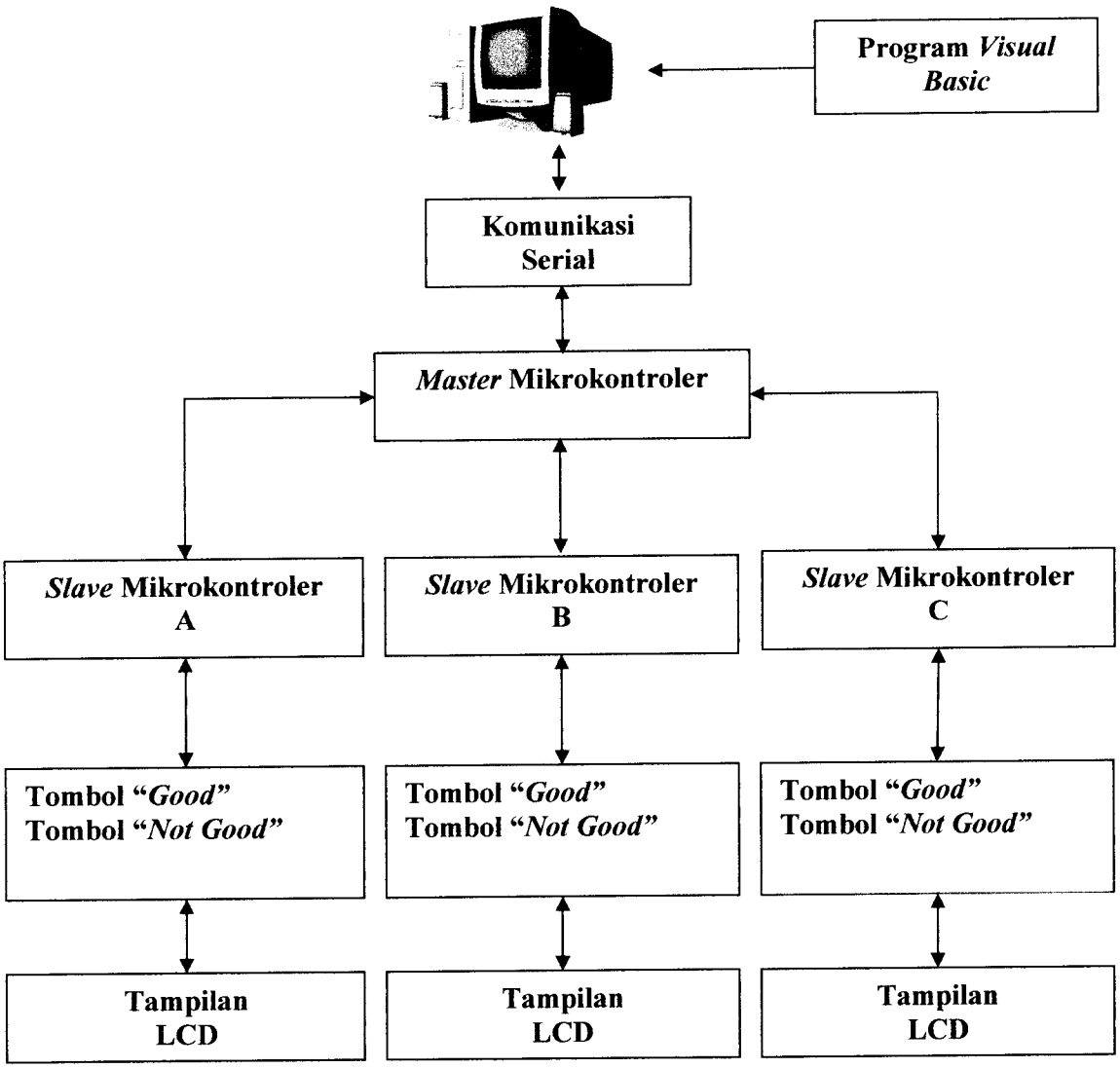
3.2.1. Rangkaian LCD.

Rangkaian tampilan berfungsi untuk menampilkan karakter yang akan ditampilkan. Rangkaian ini berupa modul tampilan kristal cair matriks titik. Modul ini terdiri dari beberapa bagian, yaitu :

- Bagian pengendali
- Bagian penggerak segment
- Bagian tampilan kristal cair

Modul ini memiliki 20 kaki masukkan yang terdiri dari : 8 bit masukan data (DB0 – DB7), masukan pilihan register (RS) sebanyak 1 bit, masukan pengendali sinyal baca dan tulis (RW) 1 bit, masukan sinyal enable (E) 1 bit, masukan tegangan

catu positif (+ 5 V) yaitu masukan V_{DD} , masukan catu tegangan 0 volt (V_{SS}), masukan tegangan pengatur kecerahan tampilan (Vee).



Gambar 3.1. Diagram blok alat kontrol produksi

Saluran data 8 bit DBO – DB7 merupakan sarana untuk memasukkan data alamat ROM dan RAM internal serta data tampilan kedalam modul. Masukan sinyal RS digunakan untuk memilih register-register yang berada dalam modul yaitu register perintah dan register data tampilan. Masukan sinyal E digunakan untuk mengaktifkan modul agar siap menerima data yang dikirimkan kepada modul. Bagian pengendali

berfungsi mengendalikan bekerjanya modul ini yaitu mengatur penerima data dari luar dan menampilkan data pada tampilan.

Untuk Cara kerja modul tampilan kristal cair adalah sebagai berikut :

1. Untuk proses masukkan data perintah internal :

   Data 8 bit ditempatkan pada saluran data DB_0 – DB_7, masukan sinyal RS diberi masukan logika rendah untuk memilih register perintah internal, kemudian masukan sinyal R/W diberi logika rendah untuk melaksanakan operasi penulisan terhadap modul. Selanjutnya masukan sinyal E diberi masukan perubahan dari logika tinggi ke rendah. Sinyal ini berfungsi untuk memulai modul untuk segera beroperasi sesuai perintah yang dimasukkan.

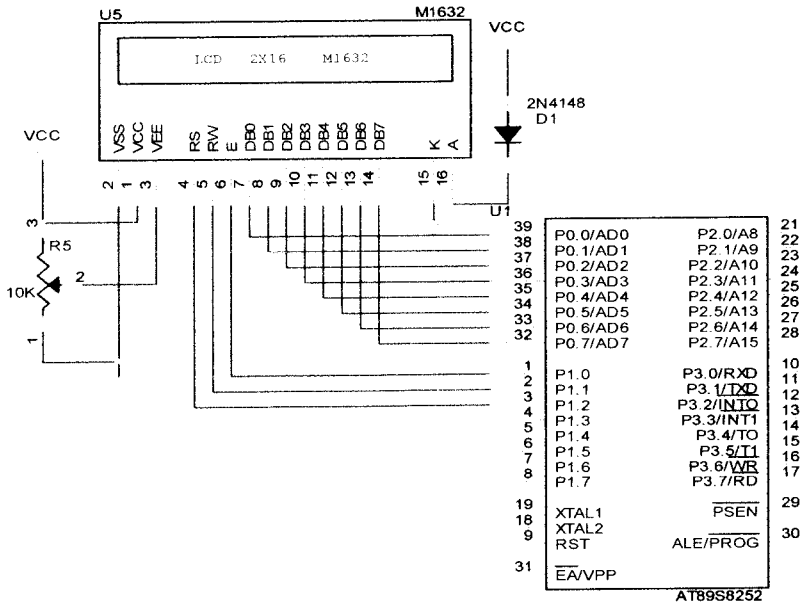
2. Untuk proses memasukkan data tampilan :

   Data 8 bit ditempatkan pada saluran data DB_0 – DB_7, masukan sinyal RS diberi masukan logika tinggi untuk memilih register tampilan, kemudian masukan sinyal R/W diberi logika rendah untuk melaksanakan operasi penulisan terhadap modul. Selanjutnya masukan sinyal E diberi masukan perubahan dari logika tinggi ke rendah. Sinyal E diberi masukan perubahan dari logika tinggi ke rendah. Sinyal E ini berfungsi untuk memulai kerja sesuai perintah yang telah dituliskan pada modul. Adapun hubungan mikrokontroler dengan LCD ditunjukan pada Gambar 3.2.

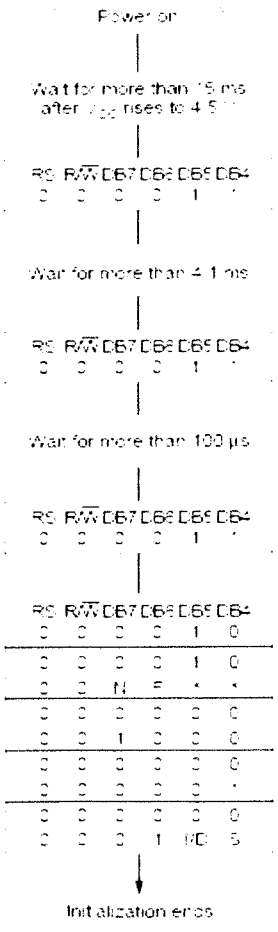
   Sedangkan alur inisialisasi penulisan LCD diperlihatkan pada Gambar 3.3. Sebagai contoh dalam penulisan LCD untuk memunculkan tulisan "Nanang Hidayat" dalam bahasa *assembly* adalah dengan cara :

```
rs      bit     p1.2
rw      bit     p1.1
e       bit     p1.0
atas    bit     24h
```

```
;++++++++++++++++++++++++++++++++++
;++++++ Insisalisasi LCD ++++++++++
;++++++++++++++++++++++++++++++++++

        lcall   tunda
        clr     rs
        mov     a,#00110000b
        lcall   tulis
        lcall   tunda
        mov     a,#00110000b
        lcall   tulis
        lcall   tunda
        mov     a,#00110000b
        lcall   tulis
        lcall   tunda

        mov     a,#00111111b    ; function set
        lcall   tulis
        mov     a,#00000001b    ; display clear
        lcall   tulis
        mov     a,#00000010b    ; return home position
        lcall   tulis
        mov     a,#00001100b    ; display off
        lcall   tulis
        mov     a,#00000110b    ; entry mode set

        clr     rs
        mov     a,#80h
        acall   tulis
        setb    rs
        mov     dptr,#good
        mov     a,#00h
        mov     atas,#16
rep:    clr     a
        movc    a,@a+dptr
        lcall   tulis
        inc     dptr
        djnz    atas,rep

tulis:  clr     rw      ;sub rutin untuk program tulis
        setb    e
        mov     p0,a
        acall   tunda
        clr     e
        setb    rw
        ret


tunda:  mov     20h,#0ffh
ul1:    mov     21h,#0ffh
ul2:    nop
        djnz    21h,ul2
        djnz    20h,ul1
        ret

good:   db      'NANANG HIDAYAT'
end.
```



Gambar 3.2. Hubungan mikrokontoler dengan LCD.



Gambar 3.3. Alur inisialisasi LCD

3.2.2. Rangkaian *input push button.*

Pada rangkaian ini tombol *push button* yang digunakan terdiri atas tombol *GOOD* dan tombol *NOT GOOD*. Tombol *GOOD* terhubung pada port P3.0 rangkaian *Slave* Mikrokontroler sedangkan tombol *NOT GOOD* terhubung pada port P3.1 rangkaian *Slave* Mikrokontroler. Tombol ini bersifat *active low* dimana ketika berlogika 0 maka tombol akan aktif dan ketika berlogika 1 maka tombol tidak aktif.

3.2.3. Rangkaian *slave* mikokontroler.

Rangakian ini adalah rangkaian mikrokontroler dengan input berupa tombol dan output berupa tampilan dot matriks (LCD). Cara kerja rangkaian ini adalah ketika salah satu tombol *GOOD* atau *NOT GOOD* ditekan maka program akan menjalankan perintah penambahan. Data penambahan selanjutnya ditampilkan pada rangkaian LCD dimana baris pertama untuk data *GOOD* dan baris kedua untuk data *NOT GOOD*. Selanjutnya data-data tersebut dikirimkan ke komputer melalui rangkaian *Master* Mikrokontroler. Proses ini berjalan berulang-ulang ketika ada aktifitas tombol ditekan. Adapun rangkaian skematik dari *Slave* Mikrokontroler ditunjukan pada Gambar 3.4.

Sedangkan program utama *slave* mikrokontroler dalam bahasa *assembler* adalah sebagai berikut :

```
START:
 MOV   DATABUS, #0FFH
 MOV   IE, #0
 CLR   LCDCE
 MOV   SP, #STACKHERE
 MOV   R0, #GD
 MOV   A, #0
 MOV   B, #4
 CALL  STRFILL
 MOV   CHGFLAG, #1
 CALL  LCDINIT
 MOV   A, #0
 MOV   DPTR, #LCDMESSAGE1
 CALL  LCDPRINT
 MOV   A, #40H
 MOV   DPTR, #LCDMESSAGE2
```

```
 CALL   LCDPRINT
 MOV    B, #10H
 CALL   DELAYLONG2
 SJMP   SENDQUALITYDATA

REPEATPROCESS:
 MOV    CHGFLAG, #0
 JNB    GDBTN, GDBUTTON
 JNB    NGBTN, NGBUTTON
 SJMP   SENDQUALITYDATA

GDBUTTON:
 MOV    R0, #GD
 SJMP   BUTTONPRESSED

NGBUTTON:
 MOV    R0, #NG

BUTTONPRESSED:
 MOV    B, #10H
 CALL   DELAYLONG
 JNB    GDBTN, $
 JNB    NGBTN, $
 MOV    CHGFLAG, #1
 CALL   INCWORD

SENDQUALITYDATA:
 CALL   SENDQLTDATA

SHOWQLTDATA:
 MOV    A, CHGFLAG
 JZ     LCDBUFFERREADY
 MOV    R1, #GD
 MOV    R0, #TMPDWORD
 MOV    B, #4
 CALL   STRCPY
 MOV    R0, #LCDR1+11
 MOV    R1, #TMPDWORD+0
 MOV    B, #5
 CALL   DCMLWORD
 MOV    R0, #LCDR2+11
 MOV    R1, #TMPDWORD+2
 MOV    B, #5
 CALL   DCMLWORD

LCDBUFFERREADY:
 CALL   LCDRFSH
 MOV    B, #20H
 CALL   DELAYLONG
 SJMP   REPEATPROCESS
 END
```

3.2.4. Rangkaian *master* mikrokontroler.

Rangkaian ini adalah rangkaian mikrokontroler dengan ditambah rangkaian komukasi serial dengan menggunakan IC MAX-232. Cara kerja rangkaian ini adalah ketika mendapatkan data dari salah satu *Slave* Mirkokontroler maka rangkaian *master* akan menggirimkan data tersebut menuju komputer untuk ditampilkan pada perangkat lunak Visual Basic. Pada perangkat lunak *visual basic* sendiri data-data *GOOD* dan

NOT GOOD akan ditampilkan secara langsung dan berulang-ulang. Adapun rangkaian skematik dari *Master* Mikrokontroler ditunjukan pada Gambar 3.5.

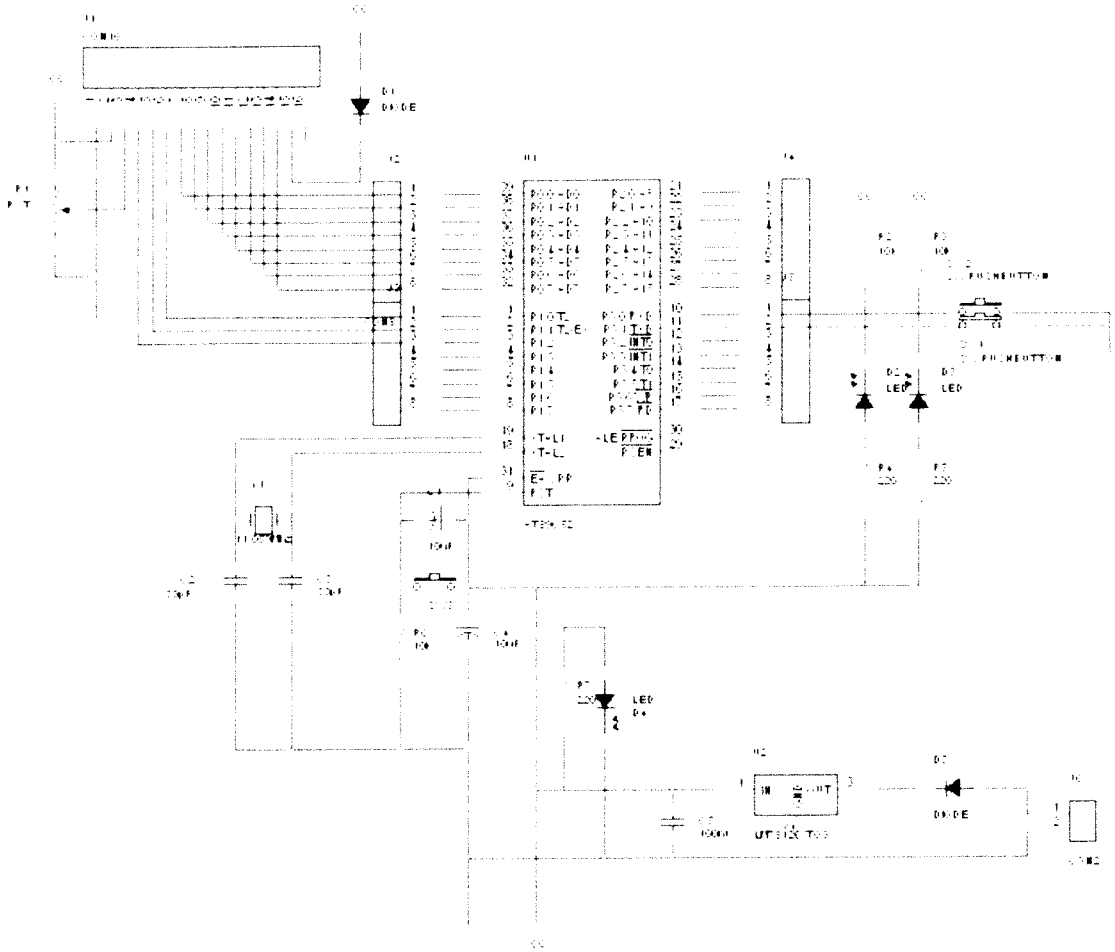
Sedangkan program utama *master* mikrokontroler dalam bahasa *assembler* adalah sebagai berikut :

```
START:
 MOV   IE, #0
 MOV   SP, #STACKHERE
 MOV   R0, #GD1
 MOV   A, #0
 MOV   B, #12
 CALL  STRFILL
 MOV   SLVNBR, #0
 MOV   TMOD, #20H
 MOV   TH1, #0F4H
 MOV   SCON, #50H
 CALL  DELAY256
 SETB  TR1
 SETB  EA
 SETB  TI


REPEATPROCESS:
 MOV   B, #10H

REPRCVSLVDATA:
 CALL  RECEIVESLAVEDATA
 JNC   SENDCRTSLVDATA
 DJNZ  B, REPRCVSLVDATA

SENDCRTSLVDATA:
 CALL  SENDQUALITYDATA
 MOV   B, #8
 CALL  DELAYLONG

QUALITYDATASENT:
 INC   SLVNBR
 MOV   A, SLVNBR
 CLR   C
 SUBB  A, #3
 JC    SLVNBRINCREMENTED
 MOV   SLVNBR, #0

SLVNBRINCREMENTED:
 SJMP  REPEATPROCESS
 END
```

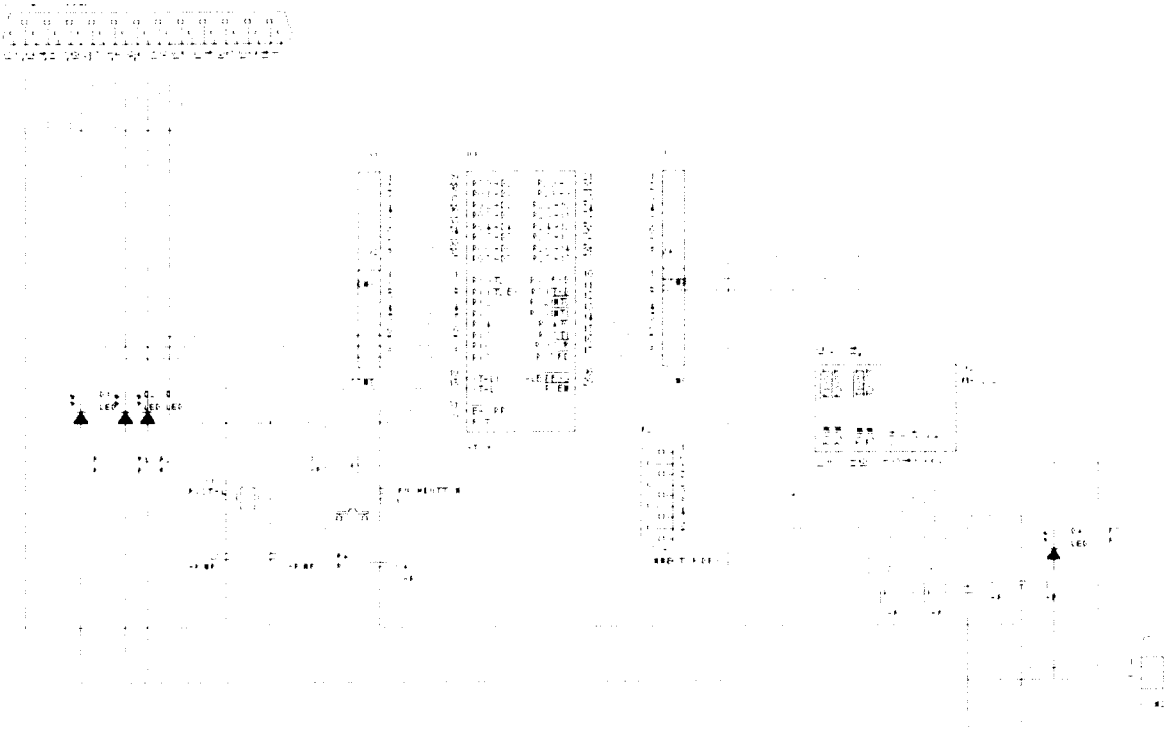
3.2.5. Rangkaian komunikasi serial MAX-232.

IC MAX-232 ini digunakan sebagai konektor komunikasi data serial dari komputer ke *Master* Mikrokontroler dan sebaliknya. Pada saat komputer megeluarkan data/sinyal melalui fasilitas keluaran serialnya (port serial), maka RS-232 ini akan mengirimkan sinyal lewat pin TXD (pin kirim data) ke mikrokontroler. Sedangkan

bila ada masukan dari mikrokontroler maka sinyal akan diterima oleh pin RXD (pin terima data) dan akan dikirimkan ke komputer.



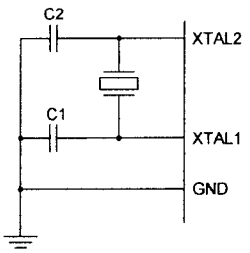
Gambar 3.4. Rangkaian *slave* mikrokontroler.



Gambar 3.5. Rangkaian *master* mikrokontroler.

3.2.6. Rangkaian mikrokontroler AT89S52

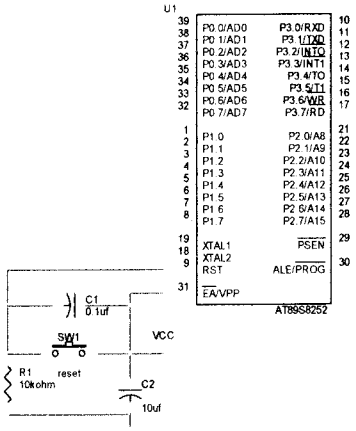
Pada rangkaian Mikrokontroler AT89S52 terdapat rangkaian osilator. Rangkaian osilator seperti yang diperlihatkan pada Gambar 3.6 menggunakan kristal frekuensi sebesar 11,059 MHz



Gambar 3.6. Rangkaian osilator

Selain itu juga terdapat rangaian reset. Mikrokontroler direset pada transisi tegangan rendah ke tegangan tinggi dan mengeksekusi program pada saat reset (RST) dalam keadaan logika rendah. Oleh karena itu pada pin RST dipasang kapasitor yang terhubung ke VCC dan resistor ke ground yang akan menjaga RST bernilai 1 pada

saat pengisian kapasitor dan akan kembali sesaat kemudian, dengan demikian mikrokontroler akan direset setiap kali diberi catu daya. Adapun rangkaian reset ditunjukan pada Gambar 3.7 :



Gambar 3.7. Rangkaian reset.

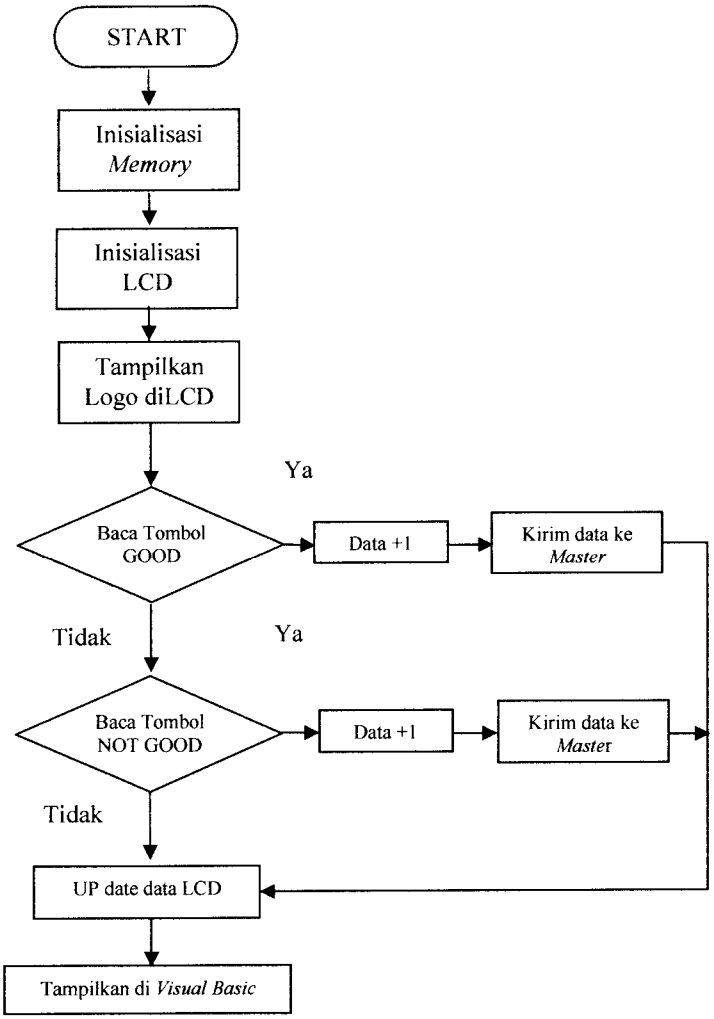
3.3. Alur Kerja Rangkaian.

Alur kerja rangkaian dapat dilihat pada Gambar 3.8.

Penjelasan *flow chart* proses kerja rangkaian adalah sebagai berikut :

1. Pertama kali alat dihidupkan.
2. Setelah itu dilanjutkan dengan proses inisialisasi *memory*.
3. Setelah itu dilanjutkan dengan proses inisialisasi LCD.
4. Proses selanjutnya adalah menampilkan logo pada LCD.
5. Proses dilanjutkan dengan pembacaan tombol *GOOD* dan *NOT GOOD*.
6. Apabila ada masukan maka dilakukan proses *counter GOOD* +1 dan *counter NOT GOOD* +1.
7. Setelah ada masukan tiap-tiap tombol maka proses selanjutnya adalah pengiriman data.

8. Data yang dikirim melalui port serial menuju komputer. Data yang didapat oleh komputer lalu ditampilkan pada program *quality monitor*.



Gambar 3.8. Alur kerja rangkaian.

BAB IV

UJI ALAT, ANALISA DAN PEMBAHASAN

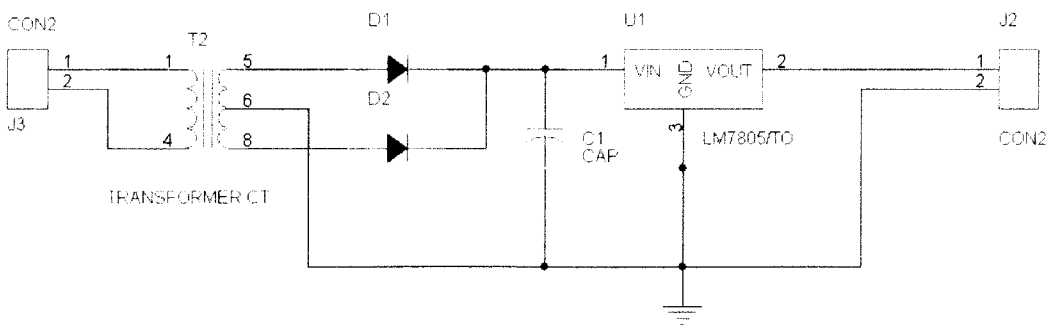
4.1. Uji Alat.

Pengujian alat merupakan bagian terpenting dalam penyusunan laporan ini. Dalam melakukan pengujian dimaksudkan untuk mendapatkan hasil perancangan dan implementasi, dengan demikian dapat diketahui sejauh mana alat dapat bekerja. Dengan pengamatan pula dapat diambil suatu analisa dan kesimpulan dalam keseluruhan alat yang dibuat.

4.2. Analisa Rangkaian.

4.2.1. Analisa rangkaian catu daya.

Rangkaian catu daya digunakan untuk memberikan daya kepada tiap-tiap rangkaian mikrokontroler. Tegangan yang dikeluarkan perlu dilakukan pengukuran untuk mengetahui apakah sudah sesuai, karena apabila tidak sesuai dimungkinkan alat tidak dapat bekerja. Rangakaian catu daya yang digunakan menghasilkan keluaran sebesar 5 Volt DC. Adapun rangkaian catu daya ditunjukan pada Gambar 4.1.



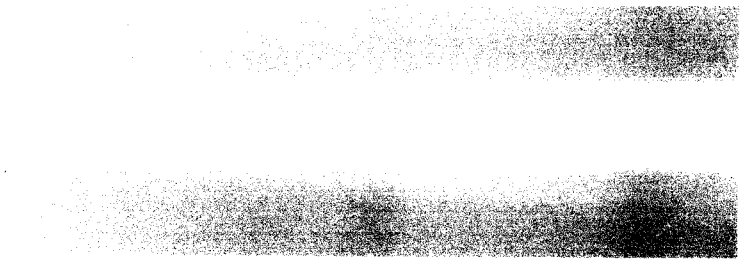
Gambar 4.1. Rangkaian catu daya.

Pada pengujian yang dilakukan dengan menggunakan *osciloscop* dapat diketahui bahwa gelombang yang dihasilkan setelah melewati trafo adalah gelombang sinus. Arus yang keluar dari trafo masih berupa arus AC oleh karena itu diperlukan dioda yang berfungsi sebagai penyearah dari arus AC menjadi arus DC. Setelah diberi dioda gelombang yang dihasilkan adalah gelombang segitiga dikarenakan terjadi *ripple*.

Dan untuk mengatasi *ripple* yang terjadi maka diperlukan kapasitor. Kapasitor disini berfungsi sebagai penapis atau mengurangi *ripple* yang terjadi. Semakin besar kapasitor yang diberikan maka *ripple* yang tejadi semakin kecil. Seperti yang terlihat pada Gambar 4.2. :

Gambar 4.2. Gelombang sebelum terregulasi

Adapun besar arus setelah melewati regulator 7805 adalah sebesar 5 volt DC. Bentuk gelombang setelah melewati regulator 7805 berupa garis lurus. Seperti yang terlihat pada Gambar 4.3 :



Gambar 4.3. Gelombang terregulasi

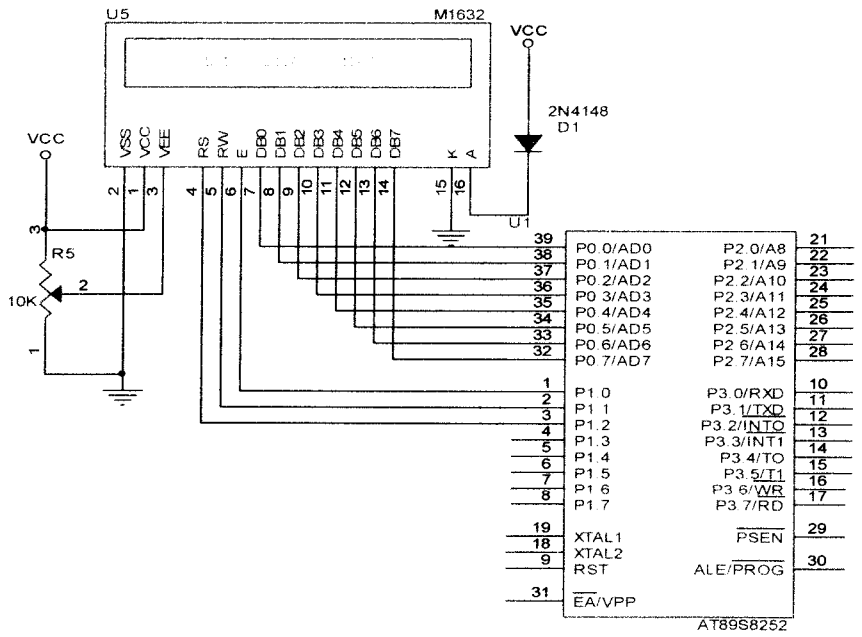
4.2.2. Analisa rangkaian mikrokontroler AT89S52.

Mikrokontroler merupakan pemroses data utama dalam perancangan ini. Pengujian rangkaian mikrokontroler dilakukan dengan cara menghubungkan keempat portnya ke rangkaian led dan men-*download* program sederhana untuk menyalakan led tersebut.

Dari hasil pengujian, led pada semua port menyala sesuai program, sehingga rangkaian mikrokontroler ini dapat digunakan sebagai pemroses utama.

4.2.3. Analisa rangkaian LCD (*liquid crystal display*).

Rangkaian LCD digunakan untuk menampilkan data hasil *counter Good* dan *counter Not Good.* Pengujian dilakukan dengan menampilkan data pada LCD, dan hasilnya dilihat langsung. Dari hasil pengamatan, LCD bisa menampilkan karakter dengan baik. Adapun rangkaian LCD ditunjukan pada Gambar 4.4.

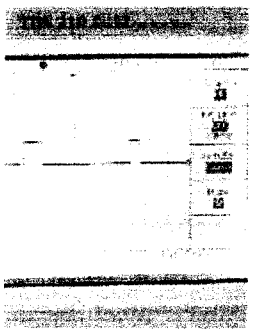


Gambar 4.4. Rangkaian LCD.

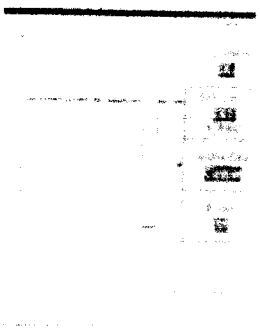
4.2.4. Analisa rangkaian komunikasi serial (MAX-232).

Rangkaian komunikasi serial MAX-232 digunakan untuk *interfacing* perangkat keras dan perangkat lunak yang tersambung dengan komputer. Rangkaian ini sebenarnya adalah sebuah IC dengan kaki masukan sebanyak 16. IC MAX-232 ini digunakan sebagai konektor komunikasi data serial dari komputer ke Master Mikrokontroler dan sebaliknya. Pada saat komputer megeluarkan data/sinyal melalui fasilitas keluaran serialnya (port serial), maka RS-232 ini akan mengirimkan sinyal lewat pin TXD (pin kirim data) ke mikrokontroler. Sedangkan bila ada masukan dari mikrokontroler maka sinyal akan diterima oleh pin RXD (pin terima data) dan akan dikirimkan ke komputer.

Adapun gambar gelombang yang terjadi pada saat komunikasi serial, mengirim dan menerima data ditunjukan pada Gambar 4.5 dan Gambar 4.6 :



Gambar 4.5 Gelombang pada saat mengirim data.



Gambar 4.6 Gelombang pada saat menerima data.

4.2.5. Analisa rangkaian tombol.

Rangkaian tombol digunakan sebagai input. Dimana ketika tombol ditekan maka sistem akan berjalan, kondisi tombol adalah *active low* dimana tombol akan terpicu ketika mendapat masukan nol, sedangkan bila masukan masih tinggi atau 1 maka tombol tidak akan bereaksi.

Adapun logika keadaan tombol ditunjukkan pada Tabel 4.1.

Tabel 4.1. Logika keadaan tombol.

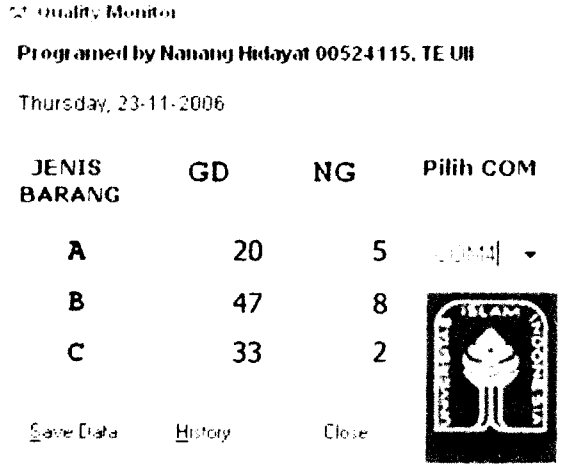
Logika keadaan	Status tombol	Keterangan
0	Aktif	Data yang ada akan ditambah dengan 1
1	Tidak aktif	Tidak melakukan instruksi apa-apa.

4.3.

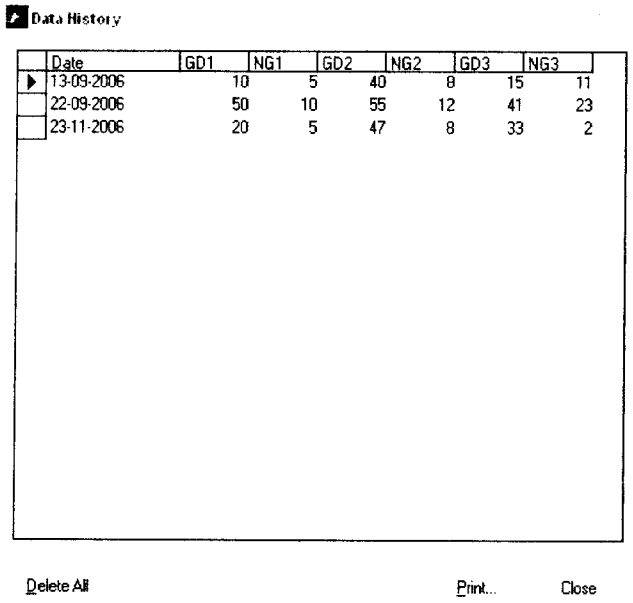
4.4. Analisa perangkat lunak.

Untuk pengujian kinerja perangkat lunak, dilakukan dengan memberikan masukan pada masing-masing *slave sitem*. Yaitu dengan menekan salah satu tombol *Good* atau *Not Good*. Setelah ada masukan maka pada perangkat lunak yang dibuat akan tertera hasil yang telah dilakukan. Adapun hasil yang ditampilkan dapat dilihat pada Gambar 4.7. :

Selanjutnya apabila telah selesai melakukan proses perhitungan maka selanjutnya data yang diperoleh dapat disimpan di program *database*. Fungsi dari *data base* adalah untuk menyimpan hasil-hasil yang telah dilakukan. Seperti yang terlihat pada Gambar 4.8. :



Gambar 4.7 Tampilan awal perangkat lunak



Gambar 4.8. Tampilan *database*

Setelah data disimpan maka selanjutnya diberikan pilihan untuk kembali ke proses awal atau dapat mencetak data yang ada. Adapun tampilan halaman yang siap dicetak ditunjukan pada Gambar 4.9.

Quality Data Report

Printed on Thursday, 23-11-2006

Tanggal	GD1	IIG1	GD2	IIG2	GD3	IIG3
13-09-2006	10	5	40	8	15	11
22-09-2006	50	10	55	12	41	23
23-11-2006	20	5	47	8	33	2

Programed by Nanang Hidayat 00524115, TE UII 3 record(s)

Gambar 4.9. Tampilan yang siap dicetak

BAB V

PENUTUP

5.1. Kesimpulan

Dari uraian yang telah disampaikan pada bab-bab sebelumnya maka dapat diambil kesimpulan sebagai berikut :

1. Mikrokontroler AT89S52 dapat digunakan sebagai alat inspeksi dalam dunia industri khususnya pada bagian produksi.
2. Rangkaian mikrokontroler AT89S52 digunakan sebagai *interface* dengan perangkat lunak yang terhubung pada unit komputer.
3. Perangkat *Visual Basic* dapat menampilkan data yang terjadi pada saat itu secara langsung.
4. Kelebihan dari alat kontrol kualitas produksi berbasis mikrokontroler adalah data yang didapat dapat dipantau secara *real time* dan adanya efisensi waktu dan tenaga yang dibutuhkan.
5. Alat kontrol kualitas produksi dapat berjalan dengan baik, dapat dilihat pada Gambar 4.7.

5.2. Saran

Penelitian ini dapat dikembangkan lagi untuk mencapai hasil yang lebih baik. Beberapa pengembangan yang dimungkinkan adalah:

1. Alat inspeksi berbasis mikrokontroler pada unit produksi ini dapat dikembangkan dengan menambah beberapa rangkaian s*lave* mikrokontroler sehingga terbentuk suatu rangkaian *multi processor.*
2. Input manual berupa tombol dapat dikembangkan dengan input berupa sensor optik.

DAFTAR PUSTAKA

1. Budiharti, Widodo, Firmansyah, Sigit 2005, *Elekronika Digital dan Mikroprosesor,* Andi, Yogyakarta.
2. Malviono, Albert Paul. 2003, *Prinsip-Prinsip Elektronika,* Terjemahan Alb Joko Santoso, Salemba Teknika, Jakarta.
3. Putra Agfianto Eko. 2002, *Teknik Antarmuka Komputer: konsep dan Aplikasi,* Graha Ilmu, Yogyakarta.
4. Prasetia, Retna, Widodo, Catur Edi.2004, *Interfacing Port Pararel dan Port Serial Komputer dengan Visual Basic 6.0,* Andi, Yogyakarta.

www.atmel.com

www.datasheetarhive.com

www.ilmukomputer.com

www.vb.net

LAMPIRAN

```
;===============================================================
;  Title       : QUALITY CONTROL, MASTER
;  Name        : NANANG, 08122862521
;  Auhor       : Nanang Hidayat
;  Version     : 1.0
;===============================================================

DSEG AT 08
GD1:            DS 2
NG1:            DS 2
GD2:            DS 2
NG2:            DS 2
GD3:            DS 2
NG3:            DS 2
PORTBUFFER:     DS 4
READBUFFER:     DS 4
SENDBUFFER:     DS 5
SLVNBR:                 DS 1
QLTNBR:                 DS 1
STACKHERE:      DS 1

CSEG
ORG     00h
LJMP    START

ORG     40H
DELAY256:
 PUSH   B
 MOV    B, #0
 DJNZ   B, $
 POP    B
 RET

DELAYLONG:
 ACALL DELAY256
 DJNZ   B, DELAYLONG
 RET

DELAYLONG2:
 PUSH   B
 MOV    B, #0
 ACALL DELAYLONG
 POP    B
 DJNZ   B, DELAYLONG2
 RET

STRCPY:
 PUSH   ACC
 PUSH   0
 PUSH   1

STRCPYCRT:
 MOV    A, @R1
 MOV    @R0, A
 INC    0
 INC    1
 DJNZ   B, STRCPYCRT
 POP    1
 POP    0
 POP    ACC
 RET

STRFILL:

 PUSH   B
 PUSH   0
```

```
STRFILLREP:
 MOV   @R0, A
 INC   0
 DJNZ  B, STRFILLREP
 POP   0
 POP   B
 RET

SERIALOUT:
 JNB   TI, $
 MOV   SBUF, A
 CLR   TI
 RET

READPORTS:
 MOV   PORTBUFFER+0, P0
 MOV   PORTBUFFER+1, P1
 MOV   PORTBUFFER+2, P2
 MOV   PORTBUFFER+3, P3
 RET

WAITSLVSTART:
 PUSH  B
 MOV   B, #0

WSLVB7HI:
 CALL  READPORTS
 MOV   A, @R1
 JB    ACC.7, SLVB7HIOK
 DJNZ  B, WSLVB7HI
 SJMP  SLVB7TO

SLVB7HIOK:
 MOV   B, #0

WSLVB7LO:
 CALL  READPORTS
 MOV   A, @R1
 JNB   ACC.7, SLVB7LOOK
 DJNZ  B, SLVB7LOOK

SLVB7TO:
 SETB  C
 SJMP  SLVSTARTOK

SLVB7LOOK:
 CLR   C

SLVSTARTOK:
 POP   B
 RET

READSLVHALF:
 PUSH  B
 PUSH  2
 MOV   B, #0

WSLVB6LO:
 CALL  READPORTS
 MOV   A, @R1
 JNB   ACC.6, SLVB6LOOK
 DJNZ  B, WSLVB6LO
 SJMP  SLVB6TO

SLVB6LOOK:
```

```
 MOV   R2, A
 MOV   B, #0

WSLVB6HI:
 CALL  READPORTS
 MOV   A, @R1
 JB    ACC.6, SLVB6HIOK
 DJNZ  B, WSLVB6HI

SLVB6TO:
 SETB  C
 SJMP  SLVHLFREAD

SLVB6HIOK:
 MOV   A, R2
 ANL   A, #0FH
 CLR   C

SLVHLFREAD:
 POP   2
 POP   B
 RET

RECEIVESLAVEDATA:
 PUSH  ACC
 PUSH  B
 PUSH  0
 PUSH  1
 MOV   R0, #READBUFFER
 MOV   A, #PORTBUFFER
 ADD   A, SLVNBR
 MOV   R1, A
 MOV   B, #10H

REPWAITSLVSTART:
 CALL  WAITSLVSTART
 JNC   SLVSTROBERCVD
 DJNZ  B, REPWAITSLVSTART
 SJMP  SLVDATARCVD

SLVSTROBERCVD:
 MOV   B, #4

READSLVBYTE:
 PUSH  B
 CALL  READSLVHALF
 JC    SLVBYTETO
 MOV   B, A
 CALL  READSLVHALF
 JC    SLVBYTETO
 SWAP  A
 ORL   A, B

SLVBYTETO:
 POP   B
 JC    SLVDATARCVD
 MOV   @R0, A
 INC   R0
 DJNZ  B, READSLVBYTE
 MOV   R0, #GD1
 MOV   A, SLVNBR
 RL    A
 RL    A
 ADD   A, R0
 MOV   R0, A
 MOV   R1, #READBUFFER
```

```
 MOV   B, #4
 CALL  STRCPY
 CLR   C

SLVDATARCVD:
 POP   1
 POP   0
 POP   B
 POP   ACC
 RET

SETPARITYBIT:
 PUSH  B
 PUSH  ACC
 MOV   B, #4
 CLR   C

SPCHKBYTELSB:
 JNB   ACC.0, SPPRTBITSET
 CPL   C

SPPRTBITSET:
 RR    A
 DJNZ  B, SPCHKBYTELSB
 POP   ACC
 MOV   ACC.4, C
 POP   B
 RET

SENDQUALITYDATA:
 PUSH  ACC
 PUSH  B
 PUSH  0
 PUSH  1
 MOV   QLTNBR, #0

FMTSLAVENUMBER:
 MOV   R0, #SENDBUFFER
 MOV   A, SLVNBR
 ANL   A, #3
 ORL   A, #80H
 ORL   A, QLTNBR
 MOV   @R0, A
 INC   R0
 MOV   A, QLTNBR
 JNZ   SQDSELECTNGDATA

SQDSELECTGDDATA:
 MOV   R1, #GD1
 SJMP  SQDDATASETSELECTED

SQDSELECTNGDATA:
 MOV   R1, #NG1

SQDDATASETSELECTED:
 MOV   A, SLVNBR
 RL    A
 RL    A
 ADD   A, R1
 MOV   R1, A
 MOV   B, #2

FMTQDBYTES:
 PUSH  B
 MOV   A, @R1
 PUSH  ACC
```

```
  ANL   A, #0FH
  CALL  SETPARITYBIT
  MOV   @R0, A
  POP   ACC
  INC   R0
  SWAP  A
  ANL   A, #0FH
  CALL  SETPARITYBIT
  MOV   @R0, A
  INC   R0
  INC   R1
  POP   B
  DJNZ  B, FMTQDBYTES
  MOV   B, #5
  MOV   R0, #SENDBUFFER

SQDSENDEACHBYTE:
  MOV   A, @R0
  CALL  SERIALOUT
  INC   R0
  DJNZ  B, SQDSENDEACHBYTE
  MOV   A, QLTNBR
  XRL   A, #40H
  MOV   QLTNBR, A
  JNZ   FMTSLAVENUMBER
  POP   1
  POP   0
  POP   B
  POP   ACC
  RET


;=========================================
;MAIN PROGRAM
;=========================================

START:
  MOV   IE, #0
  MOV   SP, #STACKHERE
  MOV   R0, #GD1
  MOV   A, #0
  MOV   B, #12
  CALL  STRFILL
  MOV   SLVNBR, #0
  MOV   TMOD, #20H
  MOV   TH1, #0F4H
  MOV   SCON, #50H
  CALL  DELAY256
  SETB  TR1
  SETB  EA
  SETB  TI


REPEATPROCESS:
  MOV   B, #10H

REPRCVSLVDATA:
  CALL  RECEIVESLAVEDATA
  JNC   SENDCRTSLVDATA
  DJNZ  B, REPRCVSLVDATA

SENDCRTSLVDATA:
  CALL  SENDQUALITYDATA
  MOV   B, #8
  CALL  DELAYLONG
```

```
QUALITYDATASENT:
 INC    SLVNBR
 MOV    A, SLVNBR
 CLR    C
 SUBB   A, #3
 JC     SLVNBRINCREMENTED
 MOV    SLVNBR, #0

SLVNBRINCREMENTED:
 SJMP   REPEATPROCESS
 END
```

```
;===============================================================
;  Title      : QUALITY CONTROL SLAVE
;  Name       : NANANG, 08122862521
;  Auhor      : Nanang Hidayat
;  Version    : 1.0
;===============================================================

LCDDP           EQU P0
LCDRS           BIT P1.2
LCDRW           BIT P1.1
LCDCE           BIT P1.0
GDBTN           BIT P3.0
NGBTN           BIT P3.1
DATABUS         EQU P2

DSEG AT 08
LCDR1:          DS 10H
LCDR2:          DS 10H
GD:             DS 2
NG:             DS 2
TMPDWORD:       DS 4
CHGFLAG:        DS 1
STACKHERE:      DS 1

CSEG
ORG     00h
LJMP    START

ORG     40H
LCDINITDATA:    DB 38h, 06h, 0Ch
LCDMESSAGE1:    DB 'QUALITY  CONTROL'
LCDMESSAGE2:    DB 'NANANG, UII 2000'
LCDFMT1:        DB 'GOOD    :       '
LCDFMT2:        DB 'NOT GOOD:       '
;               0123456789ABCDEF

DELAY256:
 PUSH   B
 MOV    B, #0
 DJNZ   B, $
 POP    B
 RET

DELAYLONG:
 ACALL DELAY256
 DJNZ   B, DELAYLONG
 RET

DELAYLONG2:
 PUSH   B
 MOV    B, #0
 ACALL DELAYLONG
 POP    B
 DJNZ   B, DELAYLONG2
 RET

STRCPY:
 PUSH   ACC
 PUSH   0
 PUSH   1

STRCPYCRT:
 MOV    A, @R1
 MOV    @R0, A
 INC    0
 INC    1
```

```
 DJNZ   B, STRCPYCRT
 POP    1
 POP    0
 POP    ACC
 RET

STRCPYX:
 PUSH   ACC
 PUSH   DPH
 PUSH   DPL
 PUSH   0

STRCPYXCRT:
 MOV    A, #0
 MOVC   A, @A+DPTR
 MOV    @R0, A
 INC    0
 INC    DPTR
 DJNZ   B, STRCPYXCRT
 POP    0
 POP    DPL
 POP    DPH
 POP    ACC
 RET

STRFILL:

 PUSH   B
 PUSH   0

STRFILLREP:
 MOV    @R0, A
 INC    0
 DJNZ   B, STRFILLREP
 POP    0
 POP    B
 RET

STRHALF:
 PUSH   B
 ANL    A, #0FH
 ADD    A, #30H
 MOV    B, A
 SUBB   A, #3AH
 MOV    A, B
 JC     STROKAY
 ADD    A, #07H

STROKAY:
 POP    B
 RET

STRBYTE:
 MOV    B, A
 ACALL STRHALF
 XCH    A, B
 SWAP   A
 ACALL STRHALF
 XCH    A, B
 RET

STRWORD:
 PUSH   ACC
 PUSH   B
 PUSH   0
 PUSH   B
```

```
  CALL   STRBYTE
  MOV    @R0, A
  DEC    R0
  MOV    @R0, B
  DEC    R0
  POP    B
  XCH    A, B
  CALL   STRBYTE
  MOV    @R0, A
  DEC    R0
  MOV    @R0, B
  POP    0
  POP    B
  POP    ACC
  RET

DIVTENBYTE:

  PUSH   0
  PUSH   ACC
  SWAP   A
  ANL    A, #0FH
  XCH    A, B
  ANL    A, #0FH
  SWAP   A
  ORL    A, B
  MOV    B, #10
  DIV    AB
  SWAP   A
  MOV    R0, A
  POP    ACC
  ANL    A, #0FH
  XCH    A, B
  ANL    A, #0FH
  SWAP   A
  ORL    A, B
  MOV    B, #10
  DIV    AB
  ORL    A, R0
  POP    0
  RET

DIVTENWORD:
  PUSH   ACC
  PUSH   0
  INC    R0
  MOV    A, @R0
  MOV    B, #0
  CALL   DIVTENBYTE
  MOV    @R0, A
  DEC    R0
  MOV    A, @R0
  CALL   DIVTENBYTE
  MOV    @R0, A
  POP    0
  POP    ACC
  RET

DCMLBYTE:
  PUSH   ACC
  PUSH   B
  PUSH   0
  PUSH   ACC
  MOV    A, R0
  ADD    A, B
  MOV    R0, A
```

```
  POP    ACC

DCMLBYTE0:
  PUSH   B
  MOV    B, #0
  CALL   DIVTENBYTE
  XCH    A, B
  CALL   STRHALF
  MOV    @R0, A
  XCH    A, B
  POP    B
  DEC    R0
  DJNZ   B, DCMLBYTE0
  POP    0
  POP    B
  POP    ACC
  RET

DCMLWORD:
  PUSH   ACC
  PUSH   B
  PUSH   0
  PUSH   1
  XCH    A, R0
  XCH    A, R1
  XCH    A, R0
  MOV    A, B
  DEC    A
  ADD    A, R1
  MOV    R1, A

DCMLWORD0:
  PUSH   B
  CALL   DIVTENWORD
  MOV    A, B
  CALL   STRHALF
  MOV    @R1, A
  POP    B
  DEC    R1
  DJNZ   B, DCMLWORD0
  POP    1
  POP    0
  POP    B
  POP    ACC
  RET

INCWORD:

  PUSH   ACC
  PUSH   0
  MOV    A, @R0
  ADD    A, #1
  MOV    @R0, A
  INC    R0
  MOV    A, @R0
  ADDC   A, #0
  MOV    @R0, A
  POP    0
  POP    ACC
  RET

LCDWAIT:
  PUSH   B
  MOV    B, #20H
  DJNZ   B, $
  POP    B
```

```
  RET

LCDWRCODE:
  CLR    LCDRS
  SJMP   LCDWRITE

LCDWRDATA:
  SETB   LCDRS

LCDWRITE:
  CLR    LCDRW
  SETB   LCDCE
  MOV    LCDDP, A
  CLR    LCDCE
  RET

LCDSETDDR:
  ORL    A, #80H
  ACALL LCDWRCODE
  ACALL LCDWAIT
  RET

LCDRSTBUFFER:
  PUSH   ACC
  PUSH   B
  PUSH   0
  PUSH   1
  MOV    R0, #LCDR1              ;format 1st row buffer
  MOV    DPTR, #LCDFMT1
  MOV    B, #16
  ACALL STRCPYX
  MOV    R0, #LCDR2              ;format 2nd row buffer
  MOV    DPTR, #LCDFMT2
  MOV    B, #16
  ACALL STRCPYX
  POP    1
  POP    0
  POP    B
  POP    ACC
  RET

LCDINIT:                         ;sub rutin inisialisasi LCD
  MOV    B, #3
  MOV    DPTR, #LCDINITDATA      ;send LCD init. data

LCDSENDCTRL:
  MOV    A, #0
  MOVC   A, @A+DPTR
  ACALL LCDWRCODE
  ACALL DELAY256
  INC    DPTR
  DJNZ   B, LCDSENDCTRL
  MOV    A, #01            ;clear display
  ACALL LCDWRCODE
  MOV    B, #0
  ACALL DELAYLONG
  ACALL LCDRSTBUFFER
  RET

LCDRFSH:
  PUSH   ACC
  PUSH   B
  PUSH   0
  MOV    R0, #LCDR1
  MOV    B, #2
  MOV    A, #0
```

```
LCDRFSH0:
  ACALL LCDSETDDR
  PUSH  B
  MOV   B, #10H

LCDRFSH1:
  MOV   A, @R0
  ACALL LCDWRDATA
  ACALL LCDWAIT
  INC   0
  DJNZ  B, LCDRFSH1
  POP   B
  MOV   A, #40H
  DJNZ  B, LCDRFSH0
  POP   0
  POP   B
  POP   ACC
  RET

LCDPRINT:
  PUSH  B
  ACALL LCDSETDDR
  MOV   B, #10H

LCDPRINT1:
  MOV   A, #0
  MOVC  A, @A+DPTR
  ACALL LCDWRDATA
  ACALL LCDWAIT
  INC   DPTR
  DJNZ  B, LCDPRINT1
  POP   B
  RET

SENDQLTHALF:
  PUSH  ACC
  ANL   A, #0FH
  MOV   DATABUS, A
  CALL  DELAY256
  SETB  DATABUS.6
  CALL  DELAY256
  POP   ACC
  RET

SENDQLTDATA:
  PUSH  ACC
  PUSH  B
  PUSH  0
  MOV   R0, #GD
  MOV   B, #4

SENDQLTBYTE:
  MOV   A, @R0
  CALL  SENDQLTHALF
  SWAP  A
  CALL  SENDQLTHALF
  INC   R0
  DJNZ  B, SENDQLTBYTE
  MOV   DATABUS, #0FFH
  MOV   B, #8
  CALL  DELAYLONG
  POP   0
  POP   B
  POP   ACC
  RET
```

```
;=========================================
;MAIN PROGRAM
;=========================================

START:
 MOV   DATABUS, #0FFH
 MOV   IE, #0
 CLR   LCDCE
 MOV   SP, #STACKHERE
 MOV   R0, #GD
 MOV   A, #0
 MOV   B, #4
 CALL  STRFILL
 MOV   CHGFLAG, #1
 CALL  LCDINIT
 MOV   A, #0
 MOV   DPTR, #LCDMESSAGE1
 CALL  LCDPRINT
 MOV   A, #40H
 MOV   DPTR, #LCDMESSAGE2
 CALL  LCDPRINT
 MOV   B, #10H
 CALL  DELAYLONG2
 SJMP  SENDQUALITYDATA

REPEATPROCESS:
 MOV   CHGFLAG, #0
 JNB   GDBTN, GDBUTTON
 JNB   NGBTN, NGBUTTON
 SJMP  SENDQUALITYDATA

GDBUTTON:
 MOV   R0, #GD
 SJMP  BUTTONPRESSED

NGBUTTON:
 MOV   R0, #NG

BUTTONPRESSED:
 MOV   B, #10H
 CALL  DELAYLONG
 JNB   GDBTN, $
 JNB   NGBTN, $
 MOV   CHGFLAG, #1
 CALL  INCWORD

SENDQUALITYDATA:
 CALL  SENDQLTDATA

SHOWQLTDATA:
 MOV   A, CHGFLAG
 JZ    LCDBUFFERREADY
 MOV   R1, #GD
 MOV   R0, #TMPDWORD
 MOV   B, #4
 CALL  STRCPY
 MOV   R0, #LCDR1+11
 MOV   R1, #TMPDWORD+0
 MOV   B, #5
 CALL  DCMLWORD
 MOV   R0, #LCDR2+11
 MOV   R1, #TMPDWORD+2
 MOV   B, #5
 CALL  DCMLWORD
```

```
MAIN PROGRAM :
'SG00.00NN 000P.0000 000P.0000 000P.0000 000P.0000
'S  : slave (1 = slave data, 0 = counter data)
'G  : 0 -> Not Good
'G  : 1 -> Good
'P  : parity bit (even parity)
'NN : Slave number

Dim DebugMode As Boolean, DbgFileTxt$, DbgFileNbr%
Dim DataValid As Boolean
Dim Serial%(0 To 4), RdFlag%(0 To 4), ByteNumber%
Dim tmpBuffer$
Private Sub COMOpen(ComNumber%)
  On Error GoTo ErrOpen
  If MSComm1.PortOpen Then MSComm1.PortOpen = False
  MSComm1.CommPort = ComNumber%
  MSComm1.Settings = "2400, N, 8, 1"
  MSComm1.InputMode = comInputModeText
  MSComm1.ParityReplace = ""
  MSComm1.Handshaking = comNone
  MSComm1.PortOpen = True
  ByteNumber = 0
ErrOpen:
End Sub
Private Sub COMClose()
  On Error GoTo COMCloseErr
  MSComm1.PortOpen = False
  Exit Sub

COMCloseErr:
End Sub
Private Sub ResetSerialBuffer()
  For i% = 0 To 4
    Serial(i%) = 0
    RdFlag(i%) = 0
  Next i%
End Sub
Private Sub ExtractReceivedData()
  'vld% = 0 --> data tidak valid
  'vld% = 1 --> data valid
  'qlt% = 0 --> GD data
  'qlt% = 1 --> NG data
  'qchi% = data kualitas, hi-byte
  'qclo% = data kualitas, lo-byte

  vld% = 1
  If (Serial(0) And &H80) = 0 Then vld% = 0
  If (Serial(0) And &H3C) <> 0 Then vld% = 0
  For i% = 0 To 4
    If RdFlag(i%) = 0 Then
      vld% = 0
      Exit For
    End If
  Next i%
  For i% = 1 To 4
    crtbyte% = Serial(i%)
    If (crtbyte% And &HE0) <> 0 Then
```

```
        vld% = 0
      Else
        crtprty% = crtbyte% And &H10
        crtbyte% = crtbyte% And 15
        ones% = 0
        For j% = 1 To 4
          If (crtbyte% And 1) <> 0 Then ones% = ones% + 1
          crtbyte% = crtbyte% \ 2
        Next j%
        If (crtprty% = 0) And ((ones% And 1) <> 0) Then vld% = 0
        If (crtprty% = 1) And ((ones% And 1) = 0) Then vld% = 0
      End If
      If vld% = 0 Then Exit For
    Next i%
    If (Serial(0) And &H40) = 0 Then
      qlt% = 0
    Else
      qlt% = 1
    End If
    slv% = Serial(0) And 3
    If vld% = 1 Then
      qclo% = 16 * (Serial(2) And 15) + (Serial(1) And 15)
      qchi% = 16 * (Serial(4) And 15) + (Serial(3) And 15)
      QCData(slv%, qlt%) = (256& * qchi% + qclo%) And &HFFFFFFFF
      DataValid = True
    Else
      DataValid = False
    End If
End Sub
Private Sub ShowQCData(SlaveNbr%)
    gdtxt$ = Format(QCData(SlaveNbr, 0), "#,##0")
    ngtxt$ = Format(QCData(SlaveNbr, 1), "#,##0")
    LblGd(SlaveNbr).Caption = gdtxt$
    LblNG(SlaveNbr).Caption = ngtxt$
End Sub
Private Sub SaveData()
    With RSQlt
      If .RecordCount > 0 Then
        .MoveFirst
        Do While Not .EOF
          If Format(RSQlt!tgl, "dd-mm-yyyy") = Format(Now, "dd-mm-yyyy")
Then
            datefound% = 1
            Exit Do
          End If
          .MoveNext
        Loop
      End If
      If datefound% = 0 Then .AddNew
      RSQlt!tgl = Now
      RSQlt!gd1 = QCData(0, 0)
      RSQlt!ng1 = QCData(0, 1)
      RSQlt!gd2 = QCData(1, 0)
      RSQlt!ng2 = QCData(1, 1)
      RSQlt!gd3 = QCData(2, 0)
      RSQlt!ng3 = QCData(2, 1)
      .UpdateBatch
```

```
    End With
  End Sub
  Private Sub tmrDate_Timer()
    LblDate.Caption = Format(Now, "dddd, dd-mm-yyyy")
  End Sub
  Private Sub MSComm1_OnComm()
    evtCom = MSComm1.CommEvent
    If (evtCom = comEvReceive) Then
      temp$ = MSComm1.Input
      ascx% = Asc(temp$)
      If DebugMode = True Then
        wrtxt$ = Hex(ascx%)
        wrtxt$ = String(2 - Len(wrtxt$), "0") + wrtxt$ + " "
        tmpBuffer = tmpBuffer + wrtxt$
        If Len(tmpBuffer) >= 512 Then
          Put #DbgFileNbr, , tmpBuffer
          tmpBuffer = ""
        End If
      End If
      If (ascx% And &H80) <> 0 Then
        ResetSerialBuffer
        ByteNumber = 0
      End If
      If ByteNumber <= 4 Then
        Serial(ByteNumber) = ascx%
        RdFlag(ByteNumber) = 1
      End If
      If ByteNumber >= 4 Then
        ExtractReceivedData
        If DataValid Then ShowQCData (Serial(0) And 3)
        ResetSerialBuffer
      Else
        ByteNumber = ByteNumber + 1
      End If
    End If
  End Sub
  Private Sub CmbCOM_Click()
    ResetSerialBuffer
    For i% = 0 To 2
      For j% = 0 To 1
        QCData(i%, j%) = 0
      Next j%
    Next i%
    For i% = 0 To 2
     ShowQCData i%
    Next i%
    COMOpen CmbCOM.ListIndex + 1
    If MSComm1.PortOpen Then
      MsgBox "Connected."
    Else
      MsgBox "Connection error."
    End If
  End Sub
  Private Sub CmbCOM_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
  End Sub
  Private Sub CmdSave_Click()
```

```
    SaveData
  End Sub
  Private Sub CmdHst_Click()
    FrmHst.Show 1
  End Sub
  Private Sub CmdClose_Click()
    Unload Me
  End Sub
  Private Sub Form_Load()
    DbgFileTxt = App.Path + "\Debug.txt"
    DbgFileNbr = FreeFile
    Open DbgFileTxt For Binary As #DbgFileNbr
    If LOF(DbgFileNbr) > 0 Then
      Close #DbgFileNbr
      Kill DbgFileTxt
      Open DbgFileTxt For Binary As #DbgFileNbr
    End If

    LblInfo.Caption = txtInfo
    With CmbCOM
      .Clear
      For i% = 1 To 6
        .AddItem "COM" + Trim(Str(i%))
      Next i%
      '.AddItem "COM1"
      '.AddItem "COM2"
      .ListIndex = 0
    End With
    dbOpen
  End Sub
  Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    If (Shift And vbCtrlMask) And (KeyCode = vbKeyD) Then DebugMode = Not
  DebugMode
    If DebugMode Then
      Me.Caption = "Quality Monitor (Debug Mode)"
    Else
      Me.Caption = "Quality Monitor"
    End If
  End Sub
  Private Sub Form_Unload(Cancel As Integer)
    Close #DbgFileNbr
    dbClose
    COMClose
  End Sub

  MOD CODE :

  Public Const txtInfo$ = "Programed by Nanang Hidayat 00524115, TE UII"
  Public dbFile$
  Public db As ADODB.Connection
  Public RSQlt As ADODB.Recordset
  Public QCData&(0 To 2, 0 To 1)
  Public Sub dbOpen()
    dbFile = App.Path + "\History.mdb"

    Set db = New ADODB.Connection
    Set RSQlt = New ADODB.Recordset
```

```
    db.CursorLocation = adUseClient
    db.Open "Provider=Microsoft.Jet.OLEDB.3.51;Data Source=" + dbFile
    RSQlt.Open "SELECT * FROM qlt ORDER BY tgl", db, adOpenStatic,
 adLockBatchOptimistic
 End Sub
 Public Sub dbClose()
    RSQlt.Close
    db.Close
    Set RSQlt = Nothing
    Set db = Nothing
 End Sub


 HISTORY CODE :
 Dim RSHist As ADODB.Recordset
 Private Sub ResetButtons()
    If RSHist.RecordCount > 0 Then
       CmdDel.Enabled = True
       CmdPrint.Enabled = True
    Else
       CmdDel.Enabled = False
       CmdPrint.Enabled = False
    End If
 End Sub
 Private Sub CmdDel_Click()
    If MsgBox("Delete all data?", vbYesNoCancel + vbQuestion, "Confirmation") =
vbYes Then
       db.Execute "delete * from qlt"
       RSQlt.Requery
       RSHist.Requery
       ResetButtons
    End If
 End Sub
 Private Sub CmdPrint_Click()
    Dim RSReport As ADODB.Recordset

    Set RSReport = RSHist.Clone
    Set DRHist.DataSource = RSReport
    DRHist.Sections(1).Controls(2).Caption = "Printed on " + Format(Now, "dddd,
dd-mm-yyyy")
    DRHist.Sections(5).Controls(2).Caption = txtInfo
    DRHist.Sections(5).Controls(3).Caption = Trim(RSReport.RecordCount) + "
record(s)"
    DRHist.Show 1
    RSReport.Close
    Set RSReport = Nothing
 End Sub
 Private Sub CmdClose_Click()
    Unload Me
 End Sub
 Private Sub Form_Load()
    Set RSHist = New ADODB.Recordset
    RSHist.Open "select * from qlt order by tgl", db, adOpenStatic,
adLockBatchOptimistic
    Set DGQ.DataSource = RSHist
    ResetButtons
 End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
   RSHist.Close
   Set RSHist = Nothing
End Sub
```

Features

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

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

AT89S52

1. Description

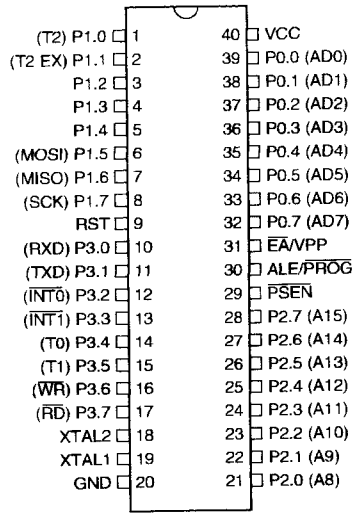
The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

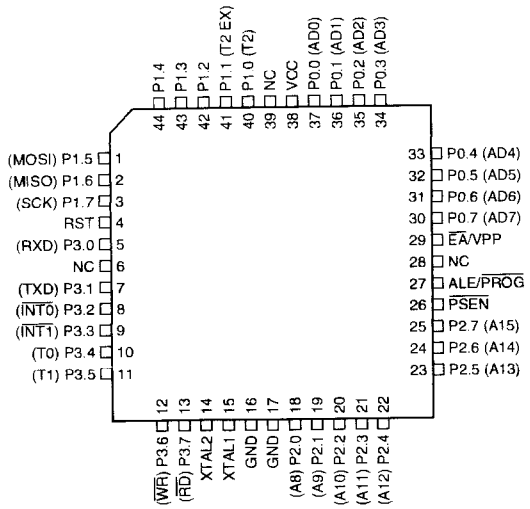
1919C–MICRO–3/05

2. Pin Configurations

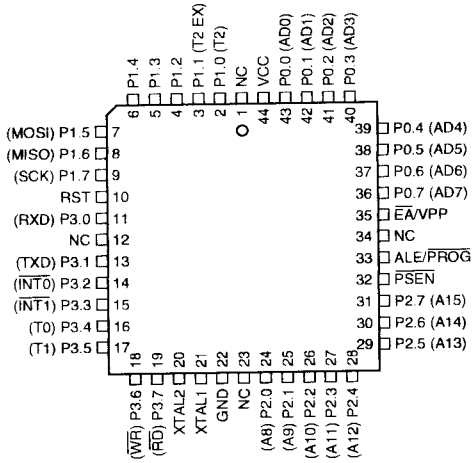
2.1 40-lead PDIP



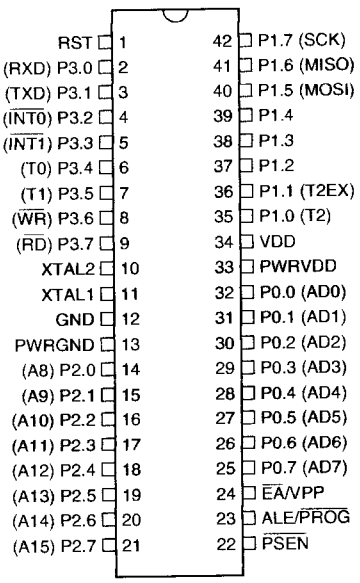
2.2 44-lead TQFP



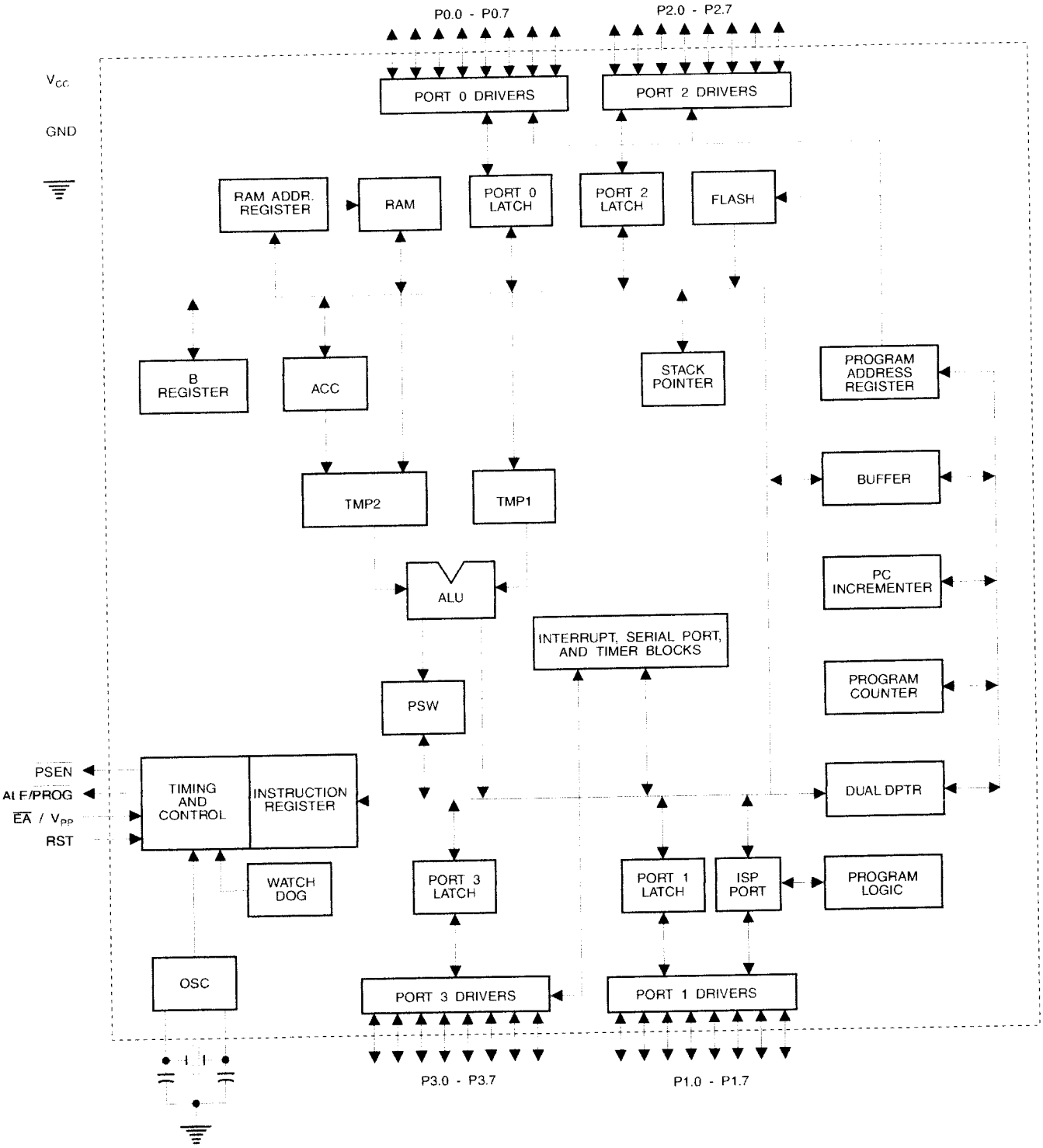
2.3 44-lead PLCC



2.4 42-lead PDIP



3. Block Diagram



4. Pin Description

4.1 VCC

Supply voltage.

4.2 GND

Ground.

4.3 Port 0

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

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

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

4.4 Port 1

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

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

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

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

4.5 Port 2

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

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

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

4.6 Port 3

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

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

4.7 RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

4.8 ALE/$\overline{PROG}$

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

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

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

Table 5-1. AT89S52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

Table 5-3. AUXR: Auxiliary Register

AUXR — Address = 8EH — Reset Value = XXX00XX0B

Not Bit Addressable

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

– Reserved for future expansion

DISALE Disable/Enable ALE

DISALE	*Operating Mode*
0	ALE is emitted at a constant rate of 1/6 the oscillator frequency
1	ALE is active only during a MOVX or MOVC instruction

DISRTO Disable/Enable Reset out

DISRTO	
0	Reset pin is driven High after WDT times out
1	Reset pin is input only

WDIDLE Disable/Enable WDT in IDLE mode

WDIDLE	
0	WDT continues to count in IDLE mode
1	WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should **ALWAYS** initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

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

Table 5-4. AUXR1: Auxiliary Register 1

AUXR1 — Address = A2H — Reset Value = XXXXXXX0B

Not Bit Addressable

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

– Reserved for future expansion

DPS Data Pointer Register Select

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

6. Memory Organization

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

6.1 Program Memory

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

On the AT89S52, if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

6.2 Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

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

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

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

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

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

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

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

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

7.1 Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When

WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 98xTOSC, where TOSC = 1/FOSC. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

7.2 WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

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

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

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

8. UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

9. Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

10. Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 5-2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 10-1. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 10-1. Timer 2 Operating Modes

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

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

10.1 Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 10-1.

10.2 Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 10-2). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 10-1. Timer in Capture Mode

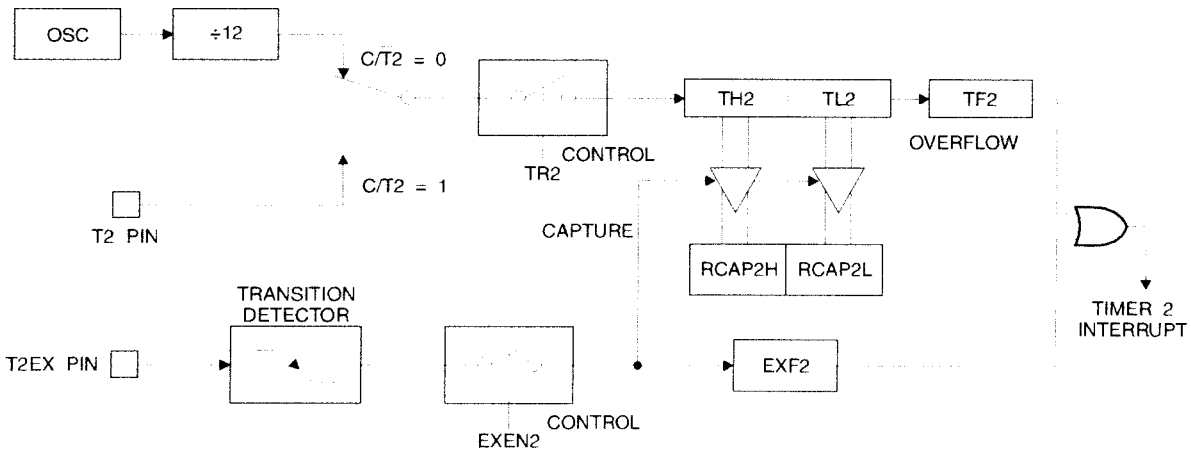


Table 10-2. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

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

Symbol	Function
–	Not implemented, reserved for future
T2OE	Timer 2 Output Enable bit
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter

Figure 10-2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture ModeRCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 10-2. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

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

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

Figure 10-2. Timer 2 Auto Reload Mode (DCEN = 0)

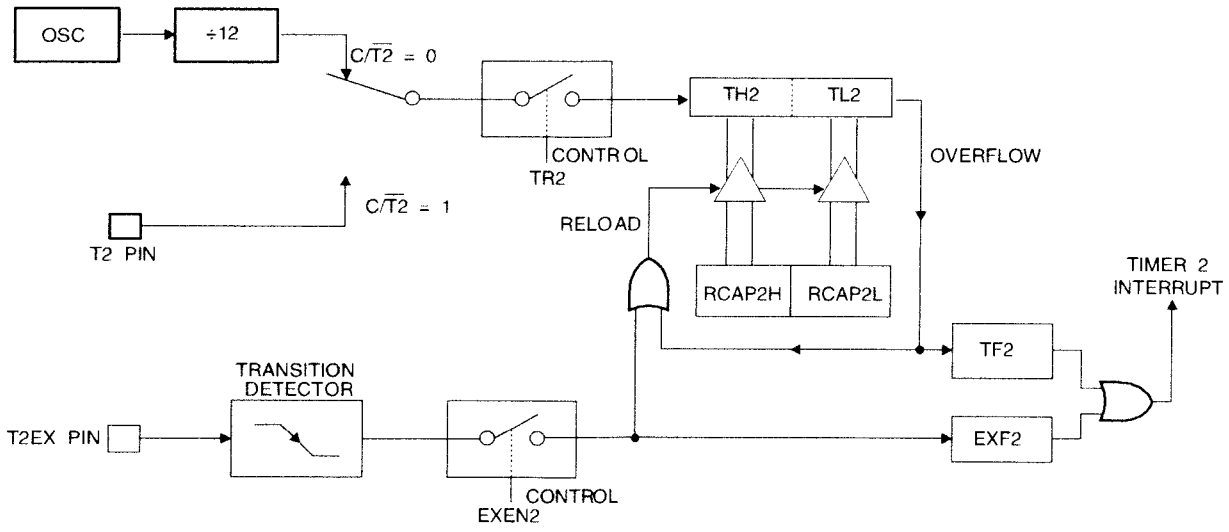
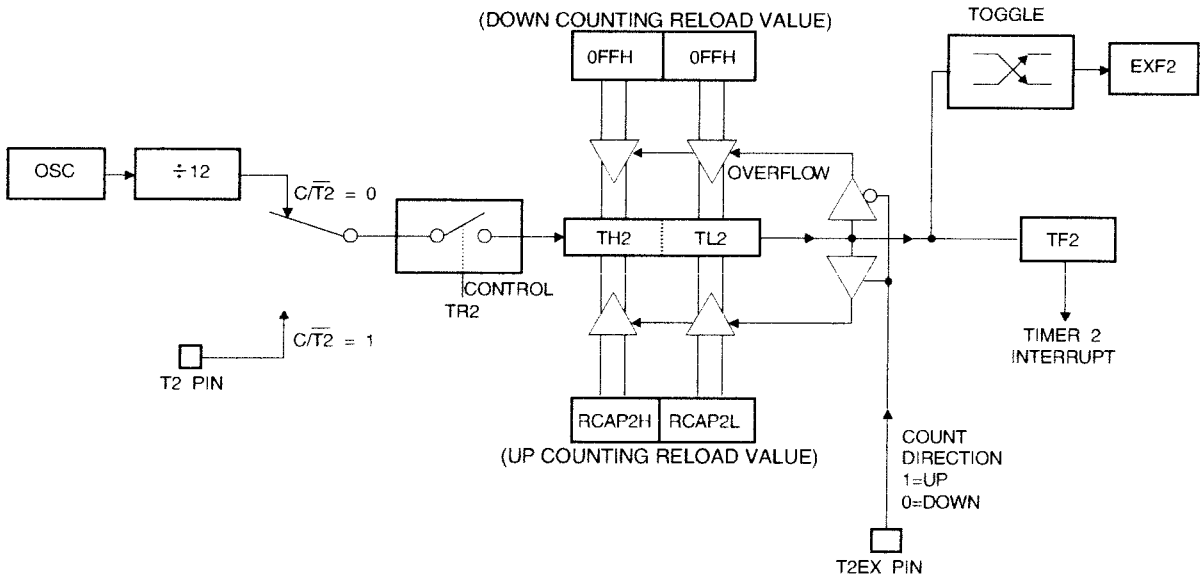


Figure 10-3. Timer 2 Auto Reload Mode (DCEN = 1)



11. Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 5-2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 11-1.

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

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

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

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

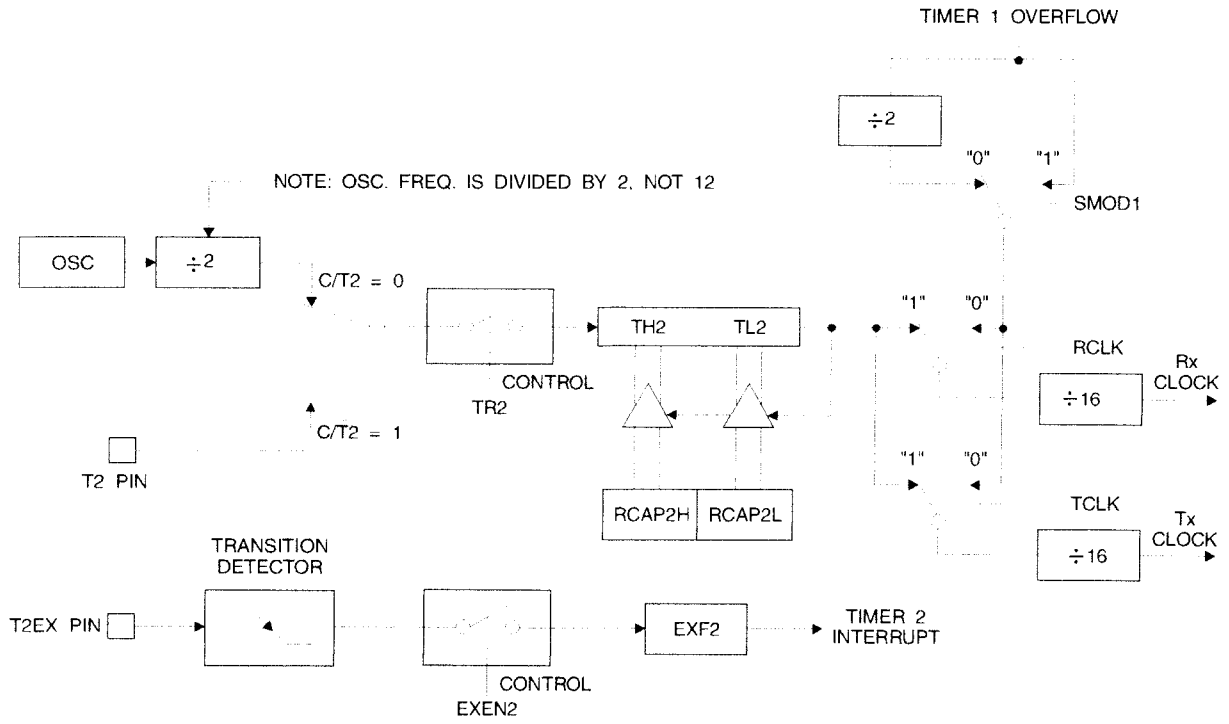
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536\text{-RCAP2H,RCAP2L})]}$$

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

Timer 2 as a baud rate generator is shown in Figure 11-1. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

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

Figure 11-1. Timer 2 in Baud Rate Generator Mode



12. Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 12-1. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz (for a 16-MHz operating frequency).

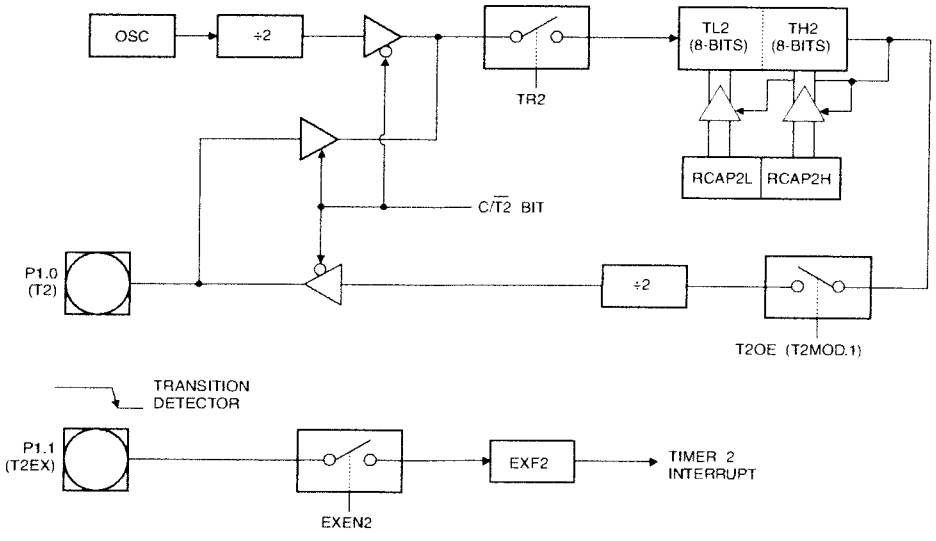
To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

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

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

Figure 12-1. Timer 2 in Clock-Out Mode



13. Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 13-1.

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

Note that Table 13-1 shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it may be used in future AT89 products.

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

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

Table 13-1. Interrupt Enable (IE) Register

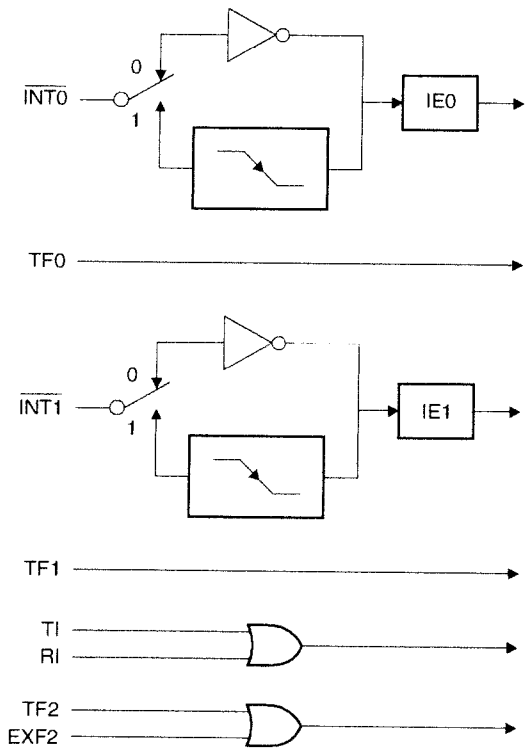
(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to reserved bits, because they may be used in future AT89 products.		

Figure 13-1. Interrupt Sources



14. Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 16-1. 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 16-2. 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.

15. Idle Mode

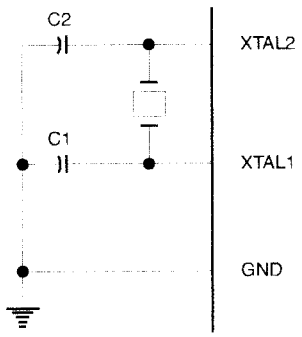
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

16. Power-down Mode

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

Figure 16-1. Oscillator Connections



Note: 1. C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 16-2. External Clock Drive Configuration

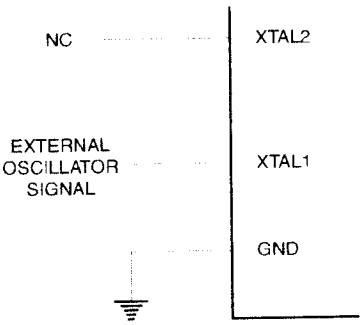


Table 16-1. Status of External Pins During Idle and Power-down Modes

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

17. Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 17-1.

Table 17-1. Lock Bit Protection Modes

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

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

18. Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S52, the address, data, and control signals should be set up according to the "Flash Programming Modes" (Table 22-1) and Figure 22-1 and Figure 22-2. To program the AT89S52, take the following steps:

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

$\overline{Data}$ Polling: The AT89S52 features $\overline{Data}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. **The status of the individual lock bits can be verified directly by reading them back.**

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

(000H) = 1EH indicates manufactured by Atmel
(100H) = 52H indicates AT89S52
(200H) = 06H

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

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

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

19. Programming the Flash – Serial Mode

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

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

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

20. Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
   a. Apply power between VCC and GND pins.
   b. Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

1. Set XTAL1 to "L" (if a crystal is not used).
2. Set RST to "L".
3. Turn V_{CC} power off.

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

21. Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 24-1.

Figure 22-1. Programming the Flash Memory (Parallel Mode)

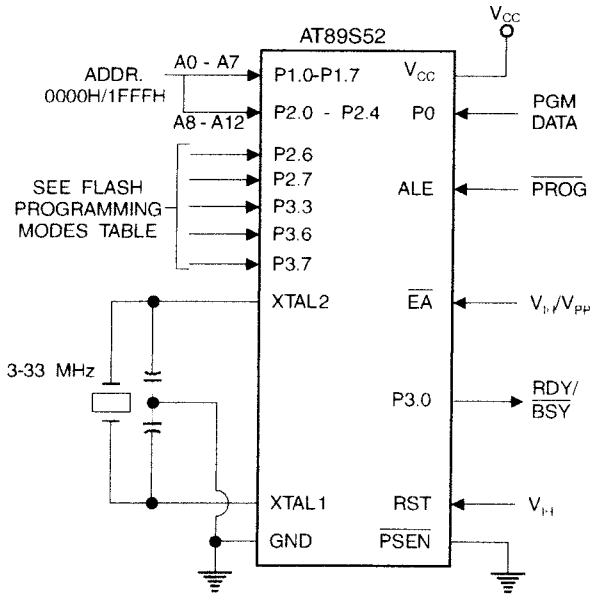


Figure 22-2. Verifying the Flash Memory (Parallel Mode)

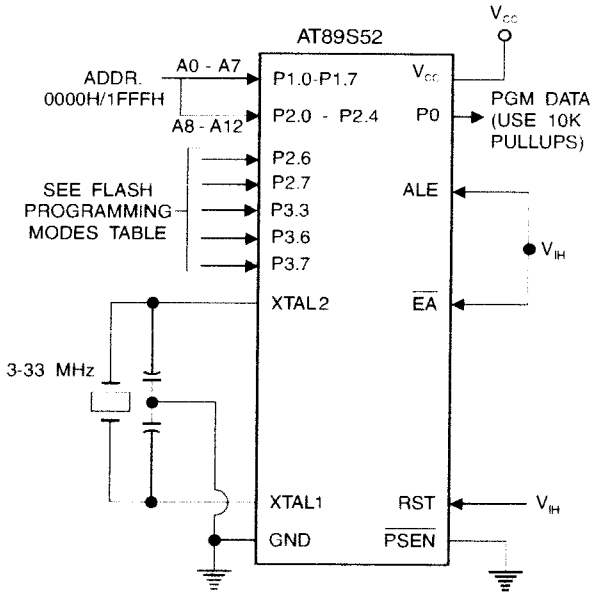
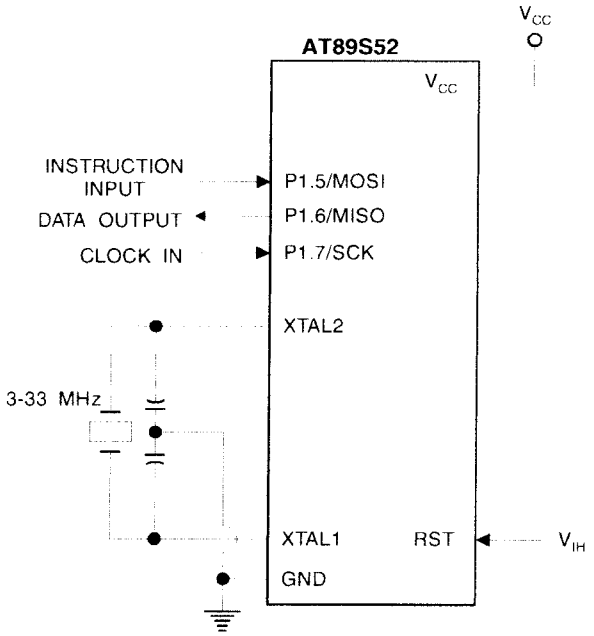


Figure 23-2. Flash Memory Serial Downloading



24. Flash Programming and Verification Waveforms – Serial Mode

Figure 24-1. Serial Programming Waveforms

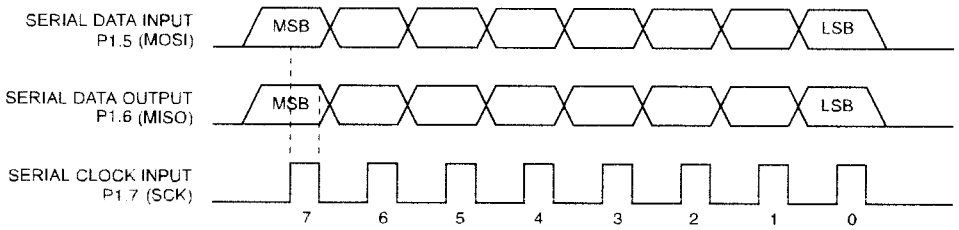


Table 24-1. Serial Programming Instruction Set

Instruction	Instruction Format Byte 1	Byte 2	Byte 3	Byte 4	Operation
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[1]	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxx A12 A11 A10 A9 A8	A7 xxx xxx0	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Note: 1. B1 = 0, B2 = 0 ---> Mode 1, no lock protection
B1 = 0, B2 = 1 ---> Mode 2, lock bit 1 activated
B1 = 1, B2 = 0 ---> Mode 3, lock bit 2 activated
B1 = 1, B2 = 1 ---> Mode 4, lock bit 3 activated

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

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

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

25. Serial Programming Characteristics

Figure 25-1. Serial Programming Timing

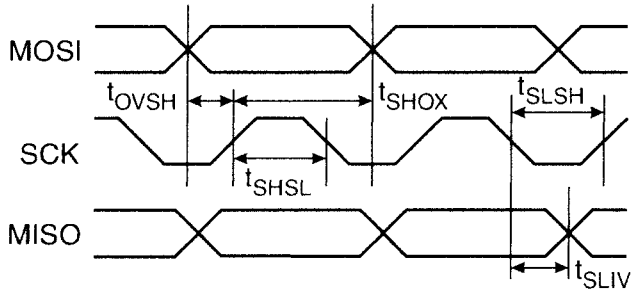


Table 25-1. Serial Programming Characteristics, T_A = -40°C to 85°C, V_{CC} = 4.0 - 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	3		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	8 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	8 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	µs

26. Absolute Maximum Ratings*

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

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

27. DC Characteristics

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

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-300	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[2]	V_{CC} = 5.5V		50	µA

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

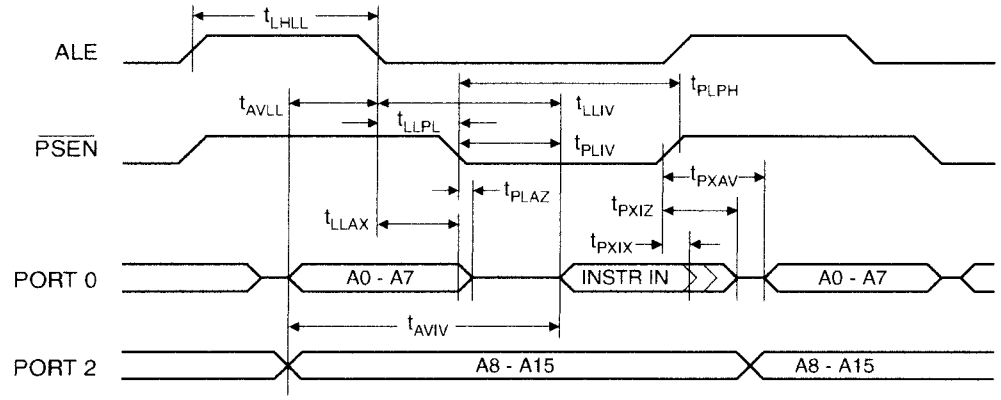
28. AC Characteristics

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

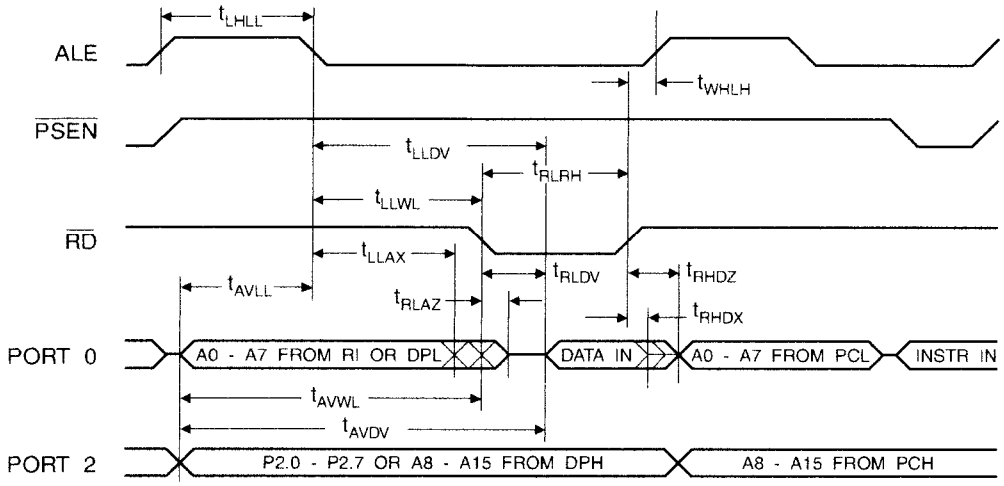
28.1 External Program and Data Memory Characteristics

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

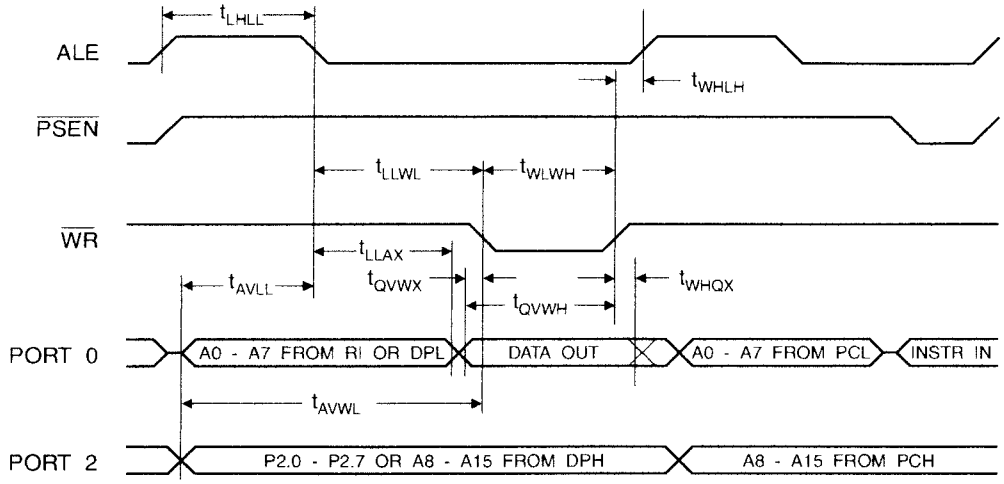
29. External Program Memory Read Cycle



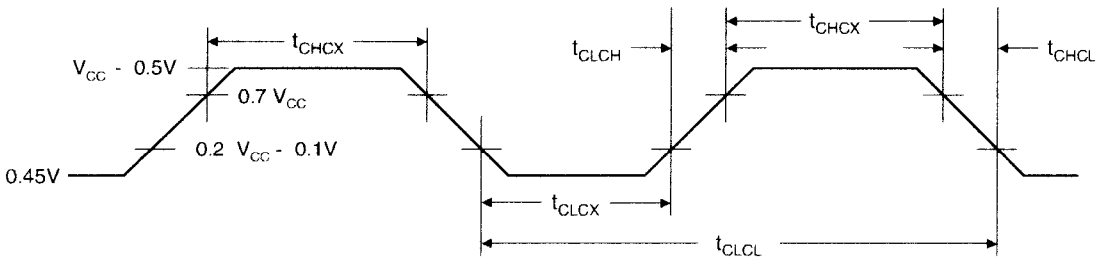
30. External Data Memory Read Cycle



31. External Data Memory Write Cycle



32. External Clock Drive Waveforms



33. External Clock Drive

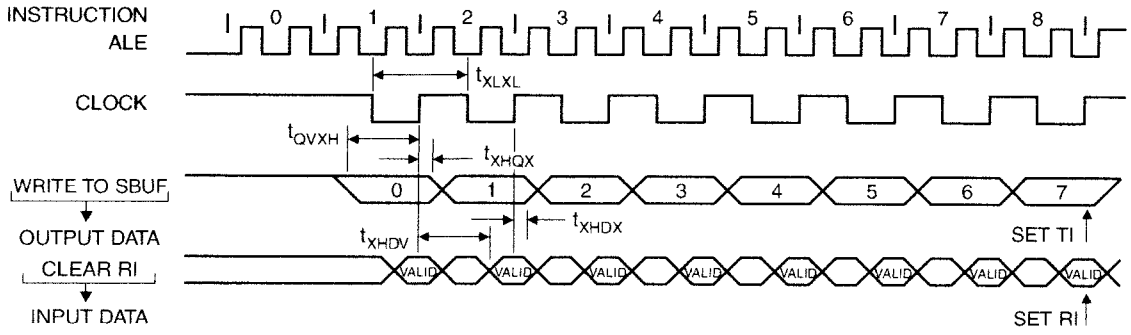
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
t_{CLCL}	Clock Period	30		ns
t_{CHCX}	High Time	12		ns
t_{CLCX}	Low Time	12		ns
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

34. Serial Port Timing: Shift Register Mode Test Conditions

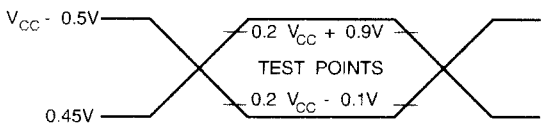
The values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

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

35. Shift Register Mode Timing Waveforms

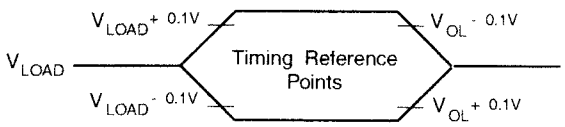


36. AC Testing Input/Output Waveforms[1]



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.

37. 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 a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

38. Ordering Information

38.1 Standard Package

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC AT89S52-24JC AT89S52-24PC AT89S52-24SC	44A 44J 40P6 42PS6	Commercial (0°C to 70°C)
		AT89S52-24AI AT89S52-24JI AT89S52-24PI AT89S52-24SI	44A 44J 40P6 42PS6	Industrial (-40°C to 85°C)
33	4.5V to 5.5V	AT89S52-33AC AT89S52-33JC AT89S52-33PC AT89S52-33SC	44A 44J 40P6 42PS6	Commercial (0°C to 70°C)

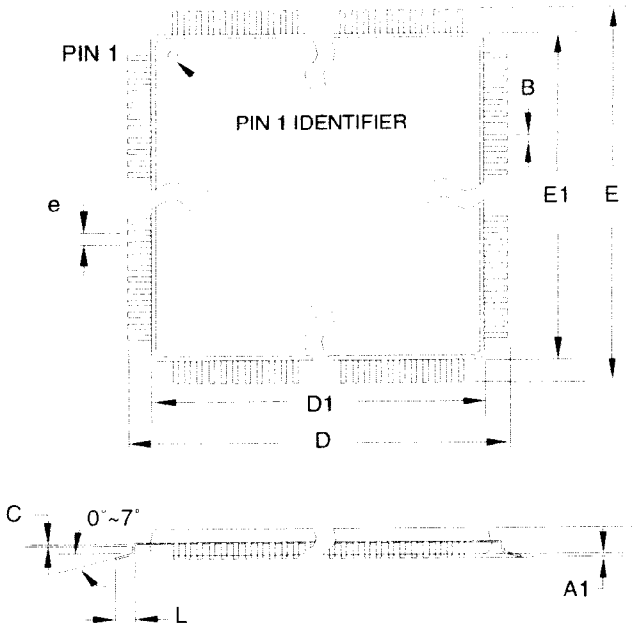
38.2 Green Package Option (Pb/Halide-free)

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AU AT89S52-24JU AT89S52-24PU	44A 44J 40P6	Industrial (-40°C to 85°C)

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
42PS6	42-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

39. Packaging Information

39.1 44A – TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

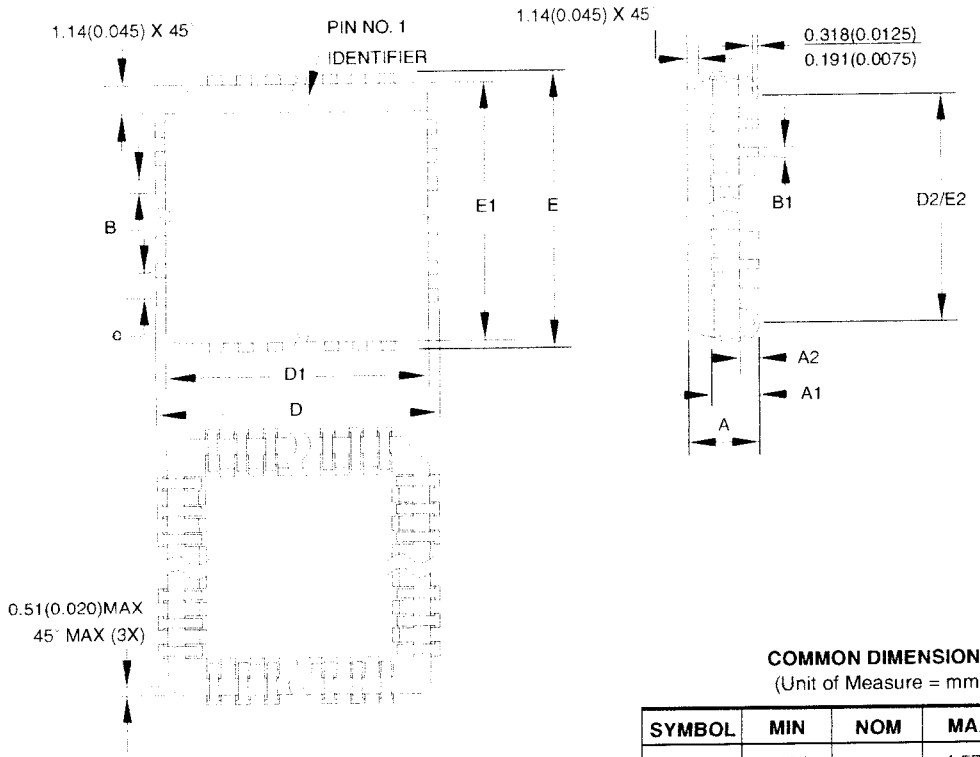
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44A**, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	44A	B

39.2 44J – PLCC



COMMON DIMENSIONS
(Unit of Measure = mm)

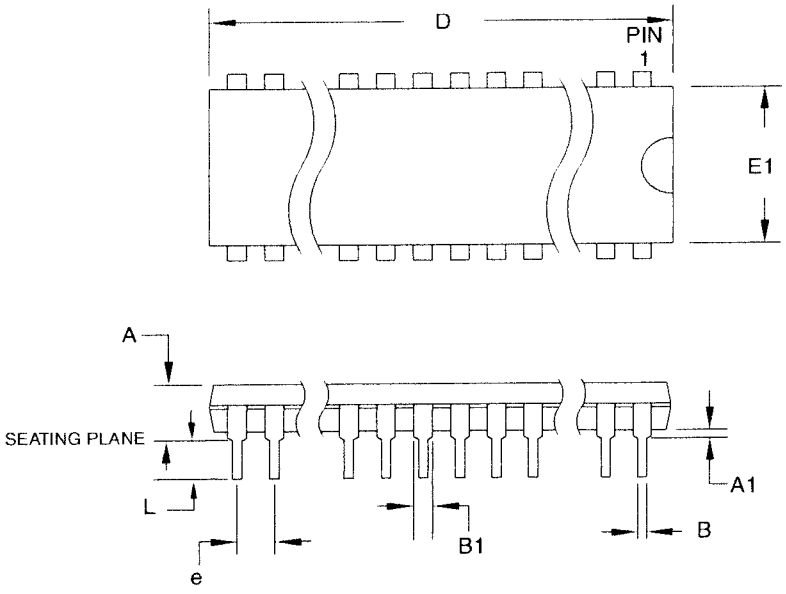
SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	–	4.572	
A1	2.286	–	3.048	
A2	0.508	–	–	
D	17.399	–	17.653	
D1	16.510	–	16.662	Note 2
E	17.399	–	17.653	
E1	16.510	–	16.662	Note 2
D2/E2	14.986	–	16.002	
B	0.660	–	0.813	
B1	0.330	–	0.533	
e	1.270 TYP			

Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010"(0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**44J**, 44-lead, Plastic J-leaded Chip Carrier (PLCC)	44J	B

39.3 40P6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

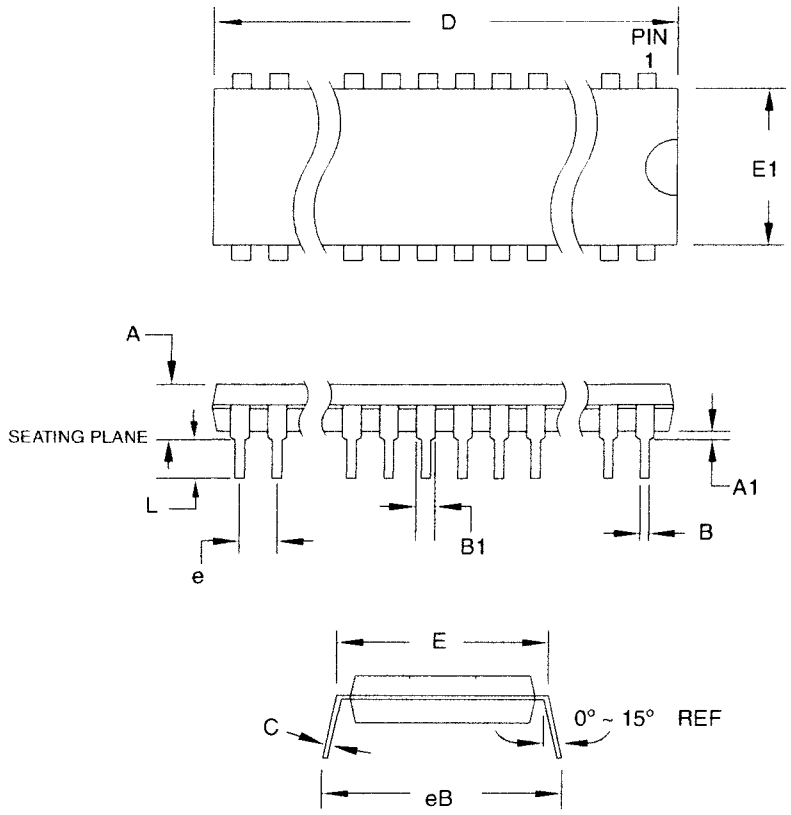
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**40P6**, 40-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	40P6	B

39.4 42PS6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.83	
A1	0.51	–	–	
D	36.70	–	36.96	Note 2
E	15.24	–	15.88	
E1	13.46	–	13.97	Note 2
B	0.38	–	0.56	
B1	0.76	–	1.27	
L	3.05	–	3.43	
C	0.20	–	0.30	
eB	–	–	18.55	
e	1.78 TYP			

Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

11/6/03

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**42PS6**, 42-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	42PS6	A

Atmel Corporation

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

Regional Headquarters

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

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

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

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

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

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

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

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

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

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

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

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

Literature Requests
www.atmel.com/literature

1919C–MICRO–3/05 xM

HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)
'99.9
Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 × 8 dot character fonts and 32 5 × 10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5 × 8 and 5 × 10 dot matrix possible
- Low power operation support:
  - — 2.7 to 5.5V
- Wide range of liquid crystal display driver power
  - — 3.0 to 11V
- Liquid crystal drive waveform
  - — A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
  - — 2 MHz (when V_{CC} = 5V)
- 4-bit or 8-bit MPU interface enabled
- 80 × 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
  - — 208 character fonts (5 × 8 dot)
  - — 32 character fonts (5 × 10 dot)

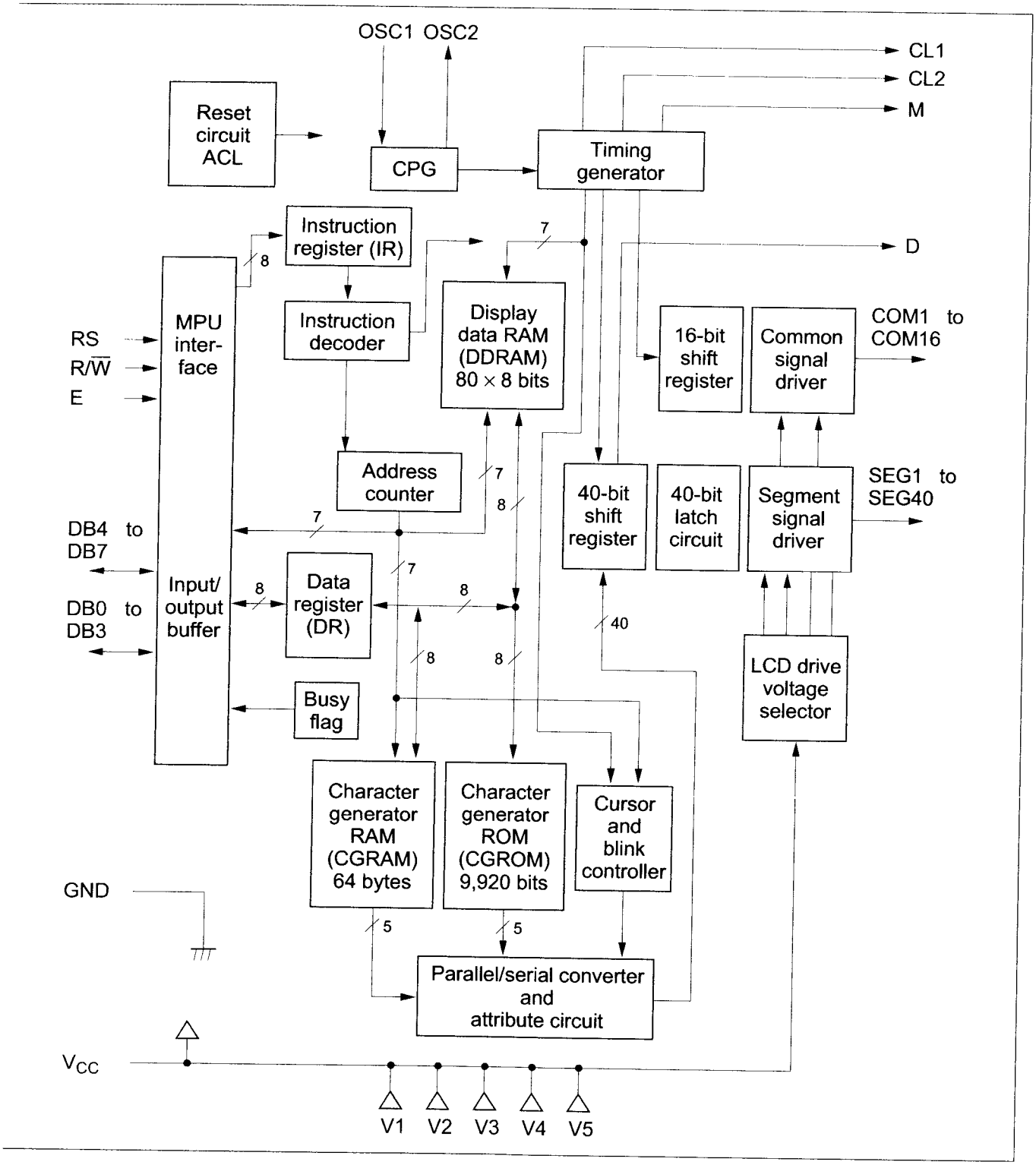
- 64 × 8-bit character generator RAM
  - — 8 character fonts (5 × 8 dot)
  - — 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
  - — 1/8 for one line of 5 × 8 dots with cursor
  - — 1/11 for one line of 5 × 10 dots with cursor
  - — 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
  - — Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

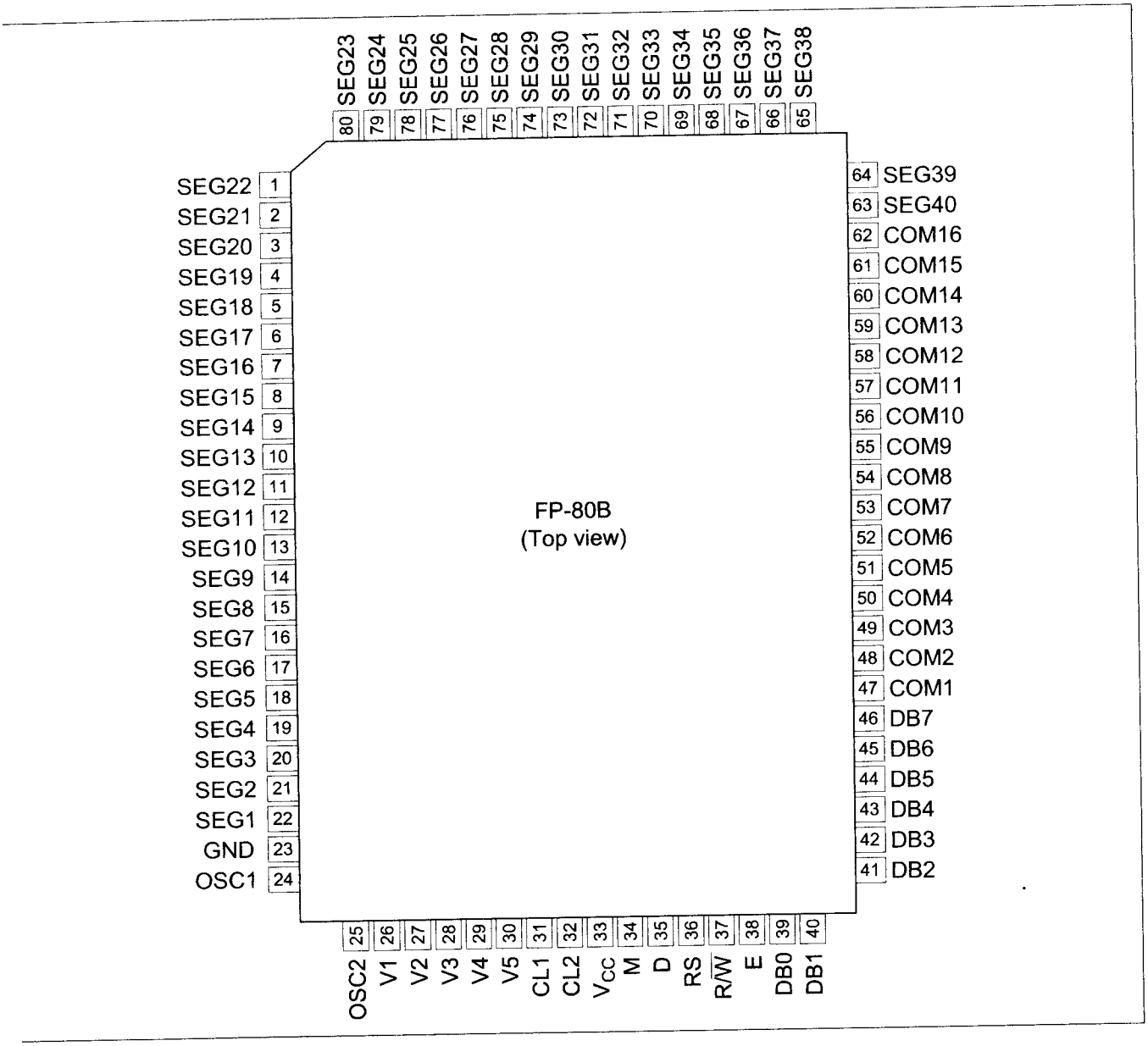
Type No.	Package	CGROM
HD44780UA00FS	FP-80B	Japanese standard font
HCD44780UA00	Chip	
HD44780UA00TF	TFP-80F	
HD44780UA02FS	FP-80B	European standard font
HCD44780UA02	Chip	
HD44780UA02TF	TFP-80F	
HD44780UBxxFS	FP-80B	Custom font
HCD44780UBxx	Chip	
HD44780UBxxTF	TFP-80F	

Note: xx: ROM code No.

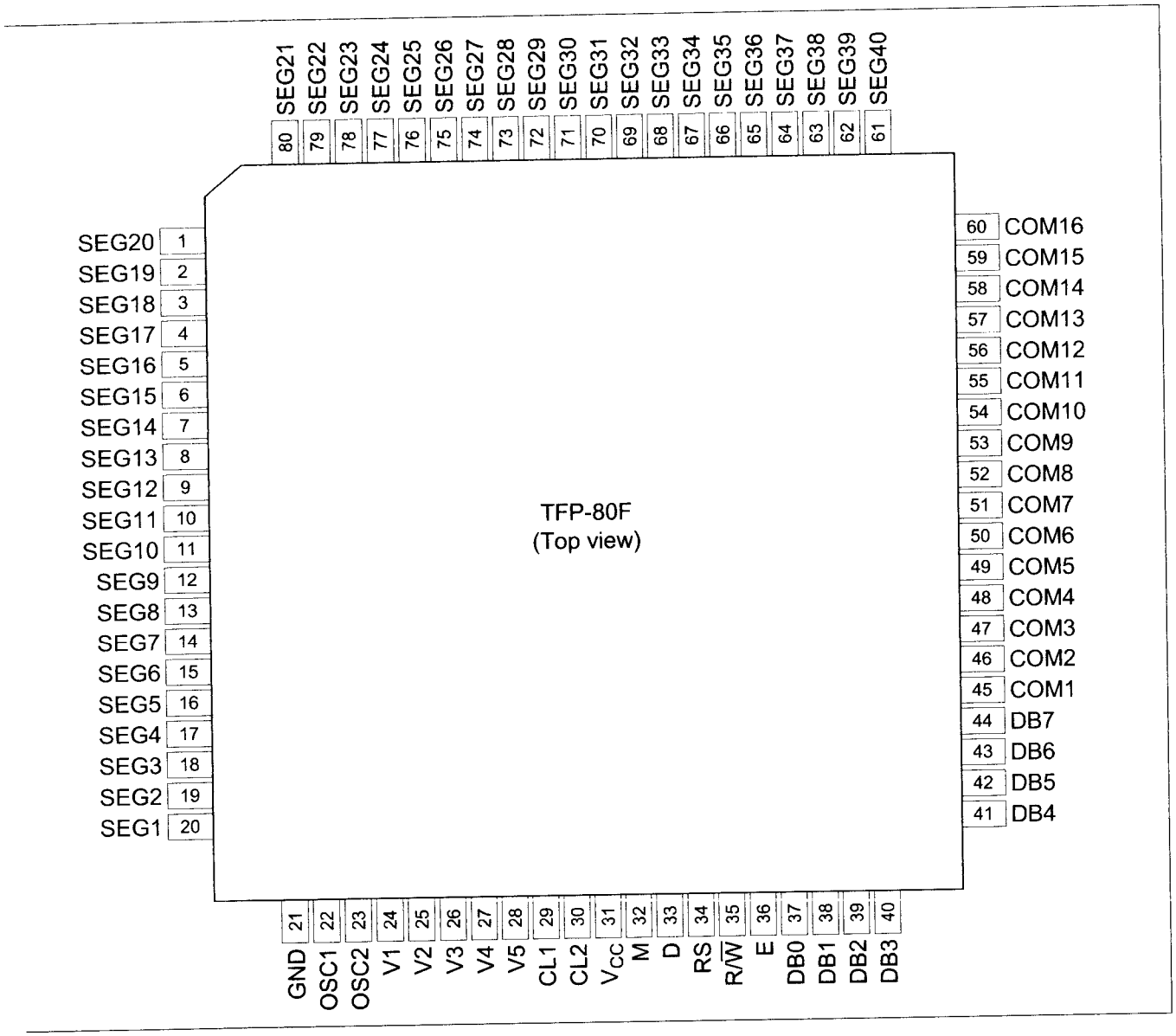
HD44780U Block Diagram



HD44780U Pin Arrangement (FP-80B)

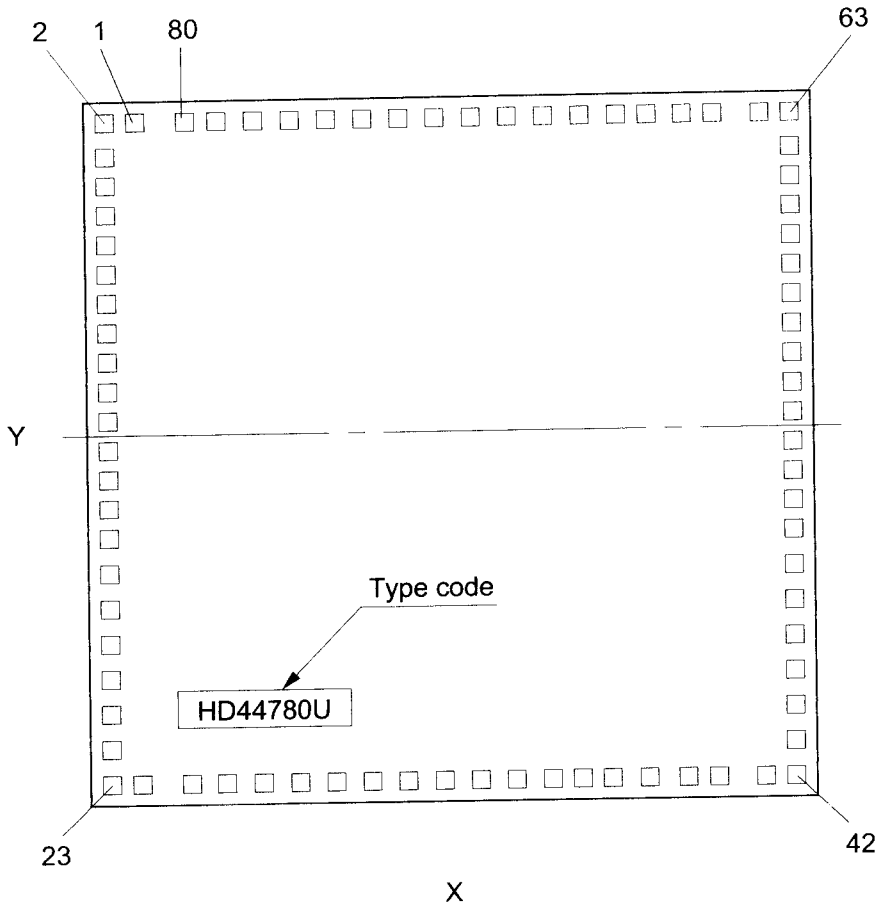


HD44780U Pin Arrangement (TFP-80F)



HD44780U Pad Arrangement

Chip size: 4.90 × 4.90 mm^2
Coordinate: Pad center (µm)
Origin: Chip center
Pad size: 114 × 114 µm^2



HCD44780U Pad Location Coordinates

		Coordinate				Coordinate	
Pad No.	Function	X (um)	Y (um)	Pad No.	Function	X (um)	Y (um)
1	SEG22	–2100	2313	41	DB2	2070	–2290
2	SEG21	–2280	2313	42	DB3	2260	–2290
3	SEG20	–2313	2089	43	DB4	2290	–2099
4	SEG19	–2313	1833	44	DB5	2290	–1883
5	SEG18	–2313	1617	45	DB6	2290	–1667
6	SEG17	–2313	1401	46	DB7	2290	–1452
7	SEG16	–2313	1186	47	COM1	2313	–1186
8	SEG15	–2313	970	48	COM2	2313	–970
9	SEG14	–2313	755	49	COM3	2313	–755
10	SEG13	–2313	539	50	COM4	2313	–539
11	SEG12	–2313	323	51	COM5	2313	–323
12	SEG11	–2313	108	52	COM6	2313	–108
13	SEG10	–2313	–108	53	COM7	2313	108
14	SEG9	–2313	–323	54	COM8	2313	323
15	SEG8	–2313	–539	55	COM9	2313	539
16	SEG7	–2313	–755	56	COM10	2313	755
17	SEG6	–2313	–970	57	COM11	2313	970
18	SEG5	–2313	–1186	58	COM12	2313	1186
19	SEG4	–2313	–1401	59	COM13	2313	1401
20	SEG3	–2313	–1617	60	COM14	2313	1617
21	SEG2	–2313	–1833	61	COM15	2313	1833
22	SEG1	–2313	–2073	62	COM16	2313	2095
23	GND	–2280	–2290	63	SEG40	2296	2313
24	OSC1	–2080	–2290	64	SEG39	2100	2313
25	OSC2	–1749	–2290	65	SEG38	1617	2313
26	V1	–1550	–2290	66	SEG37	1401	2313
27	V2	–1268	–2290	67	SEG36	1186	2313
28	V3	–941	–2290	68	SEG35	970	2313
29	V4	–623	–2290	69	SEG34	755	2313
30	V5	–304	–2290	70	SEG33	539	2313
31	CL1	–48	–2290	71	SEG32	323	2313
32	CL2	142	–2290	72	SEG31	108	2313
33	V_{CC}	309	–2290	73	SEG30	–108	2313
34	M	475	–2290	74	SEG29	–323	2313
35	D	665	–2290	75	SEG28	–539	2313
36	RS	832	–2290	76	SEG27	–755	2313
37	R/$\overline{W}$	1022	–2290	77	SEG26	–970	2313
38	E	1204	–2290	78	SEG25	–1186	2313
39	DB0	1454	–2290	79	SEG24	–1401	2313
40	DB1	1684	–2290	80	SEG23	–1617	2313

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/$\overline{W}$	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive V_{CC} –V5 = 11 V (max)
V_{CC}, GND	2	—	Power supply	V_{CC}: 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/$\overline{W}$ = 1 (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when RS = 0 and R/$\overline{W}$ = 1 (Table 1).

Table 1 Register Selection

RS	R/$\overline{W}$	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 × 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display (N = 0) (Figure 2)
  - — When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.
    When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

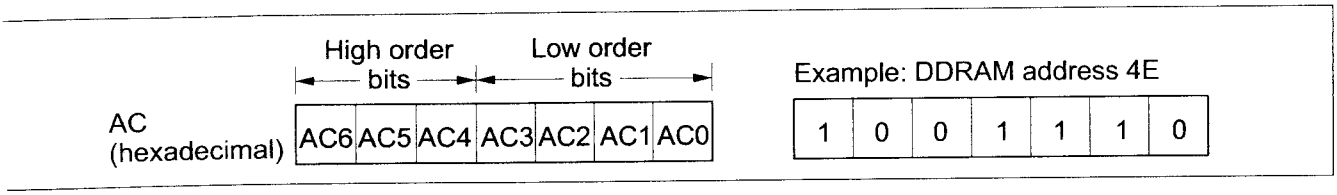


Figure 1 DDRAM Address

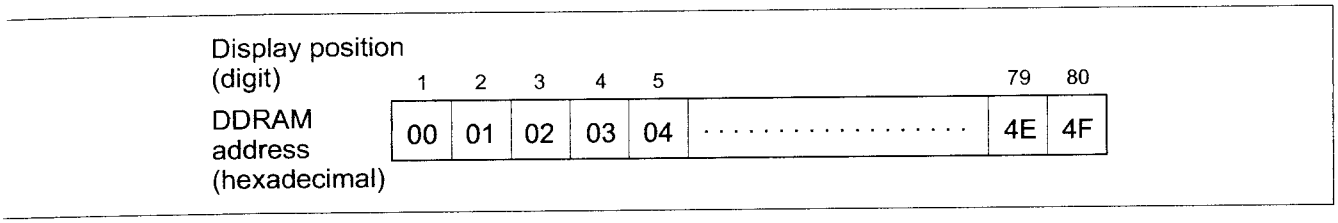


Figure 2 1-Line Display

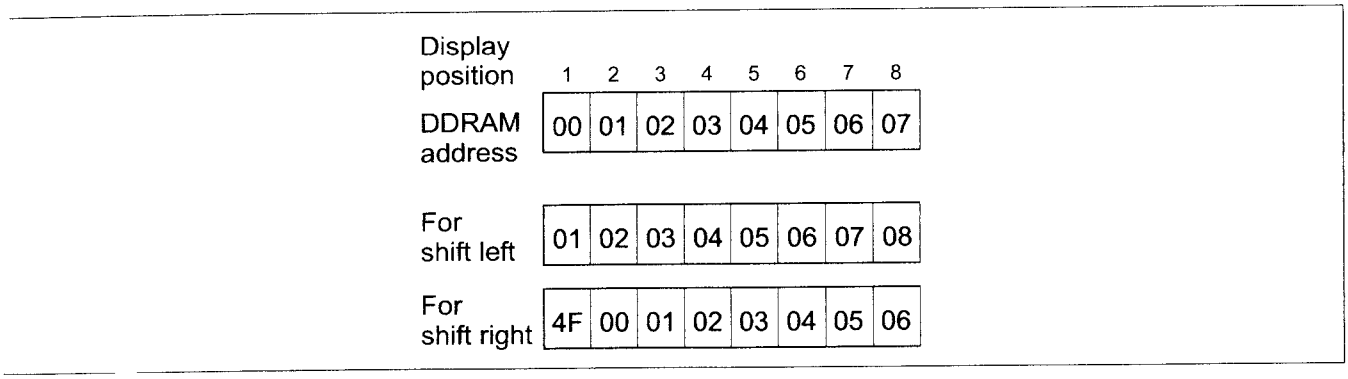


Figure 3 1-Line by 8-Character Display Example

— Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

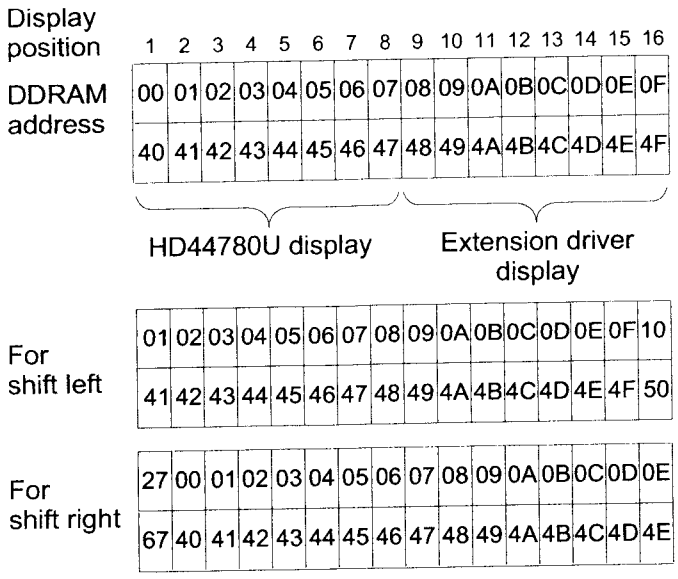


Figure 6 2-Line by 16-Character Display Example

Character Generator ROM (CGROM)

The character generator ROM generates 5 × 8 dot or 5 × 10 dot character patterns from 8-bit character codes (Table 4). It can generate 208 5 × 8 dot character patterns and 32 5 × 10 dot character patterns. User-defined character patterns are also available by mask-programmed ROM.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns by program. For 5 × 8 dots, eight character patterns can be written, and for 5 × 10 dots, four character patterns can be written.

Write into DDRAM the character codes at the addresses shown as the left column of Table 4 to show the character patterns stored in CGRAM.

See Table 5 for the relationship between CGRAM addresses and data and display patterns.

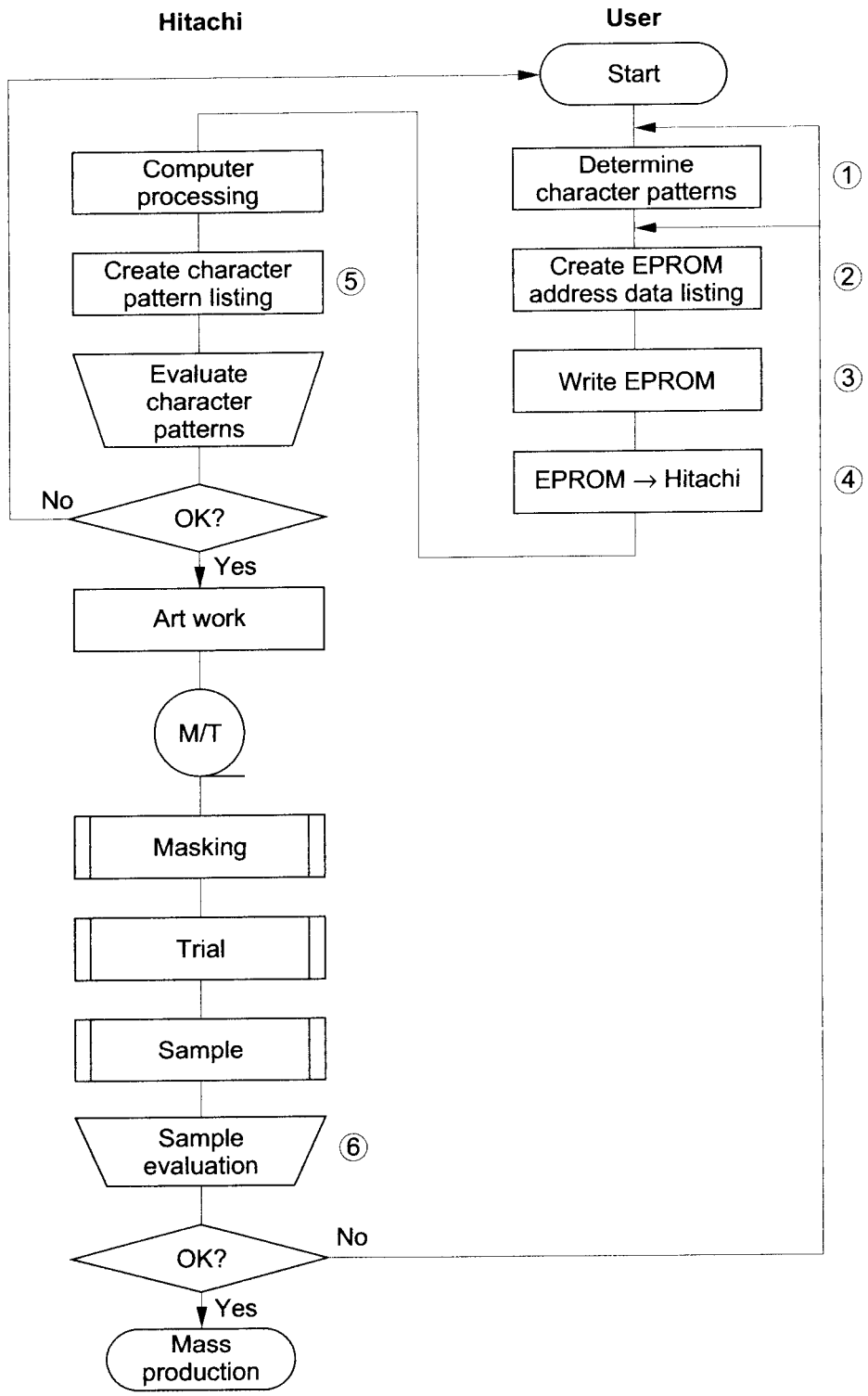
Areas that are not used for display can be used as general data RAM.

Modifying Character Patterns

- Character pattern development procedure

The following operations correspond to the numbers listed in Figure 7:

1. Determine the correspondence between character codes and character patterns.
2. Create a listing indicating the correspondence between EPROM addresses and data.
3. Program the character patterns into the EPROM.
4. Send the EPROM to Hitachi.
5. Computer processing on the EPROM is performed at Hitachi to create a character pattern listing, which is sent to the user.
6. If there are no problems within the character pattern listing, a trial LSI is created at Hitachi and samples are sent to the user for evaluation. When it is confirmed by the user that the character patterns are correctly written, mass production of the LSI proceeds at Hitachi.



Note: For a description of the numbers used in this figure, refer to the preceding page.

Figure 7 Character Pattern Development Procedure

- Programming character patterns

  This section explains the correspondence between addresses and data used to program character patterns in EPROM. The HD44780U character generator ROM can generate 208 5 × 8 dot character patterns and 32 5 × 10 dot character patterns for a total of 240 different character patterns.

  — Character patterns

  EPROM address data and character pattern data correspond with each other to form a 5 × 8 or 5 × 10 dot character pattern (Tables 2 and 3).

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)

EPROM Address												Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0	
								0	0	0	0	1	0	0	0	0	
								0	0	0	1	1	0	0	0	0	
								0	0	1	0	1	0	1	1	0	
								0	0	1	1	1	1	0	0	1	
								0	1	0	0	1	0	0	0	1	
								0	1	0	1	1	0	0	0	1	
								0	1	1	0	1	1	1	1	0	
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	← Cursor position
								1	0	0	0	0	0	0	0	0	
								1	0	0	1	0	0	0	0	0	
								1	0	1	0	0	0	0	0	0	
								1	0	1	1	0	0	0	0	0	
								1	1	0	0	0	0	0	0	0	
								1	1	0	1	0	0	0	0	0	
								1	1	1	0	0	0	0	0	0	
								1	1	1	1	0	0	0	0	0	
Character code								Line position									

Notes: 1. EPROM addresses A11 to A4 correspond to a character code.
2. EPROM addresses A3 to A0 specify a line position of the character pattern.
3. EPROM data O4 to O0 correspond to character pattern data.
4. EPROM data O5 to O7 must be specified as 0.
5. A lit display position (black) corresponds to a 1.
6. Line 9 and the following lines must be blanked with 0s for a 5 × 8 dot character fonts.

— Handling unused character patterns

1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
   a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programing a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
   b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

Table 3 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)

EPROM Address												Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0	
								0	0	0	0	0	0	0	0	0	
								0	0	0	1	0	0	0	0	0	
								0	0	1	0	0	1	1	0	1	
								0	0	1	1	1	0	0	1	1	
								0	1	0	0	1	0	0	0	1	
								0	1	0	1	1	0	0	0	1	
								0	1	1	0	0	1	1	1	1	
0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	
								1	0	0	0	0	0	0	0	1	
								1	0	0	1	0	0	0	0	1	
								1	0	1	0	0	0	0	0	0	← Cursor position
								1	0	1	1	0	0	0	0	0	
								1	1	0	0	0	0	0	0	0	
								1	1	0	1	0	0	0	0	0	
								1	1	1	0	0	0	0	0	0	
								1	1	1	1	0	0	0	0	0	

Character code — Line position

Notes: 1. EPROM addresses A11 to A3 correspond to a character code.
2. EPROM addresses A3 to A0 specify a line position of the character pattern.
3. EPROM data O4 to O0 correspond to character pattern data.
4. EPROM data O5 to O7 must be specified as 0.
5. A lit display position (black) corresponds to a 1.
6. Line 11 and the following lines must be blanked with 0s for a 5 × 10 dot character fonts.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Note: The user can specify any pattern for character-generator RAM.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)															
xxxx0001	(2)															
xxxx0010	(3)															
xxxx0011	(4)															
xxxx0100	(5)															
xxxx0101	(6)															
xxxx0110	(7)															
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0 High Low	5 4 3 2 1 0 High Low	7 6 5 4 3 2 1 0 High Low	
0 0 0 0 * 0 0 0	0 0 0	0 0 0 * * * 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 0 1 1 1 1 * * * 0 0 0 0 0	Character pattern (1) Cursor position
0 0 0 0 * 0 0 1	0 0 1	0 0 0 * * * 1 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 1 * * * 0 0 0 0 0	Character pattern (2) Cursor position
		0 0 0 * * * 0 0 1	
0 0 0 0 * 1 1 1	1 1 1	1 0 0 1 0 1 1 1 0 1 1 1 * * *	

Notes: 1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).

2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.
   Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display.
   If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.

3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).

4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.

5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data) (cont)

For 5 × 10 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0 High Low	5 4 3 2 1 0 High Low	7 6 5 4 3 2 1 0 High Low	
0 0 0 0 * 0 0 *	0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0	* * * 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 * * * 0 0 0 0 0	Character pattern Cursor position
	1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1	* * * * * * * * * * * * * * * *	
	0 0 0 0 0 0 0 1	* * *	
0 0 0 0 * 1 1 *	1 1 1 0 0 1 1 0 1 0	* * *	
	1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1	* * * * * * * * * * * * * * * *	

Notes: 1. Character code bits 1 and 2 correspond to CGRAM address bits 4 and 5 (2 bits: 4 types).

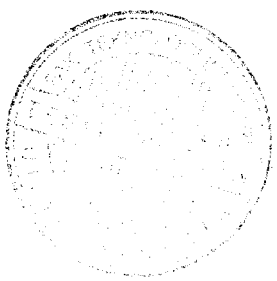
2. CGRAM address bits 0 to 3 designate the character pattern line position. The 11th line is the cursor position and its display is formed by a logical OR with the cursor.
   Maintain the 11th line data corresponding to the cursor display positon at 0 as the cursor display.
   If the 11th line data is "1", "1" bits will light up the 11th line regardless of the cursor presence.
   Since lines 12 to 16 are not used for display, they can be used for general data RAM.

3. Character pattern row positions are the same as 5 × 8 dot character pattern positions.

4. CGRAM character patterns are selected when character code bits 4 to 7 are all 0.
   However, since character code bits 0 and 3 have no effect, the P display example above can be selected by character codes 00H, 01H, 08H, and 09H.

5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.



Timing Generation Circuit

The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM, CGROM and CGRAM. RAM read timing for display and internal operation timing by MPU access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no undesirable interferences, such as flickering, in areas other than the display area.

Liquid Crystal Display Driver Circuit

The liquid crystal display driver circuit consists of 16 common signal drivers and 40 segment signal drivers. When the character font and number of lines are selected by a program, the required common signal drivers automatically output drive waveforms, while the other common signal drivers continue to output non-selection waveforms.

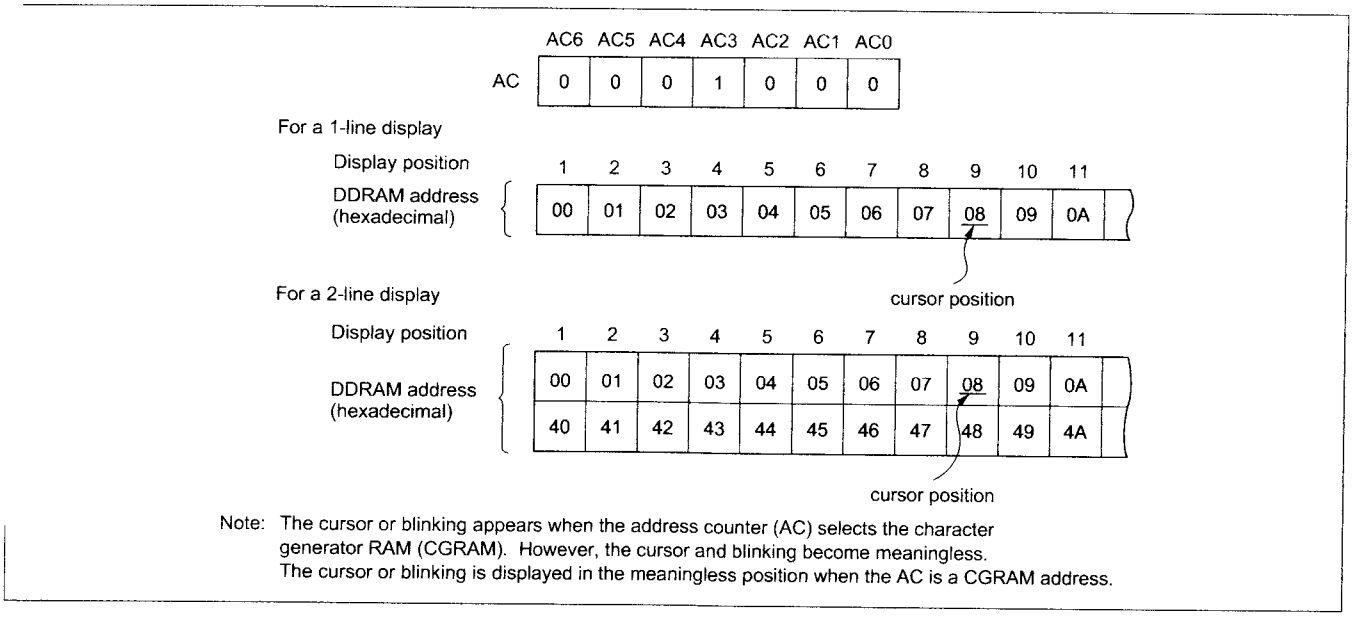
Sending serial data always starts at the display data character pattern corresponding to the last address of the display data RAM (DDRAM).

Since serial data is latched when the display data character pattern corresponding to the starting address enters the internal shift register, the HD44780U drives from the head display.

Cursor/Blink Control Circuit

The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC).

For example (Figure 8), when the address counter is 08H, the cursor position is displayed at DDRAM address 08H.



Note: The cursor or blinking appears when the address counter (AC) selects the character generator RAM (CGRAM). However, the cursor and blinking become meaningless. The cursor or blinking is displayed in the meaningless position when the AC is a CGRAM address.

Figure 8 Cursor/Blink Display Example

Interfacing to the MPU

The HD44780U can send data in either two 4-bit operations or one 8-bit operation, thus allowing interfacing with 4- or 8-bit MPUs.

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).

  The busy flag must be checked (one instruction) after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.
- For 8-bit interface data, all eight bus lines (DB0 to DB7) are used.

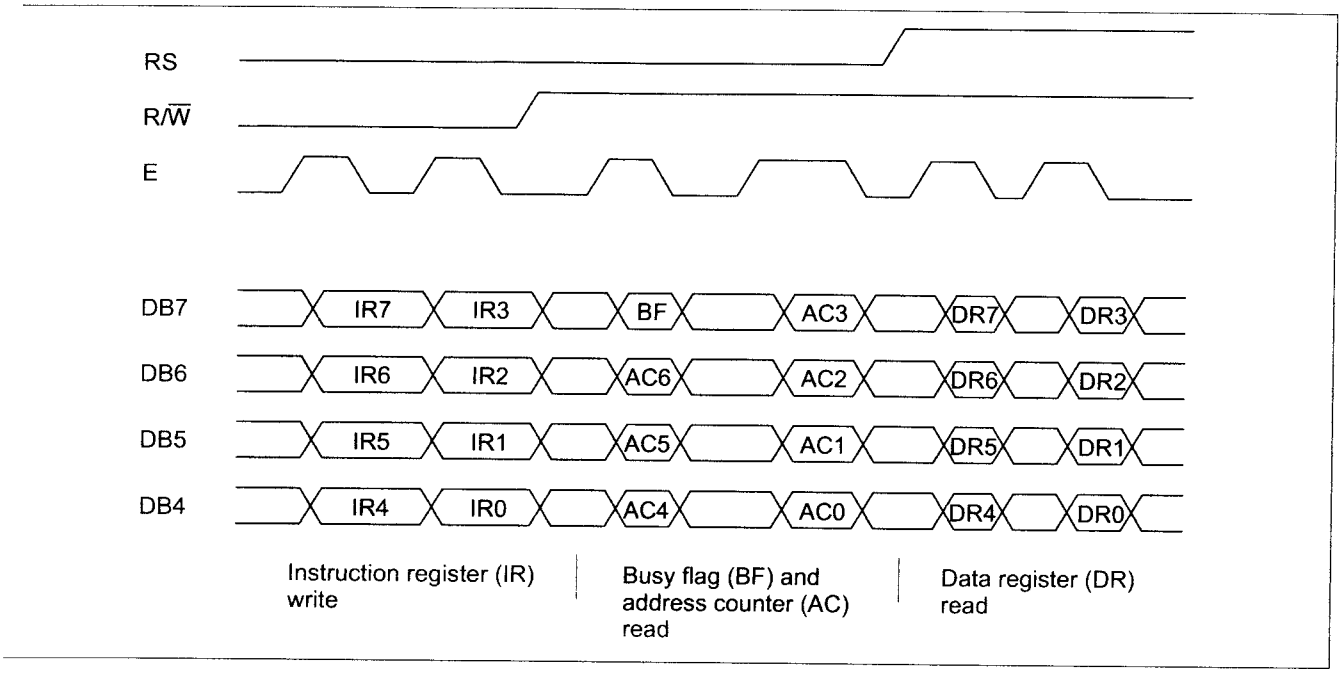


Figure 9 4-Bit Transfer Example

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends (BF = 1). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear
2. Function set:
   DL = 1; 8-bit interface data
   N = 0; 1-line display
   F = 0; 5 × 8 dot character font
3. Display on/off control:
   D = 0; Display off
   C = 0; Cursor off
   B = 0; Blinking off
4. Entry mode set:
   I/D = 1; Increment by 1
   S = 0; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initial-ization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/

write signal (R/$\overline{W}$), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state (BF = 0) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruc-tion execution time.

Table 6 Instructions

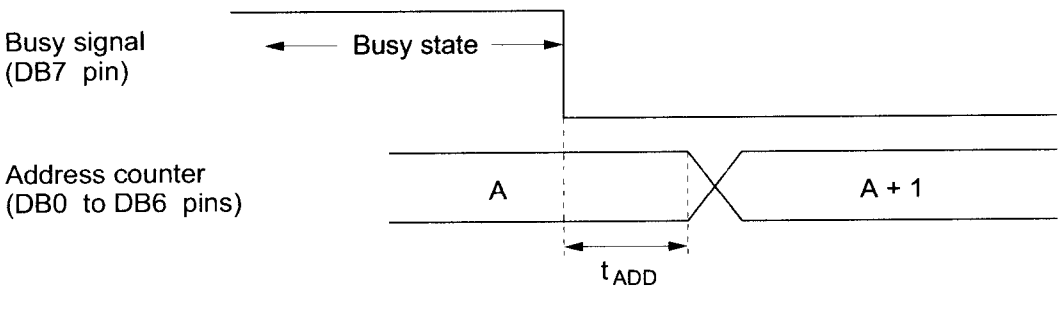
Instruction	Code RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 µs
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 µs
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 µs
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 µs
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 µs

Table 6 Instructions (cont)

Instruction	Code RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.	37 µs t_{ADD} = 4 µs*
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 × 10 dots, F = 0: 5 × 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable										DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{cp} or f_{OSC} is 250 kHz, $37\ \mu s \times \frac{270}{250} = 40\ \mu s$

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.



Note: t_{ADD} depends on the operation frequency
t_{ADD} = 1.5/(f_{cp} or f_{OSC}) seconds

Figure 10 Address Counter Update

Instruction Description

Clear Display

Clear display writes space code 20H (character pattern for character code 20H must be a blank pattern) into all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. In other words, the display disappears and the cursor or blinking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 (increment mode) in entry mode. S of entry mode does not change.

Return Home

Return home sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. The DDRAM contents do not change.

The cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

Entry Mode Set

I/D: Increments (I/D = 1) or decrements (I/D = 0) the DDRAM address by 1 when a character code is written into or read from DDRAM.

The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

S: Shifts the entire display either to the right (I/D = 0) or to the left (I/D = 1) when S is 1. The display does not shift if S is 0.

If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

Display On/Off Control

D: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

C: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5 × 8 dot character font selection and in the 11th line for the 5 × 10 dot character font selection (Figure 13).

B: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{OSC} is 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency changes according to f_{OSC} or the reciprocal of f_{cp}. For example, when f_{cp} is 270 kHz, 409.6 × 250/270 = 379.2 ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0.When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

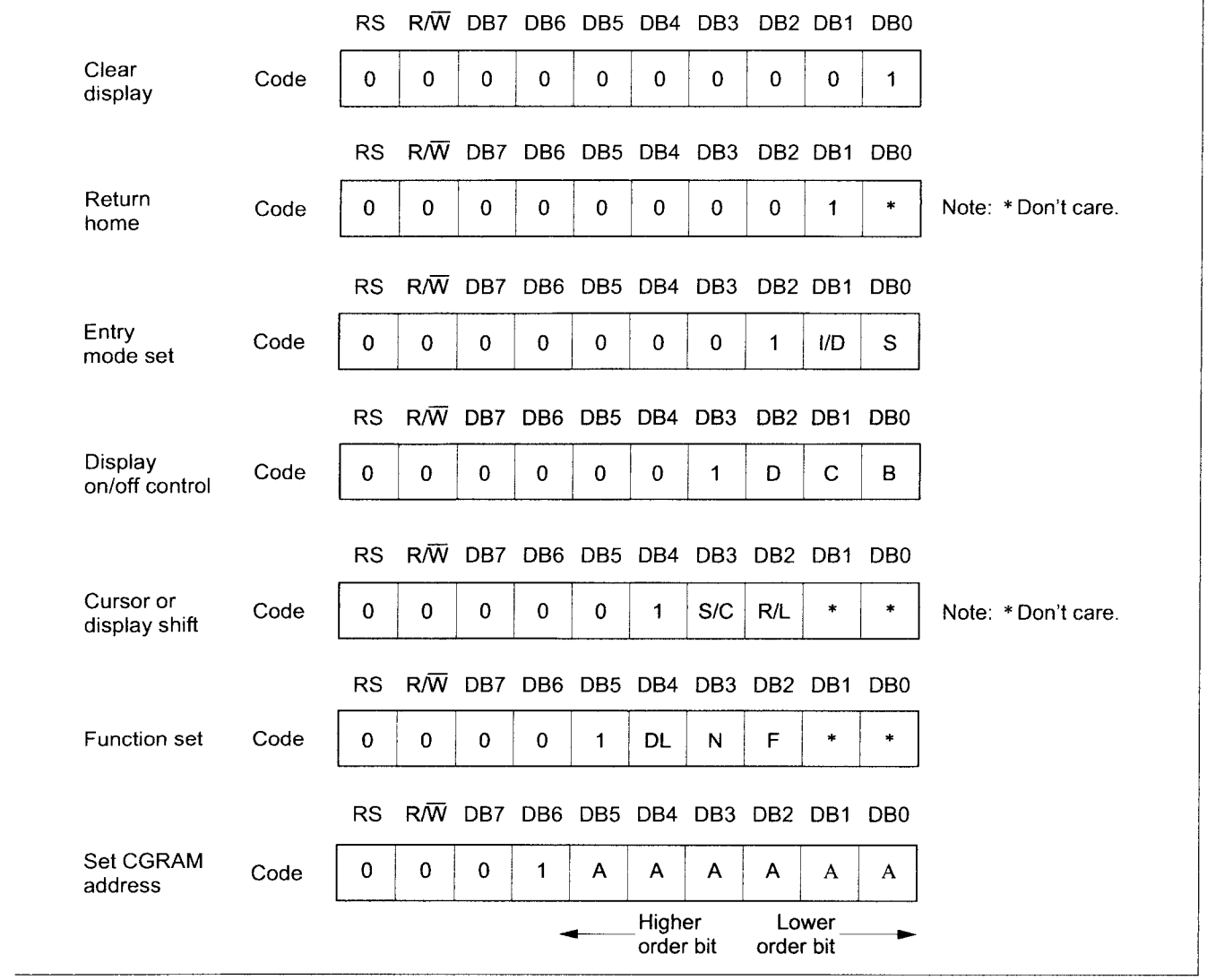


Figure 11 Instruction (1)

Set DDRAM Address

Set DDRAM address sets the DDRAM address binary AAAAAAA into the address counter.

Data is then written to or read from the MPU for DDRAM.

However, when N is 0 (1-line display), AAAAAAA can be 00H to 4FH. When N is 1 (2-line display), AAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

Read Busy Flag and Address

Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. If BF is 1, the internal operation is in progress. The next instruction will not be accepted until BF is reset to 0. Check the BF status before the next write operation. At the same time, the value of the address counter in binary AAAAAAA is read out. This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction. The address contents are the same as for instructions set CGRAM address and set DDRAM address.

Table 7 Shift Function

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

Table 8 Function Set

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

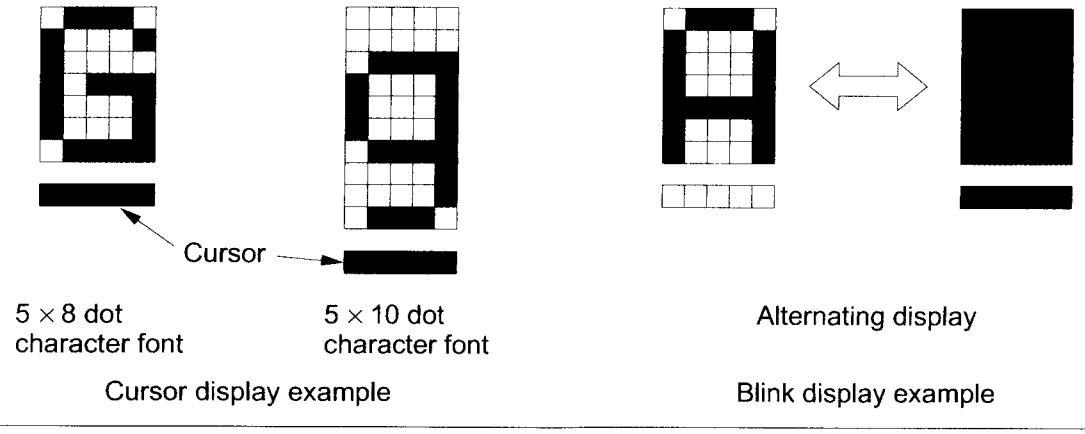


Figure 12 Cursor and Blinking

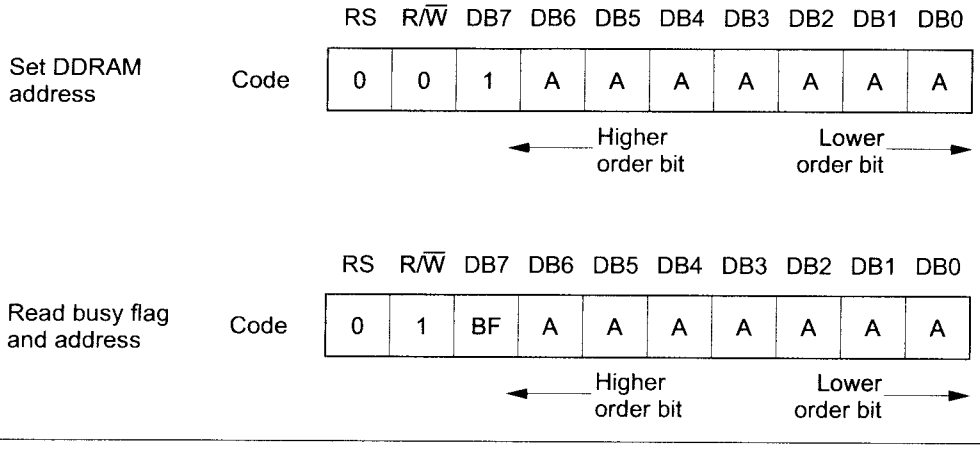


Figure 13 Instruction (2)

Write Data to CG or DDRAM

Write data to CG or DDRAM writes 8-bit binary data DDDDDDDD to CG or DDRAM.

To write into CG or DDRAM is determined by the previous specification of the CGRAM or DDRAM address setting. After a write, the address is automatically incremented or decremented by 1 according to the entry mode. The entry mode also determines the display shift.

Read Data from CG or DDRAM

Read data from CG or DDRAM reads 8-bit binary data DDDDDDDD from CG or DDRAM.

The previous designation determines whether CG or DDRAM is to be read. Before entering this read instruction, either CGRAM or DDRAM address set instruction must be executed. If not executed, the first read data will be invalid. When serially executing read instructions, the next address data is normally read from the second read. The address set instructions need not be executed just before this read instruction when shifting the cursor by the cursor shift instruction (when reading out DDRAM). The operation of the cursor shift instruction is the same as the set DDRAM address instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed regardless of the entry mode.

Note: The address counter (AC) is automatically incremented or decremented by 1 after the write instructions to CGRAM or DDRAM are executed. The RAM data selected by the AC cannot be read out at this time even if read instructions are executed. Therefore, to correctly read data, execute either the address set instruction or cursor shift instruction (only with DDRAM), then just before reading the desired data, execute the read instruction from the second time the read instruction is sent.

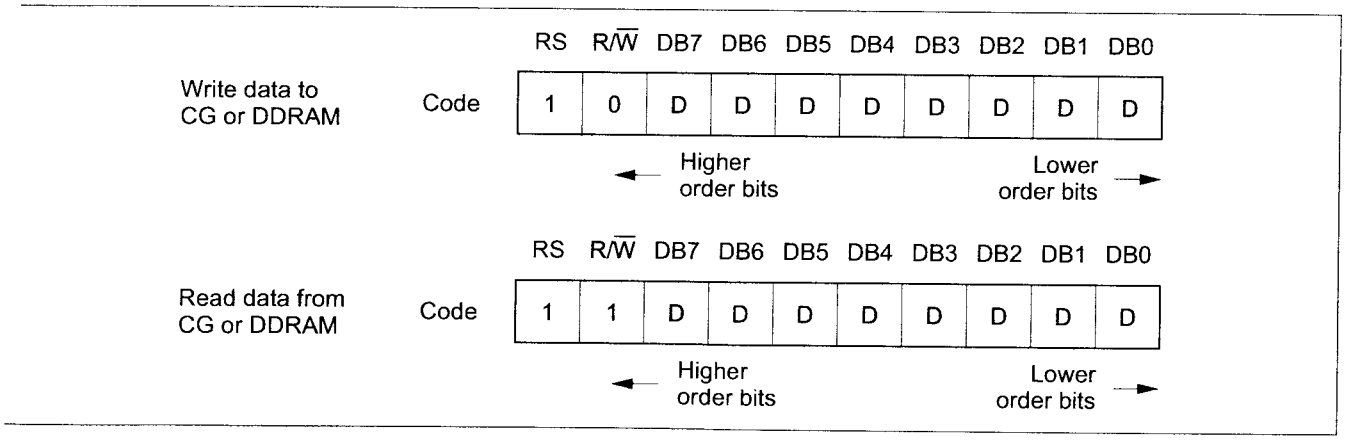


Figure 14 Instruction (3)

Interfacing the HD44780U

Interface to MPUs

- Interfacing to an 8-bit MPU

  See Figure 16 for an example of using a I/O port (for a single-chip microcomputer) as an interface device.

  In this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected to E, R/$\overline{W}$, and RS, respectively.

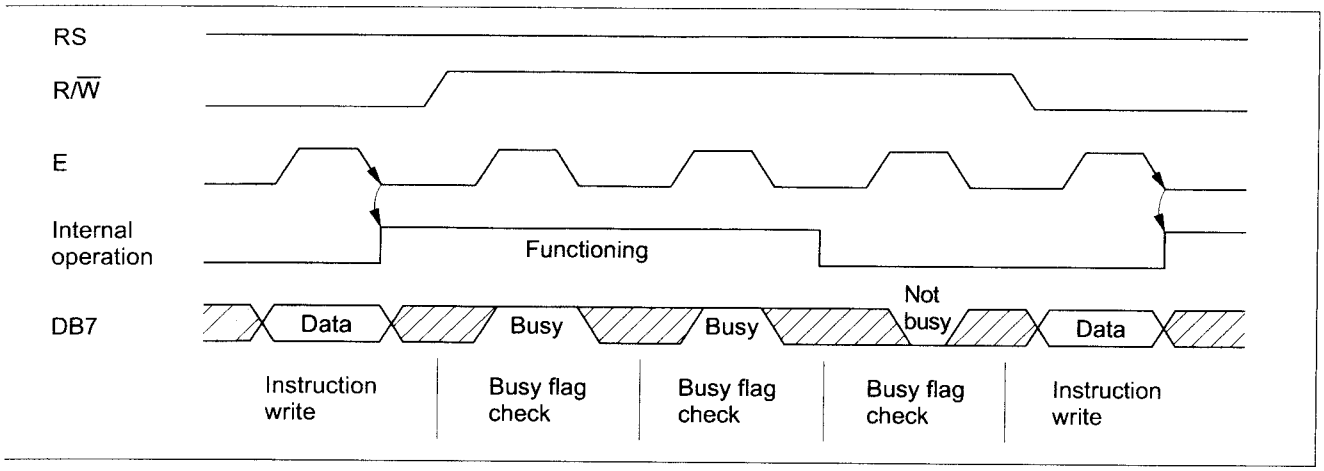


Figure 15 Example of Busy Flag Check Timing Sequence

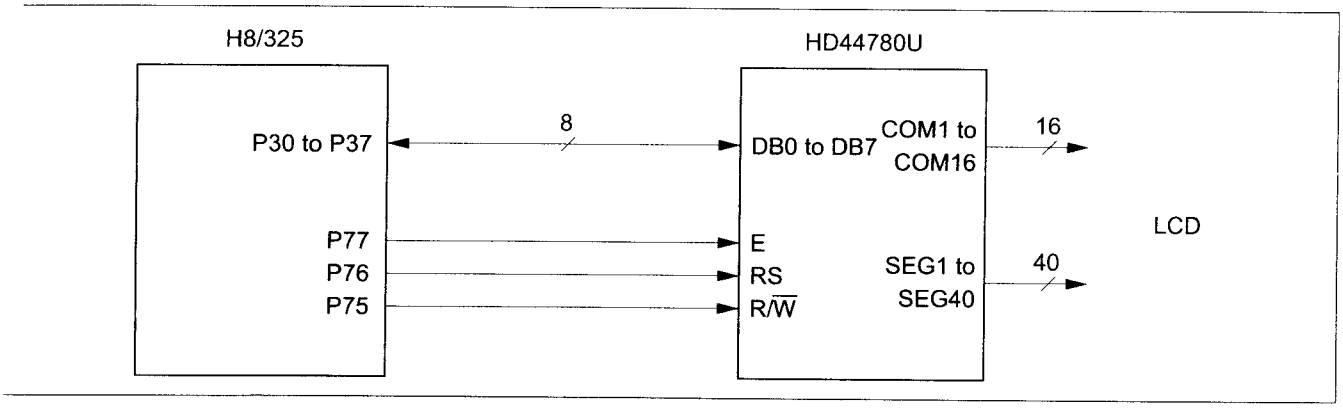


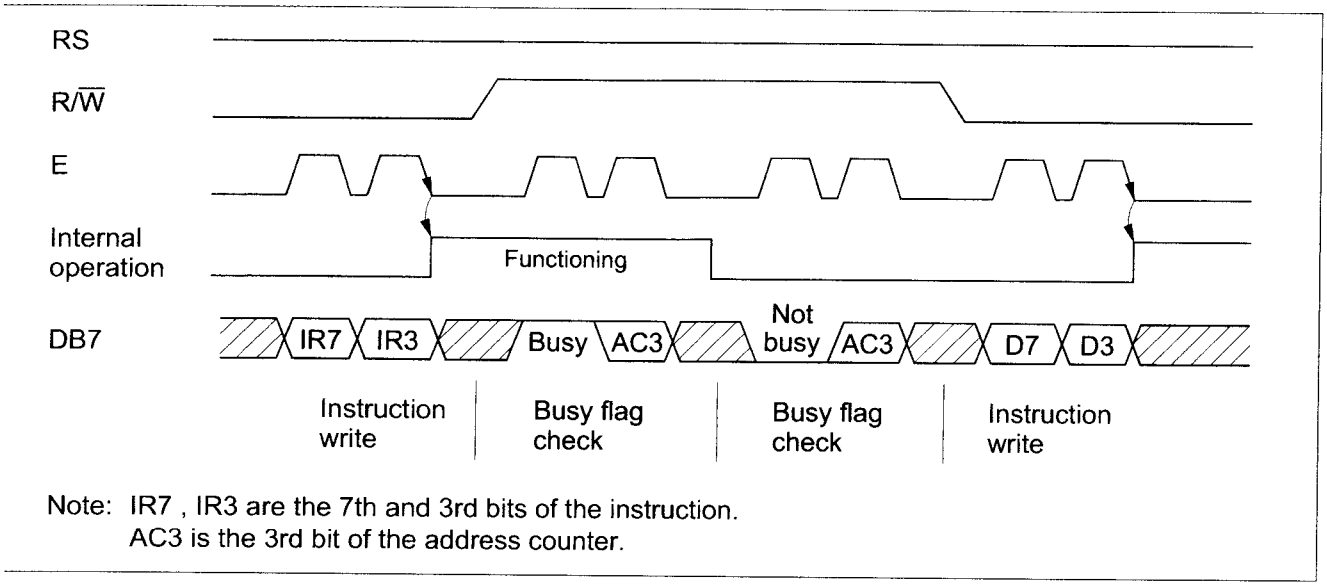
Figure 16 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

  The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 17.)

  See Figure 18 for an interface example to the HMCS4019R.

  Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.



Note: IR7 , IR3 are the 7th and 3rd bits of the instruction.
AC3 is the 3rd bit of the address counter.

Figure 17 Example of 4-Bit Data Transfer Timing Sequence

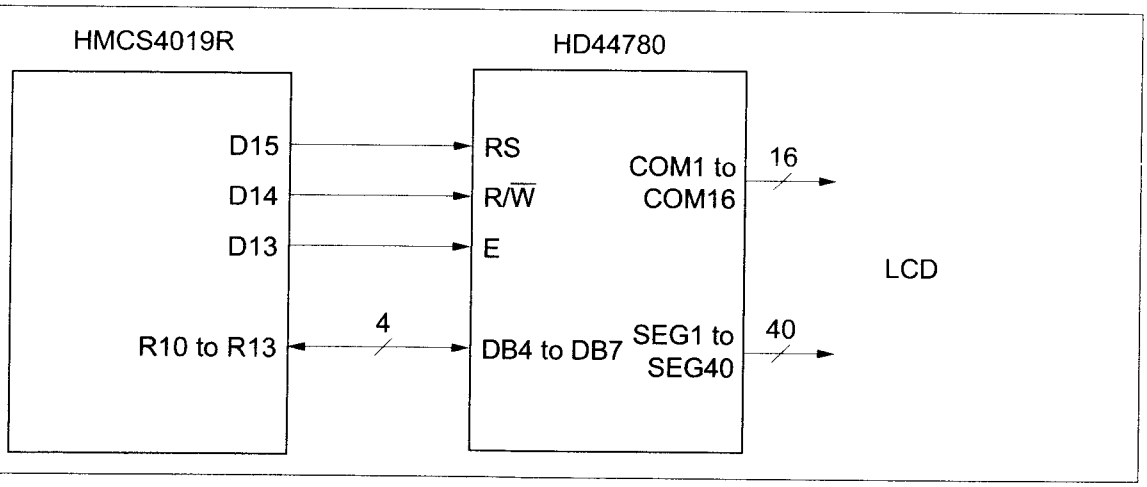


Figure 18 Example of Interface to HMCS4019R

Interface to Liquid Crystal Display

Character Font and Number of Lines: The HD44780U can perform two types of displays, 5 × 8 dot and 5 × 10 dot character fonts, each with a cursor.

Up to two lines are displayed for 5 × 8 dots and one line for 5 × 10 dots. Therefore, a total of three types of common signals are available (Table 9).

The number of lines and font types can be selected by the program. (See Table 6, Instructions.)

Connection to HD44780 and Liquid Crystal Display: See Figure 19 for the connection examples.

Table 9 Common Signals

Number of Lines	Character Font	Number of Common Signals	Duty Factor
	5 × 8 dots + cursor	8	1/8
	5 × 10 dots + cursor	11	1/11
	5 × 8 dots + cursor	16	1/16

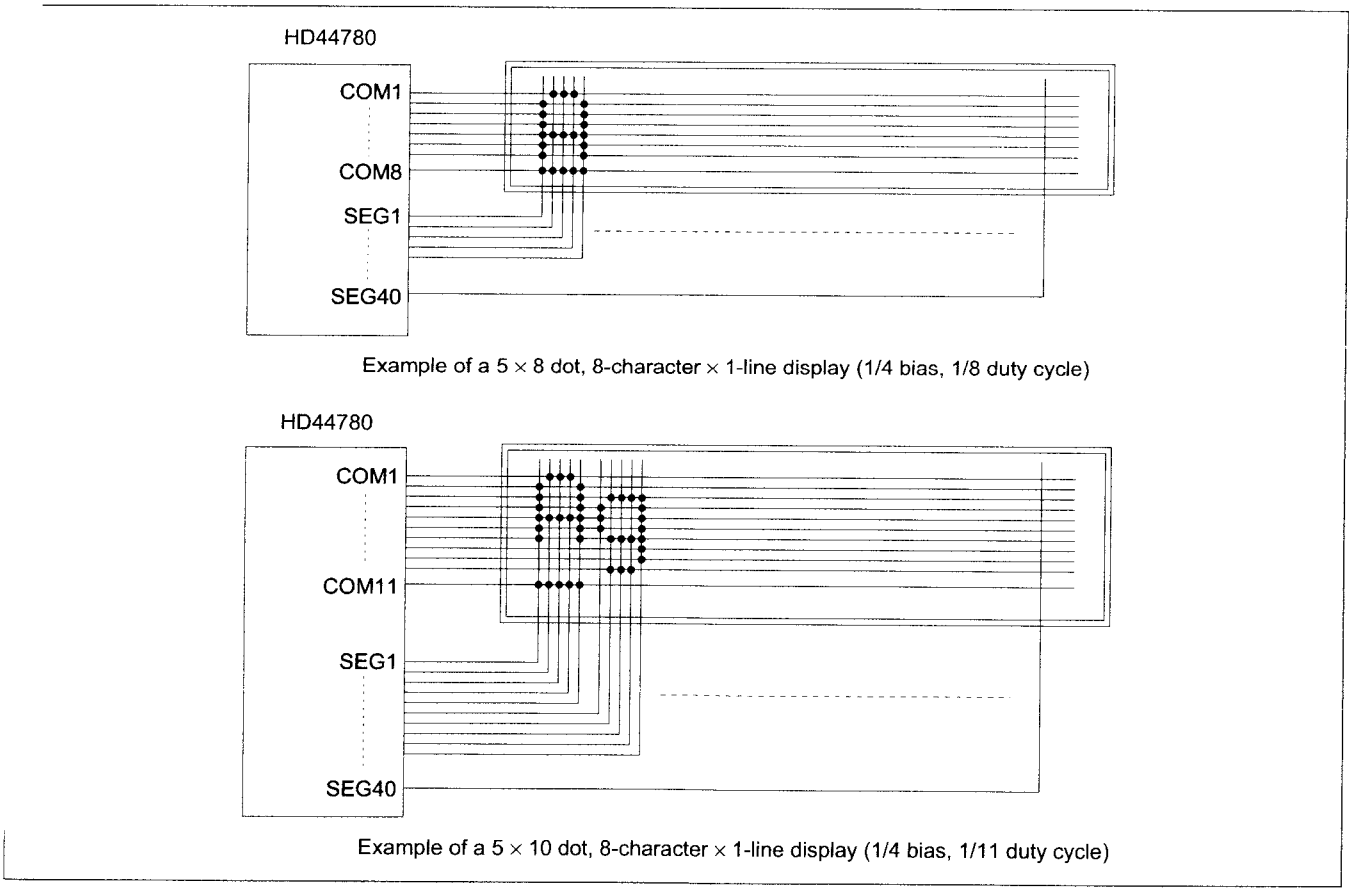
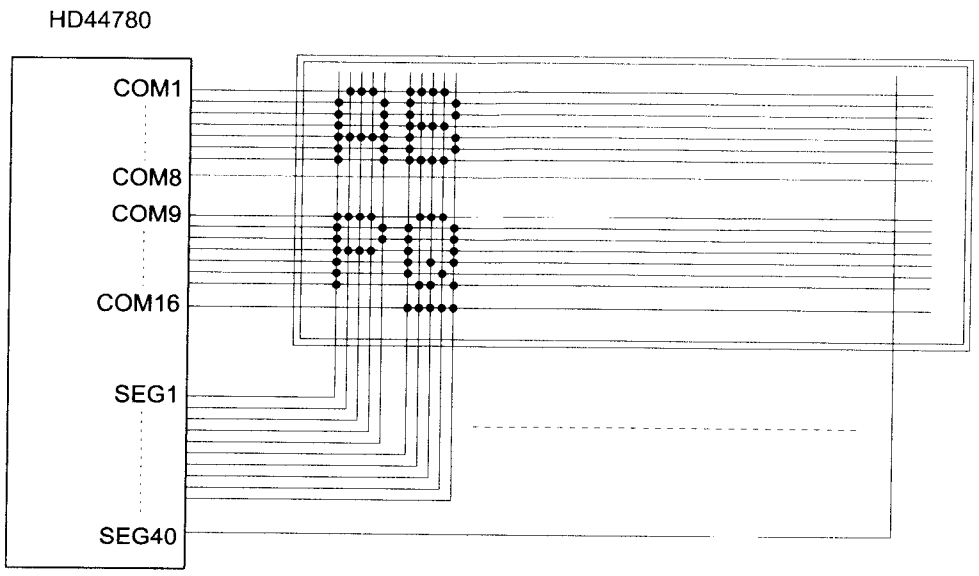


Figure 19 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 19 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state.



Example of a 5 × 8 dot, 8-character × 2-line display (1/5 bias, 1/16 duty cycle)

Figure 19 Liquid Crystal Display and HD44780 Connections (cont)

Connection of Changed Matrix Layout: In the preceding examples, the number of lines correspond to the scanning lines. However, the following display examples (Figure 20) are made possible by altering the matrix layout of the liquid crystal display panel. In either case, the only change is the layout. The display characteristics and the number of liquid crystal display characters depend on the number of common signals or on duty factor. Note that the display data RAM (DDRAM) addresses for 4 characters × 2 lines and for 16 characters × 1 line are the same as in Figure 19.

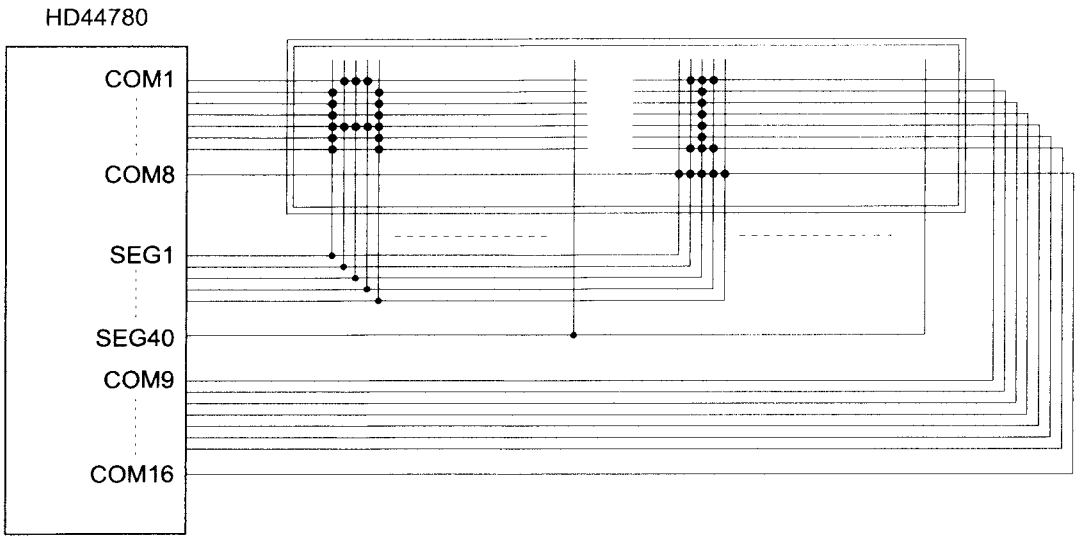


Figure 20 Changed Matrix Layout Displays

Power Supply for Liquid Crystal Display Drive

Various voltage levels must be applied to pins V1 to V5 of the HD44780U to obtain the liquid crystal display drive waveforms. The voltages must be changed according to the duty factor (Table 10).

VLCD is the peak value for the liquid crystal display drive waveforms, and resistance dividing provides voltages V1 to V5 (Figure 21).

Table 10 Duty Factor and Power Supply for Liquid Crystal Display Drive

	Duty Factor	
	1/8, 1/11	1/16
	Bias	
Power Supply	1/4	1/5
/1	V_{CC}–1/4 VLCD	V_{CC}–1/5 VLCD
/2	V_{CC}–1/2 VLCD	V_{CC}–2/5 VLCD
/3	V_{CC}–1/2 VLCD	V_{CC}–3/5 VLCD
/4	V_{CC}–3/4 VLCD	V_{CC}–4/5 VLCD
/5	V_{CC}–VLCD	V_{CC}–VLCD

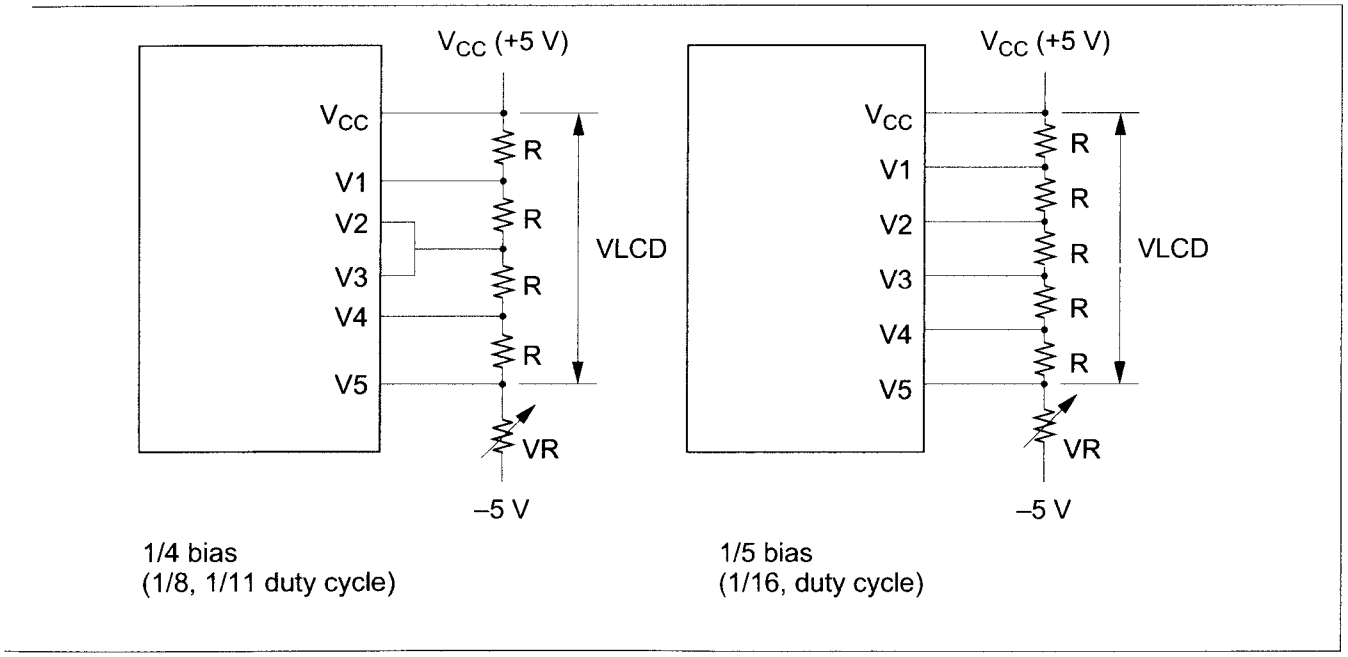
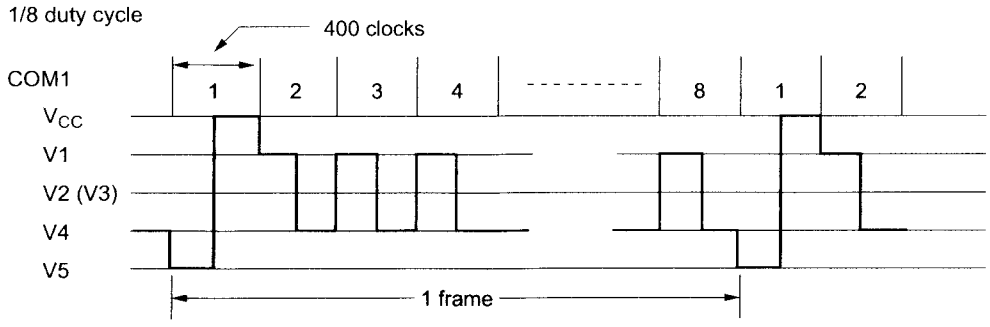


Figure 21 Drive Voltage Supply Example

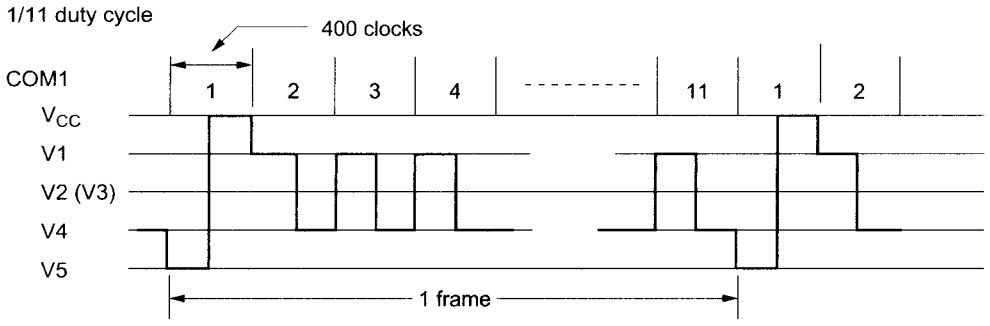
Relationship between Oscillation Frequency and Liquid Crystal Display Frame Frequency

The liquid crystal display frame frequencies of Figure 22 apply only when the oscillation frequency is 270 kHz (one clock pulse of 3.7 µs).



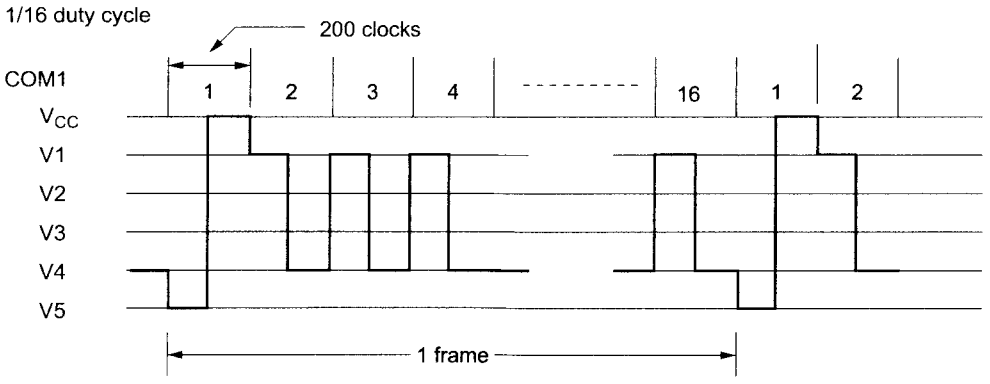
1 frame = 3.7 µs × 400 × 8 = 11850 µs = 11.9 ms

Frame frequency = $\frac{1}{11.9\ \text{ms}}$ = 84.3 Hz



1 frame = 3.7 µs × 400 × 11 = 16300 µs = 16.3 ms

Frame frequency = $\frac{1}{16.3\ \text{ms}}$ = 61.4 Hz



1 frame = 3.7 µs × 200 × 16 = 11850 µs = 11.9 ms

Frame frequency = $\frac{1}{11.9\ \text{ms}}$ = 84.3 Hz

Figure 22 Frame Frequency

Instruction and Display Correspondence

- 8-bit operation, 8-digit × 1-line display with internal reset

  Refer to Table 11 for an example of an 8-digit × 1-line display in 8-bit operation. The HD44780U functions must be set by the function set instruction prior to the display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays such as for advertising when combined with the display shift operation.

  Since the display shift operation changes only the display position with DDRAM contents unchanged, the first display data entered into DDRAM can be output when the return home operation is performed.

- 4-bit operation, 8-digit × 1-line display with internal reset

  The program must set all functions prior to the 4-bit operation (Table 12). When the power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since DB0 to DB3 are not connected, a rewrite is then required. However, since one operation is completed in two accesses for 4-bit operation, a rewrite is needed to set the functions (see Table 12). Thus, DB4 to DB7 of the function set instruction is written twice.

- 8-bit operation, 8-digit × 2-line display

  For a 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the first line has been written. Thus, if there are only 8 characters in the first line, the DDRAM address must be again set after the 8th character is completed. (See Table 13.) Note that the display shift operation is performed for the first and second lines. In the example of Table 13, the display shift is performed when the cursor is on the second line. However, if the shift operation is performed when the cursor is on the first line, both the first and second lines move together. If the shift is repeated, the display of the second line will not move to the first line. The same display will only shift within its own line for the number of times the shift is repeated.

Note: When using the internal reset, the electrical characteristics in the Power Supply Conditions Using Internal Reset Circuit table must be satisfied. If not, the HD44780U must be initialized by instructions. See the section, Initializing by Instruction.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set	0	0	0	0	1	1	0	0	*	*		Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
3	Display on/off control	0	0	0	0	0	0	1	1	1	0	_	Turns on display and cursor. Entire display is in space mode because of initialization.
4	Entry mode set	0	0	0	0	0	0	0	1	1	0	_	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	H_	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HI_	Writes I.
7	. . . . .										. . . . .	
8	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HITACHI_	Writes I.
9	Entry mode set	0	0	0	0	0	0	0	1	1	1	HITACHI_	Sets mode to shift display at the time of write.
10	Write data to CGRAM/DDRAM	1	0	0	0	1	0	0	0	0	0	ITACHI _	Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step No.	Instruction RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
11	Write data to CGRAM/DDRAM										TACHI M_	Writes M.
	1	0	0	1	0	0	1	1	0	1		
12												
13	Write data to CGRAM/DDRAM										MICROKO_	Writes O.
	1	0	0	1	0	0	1	1	1	1		
4	Cursor or display shift										MICROKO	Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*		
5	Cursor or display shift										MICROKO	Shifts only the cursor position to the left.
	0	0	0	0	0	1	0	0	*	*		
6	Write data to CGRAM/DDRAM										ICROCO	Writes C over K. The display moves to the left.
	1	0	0	1	0	0	0	0	1	1		
7	Cursor or display shift										MICROCO	Shifts the display and cursor position to the right.
	0	0	0	0	0	1	1	1	*	*		
8	Cursor or display shift										MICROCO_	Shifts the display and cursor position to the right.
	0	0	0	0	0	1	0	1	*	*		
19	Write data to CGRAM/DDRAM										ICROCOM_	Writes M.
	1	0	0	1	0	0	1	1	0	1		
20												
21	Return home										HITACHI	Returns both display and cursor to the original position (address 0).
	0	0	0	0	0	0	0	0	1	0		

Table 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction RS	R/W	DB7	DB6	DB5	DB4	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)							Initialized. No display.
2	Function set							Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
	0	0	0	0	1	0		
3	Function set							Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
	0	0	0	0	1	0		
	0	0	0	0	*	*		
	Display on/off control						_	Turns on display and cursor. Entire display is in space mode because of initialization.
	0	0	0	0	0	0		
	0	0	1	1	1	0		
5	Entry mode set						_	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
	0	0	0	0	0	0		
	0	0	0	1	1	0		
6	Write data to CGRAM/DDRAM						H_	Writes H. The cursor is incremented by one and shifts to the right.
	1	0	0	1	0	0		
	1	0	1	0	0	0		

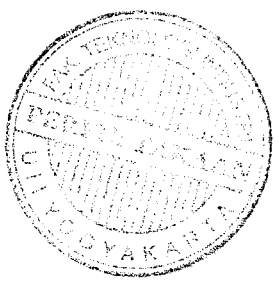
Note: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step No.	Instruction RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set	0	0	0	0	1	1	1	0	*	*	Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
3	Display on/off control	0	0	0	0	0	0	1	1	1	0	Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set	0	0	0	0	0	0	0	1	1	0	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	0	H_ Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
	.											
	.											
	.											
	.											
	.											
7	Write data to CGRAM/DDRAM	1	0	0	1	0	0	1	0	0	1	HITACHI_ Writes I.
8	Set DDRAM address	0	0	1	1	0	0	0	0	0	0	HITACHI Sets DDRAM address so that the cursor is positioned at the head of the second line.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset (cont)

Step No.	Instruction RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Display	Operation
9	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	HITACHI M_	Writes M.
10	.	.	.	.	.	.	.	.	.	.	.	
11	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	1	1	HITACHI MICROCO_	Writes O.
2	Entry mode set 0	0	0	0	0	0	0	1	1	1	HITACHI MICROCO_	Sets mode to shift display at the time of write.
3	Write data to CGRAM/DDRAM 1	0	0	1	0	0	1	1	0	1	ITACHI ICROCOM_	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
4	.	.	.	.	.	.	.	.	.	.	.	
15	Return home 0	0	0	0	0	0	0	0	1	0	HITACHI MICROCOM	Returns both display and cursor to the original position (address 0).



Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 23 and 24 for the procedures on 8-bit and 4-bit initializations, respectively.

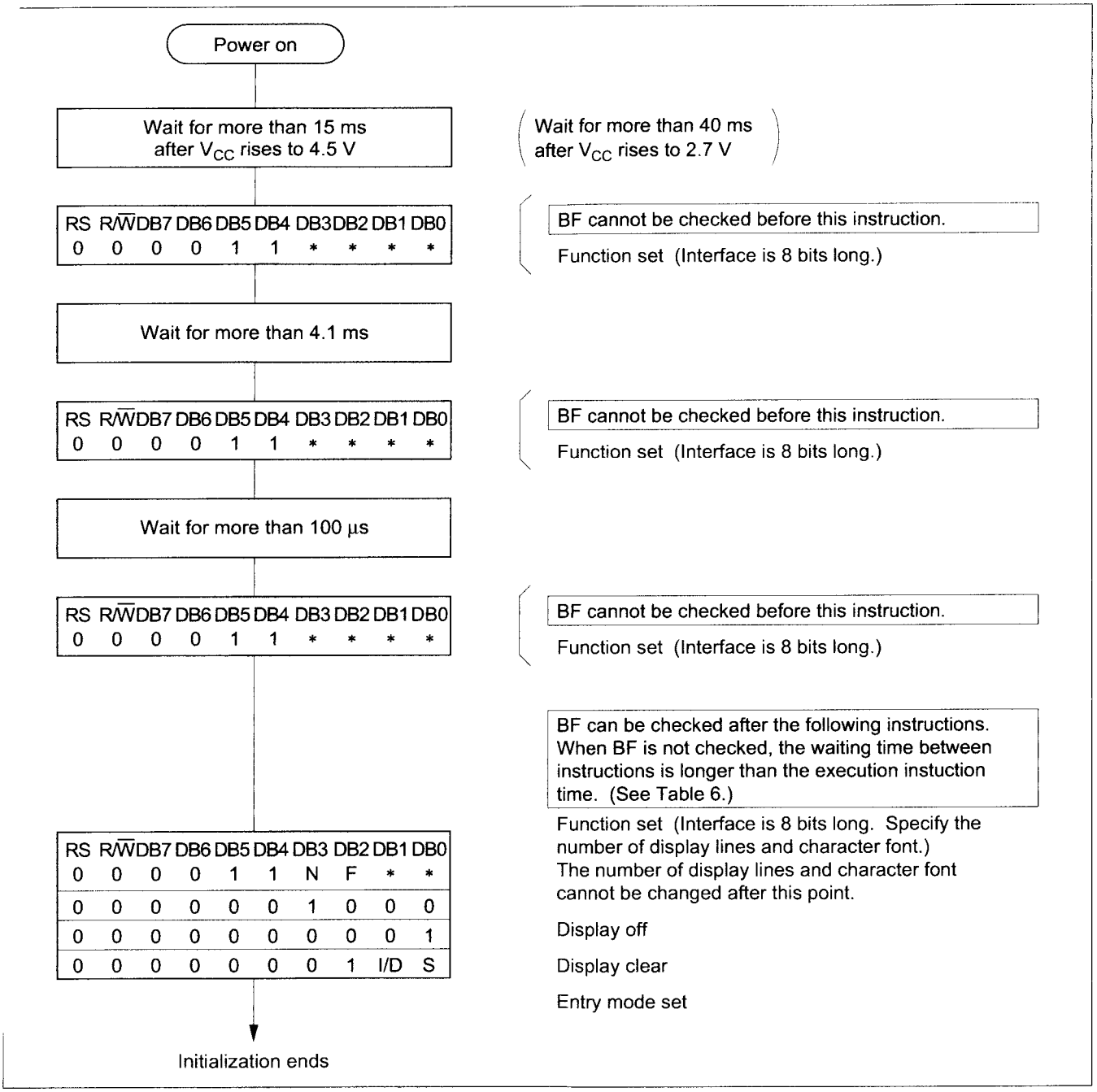


Figure 23 8-Bit Interface

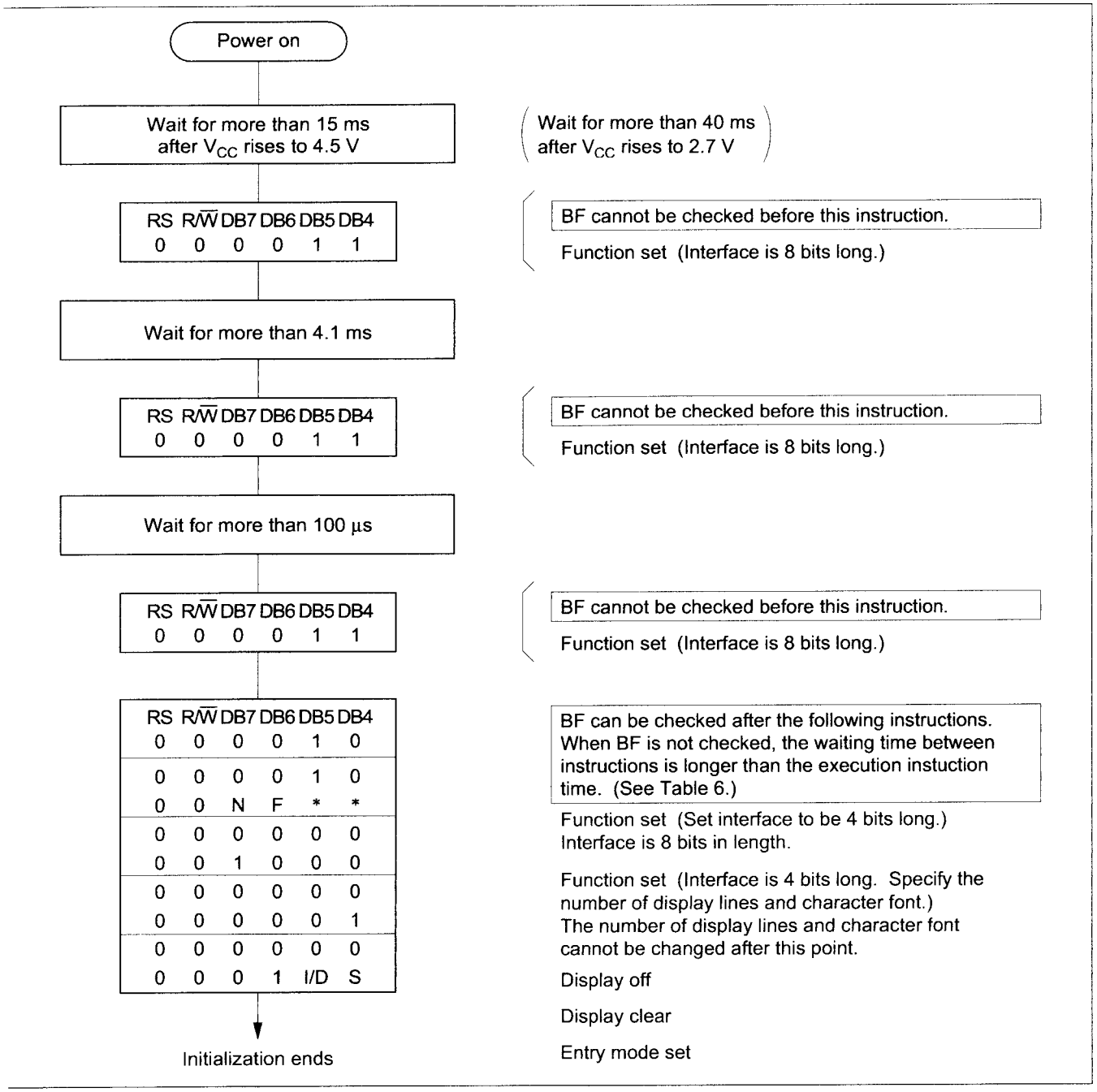


Figure 24 4-Bit Interface

Absolute Maximum Ratings*

Item	Symbol	Value	Unit	Notes
Power supply voltage (1)	V_{CC}–GND	–0.3 to +7.0	V	1
Power supply voltage (2)	V_{CC}–V5	–0.3 to +13.0	V	1, 2
Input voltage	Vt	–0.3 to V_{CC} +0.3	V	1
Operating temperature	T_{opr}	–30 to +75	°C	
Storage temperature	T_{stg}	–55 to +125	°C	4

Note: * If the LSI is used above these absolute maximum ratings, it may become permanently damaged. Using the LSI within the following electrical characteristic limits is strongly recommended for normal operation. If these electrical characteristic conditions are also exceeded, the LSI will malfunction and cause poor reliability.

DC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –30 to +75°C*[3])

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	VIH2	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	VOH1	$0.75V_{CC}$	—	—	V	$-I_{OH}$ = 0.1 mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	$0.2V_{CC}$	V	I_{OL} = 0.1 mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.8V_{CC}$	—	—	V	$-I_{OH}$ = 0.04 mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.2V_{CC}$	V	I_{OL} = 0.04 mA	8
Driver on resistance (COM)	R_{COM}	—	2	20	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	R_{SEG}	—	2	30	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/$\overline{W}$)	$-I_p$	10	50	120	µA	V_{CC} = 3 V	
Power supply current	I_{CC}	—	150	300	µA	R_f oscillation, external clock V_{CC} = 3 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics (V_{CC} = 2.7 to 4.5 V, T_a = –30 to +75°C*[3])

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rcp}	—	—	0.2	µs		
	External clock fall time	t_{fcp}	—	—	0.2	µs		
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	R_f = 75 kΩ, V_{CC} = 3 V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	25		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	360		
Data hold time	t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	–1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	200		

Power Supply Conditions Using Internal Reset Circuit

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time	t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time	t_{OFF}	1	—	—		

DC Characteristics (V_{CC} = 4.5 to 5.5 V, T_a = –30 to +75°C*[3])

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	–0.3	—	0.6	V		6
nput high voltage (2) OSC1)	VIH2	V_{CC}–1.0	—	V_{CC}	V		15
nput low voltage (2) OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) DB0–DB7)	VOH1	2.4	—	—	V	$-I_{OH}$ = 0.205 mA	7
Output low voltage (1) DB0–DB7)	VOL1	—	—	0.4	V	I_{OL} = 1.2 mA	7
Output high voltage (2) except DB0–DB7)	VOH2	0.9 V_{CC}	—	—	V	$-I_{OH}$ = 0.04 mA	8
Output low voltage (2) except DB0–DB7)	VOL2	—	—	0.1 V_{CC}	V	I_{OL} = 0.04 mA	8
Driver on resistance COM)	RCOM	—	2	20	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Driver on resistance (SEG)	RSEG	—	2	30	kΩ	±Id = 0.05 mA, VLCD = 4 V	13
Input leakage current	I_{LI}	–1	—	1	µA	VIN = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/$\overline{W}$)	$-I_p$	50	125	250	µA	V_{CC} = 5 V	
Power supply current	I_{CC}	—	350	600	µA	R_f oscillation, external clock V_{CC} = 5 V, f_{OSC} = 270 kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	V_{CC}–V5, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	V_{CC}–V5, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics (V_{CC} = 4.5 to 5.5 V, T_a = –30 to +75°C*[3])

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		11
	External clock rise time	t_{rcp}	—	—	0.2	µs		11
	External clock fall time	t_{fcp}	—	—	0.2	µs		11
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	R_f = 91 kΩ V_{CC} = 5.0 V	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

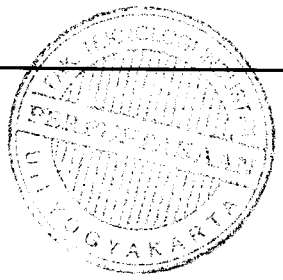
Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data set-up time	t_{DSW}	80	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/$\overline{W}$ to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data delay time	t_{DDR}	—	—	160		
Data hold time	t_{DHR}	5	—	—		



Interface Timing Characteristics with External Driver

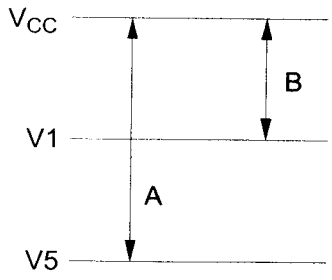
Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	–1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	100		

Power Supply Conditions Using Internal Reset Circuit

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time	t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time	t_{OFF}	1	—	—		

Electrical Characteristics Notes

1. All voltage values are referred to GND = 0 V.

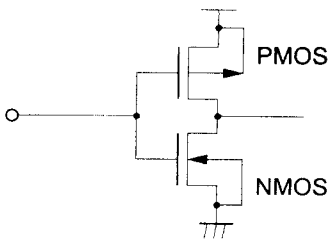


$A = V_{CC} - V5$
$B = V_{CC} - V1$
$A \geq 1.5$ V
$B \leq 0.25 \times A$

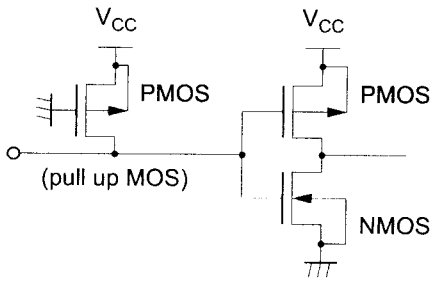
The conditions of V1 and V5 voltages are for proper operation of the LSI and not for the LCD output level. The LCD drive voltage condition for the LCD output level is specified as LCD voltage VLCD.

2. $V_{CC} \geq V1 \geq V2 \geq V3 \geq V4 \geq V5$ must be maintained.
3. For die products, specified at 75°C.
4. For die products, specified by the die shipment specification.
5. The following four circuits are I/O pin configurations except for liquid crystal display output.

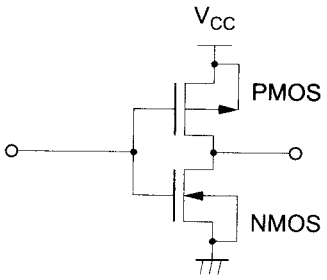
Input pin
Pin: E (MOS without pull-up)



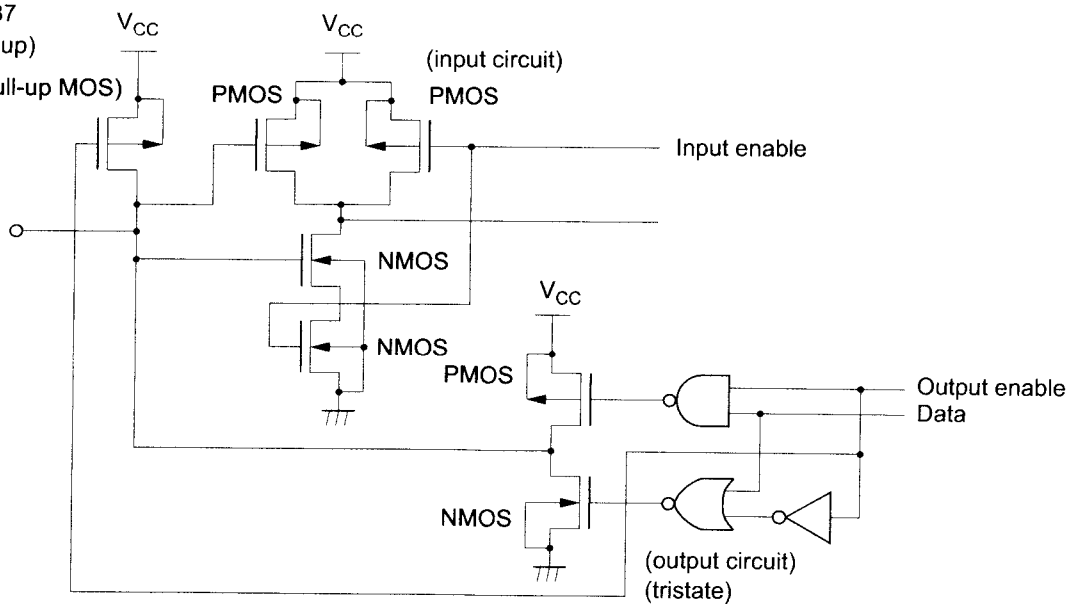
Pins: RS, R/$\overline{W}$ (MOS with pull-up)



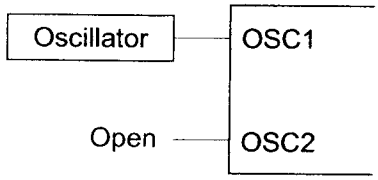
Output pin
Pins: CL1, CL2, M, D

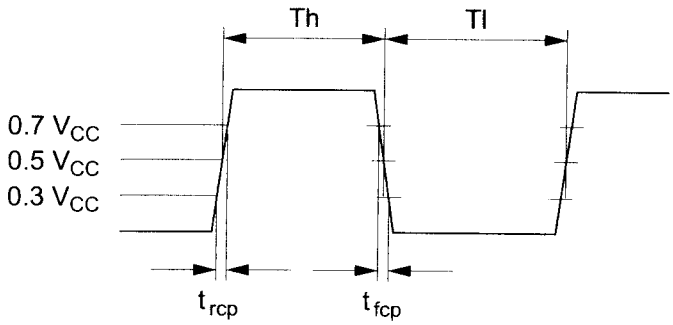


I/O Pin
Pins: DB0 –DB7
(MOS with pull-up)



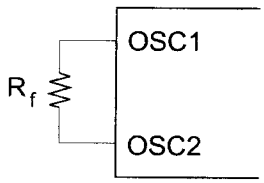
6. Applies to input pins and I/O pins, excluding the OSC1 pin.
7. Applies to I/O pins.
8. Applies to output pins.
9. Current flowing through pull–up MOSs, excluding output drive MOSs.
10. Input/output current is excluded. When input is at an intermediate level with CMOS, the excessive current flows through the input circuit to the power supply. To avoid this from happening, the input level must be fixed high or low.
11. Applies only to external clock operation.



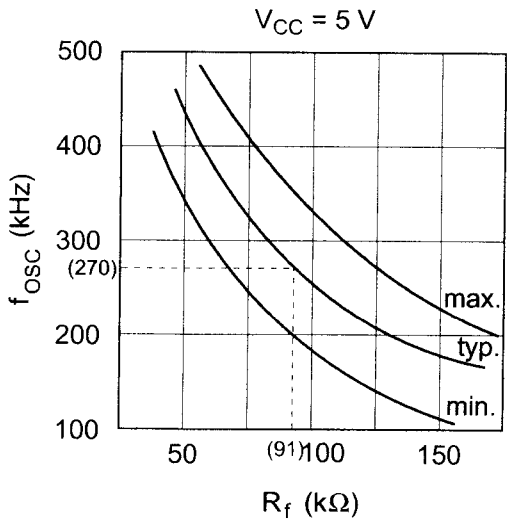


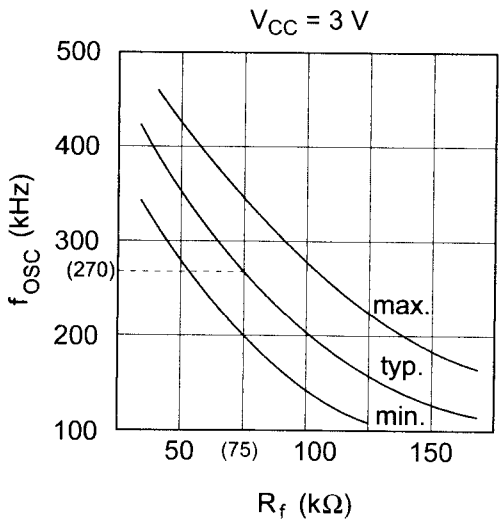
$$\text{Duty} = \frac{Th}{Th + Tl} \times 100\%$$

12. Applies only to the internal oscillator operation using oscillation resistor R_f.



R_f: 75 kΩ ± 2% (when V_{CC} = 3 V)
R_f: 91 kΩ ± 2% (when V_{CC} = 5 V)
Since the oscillation frequency varies depending on the OSC1 and OSC2 pin capacitance, the wiring length to these pins should be minimized.

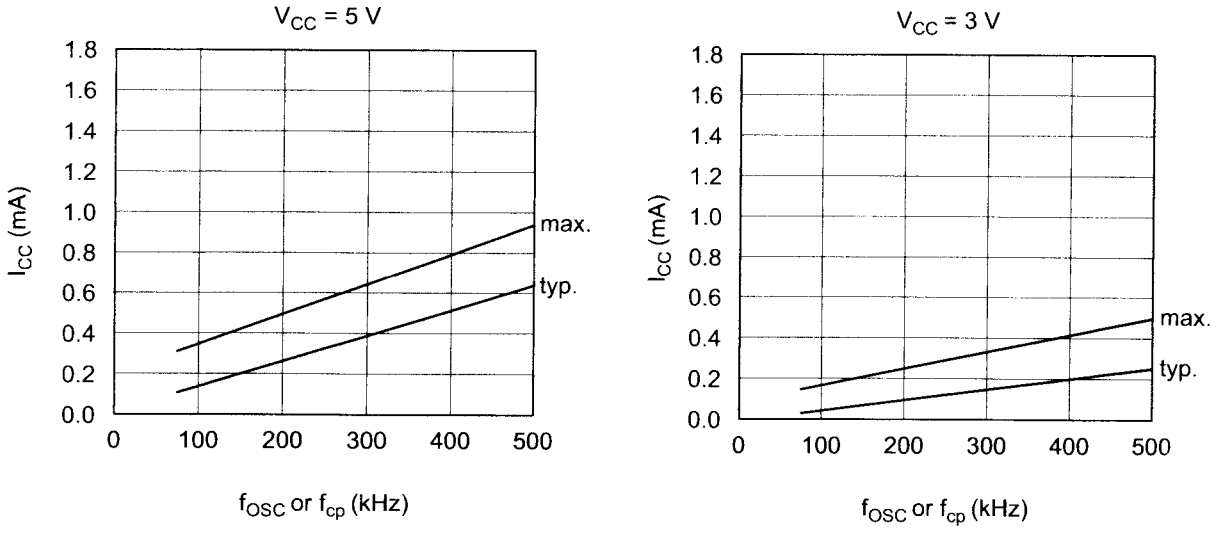




13. RCOM is the resistance between the power supply pins (V_{CC}, V1, V4, V5) and each common signal pin (COM1 to COM16).

    RSEG is the resistance between the power supply pins (V_{CC}, V2, V3, V5) and each segment signal pin (SEG1 to SEG40).

14. The following graphs show the relationship between operation frequency and current consumption.

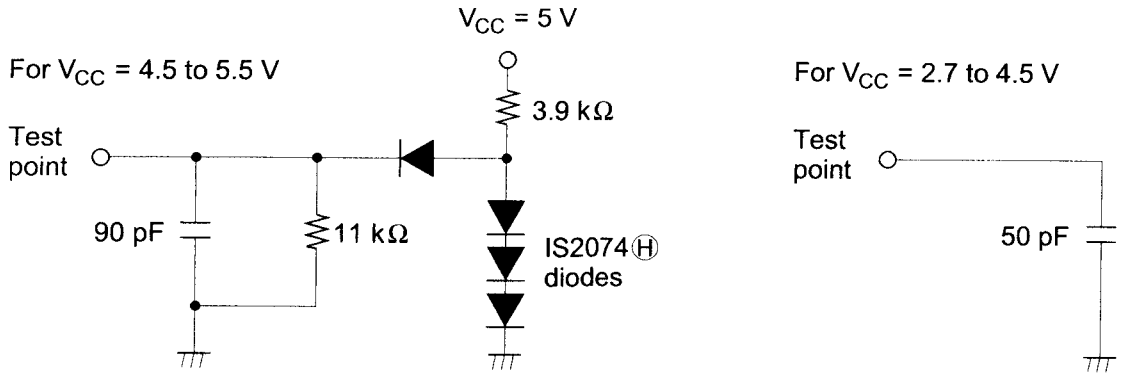


15. Applies to the OSC1 pin.

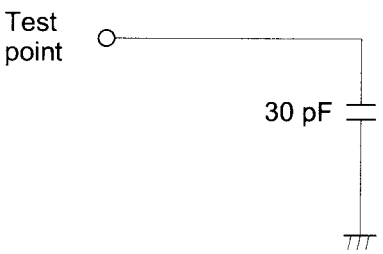
16. Each COM and SEG output voltage is within ±0.15 V of the LCD voltage (V_{CC}, V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



Timing Characteristics

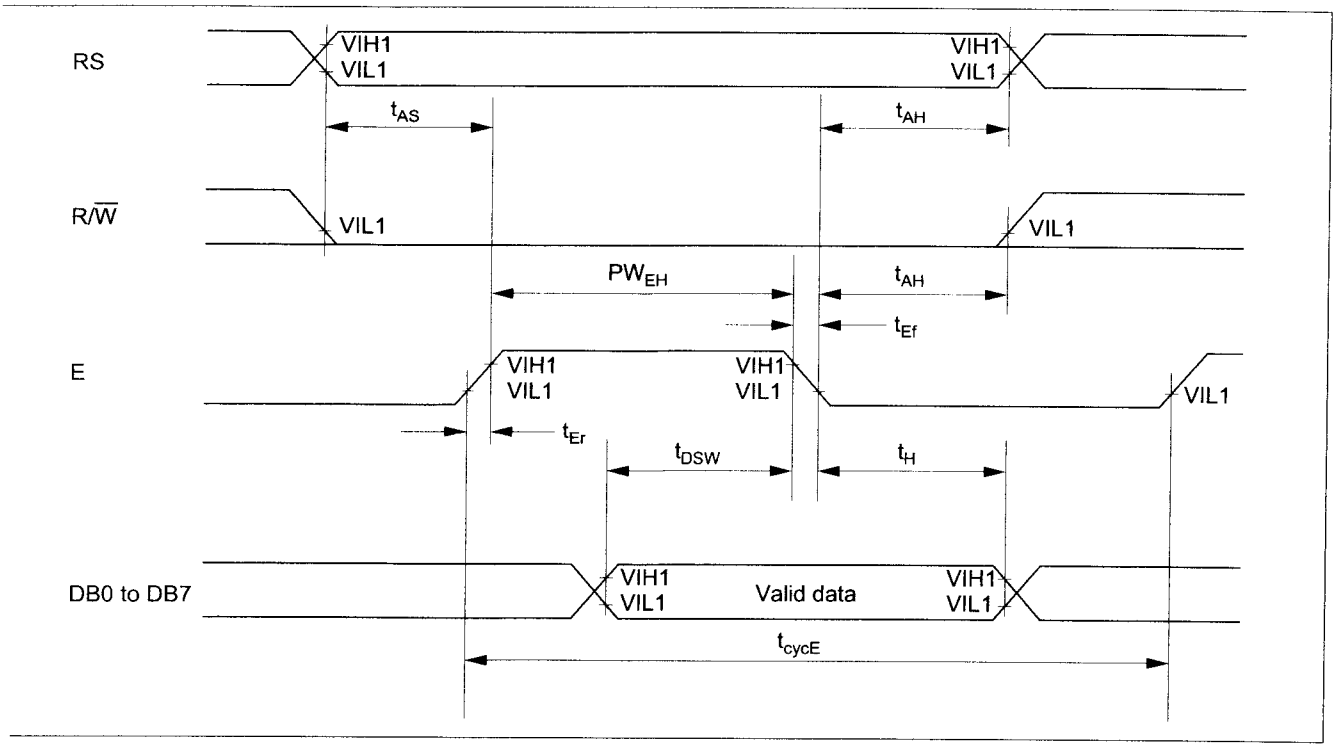


Figure 25 Write Operation

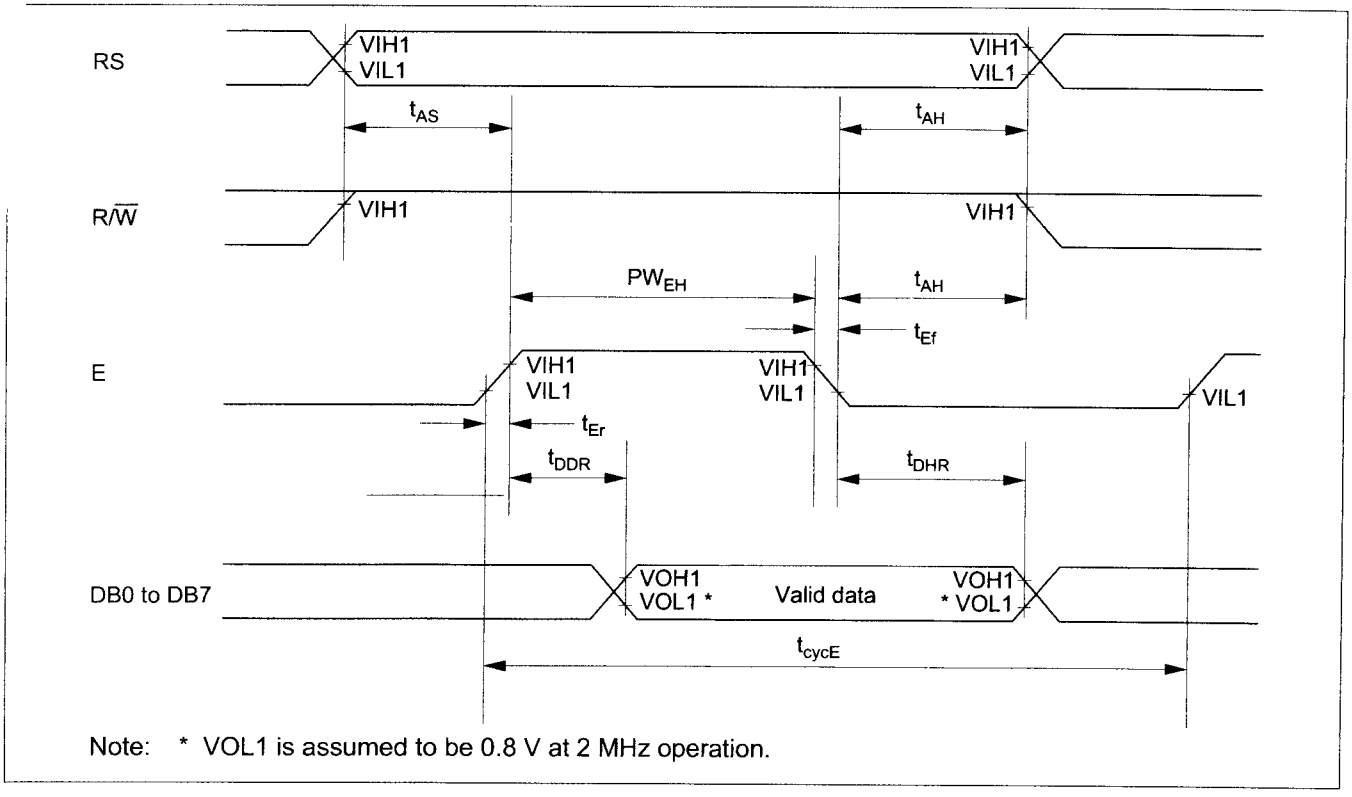


Figure 26 Read Operation

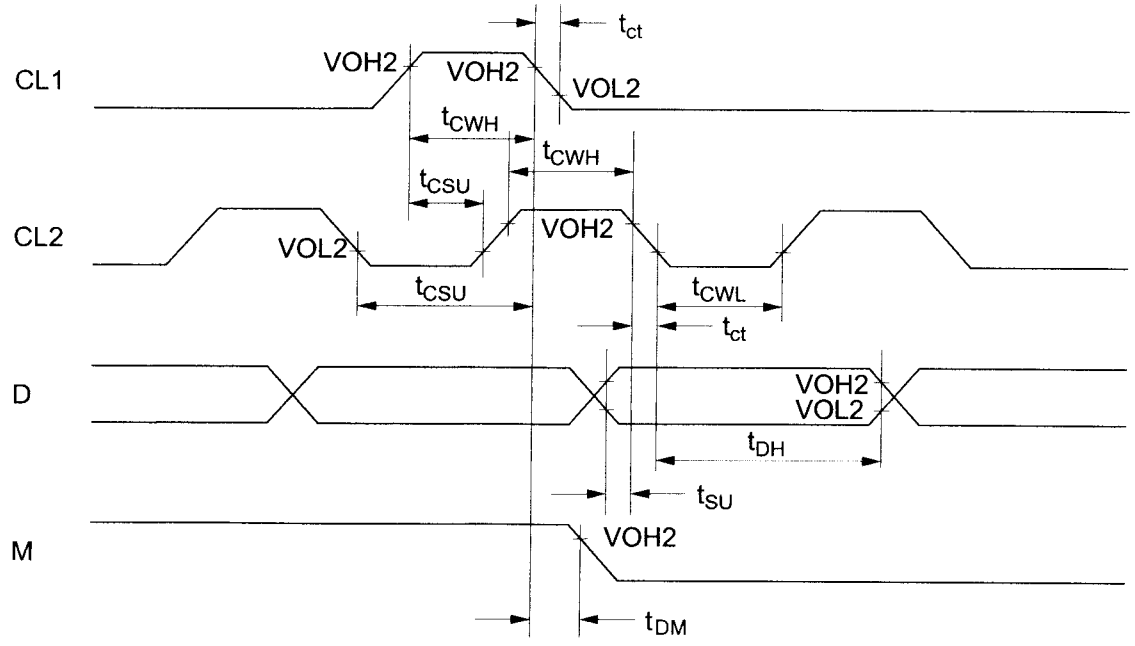
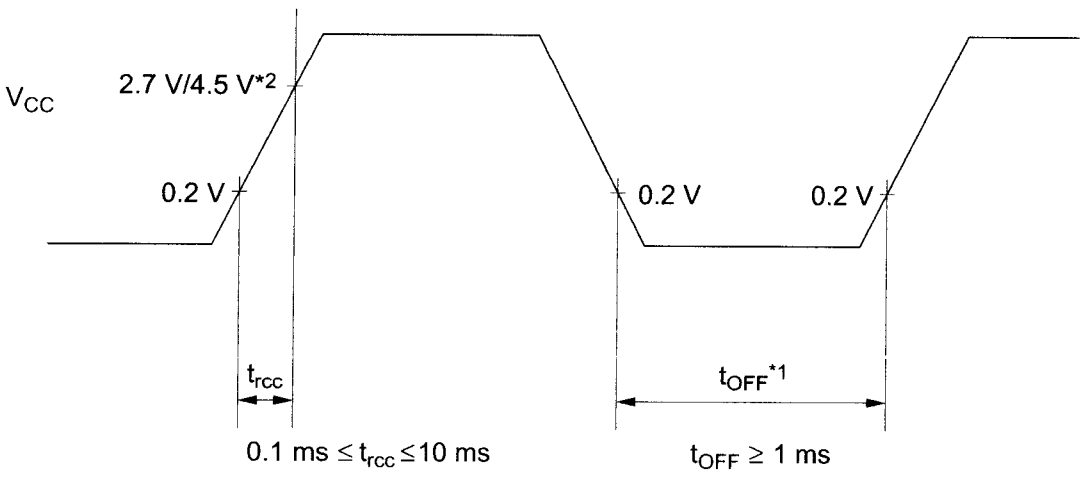


Figure 27 Interface Timing with External Driver



Notes:
1. t_{OFF} compensates for the power oscillation period caused by momentary power supply oscillations.
2. Specified at 4.5 V for 5-V operation, and at 2.7 V for 3-V operation.
3. For if 4.5 V is not reached during 5-V operation, the internal reset circuit will not operate normally.
   In this case, the LSI must be initialized by software. (Refer to the Initializing by Instruction section.)

Figure 28 Internal Power Supply Reset

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

HITACHI

Hitachi, Ltd.
Semiconductor & Integrated Circuits.
Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
Tel: Tokyo (03) 3270-2111 Fax: (03) 3270-5109

URL
NorthAmerica : http:semiconductor.hitachi.com/
Europe : http://www.hitachi-eu.com/hel/ecg
Asia (Singapore) : http://www.has.hitachi.com.sg/grp3/sicd/index.htm
Asia (Taiwan) : http://www.hitachi.com.tw/E/Product/SICD_Frame.htm
Asia (HongKong) : http://www.hitachi.com.hk/eng/bo/grp3/index.htm
Japan : http://www.hitachi.co.jp/Sicd/indx.htm

For further information write to:

Hitachi Semiconductor (America) Inc.
179 East Tasman Drive,
San Jose,CA 95134
Tel: <1> (408) 433-1990
Fax: <1>(408) 433-0223

Hitachi Europe GmbH
Electronic components Group
Dornacher Straße 3
D-85622 Feldkirchen, Munich
Germany
Tel: <49> (89) 9 9180-0
Fax: <49> (89) 9 29 30 00

Hitachi Europe Ltd.
Electronic Components Group.
Whitebrook Park
Lower Cookham Road
Maidenhead
Berkshire SL6 8YA, United Kingdom
Tel: <44> (1628) 585000
Fax: <44> (1628) 778322

Hitachi Asia Pte. Ltd.
16 Collyer Quay #20-00
Hitachi Tower
Singapore 049318
Tel: 535-2100
Fax: 535-1533

Hitachi Asia Ltd.
Taipei Branch Office
3F, Hung Kuo Building. No.167,
Tun-Hwa North Road, Taipei (105)
Tel: <886> (2) 2718-3666
Fax: <886> (2) 2718-8180

Hitachi Asia (Hong Kong) Ltd.
Group III (Electronic Components)
7/F., North Tower, World Finance Centre,
Harbour City, Canton Road, Tsim Sha Tsui,
Kowloon, Hong Kong
Tel: <852> (2) 735 9218
Fax: <852> (2) 730 0281
Telex: 40815 HITEC HX