

**RANCANG BANGUN *RUNNING TEXT* PADA LCD BERBASIS
FPGA XILINX SPARTAN II XC2S50-PQ208**

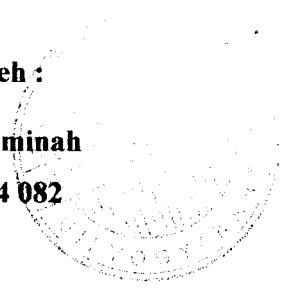
TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik
Program Studi Teknik Elektro, Fakultas Teknologi Industri
Universitas Islam Indonesia



Disusun Oleh :

NAMA : Siti Aminah
NIM : 01 524 082



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

YOGYAKARTA

2010

LEMBAR PENGESAHAN PEMBIMBING

**RANCANG BANGUN *RUNNING TEXT* PADA LCD BERBASIS FPGA
XILINX SPARTAN II XC2S50-PQ208**

TUGAS AKHIR

Disusun Oleh:

NAMA : Siti Aminah

NIM : 01 524 082

Disetujui :

Yogyakarta, April 2010

Pembimbing I



(Tito Yuwono, ST., MSc.) *XIV/147*

Pembimbing II



(Medilla Kusriyanto, ST., M.Eng.)

**RANCANG BANGUN *RUNNING TEXT* PADA LCD BERBASIS FPGA
XILINX SPARTAN II XC2S50-PQ208**

TUGAS AKHIR

Oleh

NAMA : Siti Aminah

NIM : 01 524 082

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta, _____

Tim Penguji,

Tito Yuwono, ST., M.Sc.

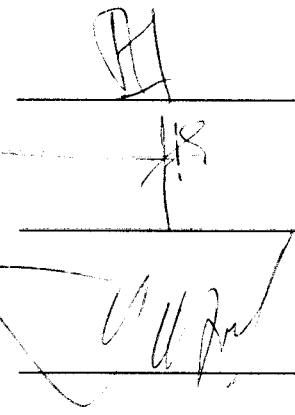
Ketua

Medilla Kusriyanto, ST., M. Eng.

Anggota I

Yusuf Aziz A, ST., M. Eng.

Anggota II



Three handwritten signatures are present, each written over a horizontal line. The first signature is at the top, the second in the middle, and the third at the bottom.

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



(Tito Yuwono, ST., M.Sc.)

MOTTO

**“ Allah tidak akan membebani seseorang, kecuali sesuai dengan kesanggupannya. Ia mendapat pahala (dari kebaikan) yang dikerjakannya dan Ia mendapatkan siksa (dari kejahatan) yang dikerjakannya”
(QS Al Baqarah [2]:286)**

**”Maka, apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh – sungguh (urusan) yang lain dan hanya kepada Tuhanmulah hendaknya kamu berharap”
(QS Al Insyirah [94] : 7- 8)**

**“Tak ada rahasia untuk menggapai sukses. Sukses itu dapat terjadi karena persiapan, kerja keras, dan mau belajar dari kesalahan”.
(Norman Vincent Peale)**

**“Yang kita cari adalah nilai kebenaran tertinggi, nilai keadilan tertinggi, nilai cinta dan kasih sayang tertinggi, nilai kesetiaan tertinggi, nilai – nilai yang tidak lagi dibatasi oleh perbedaan manusia”.
(Ary Ginanjar Agustian)**

PERSEMBAHAN

Dengan menyebut nama Allah Yang Maha Pengasih dan Penyayang,
serta dengan melantunkan sholawat dan
salam kepada sang pencetus revolusioner muslim Nabi Muhammad SAW.
Wafiah minal khoiri kulla mughlaq.

Kupersembahkan Tugas Akhir ini teruntuk:
Mama' Zubaidah tercinta, demi darah, kasih sayang, keringat kesabarannya ,
airmata yang selalu menyalurkan energi doanya.
Bapak M. Abdurrahman Badaruddin tercinta dengan ketenangannya,
bimbingan, ketauladanan, pengorbanan, kesabaran dan do'a yang terus
dipanjatkan disepertiga malamnya.
Yayu' Fatimah, Kk' Nung, Yayu' Sanah, Yayu' Yati.
Ponakan2 tersayang, adun, uki, fikri, nauval, fawwaz.
Para shahabat yang selalu setia memberikan semangatnya.
Suami dan anak-anakku kelak di istana rumah tangga.
Komplek V PPSPA Daar El Salaam Jl. Kaliurang KM 12,5.
Almamater dan Indonesiaku.

Kupersembahkan sebagai sebuah jawaban :

“Kapan lulusmu nak?”

”Sesungguhnya dalam penciptaan langit dan bumi, dan silih berganti malam dan siang benar-benar terdapat tanda-tanda bagi orang-orang yang berakal, Yaitu orang-orang yang mengingat Allah sambil berdiri, duduk, atau di kala berbaring dan mereka memikirkan tentang penciptaan langit dan bumi seraya berkata ”Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia, Maha Suci Engkau, maka peliharalah kami dari siksa neraka”.
(Ali Imran:190-191)

Katakanlah ”andaikan laut menjadi tinta untuk menulis kalimat-kalimat Tuhanku, sungguh habislah itu sebelum habis ditulis kalimat-kalimat Tuhanku, meskipun kami datangkan tambahan sebanyak itu pula”. (Al-Kahfi:109)

Terima kasih pula kepada:

- Allah SWT atas segala rahmat dan karunia-Nya pada hamba yang selalu berbuat khilaf dan dosa.
- Nabi Muhammad SAW, yang kuantikan syafa'atnya kelak. *Waftah Minal Khoiro Kulla Mughlaq.*
- Bapak, mama', spuntene nggih, nembe lulus sekarang. Makasih buanget untuk smuanya, sepuntene maleh, dereng saget bangga'ke mama' n bapa'..smoga ridho dan ikhlasmu yang membawaku menjadi anak yang berbakti dan manfaat untuk umat..amien..
- Yayu' fatimah, kk nung, yayu' sanah, yayu' yati, semoga kita selalu menjadi kluarga dan saudara yang sakinah, mawaddah warohmah, tak akan pernah bercerai berai oleh apapun yang bisa membuatnya terpisah..hidup ini penuh warna..LOP YU FULL..
- Adun, uki, fikri, naufal, fawwaz, dan satu calon ponakan yang akan lahir, nda' boleh nakal ya na', smoga menjadi manusia yang bermanfaat bagi umat.
- Le'Seh, Le'Emi, Bunda Maria, Le' Gio, Le' Emi, Le' Yeng, Le' Is, Le' Jam, Le' Nah, le' centhil, le' dar, Le' Sur, Bulek Yati..
- Mbah Putri, Mbah Kakung di Magelang, Pagaram, n Purworejo, Smoga Allah Selalu memberikan kasih sayang-Nya ditempat yang indah di syurga..
- Seluruh keluarga besar di Magelang, Pagaram Sum-Sel, Magelang, Kutoarjo, Sarang Rembang, Gamping YK..
- Bapak KH. Masykur Muhammad, L.c., Ibu Hj. Sukainah, mba Ema + Mas Sakho'= D'Rikhfi, Mas Aim, Mas Avi, Mas Iad, dan Seluruh keluarga besar nDalem Pondok Pesantren Sunan Pandan Aran Jl. Kaliurang KM 12,5 Jogjakarta..he..he..sekalian promosi..
- Pipit, Iis, Ipek, SitiJeng..lopyupull, suwun atas bantuan materiil dan spiritualnya, takkan terbalaskan oleh apapun, smoga Allah membalasnya dengan Rahmat dan Ridho-NYA.Amien.
- Seluruh dosen pengampu yang pernah kuambil mata kuliahnya berapapun sks-nya (BuBudi, PakWahyudi, PakTito, PakMed, PakYusuf, PakWidi, PakEka, PakHendra, PakNandra, BuAna, BuIzzah, PakBudi, PakZain, PakImam, PakSidiq, PakParman, PakHalim), Mas2 Lab. TE UII(MasHeri, MasAgung, MasTri, Aan, MasNur), Pak2 Satpam, Pak2 Dapur, Cleaning Servicer.
- Dewi, wina, Adi, Himawan, Heru, Muhtar, Gunawan,V3, Lalu, Sigit, Gagat, dan smua rekan2 Teknik Elektro UII dari 1997 sampai 2006.
- Guru2ku di (TK. Roudhotul Athfal 'Aisyiyah Pagaram, MI. Al-Anshor Pagaram, MTs As-Salafiyah Pati Jateng, MA. Ali Maksum Krapyak Jogjakarta).
- Nak-Anak Kelas, I'dad D, Ula (C, D, E & F) MA + MTs PP. Sunan Pandan Aran, TPA masjid Miftahul Jannah Klabanan, atas kesabaran yang diajarkan..
- Komplek V PPSPA Daar El Salaam, lt. Langitan dan lt. Pribumi. Dari Jaman ada kang udin ma kang jadid, dan kang anas yang selalu jadi cadangan, mpe sekarang jamannya muntaha, nadzif n manshur.
- Ihda + mas wawan, yang selalu memberikan semangat dan bantuannya.
- Winda, entah bagaimana kabarmu saat ini..smoga sll baik2 saja..
- Ndu khafidhoh, yuk bib+k'agus=faiZ..terimakasih sudah menjadi keluarga untukku.
- Hanik dan smua temen2 kos-nya kartini E8c karang malang, suwun bgt, bisa menginap dsaat aku tersesat ditengah perjalanan malamku..
- Iip dan abiy yang telah memberikan cerita indah disebagian waktu hidupku..tengs tu yu men..:-)..
- Lagu2, artikel2, buku2, teman2 dunia "mayaku" yang bangkitkan imajinasi hayalku, dan smua yang membantu dan memulihkan semangatku, dan smua yang tak dapat kusebutkan satu persatu, makasih

KATA PENGANTAR



Assalamu 'alaikum Wr. Wb.

Tak ada kata yang lebih layak diucapkan di akhir pembuatan Tugas Akhir ini selain ucapan syukur kepada Allah S.W.T yang memberikan kemampuan dan kesabaran kepada penulis untuk menyelesaikannya, meskipun sempat beristirahat untuk mengobati “kelelahan mental” dan mengumpulkan kembali semangat yang tercecer. Untunglah Allah seringkali menghibur, mengingatkan, mengkritik, dan menyapa kecengengan penulis dengan ayat-ayat-Nya yang hidup disekitar, maupun melalui keindahan firman-Nya dalam Al-Qur'an yang menemani malam-malam yang telah dilalui. Pada kesempatan kali ini dengan segala kerendahan hati, penulis haturkan rasa terima kasih yang tulus kepada:

1. Bapak Tito Yuwono, ST., M.Sc., Selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia dan selaku Dosen Pembimbing I.
2. Bapak Medilla Kusriyanto, ST., M. Eng., Selaku Dosen Pembimbing II untuk semua saran-saran dan bimbingannya sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Segenap Dosen Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Terima kasih atas bimbingan dan ilmu yang telah diberikan.

4. Mas Agung, Mas Tri, Mas Heri, Aan, selaku Laboran Jurusan Teknik Elektro Universitas Islam Indonesia, atas diskusi, bantuannya, pinjaman peralatan dan tempatnya, dan seluruh karyawan UII.
5. Mama' dan Papa', atas kasih sayang, air mata, pengorbanan, kesabaran, keikhlasan, perhatian, nasehat, bimbingan, dan doa-doa malamnya.
6. Yuyu' Fatimah, Kk Nung, Yuyu' Sanah, Yuyu' Yati, atas kasih sayang, dukungan, do'a dan kiriman bulanannya, dan seluruh keluarga besar di Pagaram.
7. Kelurga besar Lek Tahrir AR di Rembang, keluarga Magelang, keluarga Kutoarjo, keluarga Pagaram.
8. Keluarga besar Bapak KH. Masykur Muhammad, Lc di Jogjakarta atas tempat tinggal, bimbingan dan do'anya selama penulis kuliah.
9. Keluarga Bapak H. Abdul Muin di Pekalongan dan keluarga besar Bapak Sadono di Bantul.
10. Best of the best friends, Ndu, Wina, Dewi, V3, Adi, Lu.,Hima, Heru, Bul, Dol, Ape'.
11. D'Big Family Of Komplek V PPSPA Daar El Salaam: K' Udin, K' Jadid, K' Anaz, Mun, Mb Nisa', Mb Faw, Nadzif, Ihda, Winda, Unun, Nani, D' Iis, Iffah, Pipot, Nafi', Ca_maY, Ipe, Siti, Wafa, Matoel, dan semua warga Langitan dan Pribumi "Daar El Salaam *Community*", makasih atas persaudaraan yang tak kan pernah diputuskan oleh apapun.
12. Rekan-rekan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu

yang telah memberikan dukungan dan do'a sehingga tugas akhir ini terselesaikan.

Penulis menyadari bahwa Tugas Akhir ini tidak luput dari kesalahan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun akan penulis terima dengan lapang hati.

Akhirnya, harapan penulis semoga Tugas Akhir ini dapat bermanfaat bagi kita semua, sekian dan terima kasih.

Wassalamu 'alaikum Wr.Wb.

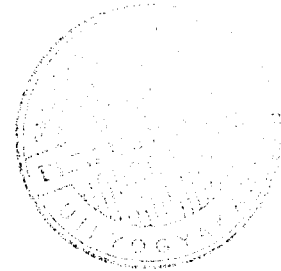
Yogyakarta, April 2010

Penulis

ABSTRAKSI

Sistem digital dewasa ini telah menjadi semakin kompleks seiring dengan perkembangan teknologi IC yang dapat membuat komponen dalam satu *chip*. Desain Verilog HDL ditujukan untuk desain pada sistem digital yang memuat ribuan sampai jutaan gerbang pada satu *chip* IC. FPGA merupakan salah satu *chip* IC yang dapat digunakan untuk implementasi sistem digital yang lebih rumit. LCD merupakan komponen penting yang menjadi *display* pada rancang bangun elektronika. Pada tugas akhir ini dirancang LCD yang dapat diimplementasikan pada FPGA dikendalikan dengan pengiriman data 4 bit. Pada perancangan ini, bahasa pemrograman yang digunakan adalah Verilog HDL dengan arsitektur struktural pada perancangan top level-nya. Rancangan diimplementasikan pada FPGA Xilinx Spartan II XC2S50-PQ208. *Software* yang digunakan untuk sintesis dan implementasi rancangan pada FPGA adalah Xilinx ISE 7.1i. Penggunaan FPGA untuk implementasi ini adalah 23 slice dari 768 (2%), slice flip-flop 41 dari 1536 (2%), 26 LUT dari 1536 (1%), dan 9 IOB dari 144 (6%), 1 BRAM dari 8 (12%), 1 GCLK dari 4 (25%). Teks yang ditampilkan pada LCD adalah “TE UII JOGJA”.

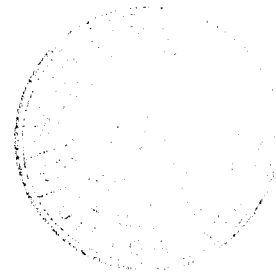
Kata kunci: FPGA, LCD, *running text*.



DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	i
LEMBAR PENGESAHAN PENGUJI.....	ii
HALAMAN PERSEMBAHAN.....	iii
HALAMAN MOTTO.....	iv
KATA PENGANTAR	vii
ABSTRAKSI	ix
DAFTAR ISI	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xiv
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan.....	3
BAB II	5
STUDI PUSTAKA	5
2.1. Tinjauan Pustaka	5
2.2. Rencana Penelitian	9
BAB III	12
PERANCANGAN SISTEM	12
3.1. Perancangan Perangkat Keras.....	12
3.1.1. FPGA (<i>Field Programmable Gate Array</i>)	14
3.1.2. FPGA Keluarga Xilinx Spartan II	14
3.1.2.1. <i>Struktur dasar keluarga Xilinx Spartan II</i>	15
3.1.2.2. Konfigurasi blok FPGA Spartan II.....	15
3.1.2.3. Programmable Routing Matrix.....	18
3.1.2.4. Mode Operasi.	20
3.1.2.5. <i>Board Pegassus</i>	21
3.1.3. LCD (<i>Liquid Crystal Display</i>)	28

3.2. Perancangan program	32
3.2.1. Xilinx ISE 7.1i (Xilinx Foundation Series).....	32
3.2.2. Verilog HDL.....	33
3.2.3. Program Utama	36
BAB IV	39
HASIL PENGUJIAN DAN PEMBAHASAN.....	39
4.1. Pendahuluan.....	40
4.2. Pengujian perangkat keras	40
4.2.1. Analisis rangkaian LCD dengan menggunakan Labview Instrument NI USB-6009	40
4.3. Pengujian Perangkat Lunak	42
4.3.1. Analisis dengan Test Bench Waveform Xilinx ISE 7.1i.....	42
4.3.2. Analisis pada Software FPGA	43
BAB V	47
PENUTUP.....	47
5.1. Kesimpulan	47
5.2. Saran.....	47
DAFTAR PUSTAKA	
LAMPIRAN	



n II	Gambar 3.1. Blok Diagram Perancangan Sistem.....	13
.....	Gambar 3.2. Blok Diagram dasar keluarga Spartan II	15
.....	Gambar 3.3. Blok Diagram I/O Spartan II.....	16
6.....	Gambar 3.4. CLB pada Spartan II.....	17
.....	Gambar 3.5. Blok RAM Spartan II	18
er-Heksad	Gambar 3.6. Struktur <i>Local Routing</i>	19
	Gambar 3.7. Koneksi BUFT untuk <i>Dedicated Horizontal Bus Line</i>	20
	Gambar 3.8. Blok Diagram Pegassus	21
	Gambar 3.9. Aliran Scan JTAG pada Pegassus.....	23
	Gambar 3.10. Rangkaian Saklar <i>Pushbutton</i> , Saklar geser dan LED	24
	Gambar 3.11. <i>Common Diode Sevent segment 4 digit</i>	25
	Gambar 3.12. <i>Common Anode Sevent segment 1 digit</i>	25
	Gambar 3.13. Pin penghubung tambahan.....	26
	Gambar 3.14. LCD 2x16 karakter	29
	Gambar 3.15. Konstruksi dasar LCD	32
	Gambar 3.16. Blok diagram alir penggunaan <i>software</i> FPGA	35
	Gambar 3.17. <i>Flowchart</i> Program.....	36
	Gambar 4.1. LCD pada FPGA	39
	Gambar 4.2. Pulsa pada pin DB-4.....	39
	Gambar 4.3. Pulsa pada pin DB-5.....	40
	Gambar 4.4. Pulsa pada pin DB-6.....	41
	Gambar 4.5. Pulsa pada pin DB-7.....	41
	Gambar 4.6. <i>Timing Diagram running text</i> pada LCD.....	43

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi komunikasi memegang peranan penting dalam sistem penyebaran informasi. LCD (*Liquid Crystal Display*) sekarang menjadi salah satu perangkat penting untuk perkembangan teknologi komunikasi informasi, juga sebagai media peraga dinamis dapat menampilkan teks ataupun *image*. Untuk menyampaikan suatu informasi diperlukan teknologi tepat dan akurat yang bertujuan untuk menarik perhatian karena sifatnya informatif dan atraktif, seperti untuk informasi layanan masyarakat .

Selama ini, sistem penampil informasi yang terletak ditengah kota dan perkantoran masih menggunakan dot matrik ataupun *sevent segment*. *Running text* merupakan salah satu cara untuk menyampaikan informasi tercepat yang dapat dilihat dengan mudah oleh masyarakat. LCD merupakan salah satu perangkat yang bisa memberikan tampilan sesuai dengan yang diinginkan dan memberikan beberapa keuntungan dibandingkan dengan perangkat lain untuk menampilkan sebuah data antara lain, hemat energi, ringan, proses perancangan yang relatif lebih mudah, dan mampu menampilkan karakter yang sesuai dengan keinginan (misalnya huruf jawa atau huruf jepang), dengan pengaturan tertentu. LCD *display* grafik mampu menampilkan data dalam bentuk *image*, sedangkan teks akan menampilkan karakter.

Perancangan *running text* sebelumnya banyak menggunakan mikrokontroler, belum ada yang mengimplementasikannya pada FPGA. Pada penelitian sebelumnya, FPGA sudah diimplementasikan pada beberapa *device* seperti, jam digital, kalkulator, pengendali *traffic light*, lengan robot, *prototype IC vending machine*. Pada penelitian tugas akhir ini, dibuat rancang bangun *running text* pada LCD yang diimplementasikan di FPGA untuk menampilkan teks yang dibuat program dan di-*download* pada FPGA dengan menggunakan Xilinx ISE 7.1i .

FPGA (*Field Programmable Gate Array*) adalah IC yang terdiri dari blok-blok logika yang interkoneksinya dapat dikonfigurasi. Dengan menggunakan FPGA dapat dilakukan perancangan sistem digital yang kemudian bisa menjadi *prototype* rancangan sistem yang akan dirakit. Umumnya FPGA lebih tepat bila digunakan untuk mengimplementasikan fungsi yang membutuhkan operasi secara paralel. Fungsi logika dan interkoneksi FPGA ditentukan oleh data yang tersimpan pada sel memori statik internal.

Untuk di Indonesia, penggunaan FPGA masih sangat sedikit sekali. Di Negara-negara maju, FPGA telah banyak digunakan, karena FPGA dapat mengimplementasikan rangkaian kombinasi maupun rangkaian sekuensial, tanpa terhambat kekurangan register. Dalam FPGA bukan membuat program, tetapi lebih tepatnya adalah merekonfigurasi FPGA. Dengan menggunakan teknik penyusunan gerbang. Fungsi logika tersebut akan dibuat sedemikian rupa, yang ditulis dengan menggunakan bahasa pemrograman Verilog HDL.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut maka dapat dirumuskan permasalahan sebagai berikut: “ Bagaimana cara merancang dan mengimplementasikan “*Running Text*” pada LCD (*Liquid Crystal Display*) yang berbasis FPGA Xilinx Spartan II XC2S50-PQ208”.

1.3 Batasan Masalah

Adapun batasan masalah adalah sebagai berikut:

- a. Perancangan program *running text* menampilkan kata “TE UII JOGJA” pada LCD yang berbasis FPGA.
- b. Implementasi Verilog HDL pada FPGA menggunakan Xilinx ISE 7.1i.

1.4 Tujuan Penelitian

Tugas akhir ini bertujuan untuk merancang program *running text* yang akan ditampilkan LCD pada *prototype* FPGA Xilinx Spartan II XC2S50 PQ208 dengan menggunakan bahasa pemrograman Verilog HDL.

1.5 Manfaat Penelitian

Dengan adanya *running text* yang berbasis FPGA dapat dimanfaatkan sebagai alat untuk menyampaikan informasi yang berbasis elektronik.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir dengan judul “ Rancang bangun *Running Text* berbasis FPGA Xilinx Spartan II XC2S50-PQ208” ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini merupakan pengantar permasalahan yang dibahas seperti latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

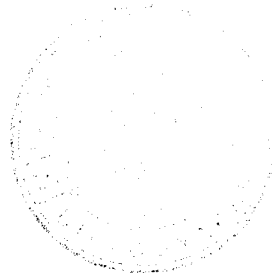
Bab ini memuat sistem kerja dan analisis perbandingan dari tugas akhir yang sejenis yang sudah dibuat dengan tugas akhir yang sedang dibuat sekarang, dan rencana penelitian.

BAB III PERANCANGAN SISTEM

Bab ini menjelaskan perancangan *hardware* dan program yang digunakan, bagian-bagian yang terlibat dalam sistem beserta cara mengimplementasikan dari setiap bagian tersebut.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini membahas mengenai perihal hasil-hasil pengujian yang diperoleh dari uji coba perakitan dibandingkan dengan dasar teori sistem.



BAB V PENUTUP

Bab ini memuat kesimpulan-kesimpulan dari proses perancangan implementasi terutama pada analisis kerja, juga memuat saran-saran pengembangan yang masih bisa dilakukan dari penelitian yang telah terlaksana.

DAFTAR PUSTAKA**LAMPIRAN**

BAB II

STUDI PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan dari penelitian yang sudah ada diperlukan untuk dilakukannya suatu pengembangan dan inovasi terhadap unjuk kerja sistem yang pernah dirancang sebelumnya dengan perancangan sistem yang ada akan dilakukan pada penelitian tugas akhir kali ini. Penelitian-penelitian yang pernah dilakukan diantaranya yaitu:

Novita Yurisdian, 2005, telah melakukan penelitian tentang perancangan pengendalian *Display Dot matrix* dengan Mikrokontroler AT89S51. Desain pada penelitian ini merupakan sistem pengatur tampilan teks berjalan yang menggunakan 7 buah *dot matrix* 5x7 berbasis AT89S51. Secara prinsip mikrokontroler berfungsi sebagai penyimpan data teks, pengatur data kontrol *segment*, pengatur data serial 8 bit, pembangkit pulsa *clock* khusus rangkaian *driver* kolom.

Pengaturan sistem *dot matrix* 5x7 menggunakan akses 8 bit data paralel yang dikendalikan menggunakan mikrokontroler memerlukan adanya rangkaian *driver* data *segment* agar akses data *segment* a – g dapat dilakukan secara paralel dan dapat digunakan sebagai sumber sinyal kendali penggerak baris 1 sampai 7. Dengan pemasangan *driver* ini pengendalian 7 buah led *matrix* 5x7 dapat dilakukan secara serempak.

Pengendalian sistem *dot matrix* 5x7 juga memerlukan rangkaian pengendali kolom *segment*. Pengaturan ini dilakukan dengan mengirim data 8 bit secara serial ke rangkaian *driver* kolom yang disusun menggunakan 5 buah gerbang TTL 8 bit serial/paralel *out shift register* IC 74LS164. Karena gerbang TTL tersebut harus dapat berjalan secara sinkron dengan laju bit data *transfer* dari mikrokontroler, maka perlu diberikan *clock* pada IC tersebut dimana pada tugas akhir yang telah dibuat ini dihubungkan dengan pin 2.0 yang berfungsi juga sebagai data *segment* a. Sedangkan laju *transfer* data 8 bit secara serial dilakukan melalui pin 3.0 yang secara spesifik memiliki fungsi tersebut. Untuk menjalankan inisialisasi teks terprogram dari arah kanan ke kiri, maka program pengatur harus diisikan ke dalam mikrokontroler melalui mode *ISP programming*. Prinsip kerja utama dari alat ini adalah pengendalian tulisan yang tertampil untuk mengubahnya harus mengubah program yang ada pada mikrokontroler.

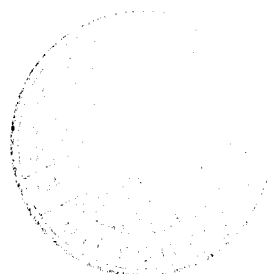
Perancangan perangkat keras ini dibangun oleh beberapa piranti seperti rangkaian pengendali utama yaitu rangkaian mikrokontroler AT89S52 dengan *oscillator* yang digunakan 11,092 MHz, ELCO dan resistor sebagai rangkaian reset, rangkaian MAX 232 yang terdiri dari IC MAX 232 dan beberapa ELCO sebagai penghubung antara mikrokontroler dengan *handphone* sehingga dapat terkoneksi. Selanjutnya, rangkaian *dot matrix* 5x7 sebanyak 8 buah sebagai penampil tulisan yang dilengkapi dengan 5 buah IC TTL 74LS164 sebagai *driver* penggeser data yang masuk ke kaki kolom data *dot matrix* sehingga tulisan bergeser.

Adisatya pramadianto, 2007. Pada penelitian yang telah dilakukan pembuatan alat yang mempunyai kemampuan untuk menyimpan memori karakter untuk ditampilkan sebanyak 1000 karakter dengan demikian memori maksimal dari SEEPROM yang digunakan adalah sebesar 1000 dikalikan dengan 5 *byte*, yaitu 5 *kilobyte*. SEEPROM yang digunakan mempunyai kapasitas memori sebesar 8192 *bytes*.

Setelah semua karakter yang dikirim oleh komputer dikodekan dan disimpan dalam memori SEEPROM, tugas selanjutnya dari mikrokontroler adalah menampilkan kode-kode huruf yang telah disimpan dalam IC SEEPROM ke penampil *dot matrix* dengan sarana bantuan dari driver baris berjumlah 7 jalur, IC demultiplexer, dan *driver* kolom sejumlah 50 jalur, kemudian tampilan pada dot matriks ini digeser penyalan dengan mikrokontroler.

Wahyu Widayanto, 2008, telah melakukan penelitian tentang pengendalian *running text* dengan SMS menggunakan mikrokontroler AT89S52. Tujuan penelitian ini adalah bagaimana mengaplikasikan teknologi SMS sebagai pengendali jarak jauh dalam menampilkan karakter berjalan pada *dot matrix display* yang lebih praktis dan canggih. Perancangan sistem ini dikendalikan oleh sebuah mikrokontroler AT89S52 yang akan mengolah masukan data dari *handphone* yang berupa data PDU (*protocol Data Unit*). Data PDU tersebut diolah dan dikonversi lagi ke tabel penyalan *dot matrix*.

Randi Pratama Putra, 2009, telah melakukan penelitian tentang perancangan *software* untuk tampilan animasi susunan dot matriks 8x8. Pada penelitiannya, *software* digunakan untuk pengenalan LED dot matriks display



dengan menggunakan bahasa pemrograman Visual Basic 6.0, dengan tiga tahap pengerjaan utama, yaitu membuat tampilan atau antarmuka, pengaturan properti kontrol program dan penulisan kode program. Tahapan pada pengerjaan penelitian yang dilakukan untuk membuat sebuah *user interface*/antarmuka yang berfungsi sebagai pengendali utama tampilan animasi pada perangkat lunak. Antarmuka yang sudah dibuat dihubungkan dengan serial port sebagai media penghubung pengirim data dari perangkat lunak ke perangkat keras dot matriks display.

Komunikasi antar *user interface*/antarmuka dengan *serial port* dibangun untuk menggunakan kontrol MSComm yang disediakan oleh Visual Basic. Setiap kontrol MSComm yang dipasang pada antarmuka hanya dapat menangani satu jalur komunikasi dengan *serial port*.

2.2 Rencana Penelitian

Rencana penelitian pada tugas akhir ini meliputi:

1. Kajian pustaka

Sebagai bahan referensi penelitian, kajian pustaka diambil dari laporan tugas akhir sebelumnya yang masih berkaitan dengan penelitian yang akan dilakukan, serta buku-buku yang mendukung penelitian tugas akhir kali ini dan berbagai informasi yang diambil dari berbagai sumber.

2. Perancangan *Hardware*

Alat yang akan dibuat adalah rangkaian LCD yang diimplementasikan pada FPGA. Komponen yang terdapat pada rangkaian ini adalah LCD, yang berfungsi untuk menampilkan *running text* yang akan dibuat. 4 buah resistor

390 Ω yang berfungsi untuk menghindari *overstressing* pada saat kontrol karakter LCD berlogika tinggi. Resistor variabel 1 K Ω berfungsi untuk mengatur *brightness* LCD. Sedangkan dioda 4148 berfungsi untuk melindungi LCD agar tidak mati ketika terjadi kesalahan pemasangan dioda.

3. Perancangan Program

Dalam proses perancangan program, dibutuhkan perangkat komputer sebagai media dalam menggunakan *software* Xilinx dengan Verilog HDL sebagai bahasa pemrogramannya. JTAG merupakan *interface* yang digunakan saat *download* program ke *board* FPGA Xilinx Spartan II XC2S5-PQ208, *output* dari FPGA Xilinx Spartan II XC2S5-PQ208 berupa LCD yang akan menampilkan tulisan berjalan sesuai dengan kode program yang telah dibuat.

Dalam pemrograman dibuat kode yang berfungsi untuk menampilkan *running text* pada LCD, kode program *running text* menggunakan kode karakter LCD dikirim per 4 bit. Data 4 bit awal yang dikirimkan adalah data 4 bit *upper nibble*, dan pengiriman data bit selanjutnya kedua adalah data 4 bit *lower nibble*.

4. Pengujian Alat

Setelah semua alat yang dirancang berhasil dibuat, harus dilakukan pengujian, agar benar-benar terjamin kualitas yang dibangun oleh suatu sistem. Pengujian alat ini meliputi *hardware* dan program dari sistem. Pada *hardware* diuji beberapa titik, yaitu pada titik tiga kontrol dan empat data bit pada LCD. Sedangkan untuk pengujian program, dilakukan simulasi program dan pengambilan *timing diagram* dengan menggunakan Xilinx ISE 7.1i.

5. Pengambilan data

Setelah pengujian alat dilakukan, pengambilan data dilakukan agar dapat dilihat hasil yang telah dicapai dari sistem yang telah dibuat. Data diperoleh dari pengujian program dan *hardware*.

6. Analisis dan pembahasan

Data dan simulasi yang telah didapat, selanjutnya dilakukan analisis dan pembahasan dari data yang telah didapatkan.

7. Penulisan laporan

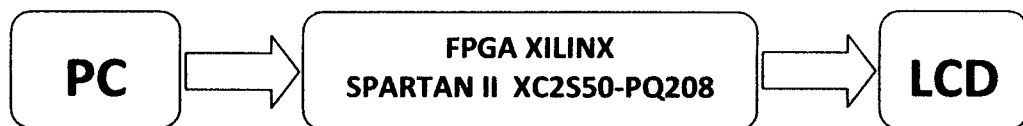
Setelah perancangan *hardware* dan program selesai dibuat, pengujian data dan analisis serta pembahasan telah didapatkan, maka tugas terakhir adalah melakukan penulisan laporan dari penelitian yang telah dilakukan, agar penelitian yang telah dilakukan dapat dikembangkan lebih lanjut dan dapat dijadikan sebagai salah satu bahan referensi pada penelitian tugas akhir berikutnya.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Perangkat Keras

Pada penelitian ini, sistem yang akan dibangun adalah perancangan “*running text*” pada LCD (*Liquid Crystal Display*) yang diimplementasikan pada FPGA. Secara keseluruhan sistem ini terdiri dari perangkat keras dan program. Gambar dibawah ini memperlihatkan perancangan sistem yang akan dibangun.



Gambar 3.1. Blok Diagram Perencanaan Sistem.

Masukan dari sistem diatas adalah berasal dari program yang kemudian akan diterjemahkan oleh komputer dan mengirimkan data bit yang dikeluarkan melalui kabel JTAG yang dihubungkan ke *board* FPGA yang telah terpasang rangkaian LCD yang ada di pin *external* dari *board* FPGA. Keluaran yang dihasilkan adalah karakter yang berbentuk teks “TE UII JOGJA” yang akan ditampilkan pada tampilan LCD. Untuk mengubah teks yang ada di tampilan , maka harus merubah kode yang ada di program, dengan mengganti kode ASCII.

3.1.1 FPGA (*Field Programmable Gate Array*)

FPGA merupakan sebuah piranti digital yang dapat diprogram untuk merepresentasikan sistem logika yang telah dirancang. Teknologi *Integrated Circuit* (IC). FPGA diperkenalkan pada tahun 1985 oleh perusahaan

semikonduktor Xilinx. FPGA adalah sebuah konsep teknologi IC yang dapat diprogram dan dihapus seperti halnya *Random Access Memory* (RAM). FPGA kemudian berkembang pesat, baik dari segi kepadatan gerbang, kecepatan dan disertai dengan penurunan harga jual.

Penemuan FPGA telah membuat peningkatan yang pesat akan pembuatan prototipe beberapa sistem digital. Salah satu produsen FPGA yang ada di pasaran adalah Xilinx, disamping produsen lainnya Actel dan Altera. Prinsip dasar dari pemrograman atau pengkonfigurasi FPGA Xilinx adalah pengubahan gambar rangkaian elektronik digital dari perangkat lunak Xilinx yang berupa file aliran bit (*bitstream*) dan di konfigurasi (di *download*) ke dalam IC FPGA Xilinx tersebut sehingga IC tersebut terkonfigurasi secara perangkat keras yang dirancang dalam perangkat lunak Xilinx. FPGA produk Xilinx sudah melewati beberapa generasi antara lain XC2000, XC3000 dan XC4000. Tiap generasi memiliki sifat dan gerbang logika, jumlah *Configurable Logic Blocks* (CLB) dan jumlah *Input/Output Blocks* (IOB).

Keuntungan-keuntungan yang dimiliki FPGA sehingga disukai oleh para penggunanya antara lain adalah :

- a. FPGA selalu dapat diprogram kembali sehingga memudahkan modifikasi tanpa harus merubah keseluruhan sistem.
- b. Sebuah rancangan secara otomatis dapat diubah dari level logika gerbang menjadi struktur *layout* dengan fasilitas yang dimilikinya, sehingga perubahan dapat dilakukan dengan mudah tanpa harus merubah rancangan awal.

- c. Simulasi hasil desain dapat dilakukan pada keluaran gerbang yang terpakai dan pada karakteristik pewaktuan yang dimiliki oleh desain yang dibuat. Hal ini sangat menguntungkan ketika waktu juga menjadi faktor yang harus diperhatikan dalam desain yang dibuat.
- d. IC FPGA keluaran terbaru mempunyai jumlah gerbang yang semakin banyak dengan fasilitas yang semakin lengkap.

3.1.2. FPGA Keluarga Xilinx Spartan II

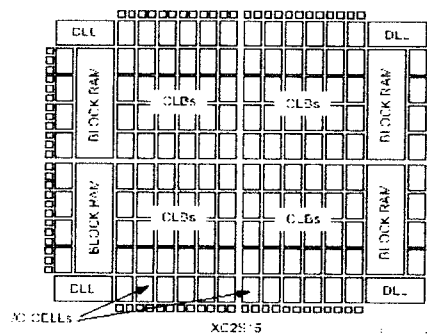
Spartan II merupakan salah satu keluarga FPGA yang dikeluarkan oleh Xilinx. Xilinx merupakan salah satu pabrik pembuat FPGA yang cukup terkenal. Keluarga Spartan II merupakan keluarga FPGA yang memiliki 15.000 sampai 20.000 gerbang. IC Xilinx ini dapat diprogram dan dihapus dengan waktu yang tidak terbatas. Keluarga Spartan ini dapat diprogram dengan mudah menggunakan *Xilinx Development System* ataupun dengan *Development System* yang lain yang dikembangkan oleh para pengguna.

Tabel 3.1. Data Keluarga Spartan II

Device	Logic Cells	System Gates (Logic and RAM)	CLB Array (R x C)	Total CLBs	Maximum Available User I/O	Total Distributed RAM Bits	Total Block RAM Bits
XC2S15	432	15.000	8 x 12	96	86	6.144	16K
XC2S30	972	30.000	12 x 18	216	132	13.824	24K
XC2S50	1.728	50.000	16 x 24	384	176	24.576	32K
XC2S100	2.700	100.000	20 x 30	600	196	38.400	40K
XC2S150	3.888	150.000	24 x 36	864	260	55.296	48K
XC2S200	5.292	200.000	28 x 42	1.176	284	75.264	56K

3.1.2.1 Struktur Dasar Keluarga Spartan II

Suatu piranti FPGA terdiri atas CLB, unit input/output, *Delay-Locked Loops* (DLLs), unit RAM dan unit *routing* yang dapat diprogram secara otomatis penuh. Susunan dan letak masing-masing bagian tersebut dapat dilihat pada Gambar 3.2.



Gambar 3.2. Blok diagram dasar keluarga Spartan II

Struktur dasar CLB terdiri dari RAM dan fungsi logika dasar. DLL digunakan untuk mengatur *clock*, perkalian *clock* dan pembagian *clock*. Pengaturan *clock* dapat dilakukan secara *eksternal (board level)* dan *internal (chip level)*. Memori terdiri dari 4 K bit yang dapat dikonfigurasi dari 1 bit ke 16 bit. Sedangkan untuk pemilihan I/O mengikuti standar I/O untuk diimplementasikan ke dalam *chip ke chip*, *chip ke memori* dan *chip ke interface*.

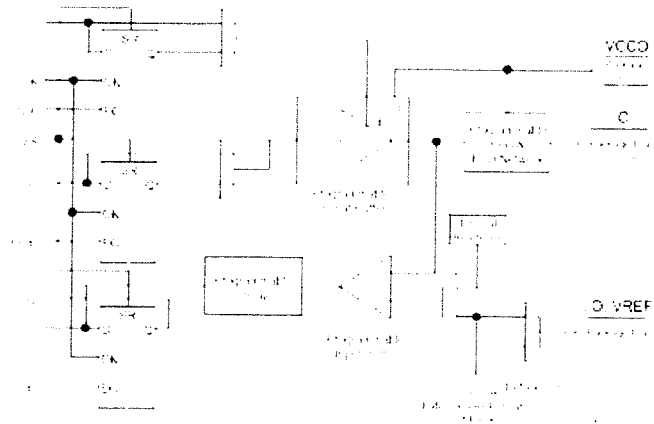
3.1.2.2 Konfigurasi Blok FPGA Spartan II

Konfigurasi blok-blok yang termuat dalam FPGA Spartan II adalah :

a. *Input/Output Blocks (IOB)*

Input/Output Blocks merupakan bagian dari FPGA yang berfungsi menghubungkan FPGA dengan piranti lain yang terkoneksi. IOB keluarga Spartan II mampu bekerja pada berbagai macam standar I/O seperti TTL, CMOS, dan

PCI. Kemampuan untuk menyesuaikan dengan berbagai macam I/O didukung dengan kemampuan tiap *pad* I/O untuk ditambahi *pull-up* dan *pull-down resistor*

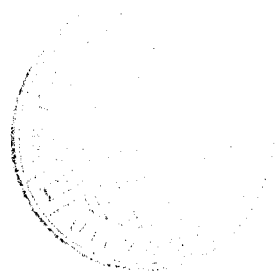


Gambar 3.3. Blok diagram I/O Spartan II

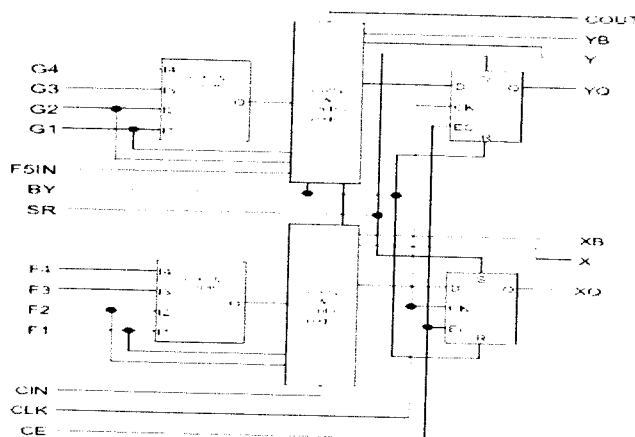
Bagian *buffer* pada Spartan II IOB *input path* akan menghubungkan sinyal *input* yang masuk secara langsung dengan logika internal atau secara tidak langsung melalui *input flip-flop* optional. Sedangkan bagian *output path* termasuk *buffer* 3 keadaan akan men-*drive* sinyal output menuju *pad output*. Sama dengan sinyal *input*, sinyal *output* ini dapat dihubungkan dengan *pad output* melalui logika *internal* maupun melalui *output flip-flop optional*.

b. *Configurable Logic Blocks (CLB)*

CLB merupakan bagian dari FPGA yang berfungsi merubah logika-logika terprogram yang dimasukkan menjadi fungsi-fungsi yang dipahami oleh FPGA dan dapat bekerja sesuai dengan program yang diinginkan. CLB Spartan II terdiri atas *Logic Cell (LC)* sebagai bangunan utama. Sebuah LC terdiri atas 4 buah *input* yang akan membangkitkan fungsi logika yang diinginkan, *carry logic* dan elemen penyimpan. Keluaran setiap LC akan men-*drive* keluaran CLB dan masukan pada *D flip-flop*. Setiap CLB pada Spartan II terdiri atas 4 LC yang tersusun dalam 2



slices yang identik. Tiap CLB ini juga memuat logika yang akan mengkombinasi generator pembangkit fungsi logika untuk 4 sampai 6 input.



Gambar 3.4. CLB pada Spartan II

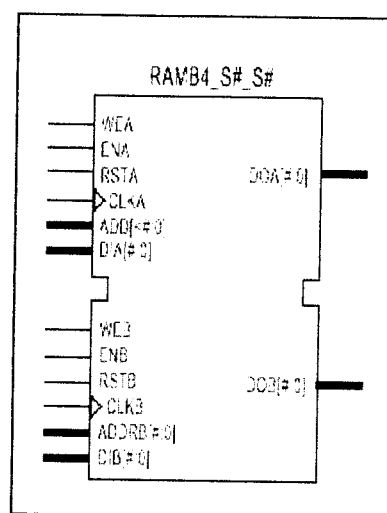
Generator pembangkit fungsi diimplementasikan dalam sebuah *look-up tables* (LUT) 4 input. LUT ini juga dapat membangkitkan sebuah RAM 16 x 1 sinkron serta membangkitkan fungsi *shift register* 16 bit. Fungsi RAM yang dibangkitkan generator ini akan melengkapi *block* RAM yang dimiliki oleh Spartan II sehingga mampu menghasilkan unit penyimpan data yang handal.

c. Delay-Locked Loops (DLLs)

Masing-masing *clock* input secara menyeluruh dikelompokkan dengan penguatan, maksudnya adalah pemenuhan digital DLL dapat menghilangkan kemiringan antara titik *clock input* dan *pin clock internal* yang dilalui oleh *hardware*. Tiap-tiap DLL dapat digerakkan oleh 2 buah jaringan *clock* secara menyeluruh. Pemantauan DLL dapat dilakukan dari *clock* masukan dan aliran *clock* serta penentuan tunda *clock* secara otomatis. Penambahan tunda yang diperkenalkan sebagaimana beberapa *clock* yang diraih secara pasti oleh sebuah

flip-flop internal salah satunya setelah periode *clock* yang masing-masing melaju terhadap *input*. Sistem tertutup ini secara efektif akan menghilangkan aliran *clock* yang tertunda, dimana hal tersebut dikerjakan oleh pembawa sistem yang mana setiap *clock* yang tersisa tepatnya dalam *flip-flop internal* akan diserempakkan dengan *edges clock*.

d. *Blocks Random Access Memory (RAM)*



Gambar 3.5. Blok RAM spartan-II

Konfigurasi FPGA spartan II sangat luas dan memiliki memori 4 K bit. Tiap-tiap blok RAM akan menempati CLB serta tiap-tiap blok dapat dikonfigurasi pada perbandingan diantara 4K x 1 dan 256 x 16 bit.

3.1.2.3 Programmable Routing Matrix

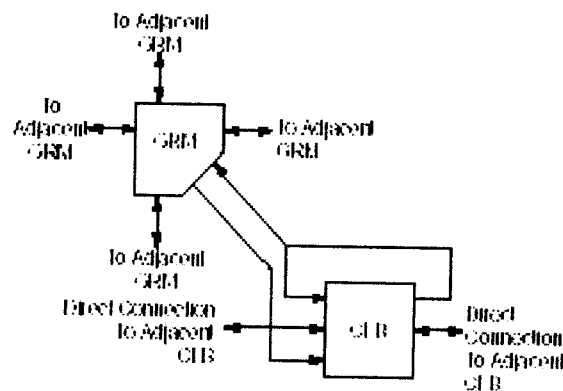
Programmable Routing Matrix merupakan cara sebuah FPGA melakukan *routing* menghubungkan CLB-CLB dan IOB-IOB yang digunakan dalam desain menjadi satu kesatuan sistem. *Routing* ini dilakukan secara otomatis penuh.

Namun untuk keperluan tertentu, optimasi jalur yang paling pendek dapat dilakukan *routing* manual.

Dalam keluarga Spartan II ada beberapa macam *routing* yang bisa digunakan yaitu:

a. *Local Routing*

Local Routing digunakan untuk mengimplementasikan hubungan antara LUT, *flip-flop* dan *General Routing Matrix* (GRM), antara jalur umpan balik internal CLB dengan LUT lain pada CLB yang sama untuk koneksi *high speed*, serta antara jalur-jalur langsung yang bisa dibuat untuk memperkecil *delay*.



Gambar 3.6. Struktur *local routing*

b. *General Purpose Routing*

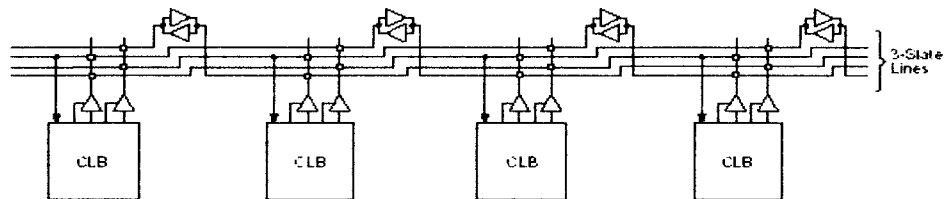
General Purpose Routing digunakan untuk melakukan koneksi vertikal dan horisontal antar kolom dan baris CLB yang digunakan.

c. *I/O Routing*

I/O Routing khusus digunakan untuk menghubungkan *array* pada CLB dengan IOB.

d. *Dedicated Routing*

Dedicated Routing digunakan untuk menghubungkan beberapa CLB, IOB ataupun LUT yang memerlukan perlakuan khusus untuk memaksimalkan performasinya.



Gambar 3.7. Koneksi BUFT untuk *Dedicated Horizontal Bus Line*

e. *Global Routing*

Global Routing digunakan untuk menghubungkan *clock* dengan bagian yang membutuhkan serta sinyal-sinyal dengan *fan-out* yang tinggi ke bagian lain.

3.1.2.4 Mode Operasi

Keluarga Spartan II dapat dioperasikan dalam 4 mode yaitu:

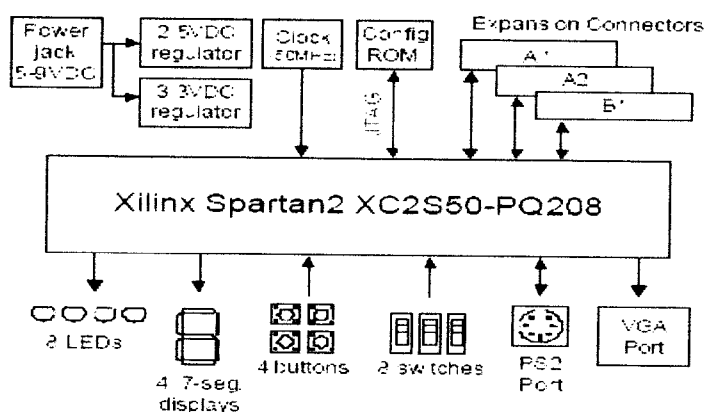
- a. *Slave Serial mode.*
- b. *Master Serial mode.*
- c. *Slave Parallel mode.*
- d. *Boundary Scan mode.*

Mode yang paling mudah digunakan adalah *Boundary Scan Mode* dimana tidak diperlukan koneksi-koneksi khusus, cukup menggunakan kabel paralel yang dikoneksikan menggunakan JTAG, maka desain dapat dengan mudah diimplementasikan ke dalam FPGA.

3.1.2 Board Pegasus

Modul Pegasus dikeluarkan oleh pengembang ke 3, yaitu DILIGENT. Dengan IC FPGA utama yaitu, Xilinx Spartan II XC2S50, dengan perangkat lunak Xilinx. Perangkat pendukung yang terdapat pada modul Pegasus diantaranya :

- 50 k gerbang Xilinx Spartan II FPGA, dengan 50 k gerbang dasar dan 200 MHz operasi *maximum*.
- XCF01S Xilinx Flash ROM.
- Berbagai macam I/O, termasuk di dalamnya 8 LED, empat *seven-segment*, empat saklar *pushbutton* dan delapan saklar geser.
- 50 MHz *oscilator* dan satu pin untuk *oscilator* tambahan.
- PS/2 dan VGA port.
- 96 *Pin* I/O dibagi dalam 3 bagian/port, yaitu : A1, A2 dan B1 yang masing-masing terdiri dari 40 *pin*.
- Seluruh I/O *pin* sudah dilengkapi pengaman.
- Port* pemrogram mode JTAG.



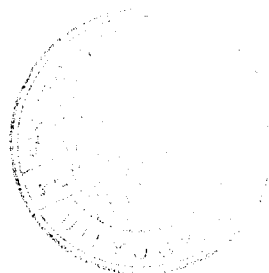
Gambar 3.8. Blok diagram Pegasus

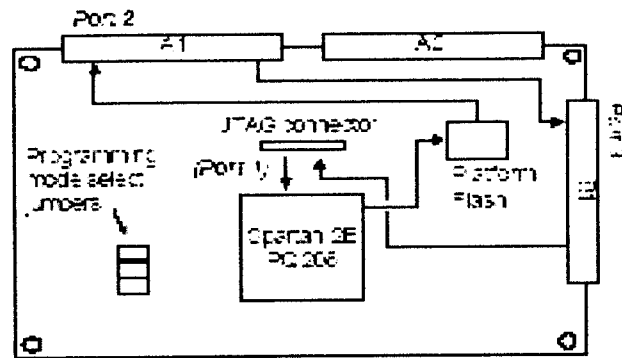
Modul Pegasus sudah dapat dibilang sangat komplit karena hanya dengan menggunakan modul ini, sudah dapat men-*download design* dan menjalankannya tanpa perlu menambah komponen lain, karena telah tersedia beberapa masukan dan keluaran.

a. *Port* JTAG dan Pengkonfigurasi Modul

Pada modul Pegasus ini selain terdapat IC FPGA utama (Spartan II XC2S50) juga terdapat IC FPGA *secondary* yaitu Spartan II XCF01S yang berfungsi sebagai IC *Flash* ROM, sehingga apabila ada suatu modul yang dapat diprogram (terdapat IC FPGA) yang terhubung ke modul Pegasus ini, maka dapat dikonfigurasi melalui *port* JTAG (*port1*), dari modul utama. Dengan menggunakan *port* JTAG dapat diketahui IC tipe, jenis dan dari keluarga FPGA yang ada pada modul utama maupun pada modul lain (tambahan) yang terhubung ke modul utama melalui *port* tambahan secara *automatic scan chain*.

Scanning dengan menggunakan JTAG sangat mudah karena perangkat lunak Xilinx melakukannya secara otomatis, Xilinx mencari melalui *port* JTAG dan membaca IC FPGA utama, kemudian apabila tidak ada modul lain yang terhubung pada *port* JTAG tambahan *port2* (A1) atau *port3* (B1) maka *buffer* pada modul Pegasus menghilangkan keberadaan *port* tambahan tersebut, sedangkan jika ada modul (memilik IC FPGA) maka *buffer* akan menyatakan bahwa ada modul yang terhubung. Saat *scanning port* JTAG membaca FPGA utama, ke flash ROM (XCF01S), lalu membaca modul yang terhubung melalui *port* tambahan tersebut, maka dapat dikonfigurasi secara bersamaan melalui 1 kabel utama, baik itu *port* USB, *Parallel* maupun *Ethernet*.





Gambar 3.9. Aliran scan JTAG pada Pegasus

b. Catu Daya

Modul Pegasus memerlukan 5 Vdc catu daya, sedangkan untuk masukan/keluaran dapat menggunakan *power* dari luar 3 Vdc sampai dengan 5 Vdc. Catu daya ini juga dipergunakan untuk mengaktifkan 8 LED, 4 display *seven segment*, sebagai masukan + 5 Vdc untuk saklar *pushbutton* dan saklar geser serta sumber *clock internal* dan *power port* tambahan.

c. Oscillator

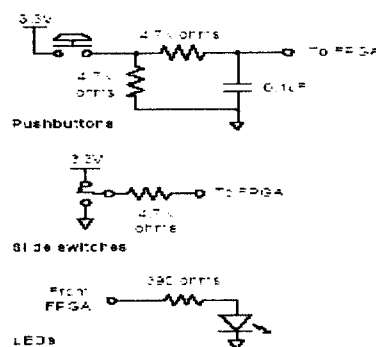
Modul Pegasus menyediakan 50 MHz *oscillator* sebagai *clock* utama untuk aplikasi-aplikasi yang akan dikonfigurasi ke Spartan II XC2S50. *Oscillator* tersebut terhubung langsung ke Spartan II XC2S50 (*pin77*).

d. Saklar Pushbutton, saklar geser indikator LED

Empat saklar *pushbutton* dan 8 saklar geser disediakan sebagai masukan. Saklar *pushbutton* dalam keadaan normal menghasilkan 0 Vdc (rendah), dan akan berubah menjadi 3 Vdc sampai dengan 5 Vdc (tinggi) ketika saklar *pushbutton* ditekan. Saklar geser menghasilkan rendah (0 Vdc) atau tinggi (3Vdc sampai dengann 5 Vdc) secara tetap tergantung dari posisinya. Saklar *pushbutton* sebagai

masukan menggunakan rangkaian RC sebagai pengaman dan agar menghasilkan masukan yang stabil, sedangkan saklar geser hanya terhubung dengan resistor secara serial.

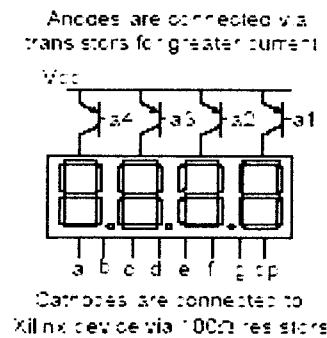
Delapan LED disediakan sebagai keluaran anoda LED terhubung ke *pin* keluaran melalui resistor $390\ \Omega$, sedangkan katoda LED terhubung langsung ke Ground. LED 9 digunakan sebagai *indikator power* FPGA dan LED ke 10 digunakan sebagai *indicator* status pemrograman JTAG.



Gambar 3.10. Rangkaian saklar *pushbutton*, saklar geser, LED

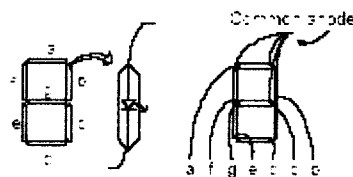
e. Seven Segment

Pada modul Pegasus terdapat 4 digit *common anoda seven-segment*. *Display seven-segment* dikonfigurasi secara *multiplexer*, jadi hanya ada 7 masukan katoda untuk mengaktifkan 28 katoda pada 4 digit *seven-segment*. Empat digit selektor *enable* berfungsi sebagai pengatur *digit* pada *seven-segment*.



Gambar 3.11. *Common anode seven-segment 4 digit*

Ketujuh anoda dari 4 *seven-segment* saling tersambung ke dalam selektor “*common anode*”, *display* ini memiliki 4 selektor yang di namakan *Anode Not* atau yang disingkat dengan AN0 sampai AN3, apabila ada sinyal atau pulsa yang mengaktifkan selektor ini maka *digit* dari selektor tersebut akan aktif. Sinyal/pulsa yang digunakan adalah sinyal/pulsa rendah (0 Vdc). Katoda dari setiap *seven-segment* terhubung ke dalam 7 keluaran, dan diberi nama CA – CG, dan akan aktif apabila ada sinyal/pulsa rendah (0 Vdc).



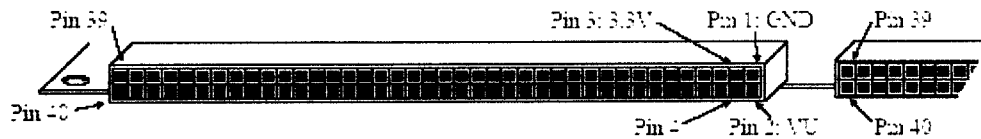
Gambar 3.12. *Common anode seven-segment 1 digit.*

Dilihat dari diagram yang terlihat maka menghasilkan *display seven-segment multiplexer*, bila anoda atau selektor (AN0 - AN3) diaktifkan maka *digit* tersebut yang akan aktif dan mendapatkan sinyal/pulsa katoda (CA – CG). Jika dilakukan secara terus menerus dan bila benar maka ke-4 *digit* akan terlihat aktif atau seolah-olah semua aktif secara bersamaan. Jika diberi sinyal rendah (0 Vdc)

secara terus menerus 1 ms – 16 ms, maka akan terlihat semua *digit* aktif. Dengan syarat, bersamaan dengan *digit enable* data untuk *digit* tersebut harus ada. *Refresh frequency* berkisar 60 Hz sampai dengan 1 KHz.

f. *Port I/O Tambahan*

Pada modul Pegasus terdapat 3 *port* tambahan (A1, A2, dan B3) dengan masing-masing *port* terdapat 40 *pin*. Masing-masing *port* memiliki GND pada *pin* 1, VU pada *pin* 2, dan 3 Vdc sampai dengan 5 Vdc pada *pin* 3. Untuk *pin* 4 sampai 35 merupakan *pin* sinyal I/O, dan *pin* 36 - 40 digunakan sebagai *port* JTAG tambahan, atau juga bisa digunakan sebagai *clock* tambahan.



Gambar 3.13. *Pin* penghubung tambahan

Tabel 3.2. Port accessory

Pegasus FPGA Pin Assignments							
Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	GND	53	VCCO	105	VCCO	157	TDO
2	TMS	54	MODE2	106	PROGRAM	158	GND
3	LLSBCLK	55	PB-IO14	107	INITIO	159	TDI
4	LCSA	56	PB-IO13	108	LMB1-DB3	160	LMA2-DB1
5	LDB7	57	BTN2	109	LMB1-DB2	161	LMA2-DB0
6	LCE	58	BTN1	110	LMB1-DB1	162	LPA-IO18
7	LDB6	59	BTNO	111	LMB1-DB0	163	LPA-IO17
8	LWE	60	AND	112	LPB-LSBCLK	164	LPA-IO16
9	LDB5	61	CE	113	LPB-CSA	165	LPA-IO15
10	LADR5	62	CD	114	LPB-DB7	166	LPA-IO14
11	GND	63	CP	115	LPB-OE	167	LPA-IO13
12	VCCO	64	GND	116	GND	168	LPA-IO12
13	VCCINIT	65	VCCO	117	VCCO	169	GND
14	LDB4	66	VCCINIT	118	VCCINIT	170	VCCO
15	LADFM	67	CC	119	LPB-DB6	171	VCCINIT
16	LDB3	68	CG	120	LPB-WE	172	LPA-IO11
17	LADR3	69	AN1	121	LPB-DB5	173	LPA-IO10
18	LDB2	70	CB	122	LPB-ADR5	174	LPA-IO9
19	GND	71	AN2	123	LPB-DB4	175	LPA-IO8
20	LADR2	72	GND	124	GND	176	LPA-IO7
21	LDB1	73	CF	125	LPB-ADR4	177	GND
22	LADR1	74	CA	126	LPB-DB3	178	LPA-IO6
23	LDB0	75	AN3	127	LPB-ADR3	179	LPA-IO5
24	LADR0	76	VCCINIT	128	VCCINIT	180	LPA-IO4
25	GND	77	GCK1	129	LPB-DB2	181	LPA-IO3
26	VCCO	78	VCCO	130	VCCO	182	GCK2
27	VS	79	GND	131	GND	183	GND
28	VCCINIT	80	GCK0	132	LPB-ADR2	184	VCCO
29	HS	81	SW7/AC2	133	LPB-DB1	185	GCK3
30	BLUE	82	SW6	134	LPB-ADR1	186	VCCINIT
31	GRN	83	SW5	135	LPB-DB0	187	LPA-IO2
32	GND	84	SW4	136	LPB-ADR0	188	LPA-IO1
33	RED	85	GND	137	GND	189	LMA1-HNT
34	PS2C	86	SW3	138	LMA2-HNT	190	GND
35	PS2D	87	SW2	139	LMA2-RESET	191	LMA1-RESET
36	LD7	88	SW1	140	LMA2-WAIT	192	LMA1-WAIT
37	LD6	89	SW0	141	LMA2-WRITE	193	LMA1-WRITE
38	VCCINIT	90	LMB1-HNT	142	LMA2-DSTB	194	LMA1-DSTB
39	VCCO	91	VCCINIT	143	VCCINIT	195	LMA1-ASTB
40	MC1-DB4	92	GND	144	VCCO	196	VCCINIT
41	LD5	93	GND	145	GND	197	VCCO
42	LD4	94	LMB1-RESET	146	LMA2-ASTB	198	GND
43	LD3	95	LMB1-WAIT	147	LMA2-DB7	199	LMA1-DB7
44	LD2	96	LMB1-WRITE	148	LMA2-DB6	200	LMA1-DB6
45	LD1	97	LMB1-DSTB	149	LMA2-DB5	201	LMA1-DB5
46	LD0	98	LMB1-ASTB	150	LMA2-DB4	202	LMA1-DB4
47	AC3	99	LMB1-DB7	151	LMA2-DB3	203	LMA1-DB3
48	AC1	100	LMB1-DB6	152	LMA2-DB2	204	LMA1-DB2
49	AC0	101	LMB1-DB5	153	DIN/D0/NO	205	LMA1-DB1
50	MODE1	102	LMB1-DB4	154	BTN3	206	LMA1-DB0
51	GND	103	GND	155	CCLK	207	TCK
52	MODE0	104	DCNE	156	VCCO	208	VCCO

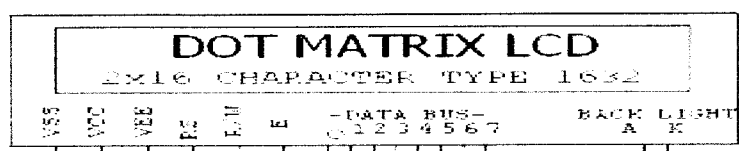
Tabel 3.3. *Pin* FPGA XC2S50

Pegasus FPGA Pin Assignments					
Pin	Function	Pin	Function	Pin	Function
1	GND	53	VCCO	105	VCCO
2	TMS	54	MODE2	106	PROGRAM
3	LLSBCLK	55	PB-I/O14	107	INIT7/O
4	LCSA	56	PB-I/O13	108	LMB1-DB3
5	LD87	57	BTN2	109	LMB1-DB2
6	LCE	58	BTN1	110	LMB1-DB1
7	LD86	59	BTNO	111	LMB1-DB0
8	LD8E	60	AND	112	LPB-LSBCLK
9	LD85	61	CE	113	LPB-CSA
10	LADR5	62	CD	114	LPB-DB7
11	GND	63	DP	115	LPB-OE
12	VCCO	64	GND	116	GND
13	VCCINIT	65	VCCO	117	VCCO
14	LD84	66	VCCINIT	118	VCCINIT
15	LADR4	67	CC	119	LPB-DB6
16	LD83	68	CG	120	LPB-WE
17	LADR3	69	AN1	121	LPB-DB5
18	LD82	70	CB	122	LPB-ADR5
19	GND	71	AN2	123	LPB-DB4
20	LADR2	72	GND	124	GND
21	LD81	73	CF	125	LPB-ADR4
22	LADR1	74	CA	126	LPB-DB3
23	LD8D	75	AN3	127	LPB-ADR3
24	LADR0	76	VCCINIT	128	VCCINIT
25	GND	77	GCK1	129	LPB-DB2
26	VCCO	78	VCCO	130	VCCO
27	VS	79	GND	131	GND
28	VCCINT	80	GCK0	132	LPB-ADR2
29	HS	81	SW7/AC2	133	LPB-DB1
30	BLUE	82	SN6	134	LPB-ADR1
31	GRN	83	SN5	135	LPB-DB0
32	GND	84	SN4	136	LPB-ADR0
33	RED	85	GND	137	GND
34	PS2C	86	SN3	138	LMA2-INT
35	PS2D	87	SN2	139	LMA2-RESET
36	LD7	88	SN1	140	LMA2-WAIT
37	LD6	89	SN0	141	LMA2-WRITE
38	VCCINIT	90	LMB1-INT	142	LMA2-DSTB
39	VCCO	91	VCCINIT	143	VCCINIT
40	MC1-DB4	92	GND	144	VCCO
41	LD5	93	GND	145	GND
42	LD4	94	LMB1-RESET	146	LMA2-ASTB
43	LD3	95	LMB1-WAIT	147	LMA2-DB7
44	LD2	96	LMB1-WRITE	148	LMA2-DB6
45	LD1	97	LMB1-DSTB	149	LMA2-DB5
46	LD0	98	LMB1-ASTB	150	LMA2-DB4
47	AC3	99	LMB1-DB7	151	LMA2-DB3
48	AC1	100	LMB1-DB6	152	LMA2-DB2
49	AC0	101	LMB1-DB5	153	DIND0/H0
50	MODE1	102	LMB1-DB4	154	BTN3
51	GND	103	GND	155	CCLK
52	MODE0	104	DONE	156	VCCO
				157	TDO
				158	GND
				159	TDI
				160	LMA2-DB1
				161	LMA2-DB0
				162	LPA-I/O18
				163	LPA-I/O17
				164	LPA-I/O16
				165	LPA-I/O15
				166	LPA-I/O14
				167	LPA-I/O13
				168	LPA-I/O12
				169	GND
				170	VCCO
				171	VCCINIT
				172	LPA-I/O11
				173	LPA-I/O10
				174	LPA-I/O9
				175	LPA-I/O8
				176	LPA-I/O7
				177	GND
				178	LPA-I/O6
				179	LPA-I/O5
				180	LPA-I/O4
				181	LPA-I/O3
				182	GCK2
				183	GND
				184	VCCO
				185	GCK3
				186	VCCINIT
				187	LPA-I/O2
				188	LPA-I/O1
				189	LMA1-INT
				190	GND
				191	LMA1-RESET
				192	LMA1-WAIT
				193	LMA1-WRITE
				194	LMA1-DSTB
				195	LMA1-ASTB
				196	VCCINIT
				197	VCCO
				198	GND
				199	LMA1-DB7
				200	LMA1-DB6
				201	LMA1-DB5
				202	LMA1-DB4
				203	LMA1-DB3
				204	LMA1-DB2
				205	LMA1-DB1
				206	LMA1-DB0
				207	TCK
				208	VCCO

3.1.3. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) yang digunakan sebagai *prototype* dari sebuah informasi. Agar terhubung dengan FPGA, LCD dilengkapi dengan 8 bit data bus (DB0 - DB7) yang digunakan untuk menyalurkan data ASCII (*American*

Standard Code for Information Interchange) maupun perintah pengatur kerjanya. Modul LCD sendiri terdiri dari *display* dan *chipset*, dimana *chipset* ini sendiri sebenarnya merupakan mikrokontroler. *Chipset* ini berfungsi untuk mengatur tampilan informasi serta berfungsi mengatur komunikasi dengan FPGA yang memakai tampilan LCD. Sebelum merancang, harus kita ketahui dahulu susunan pin dari LCD tersebut. Adapun susunan pin serta bentuk dari standar LCD 16 pin beserta fungsi dari masing-masing pin adalah seperti pada Gambar 3.2 berikut ini:



Gambar 3.14. LCD 2x16 Karakter

Tabel 3.4. Susunan LCD 2x16

NO	SIMBOL	LEVEL	FUNGSI
1	Vss	-	Power Supply 0 Volt (gnd)
2	Vcc	-	Power Supply 5 Volt \pm 10%
3	Vcc	-	Kontras LCD
4	RS	I/O	1 = Data, 0 = Instruksi
5	RW	I/O	1 = Baca, 0 = Tulis
6	EN	1 ke 0	Penyerempak (Clock)
7	DB0	I/O	Bus Data
:	:	-	Bus Data
14	DB7	I/O	Bus Data
15	A	-	Back light 4-42 V, 50-200 mA
16	K	-	Back light 0 V (gnd)

Untuk membuat suatu karakter tampilan LCD pada suatu posisi tertentu harus diketahui dahulu alamat dari LCD itu sendiri, misalnya diinginkan menulis

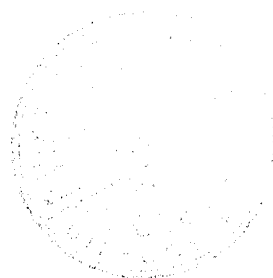
suatu kata dimulai dari baris kedua kolom pertama, berarti alamat yang digunakan pada LCD adalah $0C0h$, tanda h menunjukkan nilai tersebut dalam kode bilangan heksadesimal. Tabel 3.2 berikut merupakan peta alamat LCD dengan spesifikasi 2x16 karakter.

Tabel 3.5. Peta alamat LCD 2x16

Peta alamat LCD 2 x 16															
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Tabel 3.6. Karakter LCD

	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1
	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1
	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
xxxxx0000															
xxxxx0001	!	1	A	Q	a	q	。	ア	チ	ク	ク	ク	ク	ク	ク
xxxxx0010	"	2	B	R	b	r	「	イ	ツ	×	β	θ			
xxxxx0011	#	3	C	S	c	s	」	ウ	テ	ε	ε	ω			
xxxxx0100	\$	4	D	T	d	t	、	エ	ト	ト	ω	Ω			
xxxxx0101	%	5	E	U	e	u	・	オ	ナ	ユ	ク	ウ			
xxxxx0110	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ			
xxxxx0111	'	7	G	W	g	w	フ	キ	ヌ	ラ	q	π			
xxxxx1000	(8	H	X	h	x	イ	ク	ネ	リ	」	⌘			
xxxxx1001)	9	I	Y	i	y	ウ	ケ	ル	」	」	」			
xxxxx1010	*	:	J	Z	j	z	エ	コ	ハ	レ	i	チ			
xxxxx1011	+	;	K	[k	[オ	サ	ヒ	ロ	*	斤			
xxxxx1100	,	<	L	¥	l	l	パ	シ	フ	フ	¢	円			
xxxxx1101	-	=	M]	m]	ユ	ズ	ヘ	ン	も	÷			
xxxxx1110	.	>	N	^	n	^	ヨ	セ	ホ	ッ	ん				
xxxxx1111	/	?	O	_	o	_	ヶ	ツ	マ	°	ö	■			



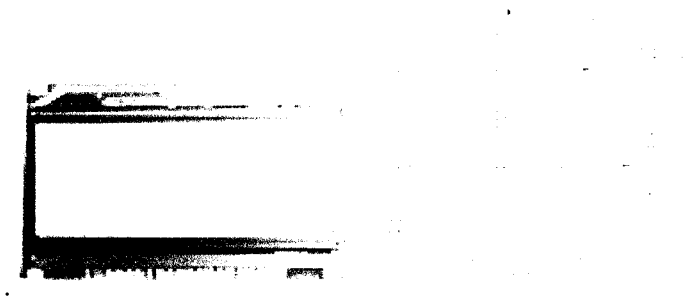
Tabel 3.7. Bilangan Desimal-Biner-Heksadesimal

Desimal	Biner	Heksadesimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Karakter LCD yang menggunakan kode ASCII memuat karakter Inggris dan karakter kana Jepang yang dipergunakan untuk membuat teks yang diinginkan. Setelah *display* diinisialisasi dan komunikasi ditetapkan, semua perintah dan pengiriman data pada karakter dengan 8 bit, menggunakan 2 kali operasi pengiriman data 4 bit. Tiap-tiap pengiriman data 8 bit harus dirubah menjadi pengiriman data 4 bit, jarak antara pengiriman pertama dan selanjutnya adalah 1 μ s, data *upper nibble* dikirimkan pertama kali, diikuti data *lower nibble*. Pada operasi tulis data 8 bit harus ada jarak antar pengiriman 40 μ s. sedangkan untuk *delay* harus dinaikkan menjadi 1.64 ms mengikuti perintah *Clear Display*.

LCD yang digunakan merupakan modul LCD dengan tampilan 2 x 16 baris dengan konsumsi daya yang rendah. LCD ini mempunyai CGROM (*Character Generator Read Onli Memory*), CGRAM (*Character Generator*

random Access Memory) dan DDRAM (*Display Data random Access Memory*), dan juga memiliki 3 *bit control* yaitu E yang merupakan *input clock*, RW sebagai *input* untuk memilih *read* atau *write* dan RS sebagai *register select*, juga memiliki 8 bit data yaitu DB0 sampai DB7 (Data Sheet, HD44780U, 2009:Page9)



Gambar 3.15. Konstruksi Dasar LCD (*Liquid Crystal Display*)

3.2. Perancangan Program

3.2.1 Xilinx (*Xilinx Foundation Series*)

Xilinx (*Xilinx Foundation Series*) adalah suatu perangkat lunak untuk merancang IC, yang nantinya hasil dari perancangan di-*download* ke *board* FPGA. Dengan menggunakan Xilinx, proses simulasi rangkaian yang telah dirancang dapat diketahui apakah benar ataukah masih ada yang mengandung kesalahan. Proses perancangan dengan menggunakan Xilinx sama seperti merancang suatu rangkaian logika secara manual, akan tetapi dengan menggunakan simulator Xilinx dapat meminimalisasi kesalahan pada proses perancangan. Untuk proses perancangan rangkaian digital, Xilinx mempunyai 3 cara, yaitu dengan menggunakan *state diagram*, HDL, dan *schematic*, pada proses

perancangan kita dapat menggunakan salah satunya atau menggabungkan ketiganya.

3.2.2 Verilog HDL

Verilog *Hardware Description Language* merupakan suatu bahasa standar yang digunakan untuk mendeskripsikan suatu rangkaian digital. Suatu kode HDL merupakan suatu representasi dari gambar rangkaian digital, dimana pada prakteknya, komputer akan lebih mudah membaca suatu kode bahasa dari pada gambar buatan manusia. Verilog HDL juga mempunyai kemampuan lebih dari sekedar menggambar rangkaian digital, karena Verilog HDL dapat melakukan pemodelan rangkaian digital menggunakan kode sekuensial yang bersifat behavioral. Bagian-bagian dari kode Verilog adalah:

- a. *Module description* merupakan bagian awal dari kode verilog yang menyatakan nama modul yang dibuat beserta daftar nama pin input dan output dari module tersebut.

```

module      and_or_demo (A, B, C, R)
input      A, B, C;
output     R;
wire      A, B, C, R;
assign     R= (A | B) & (B | C);
endmodule

```

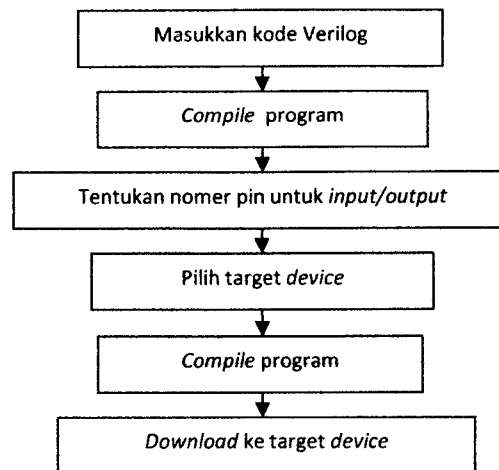
- b. *Port Declaration*, dalam bagian ini, setiap nama signal *input* dan *output* dideklarasikan. Ada tiga macam deklarasi I/O, yaitu *input*, *output* dan *bidirectional* I/O.
- c. *Signal declaration* merupakan bagian dimana sinyal-sinyal I/O maupun sinyal internal dideklarasikan. Setiap objek sinyal, baik pin *Input/Output*,

sinyal interkoneksi, maupun sinyal memori baru dideklarasikan dibagian ini. Ada dua macam tipe objek signal, yang pertama adalah tipe *reg* dan yang kedua adalah *wire*.

- d. *Parameter Definition* digunakan untuk mendefinisikan konstanta-konstanta dalam beberapa bagian kode, sehingga dapat dihasilkan kode yang lebih solid, lebih mudah dibaca, dan fleksibel terhadap modifikasi. Parameter dapat digunakan sebagai nilai yang di *assign* ke sebuah sinyal, digunakan untuk membuat kode arsitektur rangkaian yang *scalable* (lebar bit suatu sinyal dapat diubah dengan mudah) baik dalam bentuk *combinational assignment* maupun *behavioral assignment*.
- e. *Combinational assignment* merupakan *assignment* nilai dari sinyal dengan menggunakan suatu fungsi kombinasional tertentu.
- f. *Behavioral assignment* dapat menghasilkan rangkaian kombinasional yang memiliki perilaku sesuai dengan urutan *statement-statement sequential* pada deskripsi behavioral.
- g. *Component Instantiation* adalah *syntax* untuk memasukkan desain submodul kedalam modul utama dan menghubungkan sinyal internal dalam modul utama dengan pin I/O submodul tersebut.

Langkah-langkah menggunakan *software* untuk memprogram FPGA dapat dilihat pada Gambar 3.16. Program Verilog dimasukan, untuk mengetahui apakah program yang dibuat tersebut sudah benar maka program di *compile*. Jika sudah dipastikan tidak ada *error* pada program maka pin *input* dan *output* yang digunakan diisikan dengan nomor pin yang dapat dilihat pada Tabel 3.3.

pemilihan peralatan salah satunya dengan pengaturan *inteface* menggunakan JTAG. Setelah semua tahap dilakukan dan tidak terdapat *error* maka program di-*download* ke perangkat yang diinginkan. *Software* menyediakan simulasi desain yang menyajikan pendukung kinerja sistem yang ada.

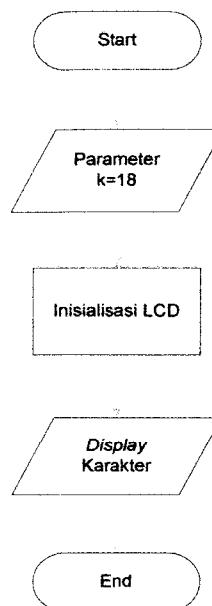


Gambar 3.16. Blok diagram alir penggunaan *software* untuk memprogram FPGA

Pada program, sistem dirancang sedemikian rupa, untuk merealisasikannya dibuat empat blok desain, yakni: *Clock*, *parameter*, inisialisasi LCD, dan *output (display)*. Pada perancangan ini menggunakan satu blok parameter, yang berfungsi untuk mengatur keluaran *clock* utama dengan status keluaran yang berubah secara konstan sehingga menghasilkan gelombang kotak atau pulsa. Blok *counter* berfungsi sebagai untuk mencacah jumlah karakter dan kontrol LCD yang akan dikeluarkan pada *display*.

3.2.3. Program Utama

Untuk menjalankan sistem, diperlukan program untuk mengendalikan perangkat keras. Perancangan program sistem "running text" pada LCD yang diimplementasikan pada FPGA ditunjukkan pada *flowchart* dibawah ini.



Gambar 3.16. *Flowchart* program.

Perancangan kode program dibawah ini dipakai untuk menampilkan *running text* pada LCD.

```

module lcd (clk, sf_ce0, lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5,
lcd_6, lcd_7);
parameter k = 18;

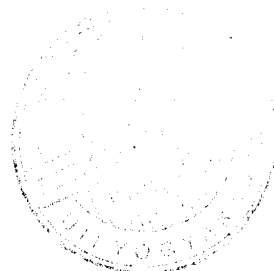
```

Kode diatas merupakan bagian awal program dibuat, yaitu dengan menentukan nama modul yang dibuat beserta nama pin masukan dan keluaran dari modul *running text* pada LCD. Sedangkan parameter berfungsi untuk mendefinisikan konstanta yang dipakai dalam *case*.

```

always @ (posedge clk) begin
count <= count + 1;

```



Pada kode diatas, *clock* yang diperlukan untuk menjalankan program pada LCD, diberikan secara terus-menerus. Karena *clock* yang diberikan disini untuk *men-drive* pada nilai cacah positif.

```

case (count[k+7:k+2])
  0: lcd_code <= 6'h03;           // Inisialisasi Power-On
  1: lcd_code <= 6'h03;
  2: lcd_code <= 6'h03;
  3: lcd_code <= 6'h02;
  4: lcd_code <= 6'h02;           // function set
  5: lcd_code <= 6'h08;
  6: lcd_code <= 6'h00;           // entry mode set
  7: lcd_code <= 6'h06;
  8: lcd_code <= 6'h00;           // kontrol display on/off
  9: lcd_code <= 6'h0E;
 10: lcd_code <= 6'h00;           // display clear
 11: lcd_code <= 6'h01;

```

Inisialisasi kontrol LCD diberikan sesuai dengan data yang ada pada *datasheet* LCD yang digunakan. Mulai dari inisialisasi power-on yang diberikan dengan kode 0: `lcd_code <= 6'h03` untuk *function set* diberikan kode 4: `lcd_code <= 6'h02` untuk *upper nibble*, sedangkan *lower nibble* diberikan kode 5: `lcd_code <= 6'h08`; dengan data biner “0010 1000” untuk mengatur lebar data, nomer baris pada *display* dan *font* karakter. Untuk pengaturan *entry mode set* yang berfungsi untuk mengatur mode geser diberikan kode 6: `lcd_code <= 6'h00`; 7: `lcd_code <= 6'h06`; dengan data biner “0000 0110” yang mengatur perpindahan *running teks* dari kanan kekiri atau dari kiri ke kanan. *Display on/off* diberikan kode 8: `lcd_code <= 6'h00`; 9: `lcd_code <= 6'h0E`; dengan data biner “0000 1110”. *Display clear* dengan kode biner “0000 0001” yang berguna untuk posisi awal *display* pada pojok kiri, waktu eksekusi dari *display clear*-nya dari 82 μ s – 1.64 ms.

```

12: lcd_code <= 6'h25;           // T
13: lcd_code <= 6'h24;
14: lcd_code <= 6'h24;           // E
15: lcd_code <= 6'h25;
16: lcd_code <= 6'h22;           //
17: lcd_code <= 6'h20;
18: lcd_code <= 6'h25;           // U
19: lcd_code <= 6'h25;
20: lcd_code <= 6'h24;           // I
21: lcd_code <= 6'h29;
22: lcd_code <= 6'h24;           // I
23: lcd_code <= 6'h29;
24: lcd_code <= 6'h22;           //
25: lcd_code <= 6'h20;
26: lcd_code <= 6'h24;           // J
27: lcd_code <= 6'h2A;
28: lcd_code <= 6'h24;           // O
29: lcd_code <= 6'h2F;
30: lcd_code <= 6'h24;           // G
31: lcd_code <= 6'h27;
32: lcd_code <= 6'h24;           // J
33: lcd_code <= 6'h2A;
34: lcd_code <= 6'h24;           // A
35: lcd_code <= 6'h21;

```

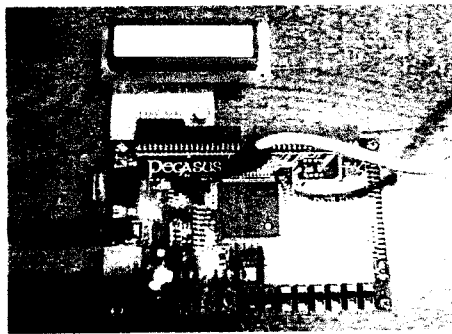
Kode diatas memberikan kode karakter yang berfungsi untuk menampilkan teks pada *display* yang berbentuk “TE UII JOGJA”. Kode tersebut memakai kode ASCII yang dikirim secara 2 kali per 4 bit-nya. Untuk karakter A data binernya adalah “0100 0001” yang dirubah dalam bentuk heksadesimal menjadi ‘h41 begitupun dengan karakter yang lain juga mempunyai kode yang berbeda-beda.

BAB IV

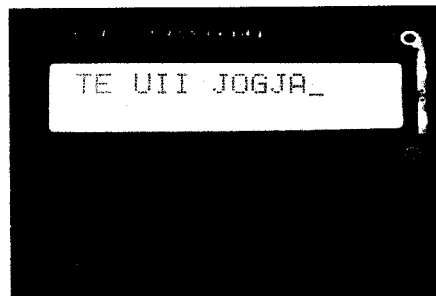
HASIL PENGUJIAN DAN PEMBAHASAN

4.1. PENDAHULUAN

Komponen utama dalam sistem rancang bangun *running text* pada LCD yang digunakan dalam penelitian ini adalah LCD, *board* FPGA Spartan II dan seperangkat komputer. Prinsip kerja dari sistem ini adalah saat FPGA yang sudah diprogram oleh seperangkat komputer dengan *software* Xilinx ISE 7.1i dan pada program telah berisi tentang pengaturan sistem penulisan karakter pada LCD dengan menggunakan bahasa pemrograman Verilog HDL. Keluaran dari FPGA akan masuk ke rangkaian LCD untuk menampilkan karakter yang telah dibuat.



Gambar 4.1. LCD pada FPGA

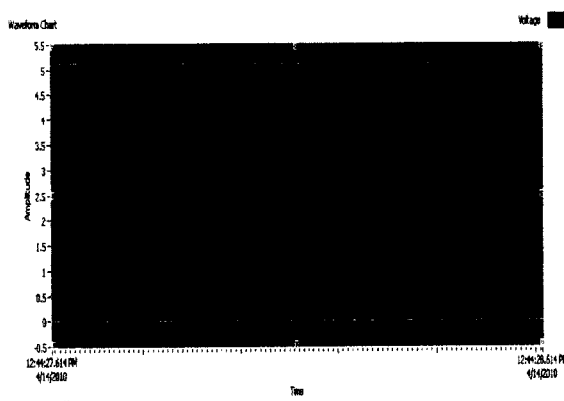


Gambar 4.2. Karakter yang muncul pada *display* LCD

4.2. Pengujian Perangkat Keras

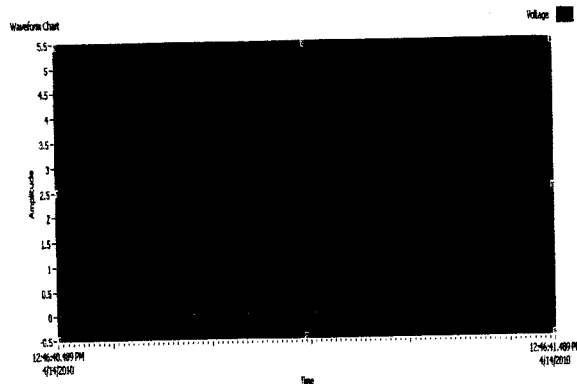
4.2.1. Analisis rangkaian LCD dengan menggunakan Labview Instrument NI USB-6009.

Pada pengujian ini digunakan LabView Instrument NI USB-6009 untuk mendapatkan sinyal rangkaian saat diaktifkan, perancangan ini menggunakan LCD LMB 162 A, 4 buah resistor 390Ω yang dihubungkan dengan *data bus* (DB 4 sampai DB 7) pada LCD, sedangkan pengujian dilakukan pada 4 pin data bus (DB 4 sampai DB 7).



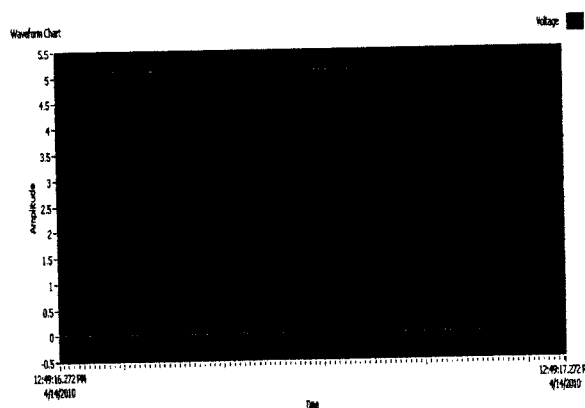
Gambar 4.3. Pulsa pada pin DB-4

Pada pengukuran DB-4 yang terhubung dengan pin 16 pada *connector pinout A1 board* FPGA ditampilkan lebar sinyal karakter yang didapatkan dengan pengukuran menggunakan penghitungan waktu *real* untuk pengiriman data bit karakter yang dilakukan dengan 2 kali per 4 bit satu kali karakter dibutuhkan waktu selama 0,78 s. Data biner untuk karakter T yaitu “0101 0100” yang dikonversikan ke bentuk heksadesimal “h54”, memerlukan waktu 0.21 s untuk siklus awal lebar data bit per karakter pada pengiriman data, dengan tegangan puncak ke puncak (Vpp) 5 Volt.



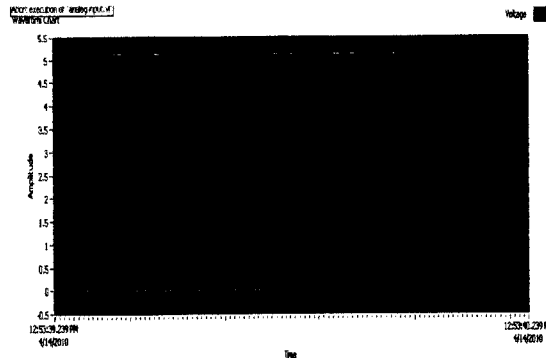
Gambar 4.4. Pulsa pada pin DB-5

Sinyal karakter yang didapatkan dari pengukuran dari DB-5 yang dihubungkan dengan pin 14 *connector A1 board* FPGA untuk pengiriman data bit karakter yang dilakukan dengan 2 kali per 4 bit satu kali karakter dibutuhkan waktu selama 0.74 s. Sedangkan siklus awal pengiriman data bit per karakter memerlukan waktu selama 0.15 s.



Gambar 4.5. Pulsa pada pin DB-6

Sinyal karakter yang didapatkan dari pengukuran dari DB-5 yang dihubungkan dengan pin 9 *connector A1 board* FPGA untuk pengiriman data bit karakter yang dilakukan dengan 2 kali per 4 bit satu kali karakter dibutuhkan waktu selama 0.65 s. Sedangkan siklus awal pengiriman data bit per karakter memerlukan waktu selama 0.5 s.



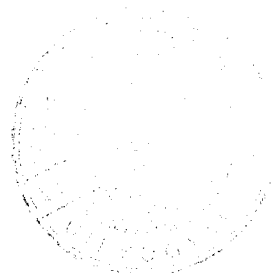
Gambar 4.6. Pulsa pada pin DB-7

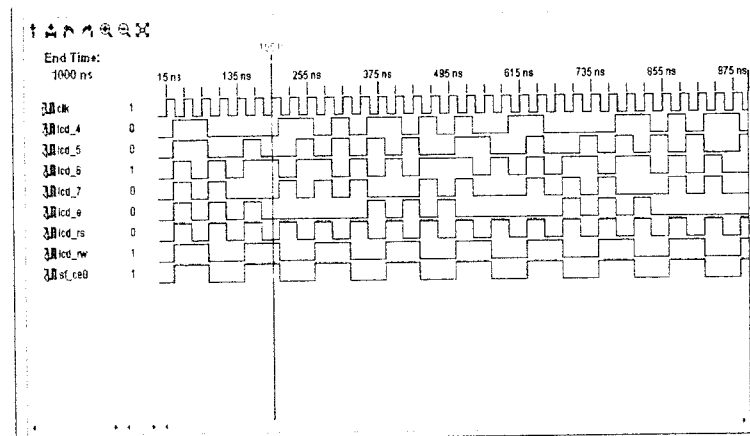
Sinyal karakter yang didapatkan dari pengukuran dari DB-5 yang dihubungkan dengan pin 7 *connector A1 board* FPGA untuk pengiriman data bit karakter yang dilakukan dengan 2 kali per 4 bit satu kali karakter dibutuhkan waktu selama 0.74 s. Sedangkan siklus awal pengiriman data bit per karakter memerlukan waktu selama 0.9 s dengan tegangan puncak ke puncak (V_{pp}) 5 Volt.

4.3 Pengujian Perangkat Lunak

4.3.1 Analisa sistem rancang bangun *running text* dengan *Test Bench Waveform* Xilinx ISE 7.1i.

Setelah program untuk mengkonfigurasi FPGA selesai dibuat, kemudian untuk memastikan kinerja dari program apakah sudah sesuai dengan yang diinginkan, maka ada baiknya program tersebut disimulasikan terlebih dahulu. Hasil dari simulasi adalah berupa *timing diagram* dari program dan untuk melakukannya digunakan *Test Bench Waveform* dari Xilinx ISE 7.1i. Ada 8 buah keluaran dari sistem yang telah dibuat dengan 1 masukan clock.

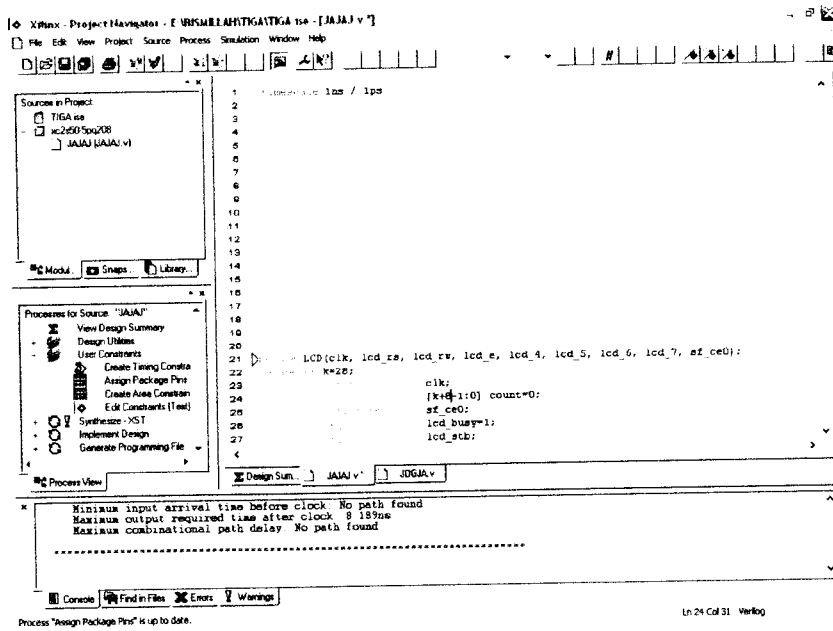




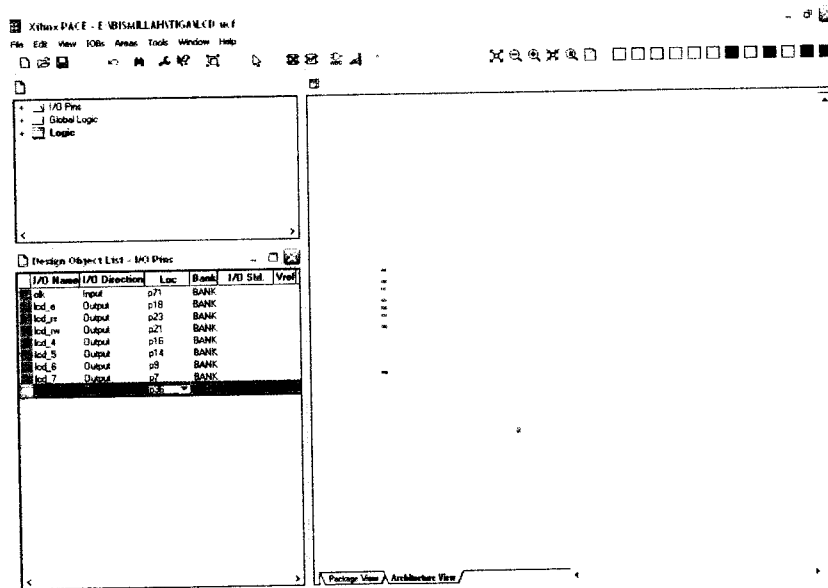
Gambar 4.7. Timing diagram running text pada LCD

4.3.2 Analisis sistem rancang bangun *running text* pada *Software* FPGA Xilinx ISE 7.1i.

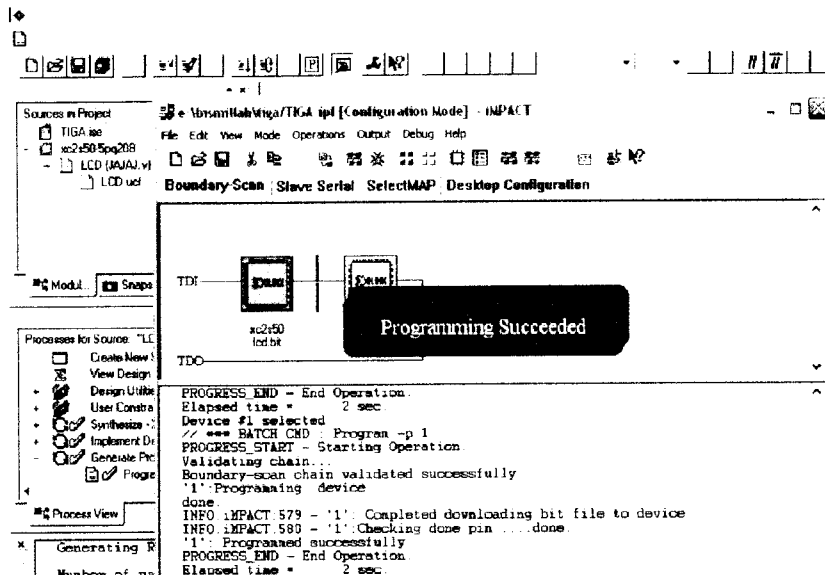
FPGA digunakan sebagai sistem utama, yaitu untuk menghubungkan dengan perangkat lunak lainnya. Sistem tersebut harus melalui proses *synthesize_XST* dan *implementation design*. Jika tidak ada *error* maka dapat dilanjutkan ke proses *generate programming file*. Rangkaian yang telah dirancang dalam bentuk *file schematic* akan diimplementasikan ke dalam *board* FPGA. Oleh karena itu, perlu diatur pemasangan kaki *input* dan *output* rangkaian *schematic* terhadap *board* FPGA. Hal ini dapat dilakukan dengan proses *assign package pin*. Didalam *assign package pin*, pengaturan *input* dan *output* dilakukan dengan cara mengisi tabel yang telah tersedia untuk memberikan alamat yang akan digunakan sebagai *input* dan *output*.



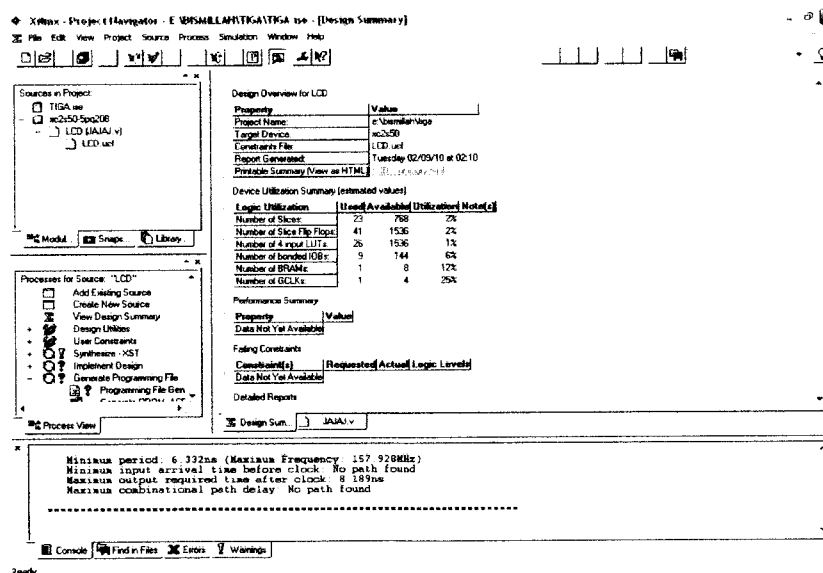
Gambar 4.11. Proses Sintesis Program



Gambar 4.12. Proses Assign pin pada board FPGA



Gambar 4.13. Proses *Download* sukses



Gambar 4.14. *Slice* FPGA yang terpakai.

Tanda *WARNING* yang muncul saat program disintesis hanyalah merupakan informasi yang memberi tahu bahwa *syntax* ada yang tidak digunakan dan bukan suatu kesalahan. Selanjutnya pada proses *assign pin* yang akan digunakan agar keluaran dapat ditampilkan pada LCD sesuai dengan inisiasi *input* dan *output* yang ada program. Suksesnya *download* program yang ada pada proses

ini menunjukkan bahwa proses ini dapat memberikan keluaran yang ada pada *device*.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan serta pengujian sistem maka dapat disimpulkan beberapa hal yaitu :

- a. Perancangan *running text* pada LCD yang diimplementasikan pada FPGA yang menampilkan kata “TE UII JOGJA” dapat terwujud dengan baik.
- b. Performansi FPGA yang digunakan adalah sebanyak 23 slice dari 768 (2 %).
- c. LCD akan tetap menampilkan *running text* walaupun kabel JTAG pada FPGA dicabut.

5.2. Saran

Perancangan *running text* pada LCD ini dinilai masih terdapat kekurangannya maka untuk pengembangan selanjutnya disarankan untuk memperhatikan hal berikut:

- a. Alat ini dapat diimplementasikan pada *device* yang lain untuk pengembangan penelitian lebih lanjut, dengan dikombinasikan pada kode program dan alat lainnya.
- b. Pada alat ini bisa ditambahkan *keypad* sebagai masukannya, yang dapat memudahkan *user* untuk mengganti teks sesuai dengan keinginan.

DAFTAR PUSTAKA

<http://www.edaboard.com>

<http://www.fpga4fun.com/text> LCD module

http://www.Xilinx.com/products/silicon_solutions/fpgas/Spartan_series/spartan2_fpga/capabilities/architecture.htm

<http://www.ndoware.com>

<http://www.pdfsea.com/down/2008/0710/7546.html>/datasheet LMB1632a

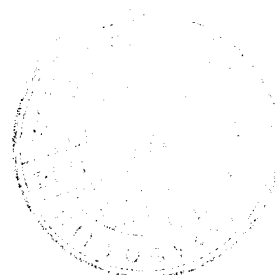
Yurisdian, Novita, 2005, *Perancangan Pengendalian Display Dot Matriks Dengan Mikrokontroler At89s51*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Setyono, Pipin, 2006, *Desain IC Mesin Penjual Otomatis Berbasis FPGA*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Fakta Sari, Dini, 2006, *Perancangan Pengendalian Lampu Lalu Lintas dan Motor Stepper menggunakan FPGA*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Pramadianto, Adisatya, 2007, *Penampil Informasi Dot Matriks dengan Animasi Terprogram Berbasis Mikrokontroler AT89C52*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Widayanto, Wahyu, 2008, *Pengendalian "Running Text" Dengan Sms Menggunakan Mikrokontroler At89s52*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.



Pratama Putra, Randi, 2009, *Perancangan Software untuk Tampilan Animasi Susunan Dot Matriks 8x8*, Skripsi Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta

Training Center Electrical Engineering. *Modul Pelatihan IC Design With Based FPGA Pegasus*. Jogjakarta: Teknik Elektro Universitas Islam Indonesia, 2006.

Firmansyah, ST., Oki, *Verilog HDL Training*, Sekolah Teknik Elektro dan Informatika (STIE), Institut Teknologi Bandung.

Widjanarka N., IR Wijaya, 2006, *Teknik Digital*, Penerbit Erlangga, Jakarta.

LAMPIRAN

TO FPGA

LCD

GND 1

VCC 3

RS 5

RW 7

E 9

DB_4 11

DB_5 13

DB_6 15

DB_7 17

R5
1KR ▲

R6=390Ω
R7=390Ω
R8=390Ω
R9=390Ω

▲ D1
4148

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

RANGKAIAN LCD PADA FPGA



Spartan-II FPGA Family Data Sheet

01 June 13, 2008

Product Specification

This document includes all four modules of the Spartan®-II FPGA data sheet.

Module 1: Introduction and Ordering Information

DS001-1 (v2.8) June 13, 2008

Introduction

Features

General Overview

Product Availability

User I/O Chart

Ordering Information

Module 2: Architectural Description

DS001-2 (v2.8) June 13, 2008

Architectural Description

- Spartan-II Array
- Input/Output Block
- Configurable Logic Block
- Block RAM
- Clock Distribution: Delay-Locked Loop
- Boundary Scan

Development System

Configuration

- Configuration Timing

Design Considerations

Module 3: DC and Switching Characteristics

DS001-3 (v2.8) June 13, 2008

- DC Specifications
 - Absolute Maximum Ratings
 - Recommended Operating Conditions
 - DC Characteristics
 - Power-On Requirements
 - DC Input and Output Levels
- Switching Characteristics
 - Pin-to-Pin Parameters
 - IOB Switching Characteristics
 - Clock Distribution Characteristics
 - DLL Timing Parameters
 - CLB Switching Characteristics
 - Block RAM Switching Characteristics
 - TBUF Switching Characteristics
 - JTAG Switching Characteristics

Module 4: Pinout Tables

DS001-4 (v2.8) June 13, 2008

- Pin Definitions
- Pinout Tables

IMPORTANT NOTE: This Spartan-II FPGA data sheet is in four modules. Each module has its own Revision History at the end of the module. Use the PDF "Bookmarks" for easy navigation in this volume.



Spartan-II FPGA Family: Introduction and Ordering Information

01-1 (v2.8) June 13, 2008

Product Specification

Introduction

Spartan®-II Field-Programmable Gate Array family offers users high performance, abundant logic resources, a rich feature set, all at an exceptionally low price. The member family offers densities ranging from 15,000 to 200,000 system gates, as shown in Table 1. System performance is supported up to 200 MHz. Features include 16K RAM (to 56K bits), distributed RAM (to 75,264 bits), selectable I/O standards, and four DLLs. Fast, flexible interconnect means that successive design iterations continue to meet timing requirements.

Spartan-II family is a superior alternative to mask-programmed ASICs. The FPGA avoids the initial cost, lengthy development cycles, and inherent risk of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary (impossible with ASICs).

Features

- Second generation ASIC replacement technology
- Densities as high as 5,292 logic cells with up to 200,000 system gates
- Streamlined features based on Virtex® FPGA architecture
- Unlimited reprogrammability
- Very low cost
- Cost-effective 0.18 micron process

- System level features
 - SelectRAM™ hierarchical memory:
 - 16 bits/LUT distributed RAM
 - Configurable 4K bit block RAM
 - Fast interfaces to external RAM
 - Fully PCI compliant
 - Low-power segmented routing architecture
 - Full readback ability for verification/observability
 - Dedicated carry logic for high-speed arithmetic
 - Efficient multiplier support
 - Cascade chain for wide-input functions
 - Abundant registers/latches with enable, set, reset
 - Four dedicated DLLs for advanced clock control
 - Four primary low-skew global clock distribution nets
 - IEEE 1149.1 compatible boundary scan logic
- Versatile I/O and packaging
 - Pb-free package options
 - Low-cost packages available in all densities
 - Family footprint compatibility in common packages
 - 16 high-performance interface standards
 - Hot swap Compact PCI friendly
 - Zero hold time simplifies system timing
- Core logic powered at 2.5V and I/Os powered at 1.5V, 2.5V, or 3.3V
- Fully supported by powerful Xilinx® ISE® development system
 - Fully automatic mapping, placement, and routing

Table 1: Spartan-II FPGA Family Members

Device	Logic Cells	System Gates (Logic and RAM)	CLB Array (R x C)	Total CLBs	Maximum Available User I/O ⁽¹⁾	Total Distributed RAM Bits	Total Block RAM Bits
XC2S15	432	15,000	8 x 12	96	86	6,144	16K
XC2S30	972	30,000	12 x 18	216	92	13,824	24K
XC2S50	1,728	50,000	16 x 24	384	176	24,576	32K
XC2S100	2,700	100,000	20 x 30	600	176	38,400	40K
XC2S150	3,888	150,000	24 x 36	864	260	55,296	48K
XC2S200	5,292	200,000	28 x 42	1,176	284	75,264	56K

Notes:
All user I/O counts do not include the four global clock/user input pins. See details in Table 2, page 4.

©2008 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, the Brand Window, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

General Overview

Spartan-II family of FPGAs have a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), surrounded by a perimeter of programmable Input/Output Blocks (IOBs). There are four Delay-Locked Loops (DLLs), one at each corner of the die. Two columns of Block RAM lie on opposite sides of the die, between the IOBs and the CLB columns. These functional elements are connected by a powerful hierarchy of versatile routing channels (see Figure 1).

Spartan-II FPGAs are customized by loading configuration data into internal static memory cells. Unlimited reprogramming cycles are possible with this approach. The values in these cells determine logic functions and interconnections implemented in the FPGA. Configuration data can be read from an external serial PROM (master

serial mode), or written into the FPGA in slave serial, slave parallel, or Boundary Scan modes.

Spartan-II FPGAs are typically used in high-volume applications where the versatility of a fast programmable solution adds benefits. Spartan-II FPGAs are ideal for shortening product development cycles while offering a cost-effective solution for high volume production.

Spartan-II FPGAs achieve high-performance, low-cost operation through advanced architecture and semiconductor technology. Spartan-II devices provide system clock rates up to 200 MHz. In addition to the conventional benefits of high-volume programmable logic solutions, Spartan-II FPGAs also offer on-chip synchronous single-port and dual-port RAM (block and distributed form), DLL clock drivers, programmable set and reset on all flip-flops, fast carry logic, and many other features.

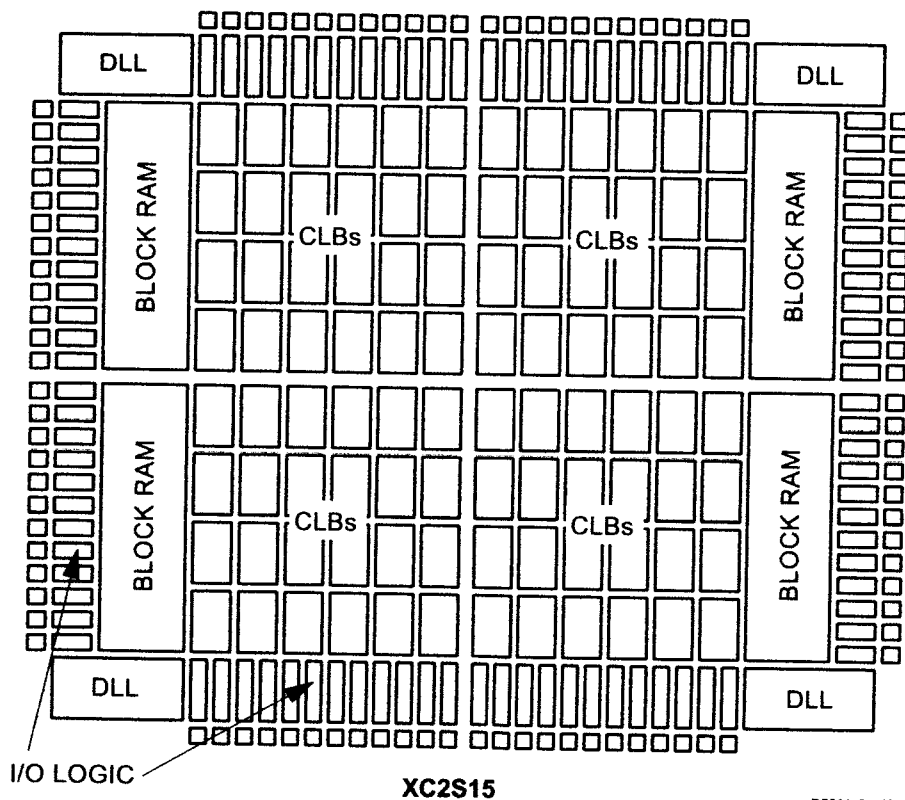


Figure 1: Basic Spartan-II Family FPGA Block Diagram

DS001_01_091800

Spartan-II Product Availability

Table 2 shows the maximum user I/Os available on the device and the number of user I/Os available for each device/package combination. The four global clock pins are usable as additional user I/Os when not used as a global clock. These pins are not included in user I/O counts.

Table 2: Spartan-II FPGA User I/O Chart⁽¹⁾

Device	Maximum User I/O	Available User I/O According to Package Type					
		VQ100 VQG100	TQ144 TQG144	CS144 CSG144	PQ208 PQG208	FG256 FGG256	FG456 FGG456
C2S15	86	60	86	(Note 2)	-	-	-
C2S30	92	60	92	92	(Note 2)	-	-
C2S50	176	-	92	-	140	176	-
C2S100	176	-	92	-	140	176	(Note 2)
C2S150	260	-	-	-	140	176	260
C2S200	284	-	-	-	140	176	284

Notes:

All user I/O counts do not include the four global clock/user input pins.
Discontinued by PDN2004-01.

SPECIFICATION

Model: LMB162A

1. BASIC SPECIFICATIONS

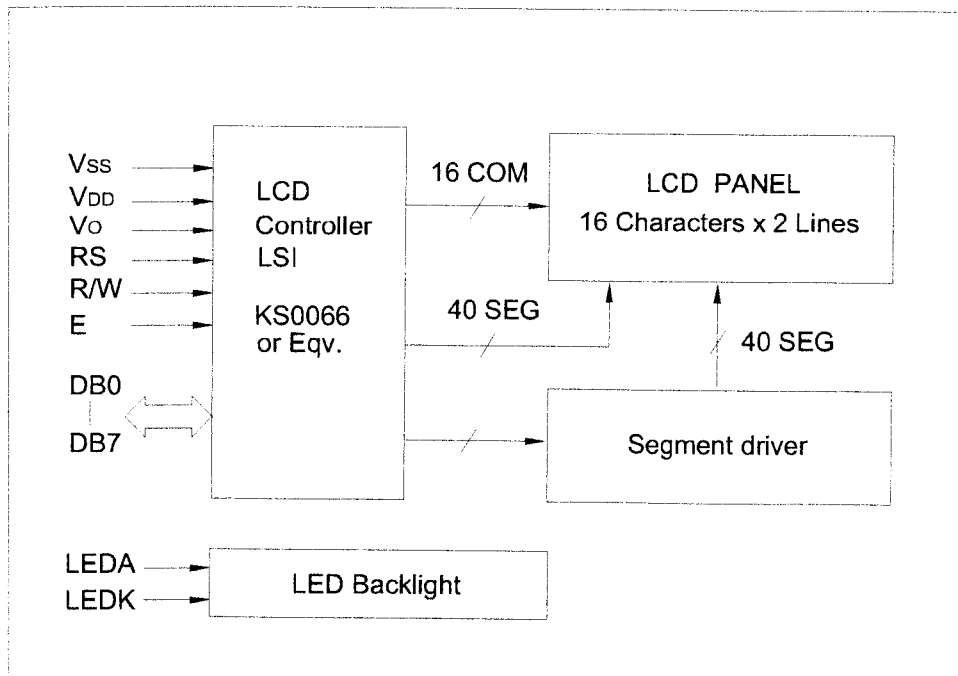
1.1 Display Specifications

LCD Mode	:	STN—Positive—Transflective
Display Color	:	Dark Blue
Background Color	:	Yellow-Green
Driving Duty	:	1/16 Duty
Viewing Direction	:	6:00
Backlight	:	LED

1.2 Mechanical Specifications

Outline Dimension	:	80.0(W) X 36.0(H) X 14.0(T)	mm
Viewing Area	:	64.6(W) X 16.0(H)	mm
Number of Characters	:	16 Characters X 2 Lines	
Character Size	:	2.95 X 5.55	mm
Dot Size	:	0.55 X 0.65	mm
Weight	:		

1.3 Block Diagram



1.4 Terminal Functions

Pin No.	Symbol	Level	Function
1	VSS	-	Ground
2	VDD	-	Power Supply for Logic (+5V)
3	VO	-	Power Supply for LCD
4	RS	H/L	Register Selection H: Display Data L: Instruction Code
5	R/W	H/L	Read/Write Selection H: Read Operation L: Write Operation
6	E	H, H→L	Enable Signal. Read data when E is "H", write data at the falling edge of E.
7	DB0	H/L	In 8-bit mode, used as low order bi-directional data bus. In 4-bit mode, open these terminals.
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	In 8-bit mode, used as high order bi-directional data bus.
12	DB5	H/L	
13	DB6	H/L	In 4-bit mode, used as both high and low order data bus.
14	DB7	H/L	
15	LEDA	--	LED Power Supply (+5V)
16	LEDK	--	LED Power Supply (0v)

2. ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Min.	Max.	Unit
Supply Voltage(Logic)	VDD-VSS	-0.3	7.0	V
Supply Voltage(LCD)	VDD-VO	-0.3	13.0	V
Input Voltage	VI	-0.3	VDD+0.3	V
Operating Temp.	Topr	-20	70	°C
Storage Temp.	Tstg	-30	80	°C

3. ELECTRICAL CHARACTERISTICS

3.1 DC Characteristics

(VDD=5.0V±10%, Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	UNIT
Supply Voltage (Logic)	VDD		4.5	5.0	5.5	V
Supply Voltage (LCD Drive)	VDD-VO		--	5.0	--	V
Input High Voltage	VIH		2.2	--	VDD	V
Input Low Voltage	VIL		-0.3	--	0.6	V
Output High Voltage	VOH	IOH=-0.2mA	2.4	--	VDD	V
Output Low Voltage	VOL	IOL=1.2mA	0	--	0.4	V
Supply Current (Logic)	IDD	VDD=5.0V	--	1.5	3.0	mA

3.2 Interface Timing Chart

(VDD=5.0V±10%, Ta=25°C)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode Refer to fig.1	E Cycle Time	tc	500	--	--	ns
	E Rise/Fall Time	tr, tf	--	--	20	
	E Pulse Width (High,Low)	tw	230	--	--	
	R/W and RS Setup Time	tsu1	40	--	--	
	R/W and RS Hold Time	th1	10	--	--	
	Data Setup Time	tsu2	80	--	--	
	Data Hold Time	th2	10	--	--	
Read Mode Refer to fig.2	E Cycle Time	tc	500	--	--	ns
	E Rise/Fall Time	tr, tf	--	--	20	
	E Pulse Width (High,Low)	tw	230	--	--	
	R/W and RS Setup Time	tsu	40	--	--	
	R/W and RS Hold Time	th	10	--	--	
	Data Output Delay Time	td	--	--	120	
	Data Hold Time	tdH	5	--	--	

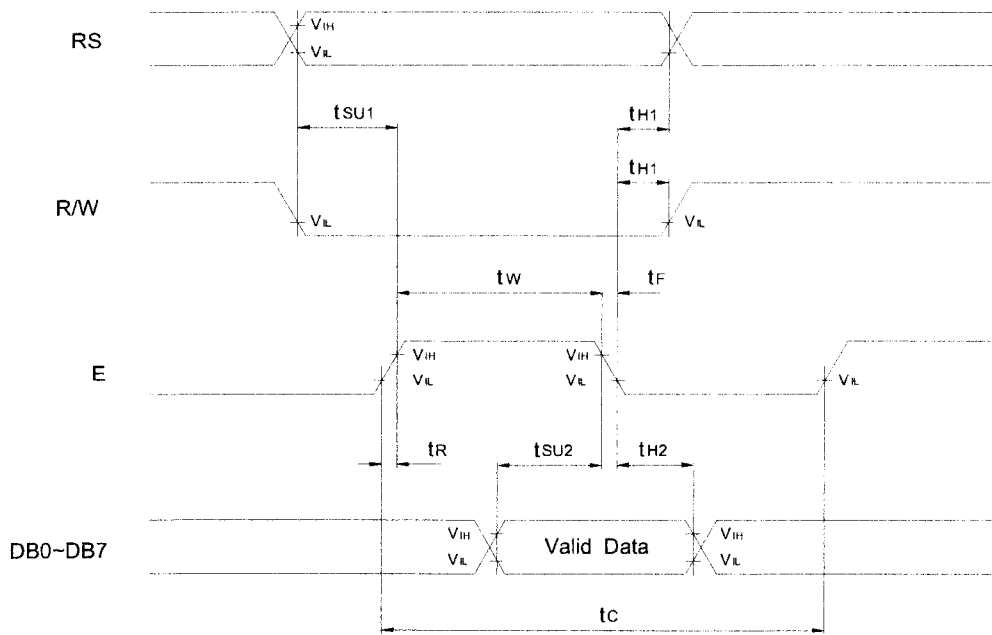


Fig.1 MPU Write Timing

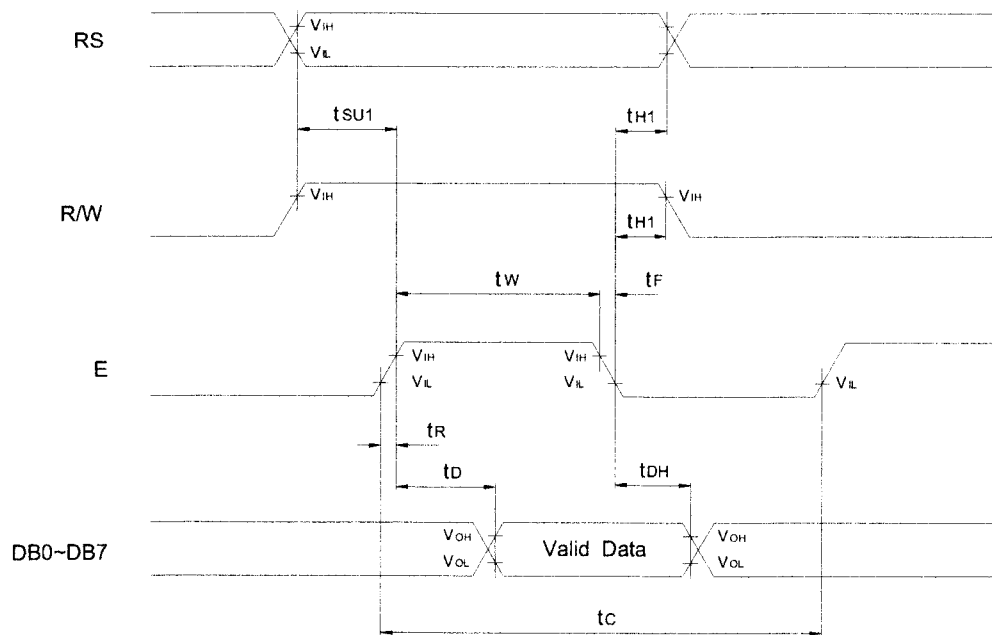
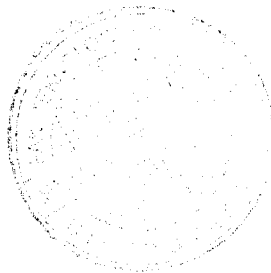
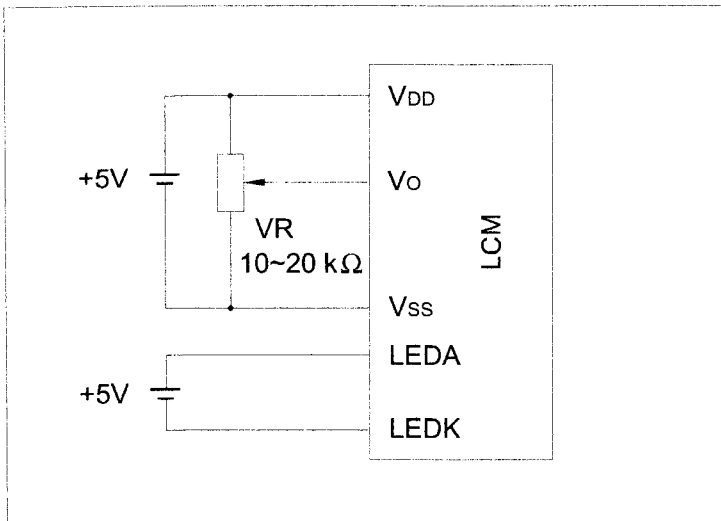


Fig.2 MPU Read Timing

3.3 LED Backlight Characteristics (Ta=25°C)

Item	Symbol	Condition	Min.	Typ.	Max.	UNIT
Forward Voltage	Vf		3.9	4.1	4.3	V
Forward Current	If	Vf=4.1V	--	110	--	mA
Peak Wave Length	λ_p	If=110mA	--	568	--	nm
Luminance	Lv	If=110mA	--	100	--	cd/m ²

3.4 Power Supply



4. OPERATING PRINCIPLES & METHODES

4.1 Register

The LCD Controller has two 8-bit registers, the Instruction register (IR) and the data register (DR).

The IR is a write only register to store instruction codes like Display Clear or Cursor Shift as well as addresses for the Display Data RAM (DD RAM) or the Character Generator RAM (CG RAM).

The DR is a read/write register used for temporarily storing data to be read/written to/from the DD RAM or CG RAM. Data written into the DR is automatically written into DD RAM or CG RAM by an internal operation of the display controller.

The DR is also used to store data when reading out data from DD RAM or CG RAM.

When address information is written into IR, data is read out from DD RAM or CG RAM to DR by an internal operation. Data transfer is then completed by reading the DR.

After performing a read from the DR, data in the DD RAM or CG RAM at the next address is sent to the DR for the next read cycle. The register select (RS) signal determines which of these two registers is selected.

Table 4.1 Selection of Registers

RS	R/W	Function
0	0	Instruction Write operation (MPU writes instruction code to IR)
	1	Read Busy flag (DB7) and Address Counter (DB0 ~ DB6)
1	0	Data Write operation (MPU writes data to DR)
	1	Data Read operation (MPU reads data from DR)

4.2 Busy Flag (BF)

When the busy flag is high or "1" the module is performing an internal operation and the next instruction will not be accepted. The busy flag outputs to DB7 when RS = 0 and a read operation is performed. The next instruction must not be written until ensuring that the busy flag is low or "0".

4.3 Address Counter (AC)

The address counter (AC) assigns addresses to the DD RAM and the CG RAM.

When the address of an instruction is written into the IR, the address information is sent from the IR to the AC. The selection of either DD RAM or CG RAM is also determined concurrently by the same instruction. After writing into or reading from the DD RAM or CG RAM the address counter (AC) is automatically increased by 1 or decreased by 1 (determined by the I/D bit in the "Entry Mode Set" command). AC contents are output to DB0 ~ DB6 when RS = 0 and a read operation is performed.

4.4 Display Data RAM (DD RAM)

The Display Data RAM (DD RAM) stores the display data represented in 8-bit character codes. Its capacity is 80 x 8 bits or 80 characters. The Display Data RAM that is not used for the display can be used as a general data RAM.

The DD RAM address (ADD) is set in the Address Counter (AC) and is represented in hexadecimal. The address counter can be written by using the "Set DD RAM Address" instruction and can be read by using the "Read Busy Flag and Address" instruction. In each case, data bits DB0-DB6 represent the DD RAM address. In the read operation, bit DB7 represents the "Busy Flag".

MSB								LSB
BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	

Relations between DD RAM addresses and positions on the liquid crystal display are shown below.

		1	2	3	---	14	15	16	← Display Position
DD RAM Addr.	Line 1	00H	01H	02H	---	0DH	0EH	0FH	
	Line 2	40H	41H	42H	---	4DH	4EH	4FH	

When display shift operation is performed, the DD RAM address moves as follows:

For left shift:

		1	2	3	---	14	15	16	← Display Position
DD RAM Addr.	Line 1	01H	02H	03H	---	0EH	0FH	10H	
	Line 2	41H	42H	43H	---	4EH	4FH	50H	

For right shift:

		1	2	3	---	14	15	16	← Display Position
DD RAM Addr.	Line 1	27H	00H	01H	---	0CH	0DH	0EH	
	Line 2	67H	40H	41H	---	4CH	4DH	4EH	

4.5 Character Generator ROM (CG ROM)

The Character Generator ROM (CG ROM) generates 5 x 7 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate up to 192 types of 5x7 dot character patterns and 32 types of 5x10 dot character patterns. Table 4.3 shows the relation between character codes and character patterns of the standard character font.

4.6 Character Generator RAM (CG RAM)

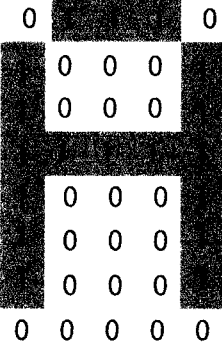
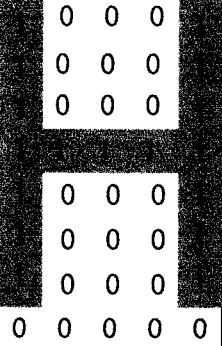
The CG RAM is a 64 x 8 bit RAM in which the user can program custom character patterns. With 5 x 7 dots, 8 types of character patterns can be written and with 5 x 10 dots 4 types of character patterns can be written. To write previously programmed characters from the CG RAM to the DD RAM, character codes 00H through 07H are used. (See character font Table 4.3). Unused CG RAM locations can be used for general purpose RAM.

The relationship between CG RAM address and data and the displayed character is shown in Tables 4.2

To program a 5 x 7 character pattern into the CG RAM location (for example, character code 01H), the following steps should be taken.

- A. Use the "Set CG RAM address" command to position the CG RAM pointer to the 1st row of character code 01H (CG RAM address=48H).
- B. Use the "Write Data to CG or DD RAM" Command to write the top row of the custom character (Only lower 5-bit of character pattern data is valid).
- C. The CG RAM address is automatically increased if the I/D bit is set in the "Entry Mode Set" command. When this is the case, return to step B until all rows of the character are written.
- D. After writing all 7 rows of data, use the "Set DD RAM address" command to return the address counter to a DD RAM location.
- E. To display the custom character written above, use the "Write Data to CG or DD RAM" command with the data being 01H to display the character in the DD RAM address.

Table 4.2 Relation between CG RAM address, character codes (DD RAM) and character patterns (5x7 dots)

Character Code (DDRAM data)	CGRAM Address	CGRAM Data	Pattern number
D7 D6 D5 D4 D3 D2 D1 D0	A5 A4 A3 A2 A1 A0	P7 P6 P5 P4 P3 P2 P1 P0	
0 0 0 0 × 0 0 0	0 0 0 0 0 0	× × × 0  0	pattern 1 cursor position
· · · · ·	· · · · ·	· · · · ·	· · · · ·
0 0 0 0 × 1 1 1	0 0 0 0 0 0	× × ×  0	pattern 8 cursor position
· · · · ·	· · · · ·	· · · · ·	· · · · ·

Notes:

1. Character code bits 0~2 correspond to CG RAM address bit 3~5 (3 bits: 8 types).
2. CG RAM address bits 0~2 designate the line position within a character pattern. The 8th line is the cursor position and display is determined by the logical OR of the 8th line and the cursor. Maintain the 8th line data, corresponding to the cursor display position, in the "0" state for cursor display. When the 8th line data is "1", bit 1 lights up regardless of cursor existence.
3. Character pattern row positions correspond to CG RAM data bits 0~4 as shown in the above (bit 4 being at the left end). Since CG RAM data bits 5~7 are not used for display, they can be used for the general data RAM as memory elements still exit.
4. As shown in Table 4.2, CG RAM character patterns are selected when character code bits 4~7 are all "0". However as character code bit 3 is an ineffective bit, the "A" in the character pattern example is selected by character code "00H" or "08H".
5. "1" for CG RAM data corresponds to selected pixels and "0" for non-selected.

Table 4.3 CGROM Character Code Table

Upper 4bit Lower 4bit	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	00 RAM (1)			0	1	P	Y	P				一	夕	三	夕	P
0001	(2)	!	1	A	Q	△	4			。	ア	チ	△	△	夕	夕
0010	(3)	"	2	B	R	△	7			「	イ	ウ	夕	夕	夕	夕
0011	(4)	#	3	C	S	△	△			」	ウ	チ	△	△	夕	夕
0100	(5)	*	4	D	T	△	△			√	工	ト	△	△	夕	夕
0101	(6)	△	5	E	U	△	△			。	才	大	工	夕	夕	夕
0110	(7)	△	6	F	V	△	△			ヲ	力	二	△	△	夕	夕
0111	(8)	△	7	G	W	△	△			ア	キ	△	△	夕	夕	夕
1000	(1)		△	H	X	△	△			△	夕	△	△	夕	夕	夕
1001	(2)		△	I	Y	△	△			△	夕	△	△	夕	夕	夕
1010	(3)		*	J	Z	△	△			△	△	△	△	夕	夕	夕
1011	(4)		+	△	△	△	△			△	△	△	△	夕	夕	夕
1100	(5)		△	△	△	△	△			△	△	△	△	夕	夕	夕
1101	(6)		△	△	△	△	△			△	△	△	△	夕	夕	夕
1110	(7)		△	△	△	△	△			△	△	△	△	夕	夕	夕
1111	(8)		△	△	△	△	△			△	△	△	△	夕	夕	夕

5. MPU INTERFACE

5.1 General

(1). The LCD controller can be operated in either 4 or 8 bits mode. Instructions/Data are written to the display using the signal timing characteristics found in section 3.2.

When operating in 4-bit mode, data is transferred in two 4-bit operations using data bits DB4~DB7. DB0~DB3 are not used. When using 4-bit mode, data is transferred twice before the instruction cycle is complete. The higher order 4 bits (contents of DB4~DB7 when interface data is 8 bits long) is transferred first, then the lower order 4 bits (contents of DB0~DB3 when interface data is 8 bits long) is transferred. Check the busy flag after 4-bit data has been transferred twice (one instruction). A 4-bit two operation will then transfer the busy flag and address counter data.

(2). When operating in 8-bit mode, data is transferred using the full 8-bit bus DB0~DB7.

5.2 Initialization

5.2.1 Initialization by the Internal Reset Circuit

The display can be initialized using the internal reset circuit when the power is turned on. The following instructions are executed in initialization. The busy flag (BF) is kept in busy state until initialization ends. The busy flag will go active 10ms after Vcc rises to 4.5V.

(1). Display Clear

(2). Function set:

DL = 1 : 8 bit interface operation

N = 0 : 1 - line display

F = 0 : 5 x 7 dot character font

(3). Display ON/OFF Control:

D = 0 : Display OFF

C = 0 : Cursor OFF

B = 0 : Blink OFF

(4). Entry Mode Set

I/D = 1 : +1 (Increment Mode)

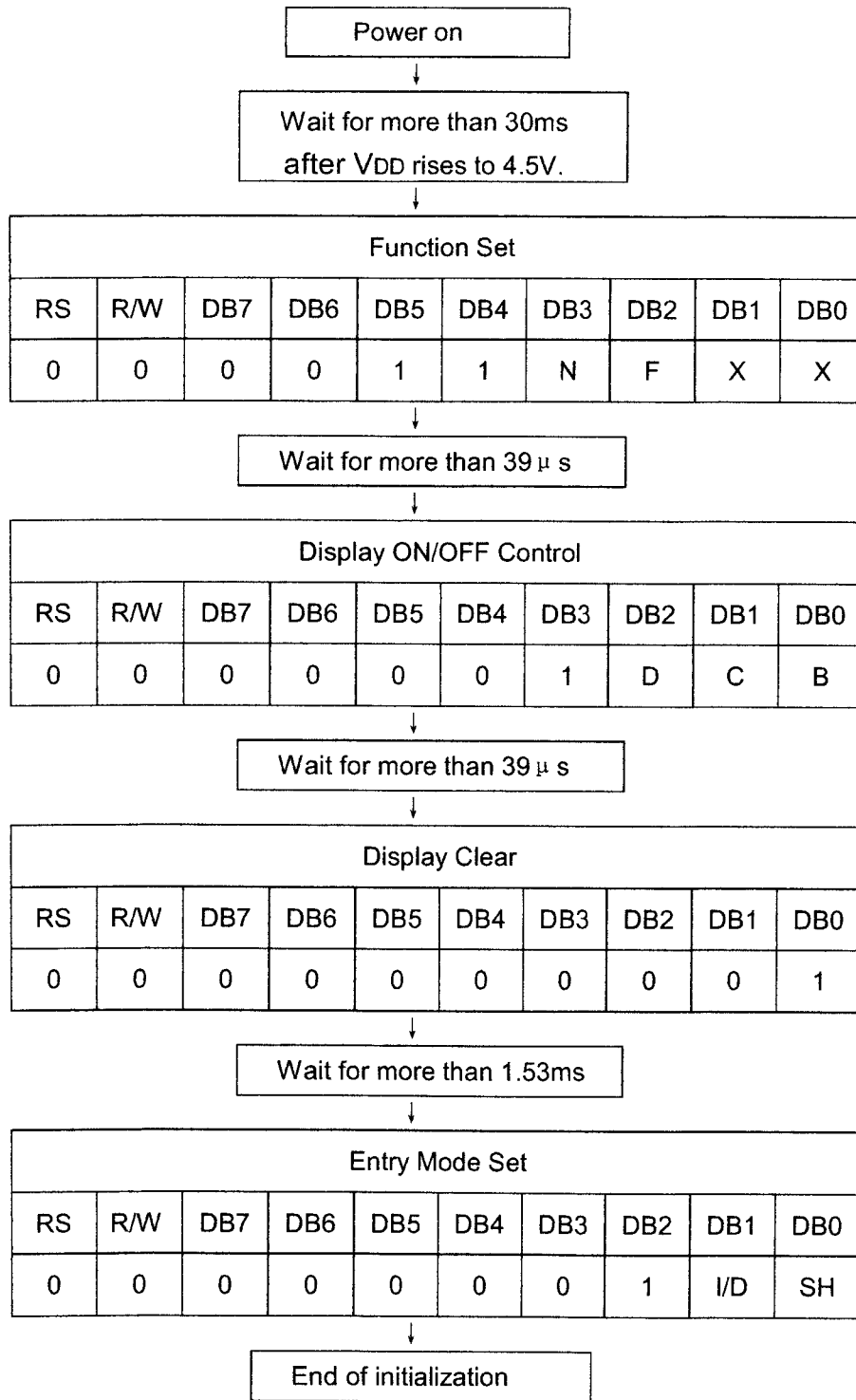
S = 0 : No Display Shift operation

If the internal power supply reset timing cannot be met ($0.1\text{ms} < \text{trcc} < 10\text{ms}$), the internal reset circuit will not operate normally and initialization will not be performed. In this case, the display must be initialized by software.

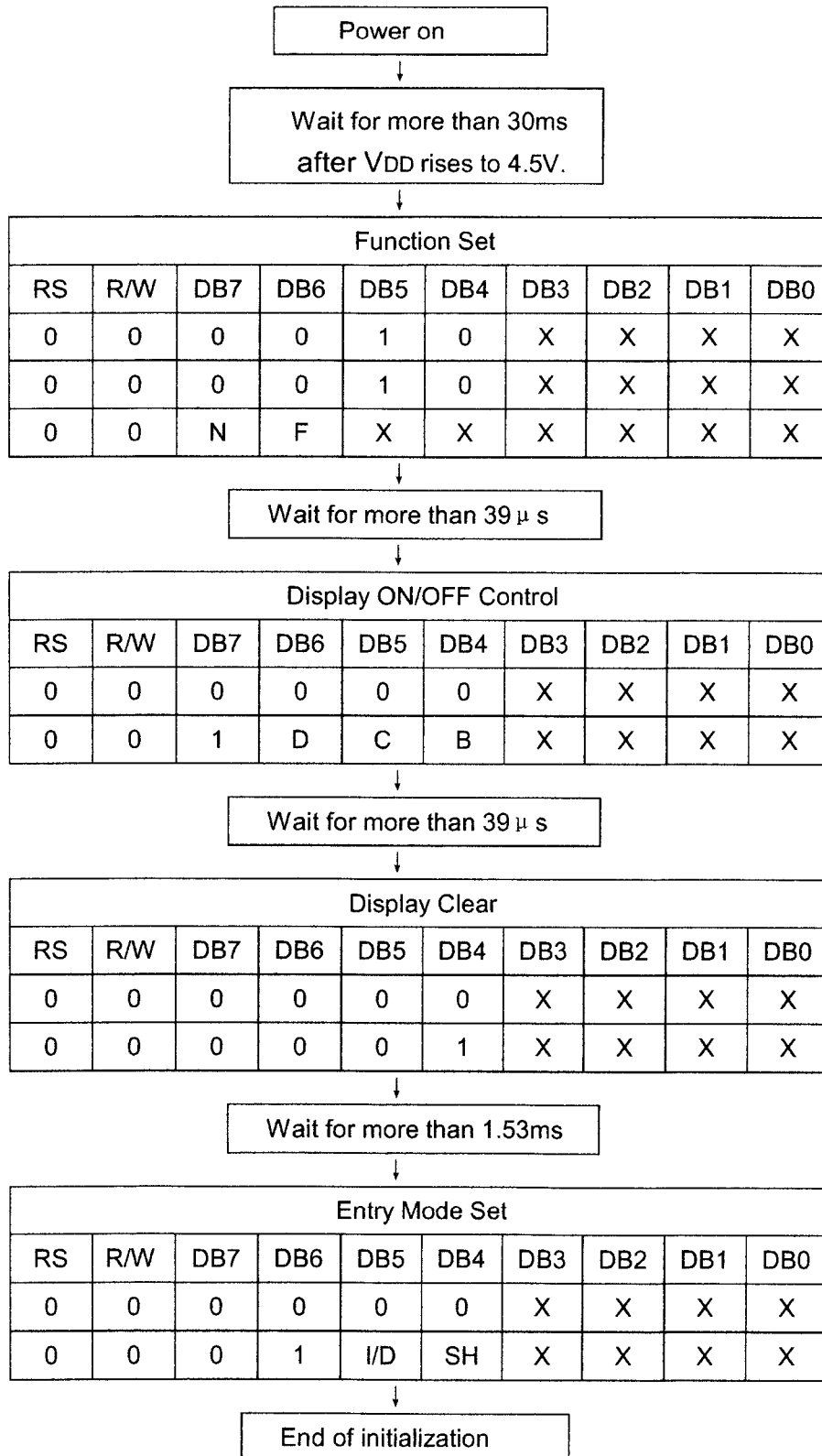
5.2.2 Software Initialization

Although software initialization is not mandatory, it is recommended that this procedure always be performed. When the internal power supply reset timing cannot be met, then the display must be initialized using one of the following procedures.

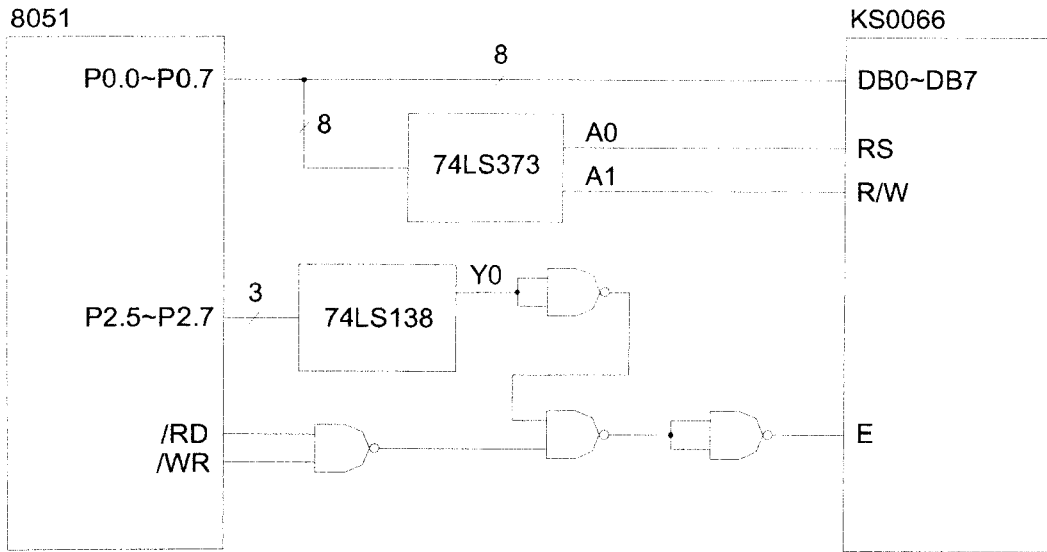
1). 8-Bit Initialization:



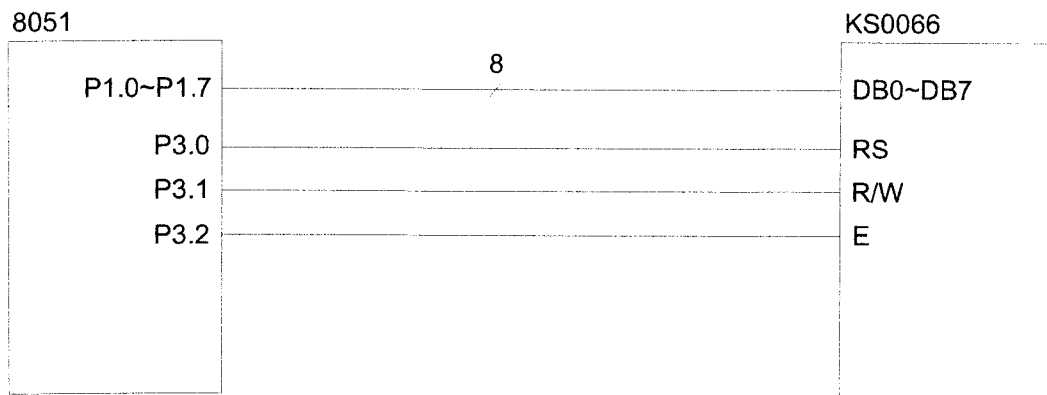
2). 4-Bit Initialization:



5.3 Connection with 8051 Family MPU



a. Application Circuit 1



b. Application Circuit 2

3. DISPLAY CONTROL INSTRUCTION

Table 6.1 Instructions

Instruction	Instruction code										Description	Execution time (fosc=270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address to 00H.	1.53ms
Return Home	0	0	0	0	0	0	0	0	0	1	Sets DDRAM address to 00H in AC and returns shifted display to its original position. The contents of DDRAM remain unchanged.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Sets cursor move direction and enable the shift of entire display. These operations are performed during data write and read.	39 μ s
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set ON/OFF of entire display (D), cursor ON/OFF(C), and blinking of cursor position character(B).	39 μ s
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Moves cursor and shifts display without changing DDRAM contents.	39 μ s
Function Set	0	0	0	0	1	DL	N	F	-	-	Sets interface data length (DL: 8-bit/4-bit), numbers of display line (N: 2-line/1-line), and display font type (F: 5x11dots/5x8dots)	39 μ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address Counter.	39 μ s
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s
Write data to CG or DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43us
Read data from CG or DD RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43us

"-" : don't care

- Note: 1. Make sure to check the busy flag before sending the instruction to the display. If the busy flag is not checked, the time between first and next instruction must be longer than the instruction execution time list in the Table 6.1.
2. After execution of CG RAM/DD RAM data write or read instruction, the RAM address counter is increased or decreased by 1. The RAM address counter is updated after the busy flag turns off.

3.1 Clear Display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing the space code "20H" to all DD RAM addresses, and set DD RAM address to "00H" into address counter. Returns cursor to the original position, namely, brings the cursor to the upper left end of the display. The execution of clear display instruction sets entry mode to increment mode (I/D = 1).

6.2 Return Home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

Sets the DD RAM address "00H" in address counter. Return display to its original position if it was shifted. DD RAM contents do not change. The cursor or the blink moves to the upper left end of the display. Contents of DD RAM remain unchanged.

6.3 Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Sets the move direction of cursor and display.

I/D: Increases (I/D = 1) or decreases (I/D = 0) the DD RAM address by 1 when a character code is written into or read from the DD RAM.

The cursor or blink moves to the right when increased by 1 and to the left when decreased by 1. The same applies to writing and reading the CG RAM.

S: Shifts the entire display when S = 1; shifts to the left when I/D = 1 and to the right when I/D = 0. Thus it looks as if the cursor stands still and only the display seems to move. The display does not shift when reading from DD RAM or writing/reading into/from CG RAM.

When S = 0, the display does not shift.

6.4 Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Controls the display ON/OFF status, Cursor ON/OFF and Cursor Blink function.

D: The entire display is ON when D = 1 and OFF when D = 0. The display data remains in the DD RAM when display is OFF, it can be displayed immediately by setting D = 1.

C: The cursor displays when C = 1 and does not display when C = 0. The cursor is displayed on the 8th line when 5x7 dots character font has been selected.

B: The character indicated by the cursor blinks when B = 1. The blink is displayed by switching between all "High" data and display characters at 0.4 sec intervals.

The cursor and the blink can be set to display simultaneously.

When B = 0, the blink is off.

6.5 Cursor or Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

Shifts the cursor position or display to the right or left without writing or reading display data. This function is used to correct or search for the display.

Note that the display shift is performed simultaneously in all lines.

The contents of address counter do not change when display shift is performed.

Table 6.2 Shift Patterns According to S/C and R/L Bits

S/C	R/L	Operation
0	0	Shifts cursor position to the left (AC is decreased by 1)
0	1	Shifts cursor position to the right (AC is increased by 1)
1	0	Shifts the entire display to the left, cursor follows the display shift.
1	1	Shifts the entire display to the right, cursor follows the display shift.

6.6 Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	-	-

Sets the interface data length, the number of lines, and character font.

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

N: Sets the number of lines

N = 0 : 1 line display (1/8 duty)

N = 1 : 2 line display (1/16 duty)

F: Sets character font.

F = 0 : 5 x 7 dots

F = 1 : 5 x 10 dots

Note: Perform the function at the head of the program before executing all instructions (except Busy flag/address read).

6.7 Set CG RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Sets the CG RAM address to the address counter. Data is then written/read to/from the CG RAM.

6.8 Set DD RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Sets the DD RAM address to the address counter. Data is then written/read to/from the DD RAM.

When in 1-line display mode (N = 0), DD RAM address is from "00H" to "4FH".

When in 2-line display mode (N = 1), DD RAM address corresponding to 1st line and 3rd line of the display is from "00H" to "27H"; the address corresponding to 2nd and 4th line of the display is from "40H" to "67H".

6.9 Read Busy Flag & Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Reads the busy flag (BF) and value of the address counter (AC). BF = 1 indicates that internal operation is in progress and the next instruction will not be accepted until BF is set to "0". The BF status should be checked before each write operation. At the same time the value of the address counter is read out. The address counter is used by both CG and DD RAM and its value is determined by the previous instruction.

6.10 Write Data to CG or DD RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Writes binary 8-bit data to the CG or DD RAM.

The previous address set instruction (CG RAM address set or DD RAM address set) determines whether the CG or DD RAM is to be written. After a write the address is automatically increased or decreased by 1, according to the entry mode. The entry mode also determines display shift.

6.11 Read Data from CG or DD RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

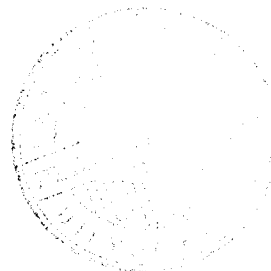
Reads binary 8-bit data from the CG RAM or DD RAM.

The previous address set instruction (CG RAM address set or DD RAM address set) determines whether the CG or DD RAM is to be read. Before entering the read instruction, you must execute either the CG RAM or DD RAM address set instruction. If you don't, the first read data will be invalidated. If RAM data is read several times without RAM address instruction set before read operation, the correct RAM data can be obtained from the second read. The "address set" instruction need not be executed just before the "read" instruction when shifting the cursor by cursor shift instruction (when reading DD RAM). The cursor shift instruction operation is the same as that of the DD RAM address set instruction.

After a read the address is automatically increased or decreased by 1, according to the entry mode; however, display shift is not executed no matter what the entry mode is.

Note: The address counter (AC) is automatically increased or decreased by 1 after a "write" instruction to either CG RAM or DD RAM. RAM data selected by the AC cannot then be read out even if "read" instructions are executed.

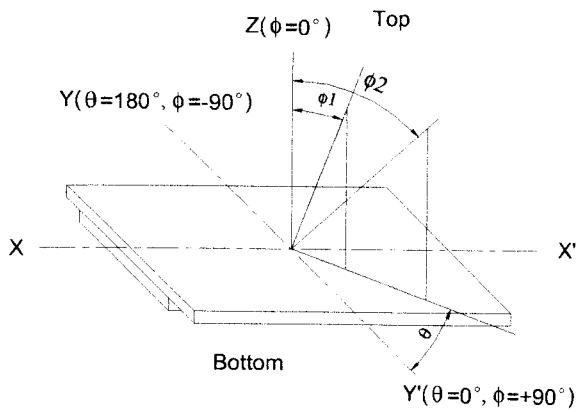
The conditions for correct data read out are: (a) Execute either the address set instruction or cursor shift instruction (only with DD RAM) or (b) The execution of the "read data" instruction from the second time when the read instruction is performed multiple times in serial.



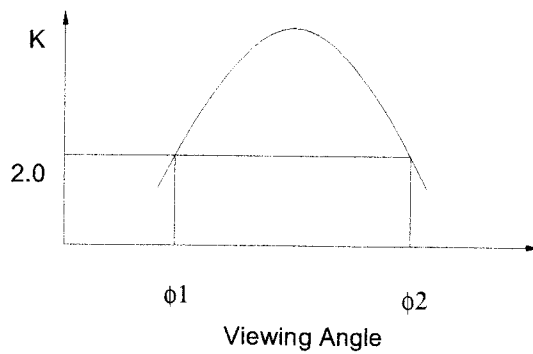
7. ELECTRO—OPTICAL CHARACTERISTICS ($T_a=25\text{ }^\circ\text{C}$)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Note
View Angle	$\Phi 2-\Phi 1$	$K \geq 2, \theta = 0^\circ$	--	80	--	Deg	Note1, Note2
Contrast	K	$\Phi = 0^\circ, \theta = 0^\circ$	3	--	--	--	Note3
Response Time	tr (rise)	$\Phi = 0^\circ, \theta = 0^\circ$	--	250	--	ms	Note3
	tf (fall)	$\Phi = 0^\circ, \theta = 0^\circ$	--	250	--	ms	

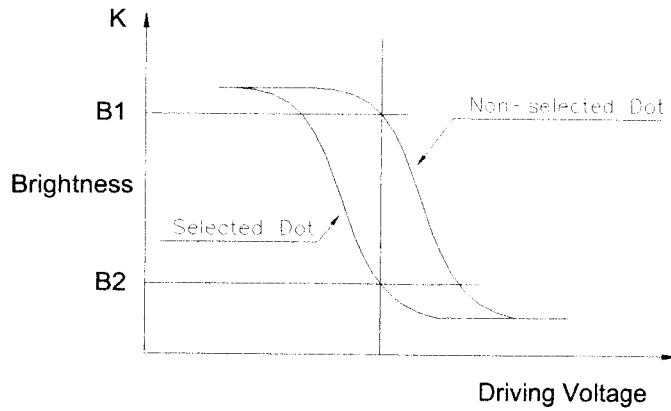
Note1: Definition of Viewing Angle θ, Φ



Note2: Definition of viewing Angle Range: $\Phi 1, \Phi 2$

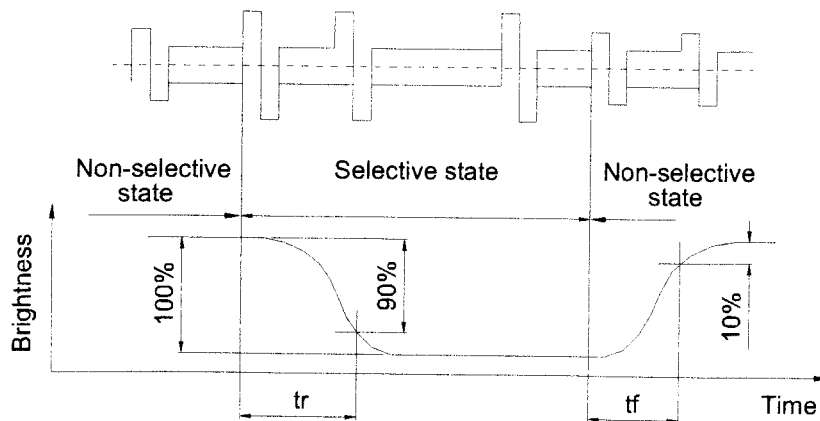


Note3: Definition of Contrast



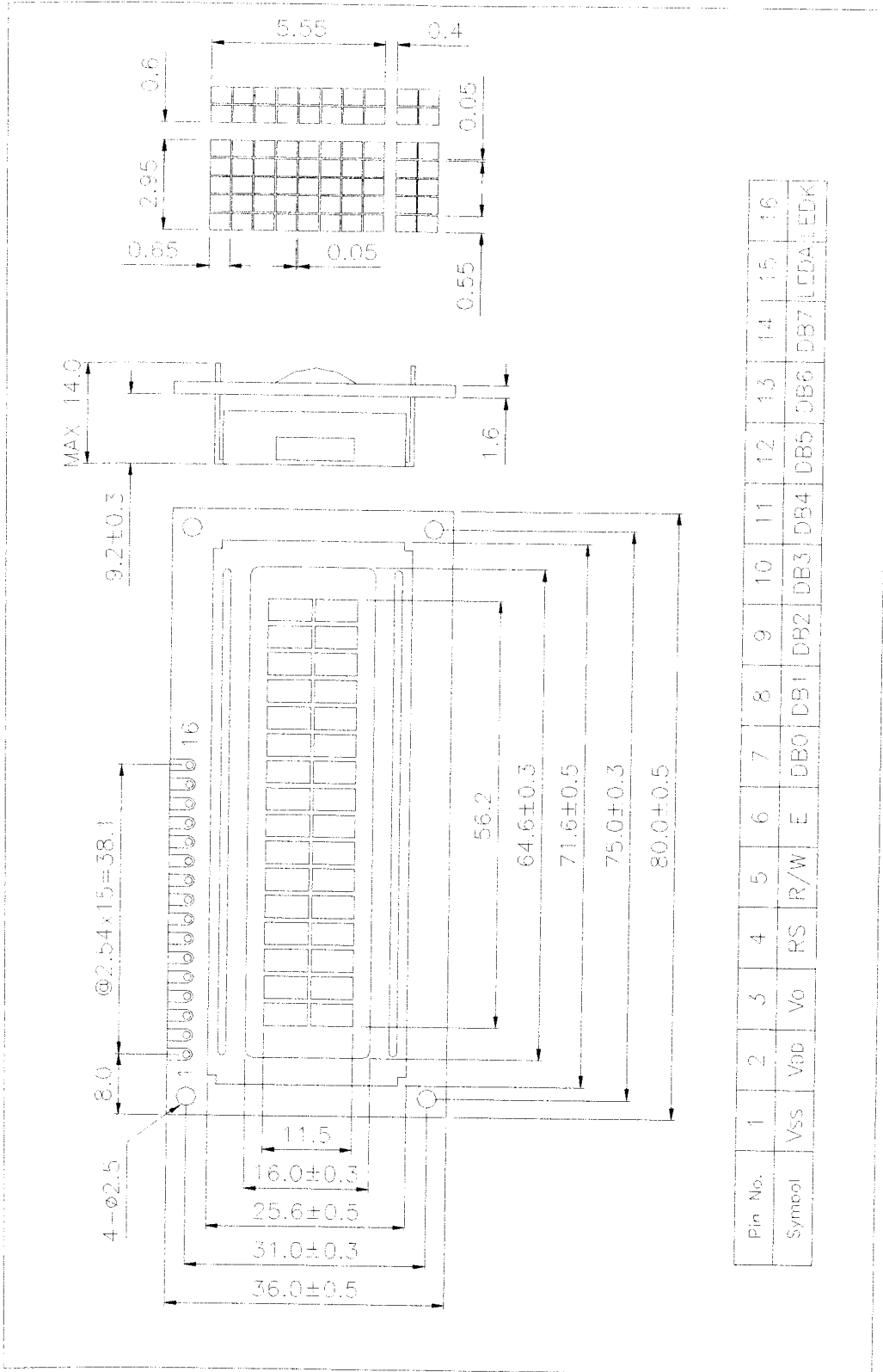
$$\text{Contrast} = \frac{\text{Brightness of non-selected dot (B1)}}{\text{Brightness of selected dot (B2)}}$$

Note4: Definition of Response Time

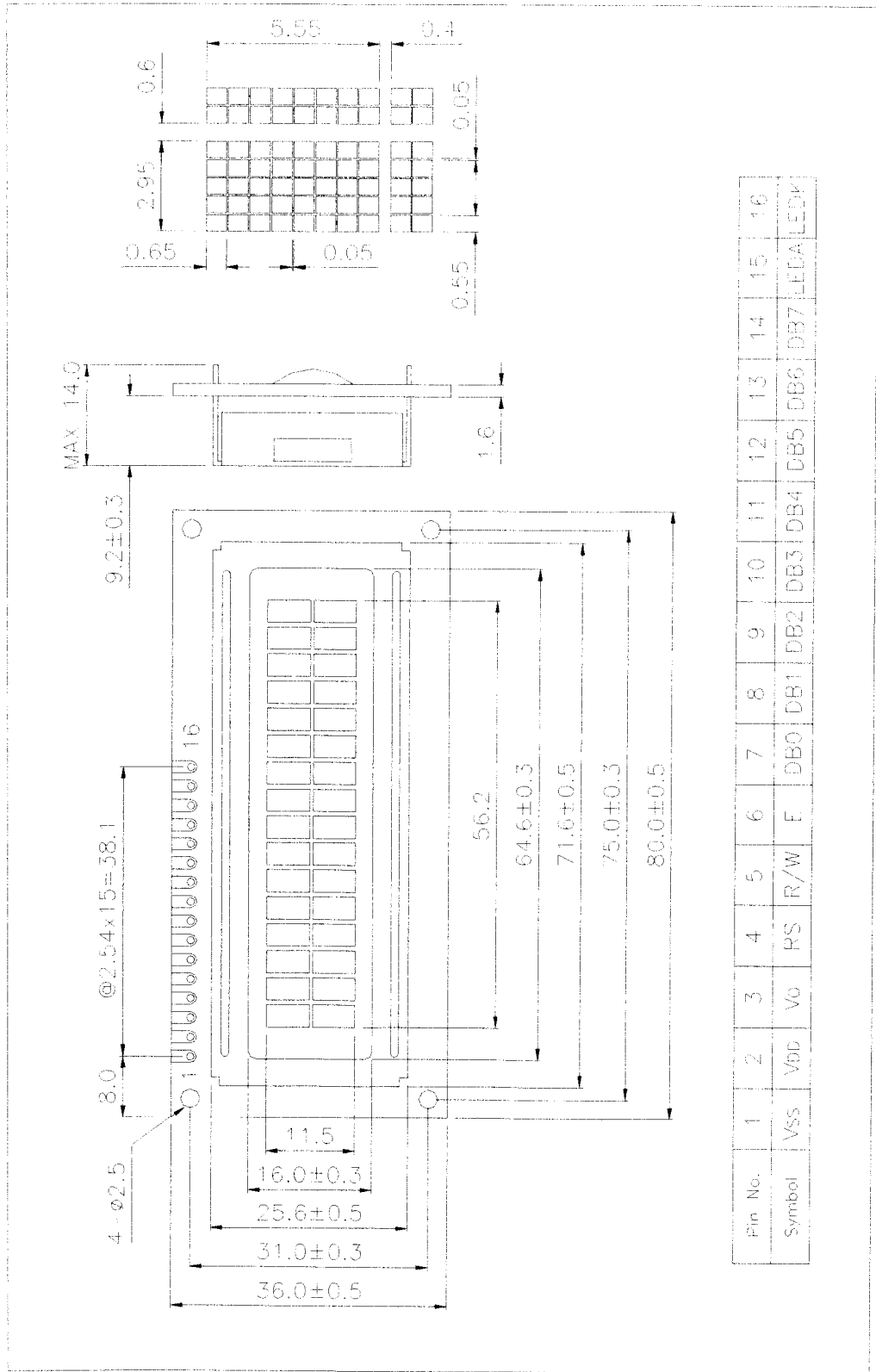


3. DIMENSIONAL OUTLINE

9



3. DIMENSIONAL OUTLINE



9. PRECAUTIONS FOR USE OF LCD MODULE

9.1 Handling Precautions

- 1) The display panel is made of glass. Do not subject it to a mechanical shock by Dropping it from a high place, etc.
- 2) If the display panel is damaged and the liquid crystal substance inside it leaks out, Be sure not to get any in your mouth. If the substance comes into contact with your Skin or clothes, promptly wash it off using soap and water.
- 3) Do not apply excessive force on the surface of display or the adjoining areas of LCD module since this may cause the color tone to vary.
- 4) The polarizer covering the display surface of the LCD module is soft and easily Scratched. Handle this polarizer carefully.
- 5) If the display surface of LCD module becomes contaminated, blow on the Surface and gently wipe it with a soft dry cloth. If it is heavily contaminated, moisten Cloth with one of the following solvents.

- Isopropyl alcohol
- Ethyl alcohol

Solvents other than those mentioned above may damage the polarizer.

Especially, do not use following.

- Water
- Ketone
- Aromatic Solvents

9. 6) When mounting the LCD module make sure that it is free of twisting, warping, and Distortion. Distortion has great influence upon display quality. Also keep the Stiffness enough regarding the outer case.
- 7) Be sure to avoid any solvent such as flux for soldering never stick to Heat-Seal. Such solvent on Heat-Seal may cause connection problem of heat-Seal and TAB.
- 8) Do not forcibly pull or bend the TAB I/O terminals.
- 9) Do not attempt to disassemble or process the LCD module.
- 10) NC terminal should be open. Do not connect anything.
- 11) If the logic circuit power is off, do not apply the input signals.
- 12) To prevent destruction of the elements by static electricity, be careful to maintain an optimum work environment.
 - Be sure to ground the body when handling the LCD module.
 - Tools required for assembly, such as soldering irons, must be properly grounded.
 - To reduce the amount of static electricity generated, do not conduct assembly and other work under dry conditions.
 - The LCD module is coated with a film to protect the display surface. Exercise care when peeling off this protective film since static electricity may be generated.

9.2 Storage Precautions

- 1) When storing the LCD module, avoid exposure to direct sunlight or to the light of Fluorescent lamps and high temperature/high humidity. Whenever possible, the