LAPORAN LATIHAN ILMIAH TAHAP AKHIR I

WXES3181

KAWALAN PAPAN KEKUNCI PS/2

OLEH:

HASINA BTE HUSSIN

WEK000193

Laporan ini merupakan sebahagian daripada keperluan Ijazah Sarjana Muda Sains Komputer dan Teknologi Maklumat Universiti Malaya

SEISI 2003/2004

ABSTRAK

Hubungan antara perkakasan dalam sesebuah komputer sebenarnya adalah satu jenis komunikasi diantara kawalan (controller). Apabila sesebuah kawalan ingin menghantar atau menerima arahan daripada kawalan yang lain, satu bentuk isyarat dihantar untuk mewakili arahan tersebut. Isyarat tersebut adalah bentuk digital bagi satu set arahan bahasa mesin yan dihantar dalam bentuk jujukan bit menggunakan sambungan bersiri khusus.

Satu kenyataan perlu diterima dimana sesebuah komputer hanya akan berfungs dengan betul apabila setiap kawalan yang wujud didalamnya menjalankan fungsi masing-masing dengan betul.

Kawalan papan kekunci adalah salah satu kawalan yang wujud didalam komputer untuk memastikan kelancaran fungsi komputer tersebut. Kawalan ini akan mentafsir dan mengesan sebarang ralat pada data input yang dihantar untuk menghasilkan output yang sepatutnya.

Kajian ini akan menghuraikan serba sedikit tentang bagaimana sesebuah kawalan papan kekunci berfungsi dari mula kekunci ditekan sehinggalah kepada pengeluaan outputnya.

PENGHARGAAN

Pertama sekali saya ingin memanjatkan kesyukuran ke hadrat Ilahi kerana dengan izin dan limpah kurnianya maka dapat saya menyiapkan tesis ini. Selawat dan salam keatas junjungan Nabi Muhammad SAW, keluarga Baginda dan para sahabat r.a.

Terima kasih yang tidak terhingga kepada penasihat saya Encik Yamani Idna Idris yang begitu banyak memberikan nasihat dan tunjuk ajar serta sanggup meluangkan masa dan tenaga sepanjang saya menyiapkan tesis ini. Juga kepada moderator, Profesor Datuk Doktor Mashkuri Haji Yaacob yang banyak memberikan idea-idea dan komen yang membina.

Limpahan kasih sayang dan penghargaan buat keluarga yang banyak memberikan sokongan moral, kebendaan dan doa yang tak putus untuk saya dalam mengharungi hidup sebagai pelajar dan membentuk diri menjadi seorang insan.

Terima kasih juga saya tujukan kepada rakan-rakan seperjuangan dan mereka yang terlibat dalam penghasilan tesis ini sama ada secara langsung atau tidak.

Hanya Allah sahaja yang dapat membalas jasa kalian.

ISI KANDUNGAN

SENARAI JADUAL

SENARAI RAJAH

PENGENALAN

1.0 PENGENALAN

Projek ini mengupas peranan kawalan Papan Kekunci dalam penerimaan dan penghantaran data dari peranti input ke peranti output. Dalam menjelaskan fungsi utama ini, akan dijelaskan bentuk sebenar dalam bahasa kod, pengkodan kepada isyarat digital, pengesanan ralat dan bentuk gelombang melawan masa.

1.1 SKOP

Kajian ilmiah ini akan merangkumi kaedah penghantaran dan penerimaan data dari peranti input ke peranti output dimana papan kekunci bertindak sebagai peranti input manakala LED 7 segmen mewakili peranti output. Walaupun nanti fungsi-fungsi lain kawalan papan kekunci akan didedahkan, kajian ini akan hanya membincangkan mengenai fungsinya sebagai antaramuka antara papan kekunci PS/2 dan LED 7 segmen.

Bab 1 akan merangkumi pengenalan kepada tajuk, masalah-masalah yang perlu diselesaikan, skop permasalahan dan kekangan terhadap pengaplikasian sistem.Bab ini juga meliputi kajian literasi yang akan menjelaskan dengan lebih mendalam tentang PS/2 dan fungsi kawalan PS/2.

Bab 2 pula akan menjelaskan tentang metodologi yang digunakan untuk tajuk ini. Dalam bab ini akan dijelaskan tentang bahasa

pengaturcaraan yang dipilih, kelebihannya, webpack yang sesuai digunakan dan keperluan sistem.

Dalam bab ke 3, analisa sistem akan dilakukan. Disini akan dibincangkan tentang keperluan perkakasan yang digunakan. Dalam tesis ini akan dibincangkan mengenai papan XSA100 dan komponen-komponen yang terdapat didalamnya.

Dalam bab 4, fasa rekabentuk akan dimulakan. Bab ini akan memberikan gambaran sebenar peranti yang terlibat dan susunannya dalam membentuk litar, fungsi setiap pin dan penerangan tentang blok diagram yang mewakili peranti kawalan.

Bab yang kelima pula menyentuh tentang perlaksanaan system. Semua peringkat rekabentuk modul-modul yang telah dibangunkan pada fasa yang sebelumnya akan dinabgunkan dan ditukarkan ke dalam bentuk pengaturcaraan VHDL.

Bab yang ke enam menerangkan bagaimana pengujian dilakukan keatas kod pengaturcaraan. Bit-bit ujian akan dimasukkan kedalam setiap modul untuk memastikan setiap komponen tersebut berfungsi seperti yang dikehendaki.

Bab 7 adalah bab yang terakhir yang membincangkan tentang keputusan yang diperolehi daripada pengujian yang dilakukan, kelebihan

dan kekurangan system dan kesimpulan bagi projek yang telah dibangunkan. Perubahan-perubahan dari segi rekabentuk system dan penggunaan perisian yang digunakan utnuk membangunkan projek ini juga akan dibincangkan.

1.2 OBJEKTIF

Objektif utama kajian ini adalah untuk memahami peranan kawalan papan kekunci dalam memproses input kepada output yang sepatutnya. Dengan ini fungsi setiap bahagian kawalan papan kekunci yang terlibat dalam proses ini akan lebih difahami. Oleh kerana bahasa pengaturcaraan VHDL digunakan, melalui kajian ini juga dapatlah diketahui kelebihan bahasa tersebut berbanding kaedah pengkodan yang lain.

1.3 KEKANGAN

Kekangan yang dihadapi akan dibincangkan dari segi:

- Rekabentuk yang terlibat adalah model kasar bagi mengkaji fungsi kawalan papan kekunci ini sahaja maka pendekatannya mungkin berbeza dengan kaedah yang digunakan dalam dunia pengkomputeran sebenar.

- Penghantaran isyarat adalah dalam bentuk teori semata-mata kerana bentuk sebenar gelombang yang dihantar tidak dapat dilihat dengan mata kasar.
- Oleh kerana gelombang input dan output adalah bersandaran dengan masa, sebarang lengahan yang tidak disengajakan mungkin menyebabkan berlakunya ralat pada output. Maka output yang diharapkan tidak dapat dihasilkan pada masa yang ditetapkan semasa teori.
- WebPack yang digunakan dalam kajian ini adalah salah satu versi yang boleh digunakan. Mungkin terdapat versi lain yang boleh digunakan pada masa akan datang bergantung kepada kesesuaian atau keperluan perisian tersebut kepada teknologi terkini.

1.4 PERANCANGAN AKTIVITI

Minggu	Aktiviti
1	Memilih tajuk tesis
2	Memahami tajuk tesis, kenalpasti objektif, skop dan pengenalan projek
3	Mencari bahan rujukan melalui internet, bahan bacaan
4	Mencari bahan rujukan melalui internet, bahan bacaan
5	Mula menulis untuk bab pertama
6	Mencari bahan untuk bab kedua
7	Mencari bahan untuk bab kedua
8	Memasang perisian yang sepatutnya untuk melarikan program aturcara Mula menulis bab kedua
9	Menulis bab kedua: mengkaji dengan lebih mendalam tentnag kaedah yang digunakan
10	Menulis bab kedua: mencari perbandingan antara perkakasan yang digunakan dengan perkakasan yang lain
11	Mencari bahan untuk bab ketiga
12	Mencari bahan untuk bab ketiga Mula menulis bab ketiga
13	Mencari bahan bab keempat VIVA
14	Menulis bab keempat Membuat semakan terakhir sebelum penghantaran

Jadual 1.1: Jadual perancangan WXES3181

Tugas/ Minggu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pilih tajuk														
Memahami tajuk														
Cari bahan														
Bab pertama														
Cari bahan														
Perisian														
Bab kedua														
Cari bahan														
Bab ketiga														
Cari bahan														
Bab keempat														
Semakan dan penghantaran														

Jadual 1.2: Garis masa untuk WXES3181

Minggu	Aktiviti
1	Pemeriksaan semula perisian yang sesuai untuk kaedah yang digunakan
2	Proses rekabentuk pengkodan
3	Proses rekabentuk pengkodan
4	Merangka bab pertama
5	Proses perlaksanaan Proses pengujian
6	Merangka dan menulis bab kedua Proses pengujian
7	Proses pengujian
8	Merangka bab ketiga Proses pengujian
9	Mencari bahan untuk perbincangan projek Proses pengujian
10	Mencari bahan untuk perbincangan projek Proses pengujian
11	Penyediaan dokumentasi lengkap projek Proses pengujian
12	Penyediaan dokumentasi lengkap projek Proses pengujian
13	Penyediaan dokumentasi lengkap projek
14	Pengantaran WXES3182

Jadual 1.3: Jadual perancangan WXES3182

Tugas/ minggu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Perisian														
Pengekodan														
Bab pertama														
Perlaksanaan														
Bab kedua														
Pengujian														
Bab ketiga														
Perbincangan														
Dokumentasi														
Penghantaran														

Jadual 1.4: Garis masa untuk WXES3182

1.5 SEJARAH PS/2

Personal system/2 atau PS/2 adalah generasi kedua IBM bagi komputer peribadi yang diperkenalkan kepada umum pada 1987. PS/2 IBM direkabentuk untuk mengekalkan kesesuaian perisian namun agak berbeza dari segi perkakasan. Ia memperkenalkan bus MikroChannel untuk komunikasi dalam sistem yang lebih pantas namun ia gagal mengekalkan bas AT terbuka (kini dikenali sebagai bas ISA), yang mana bermakna tiada satu pun daripada jutaan kad 'add-in' yang wujud dapat berfungsi. Dalam erti kata lain, IBM PS/2 yang baru ini tidak mempunyai kesesuaian IBM.

Sebagai tambahan, IBM menjadikan PS/2 dari segi teknikal dan kesahan supaya sukar diklon. Sebagai ganti, IBM menawarkan penjualan lesen Microchannel kepada sesiapa yang berkemampuan.

PS/2 model 25 dan 30 adalah berasaskan ISA atau dalam kata lain ia adalah satu bentuk sistem seperti IBM dalam bentuk faktor yang berbeza.model yang tertinggi dilengkapi dengan bas Mikrochannel. Semua model menggunakan hard drive ESDI atau SCSI, dengan 1 paras masukan, antaramuka ESDI yang mana tidak bersesuaian dengan piawai industri dan mengikat pengguna kepada IBM untuk alat ganti. PS/2 model 50 dan 60 menggunakan pemproses Intel 80286 manakala PS/2 model 70 dan 80 menggunakan pemproses Intel 80386.

Walaupun 'floppy disk' 3.5" 1.44 Megabait adalah biasa dalam industri dalam tahun 1987, PS/2 adalah produk IBM yang pertama yang menggunakannya sebagai piawai. Sistem PS/2 juga memperkenalkan spesifikasi baru bagi antaramuka tetikus. Antaramuka papan kekunci PS/2 secara elektronik serupa dengan antaramuka Baby AT namun dipakejkan kepada bentuk yang lebih kecil yang mana dicipta serupa seperti antaramuka tetikus yang baru.

Perubahan yang paling besar dilakukan oleh model PS/2 adalah piawai display. Superset VGA dan piawai grafik 8514 yang lebih maju kekal menjadi antaramuka grafik digunakan hari ini.

1.6 FUNGSI KAWALAN PAPAN KEKUNCI

Sebelum ini, komputer menggunakan kawalan papan kekunci 8042 sebagai antaramuka kepada papan kekunci. Kawalan pada papan ibu sebenarnya adalah berantaramuka dengan kawalan seperti yang terdapat dal;am papan kekunci dengan menggunakan sambungan sesiri khusus. Kebanyakan komputer moden menyepadankan cara 8042 berfungsi kepada papan ibu. Sesetengah cip super I/O lebih bersepadu dimana ia memasukkan fungsi kawalan papan kekunci kepadanya. Ini mengelakkan keperluan kepada cip kawalan papan kekunci yang berasingan.

Kawalan papan kekunci memberikan fungsi-fungsi berikut kepada komputer:

- Kawalan papan kekunci dan Interpretasi

  Kawalan papan kekunci adalah litar yang megawal papan kekunci. Apabila kod imbas (scan code) diterima, ia akan memaklumkan kawalan menggunakan sampukan peranti yang ditujukan kepada papan kekunci (IRQ1). Kawalan kemudiannya mentafsirkan kekunci yang ditekan lalu melaksanakannya dengan betul. Kawalan juga menangani kadar typematic (typematic rate) papan kekunci, iaitu kadar ulangan apabila kekunci ditekan untuk satu tempoh masa.

- Menyokong tetikus PS/2

  Kawalan papan kekunci yang mengawal input daripada tetikus jenis PS/2 bersepadu kini banyak dijumpai pada kebanyakan mesin terbaru. IBM yang mula-mula mengubahsuai rekabentuk lama apabila piawai ini dicipta bersama line PS/2.

- Capaian kepada Paras Ingatan Tinggi

  Ingatan 64K dan keatas dikenali dengan Paras Ingatan Tinggi. Capaian kepada bahagian ingatan ini boleh dilakukan oleh DOS walaupun sebenarnya bahagian

640K ini tidak wujud pada ingatan konvensional iaitu menjadi kelemahan dalam rekabentuk AT IBM yang asal. Apabila had 640K telah dicapai, kebolehgunaan kawasan ini untuk menjimatkan ingatan konvensional merupakan satu kelebihan. Capaian kepada kawasan ini melalui line 20 (line alamat ke 21) yang dikawal oleh kawalan papan kekunci. Kebanyakan cip terbaru mempunyai ciri ini bagi meningkatkan prestasinya.

METODOLOGI

2.0 PENGENALAN KEPADA VHDL

2.0.1 VHDL

VHDL adalah satu perkakasan yang sangat berkuasa. Ia dicipta khas untuk merekabentuk litar sama ada pada peringkat get atau behavioral. Walaupun adalah tidak mustahil untuk menggunakan VHDL untuk memprogramkan perisian serbaguna yang lain, ini bukanlah tujuan sebenar bahasa ini dicipta. Struktur bahasa ini sendiri menyokong rekabentuk perkakasan. Setiap litar bersepadu boleh dijelaskan dengan asas yang sangat realistic, termasuklah rekabentuk hierarki dan masalah permasaan. Selain itu, VHDL adalah bahasa pengaturcaraan yang direkabentuk dan dioptimumkan untuk menjelaskan perlakuan litar digital dan system. VHDL menggabungkan ciri-ciri seperti berikut:

2.0.2 Bahasa Permodelan Simulasi

VHDL mempunyai ciri-ciri yang tepat untuk menjelaskan perlakuan komponen elektronik sama ada pada peringkat get logik ringkas sehinggalah kepada mikropemproses lengkap dan cip. Ciri-ciri VHDL membenarkan aspek-aspek elektrikal bagi perlakuan litar dijelaskan dengan tepat. Model simulasi yang terhasil kemudian boleh digunakan sebagai blok dalam litar yang lebih besar (menggunakan skematik, gambarajah blok atau diskripsi peringkat sistem VHDL) untuk tujuan simulasi.

2.0.3 Bahasa Piawai

Penggunaan bahasa piawai seperti VHDL dapat memberi jaminan kepada kefungsian konsep rekabentuk tanpa perlu mempertimbangkan kaedah masukan

rekabentuk yang dipilih tidak disokong oleh perkakasan terbaru. Penggunaan bahasa piawai juga bermakna kita boleh mempergunakan perkakasan rekabentuk yang terbaru.

2.0.4 Bahasa Masukan Rekabentuk

Sama seperti bahasa pengaturcaraan peringkat tinggi lain yang membenarkan konsep rekabentuk kompleks didedahkan sebagai program komputer, VHDL membenarkan kelakuan bagi litar elektronik kompleks diadaptasikan kedalam sistem rekabentuk untuk sistesis litar automatik atau simulasi sistem. Seperti Pascal, C dan C++, VHDL merangkumi ciri-ciri yang berguna untuk teknik rekabentuk berstruktur dan menawarkan sejumlah besar set kawalan dan ciri-ciri perwakilan data. Apa yang membezakan VHDL dengan bahasa pengaturcaraan ini adalah VHDL menyediakan ciri-ciri yang membenarkan keserentakan dihuraikan. Ini adakah penting kerana perkakasan yang dijelaskan menggunakan VHDL sememangnya menjalankan operasi secara serentak.

2.0.5 Bahasa Pengujian

Salah satu daripada aspek terpenting VHDL adalah kebolehannya untuk menawan spesifikasi pencapaian litar, dalam bentuk yan biasanya dirujuk sebagai 'test bench'. Test bench adalah diskripsi bagi stimulus litar dan output jangkaan yang mengesahkan perlakuan litar dengan masa. Test bench sepatutnya menjadi bahagian penting bagi mana-mana projek VHDL dan patut dibentuk selari dengan lain-lain diskripsi litar.

2.0.6 Bahasa Netlist

VHDL adalah bahasa yang berkuasa bukan hanya semasa memasuki rekabentuk baru pada peringkat tinggi bahkan juga berguna sebagai bentuk peringkat rendah bagi komunikasi antara perkakasan yang berbeza dalam persekitaran rekabentuk berasaskan komputer. Ciri bahasa berstruktur VHDL membenarkannya digunakan secara berkesan sebagai bahasa netlist menggantikan lain-lain bahasa netlist.

2.1 KEGUNAAN VHDL

VHDL digunakan untuk tujuan seperti berikut:

2.1.1 Spesifikasi Rekabentuk

VHDL boleh digunakan pada bila-bila masa sama ada semasa rekabentuk peringkat tinggi, meng'capture' pencapaian dan keperluan antaramuka bagi setiap komponen dalam sesebuah sistem yang besar. Ia berguna terutamanya bagi projek besar yang melibatkan kerja berkumpulan. Dengan menggunakan pendekatan 'top-down' kepada rekabentuk, perekabentuk sistem perlu menjelaskan antaramuka bagi setiap komponen dalam sistem dan menjelaskan keperluan keserasian antara komponen dalam bentuk test bench peringkat tinggi. Difinisi antaramuka (lazimnya dijelaskan sebagai deklarasi entiti VHDL) dan spesifikasi pencapaian peringkat tinggi (test bench) kemudian boleh diagihkan kepada ahli kumpulan yang lain untuk proses perlengkapan atau pembaikan.

2.1.2 Peng'capture'an Rekabentuk

Peng'capture'an rekabentuk adalah fasa dimana perincian bagi sistem dimasukkan kedalam sistem rekabentuk berasaskan komputer. Dalam fasa ini, rekabentuk akan diekspreskan sebagai skematik atau menggunakan diskripsi VHDL. Jika kita bercdang untuk menggunakan teknologi sintesis, maka kita akan menulis bahagian VHDL bagi rekabentuk dengan menggunakan cara VHDL yang tepat untuk sintesis.

Fasa peng'capture'an mungkin melibatkan perkakasan dan kaedah masukan rekabentuk selain VHDL. Dalam kebanyakan kes, diskripsi rekabentuk yang ditulis dalam VHDL digabungkan dengan representasi yang lain seperti skematik untuk membentuk sistem yang lengkap.

2.1.3 Simulasi Rekabentuk

Apabila memasuki sistem rekabentuk berasaskan komputer, kita mungkin mahu mensimulasikan operasi litar untuk memastikan sama ada ia menepati kefungsian dan keperluan masa seperti yang dibina semasa proses spesifikasi. Jika kita perlu mencipta 1 atau lebih test bench sebagai sebahagian daripada spesifikasi rekabentuk, kita akan menggunakan simulator untuk mengenakan test bench kepada rekabentuk seperti yang ditulis untuk sintesis (simulasi fungsian).

2.1.4 Dokumentasi Rekabentuk

Pengaturcaraan berstruktur bercirikan VHDL bergabung dengan ciri-ciri pengurusan konfigurasinya menjadikan VHDL bentul semulajadi dalam mendokumenkan litar yang besar dan kompleks.

2.1.5 Sebagai Alternatif Kepada Skematik

Skematik telah lama menjadi sebahagian daripada rekabentuk sistem elektronik. Skematik mempunyai kelebihannya sendiri terutamanya apabila digunakan untuk menunjukkan perlitaran dalam bentuk gambarajah blok. Atas alasan ini banyak perkakasan rekabentuk VHDL kini menawarkan kebolehan untuk menggabungkan skematik dan representasi VHDL sebagai rekabentuk.

2.2 PERINGKAT ABSTRAK VHDL

VHDL menyokong rekabentuk pada beberapa peringkat pemisahan. Tiga daripadanya adalah sangat penting iaitu:

i) peringkat algoritma

ii) peringkat register-transfer

iii) peringkat logik

i) Peringkat algoritma

Peringkat ini menjelaskan sistem sebagai algoritma serentak. Setiap algoritma itu sendiri adalah berturutan, ini bermakna ia terdiri daripada satu set arahan yang dilaksanakan secara bergilir. Fungsi, prosedur dan proses merupakan elemen terpenting. Ia tidak bergantung kepada struktur rekabentuk.

ii) Peringkat register-transfer

Rekabentuk menggunakan peringkat register-transfer menentukan perlakuan litar dengan operasi dan pemindahan data diantara daftar. Jam 'explicit' digunakan. Rekabentuk RTL mempunyai kebarangkalian permasaan yang tepat, perlaksanaan operasi dijadualkan pada suatu tempoh.

iii) Peringkat logik

Dalam peringkat logik, perlakuan sistem dijelaskan dengan sambungan logikal dan sifat permasaannya. Semua isyarat adalah isyarat diskrit. Ia hanya boleh mempunyai nilai logikal yang tetap ('0', '1' ,'x' ,'low' ,'high' ,'undefined' ,'true' ,'false' , dll). Operasi yang boleh digunakan adalah tertakrif (AND, NAND, OR, NOR, XOR, NOT, dll).

2.3 KELEBIHAN VHDL

VHDL menawarkan kelebihan seperti berikut untuk rekabentuk digital:

Piawai:VHDL adalah piawai EKE. Seperti piawai yang lain (seperti piawai graphic x-window, piawai antaramuka komunikasi, bahasa pengaturcaraan peringkat tinggi, dll), ia mengurangkan kekeliruan dan menjadikan komunikasi antara perkakasan, pengeluar dan produk lebih mudah. Sebarang pembangunan kepada piawai mempunyai peluang yang lebih baik untuk bertahan lama kerana kesesuaiannya dengan yang lain.

Sokongan Industri: Dengan kewujudan lebih banyak perkakasan VHDL yang cekap dan berkuasa menyokong pembangunan industri elektronik. Syarikat yang menggunakan perkakasan VHDL bukan hanya dapat mempertahankan kontrak malahan rekabentuk komersil mereka.

Mudah Alih: Kod VHDL yang sama boleh disimulasikan dan digunakan dalam banyak perkakasan rekabentuk dan pada peringkat proses rekabentuk yang berbeza. Ini dapat mengurangkan kebergantungan kepada satu set rekabentuk yang berkemampuan terhad dan tidak mampu bersaing di pasaran terkini. Piawai VHDL juga menukarkan data rekabentuk dengan lebih mudah berbanding pangkalan data rekabentuk kepunyaan perkakasan rekabentuk.

Kebolehan Permodelan: VHDL dibina untuk memodelkan semua peringkat rekabentuk dari kotak elektronik ke transistor.VHDL mampu menampung bentuk kelakuan dan rutin matematik yang menjelaskan

model kompleks seperti rangkaian bergilir dan litar analog. Ia membenarkan penggunaan senibina berbilang dan bergabung dengan rekabentuk yang sama dalam pelbagai peringkat proses rekabentuk.

Sintesis dalam domain rekabentuk digital adalah satu proses penterjemahan dan pengoptimuman. Sebagai contoh, sintesis lampiran adalah proses pengambilan netlist rekabentuk dan menterjemahkannya kedalam bentuk data yan memudahkan penempatan dan laluan, membolehkan pemasaan dan saiz cip dioptimumkan. Sintesis logik pula adalah proses pengambilan bentuk input (VHDL), menterjemahkannya kepada bentuk (persamaan Boolean dan spesifik perkakasan sintesis), dan kemudian mengoptimumkan lengahan dan kawasan penyebaran.

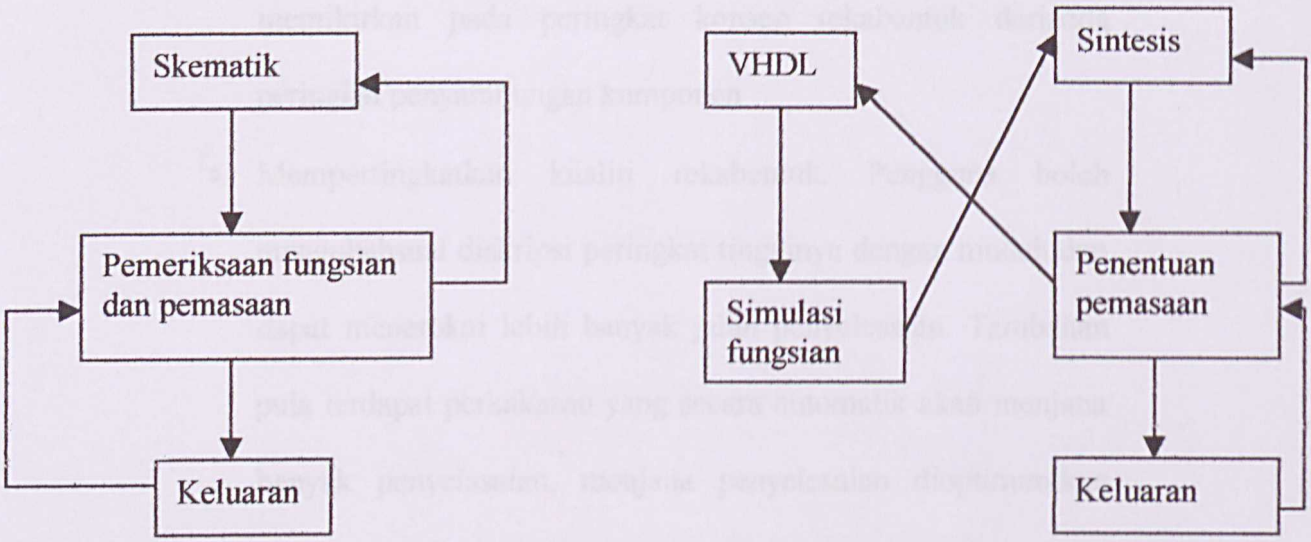
Perlu diambil perhatian bahawa terdapat blok yang menjanakan litar sesiri seperti flip flop dan sesetengah blok menjanakan litar gabungan. Setelah kod VHDL diterjemahkan kedalam bentuk internal, pengoktimuman proses boleh dilakukan berdasarkan kepada desakan seperti pecutan, kawasan, kuasa dan lain-lain.

2.4 METODOLOGI REKABENTUK YANG BARU

Pengenalan kepada VHDL dan sintesis membolehkan komuniti rekabentuk meneroka metodologi rekabentuk yang baru. Pendekatan rekabentuk tradisional seperti yang ditunjukkan dalam **Rajah 2.1** dimulakan dengan skematik lukisan dan kemudian menghasilkan simulasi

fungsian dan pemasaan berdasarkan skematik yang sama. Jika terdapat sebarang ralat pada rekabentuk, proses diulang semula untuk mengemaskini skematik. Selepas keluaran, fungsian dan catatan semula pemasaan disahkan sekali lagi dengan menggunakan skimatik yang sama.

Pendekatan rekabentuk berasaskan VHDL diilustrasikan dalam **Rajah 2.1**. Rekabentuk ini digunakan untuk mengesahkan kefungsian rekabentuk. Secara umum, mengubahsuai kod sumber VHDL adalah lebih cepat berbanding menukar skematik. Ini membolehkan perekabentuk membuat pembetulan kefungsian rekabentuk untuk meneroka lebih banyak 'trade-off' senibina dan memperolehi lebih banyak kesan keatas rekabentuk. Setelah fungsian memenuhi keperluan, kod VHDL disintesiskan untuk menjana skematik atau jumlah netlist yang sama. Netlist boleh digunakan untuk mengeluarkan litar dan mengesahkan keperluan pemasaan (sama ada sebelum atau selepas pengeluaran). Perubahan rekabentuk boleh dilakukan dengan mengubahsuai kod VHDL atau mengubah desakan (pemasaan, kawasan dll) dalam sintesis. Pendekatan rekabentuk dan metodologi telah mempertingkatkan proses rekabentuk dengan memendekkan masa rekabentuk, mengurangkan pengulangan semula rekabentuk dan meningkatkan kekompleksan rekabentuk yang boleh diuruskan oleh perekabentuk.



Pendekatan rekabentuk skematik tradisional

Pendekatan rekabentuk berasaskan VHDL

Rajah 2.1 : Perbandingan Proses Rekabentuk

Selain itu, VHDL juga menawarkan beberapa kelebihan berbanding dengan bahasa diskripsi perkakasan yang lain:

- VHDL membenarkan 1 projek dijalankan secara berkumpulan.
- VHDL menyokong banyak metodologi rekabentuk yan berbeza (cthnya rekabentuk atas-bawah, rekabentuk bawah-atas berasaskan perpustakaan, dll)
- Perekabentuk menggunakan VHDL menjadi jauh lebih produktif berbanding mereka yang menggunakan skematik atau bahasa tuju seperti ABEL atau CUPL. Masa rekabentuk dipendekkan
- Peringkat pemisahan VHDL membenarkan perekabentuk merekabentuk pada peringkat RTL (Register Transfer) dan peringkat perlakuan, maka perekabentuk hanya perlu

memikirkan pada peringkat konsep rekabentuk daripada peringkat penyambungan komponen

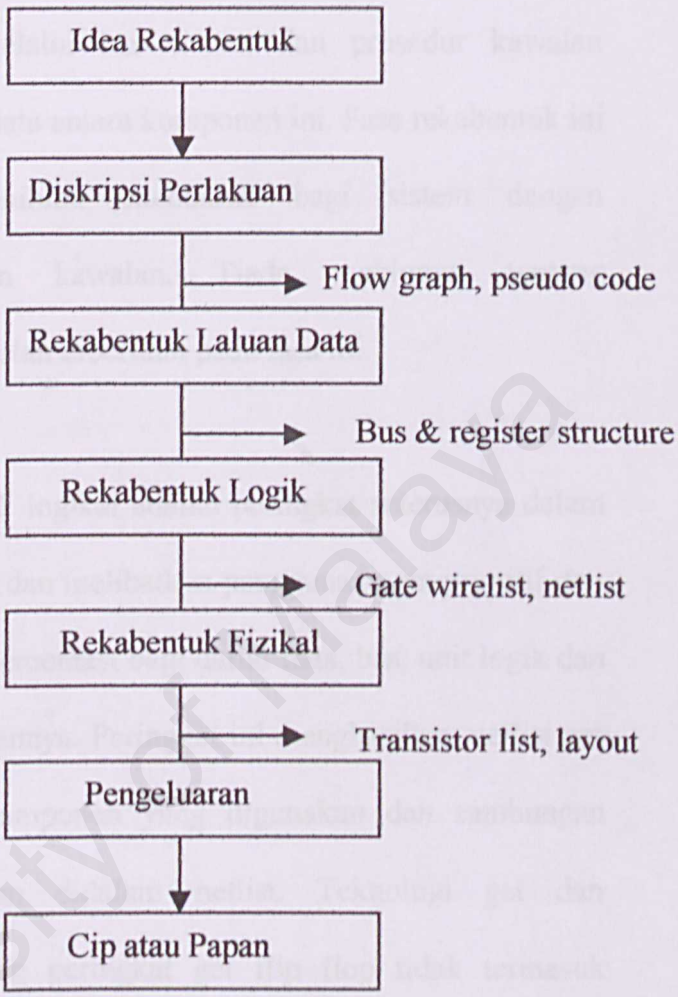
- Mempertingkatkan kualiti rekabentuk. Pengguna boleh mengubahsuai diskripsi peringkat tingginya dengan mudah dan dapat menerokai lebih banyak jalan penyelesaian. Tambahan pula terdapat perkakasan yang secara automatik akan menjana banyak penyelasaian, menjana penyelesaian dioptimumkan dengan pergantungan tertentu dan menggunakan sintesis logik automatik, pemetaan atau pengoptimuman keluaran. Kombinasi ciri-ciri diatas membolehkan kita memperolehi rekabentuk berkualiti dengan lebih cepat.

2.4.1 PROSES REKABENTUK PERKAKASAN

Dalam sistem digital biasa, aliran rekabentuk adalah seperti yang ditunjukkan dalam Rajah. Idea rekabentuk asal melalui beberapa transformasi sebelum implementasi perkakasannya diperolehi. Pada setiap peringkat transformasi, perekabentuk memeriksa hasil daripada transformasi sebelumnya, menambahkan maklumat kepadanya dan membawanya kepada peringkat transformasi yang berikutnya.

Mula-mula, perekabentuk perkakasan bermula dengan idea rekabentuk. Difinisi yang lebih kompleks bagi perkakasan yang dicadangkan kemudiannya dibina daripada idea rekabentuk yang asal. Oleh itu adalah perlu bagi perekabentuk

menjanakan difinisi perlakuan bagi sistem yang sedang dibentuk.



Rajah 2.2: Rekabentuk Sistem Digital Biasa

Fasa berikutnya dalam proses rekabentuk adalah rekabentuk bagi laluan data sistem. Dalam fasa ini, peekabentuk menentukan daftar dan unit logik yang perlu untuk implementasi sistem. Komponen ini boleh jadi disambungkan menggunakan bas searah atau 2 arah. Berdasarkan kepada

kelakuan sistem yang dicadangkan, prosedur kawalan pergerakan data antara daftar dan unit logik melalui bas kemudiannya dibina. Komponen data dalam litar bahagian data berkomunikasi melalui bas sistem dan prosedur kawalan mengawal aliran data antara komponen ini. Fasa rekabentuk ini menghasilkan senibina ekabentuk bagi sistem dengan spesifikasi aliran kawalan. Tiada maklumat tentang implementasi kawalan diberikan pada fasa ini.

Rekabentuk logikal adalah peringkat seterusnya dalam proses rekabentuk dan melibatkan penggunaan get primitif dan flip flop atau implementasi bagi daftar data, bas, unit logik dan perkakasan kawalannya. Peringkat ini menghasilkan netlist get dan flip flop. Komponen yang digunakan dan sambungan mereka ditentukan didalam netlist. Teknologi get dan penerangan tentang peringkat get flip flop tidak termasuk dalam netlist ini.

Peringkat rekabentuk berikutnya menukarkan netlist dari peringkat sebelumnya kepada senarai transistor atau keluaran. Ini melibatkan pengantian flip flop dengan transistor yang bernilai sama dan sel perpustakaan. Peringkat ini mempertimbangkan keperluan muatan dan pemasaan dalan selnya atau proses pemilihan transistor.

Langkah terakhir dalam proses rekabentuk adalah pengeluaran, yang mana menggunakan senarai transistor atau spesifikasi keluaran untuk menggabungkan membentuk perkakasan terprogram seperti FPGA yang mana digunakan dalam implementsi tesis ini atau untuk menjanakan topeng untuk pembinaan litar teragih (IC).

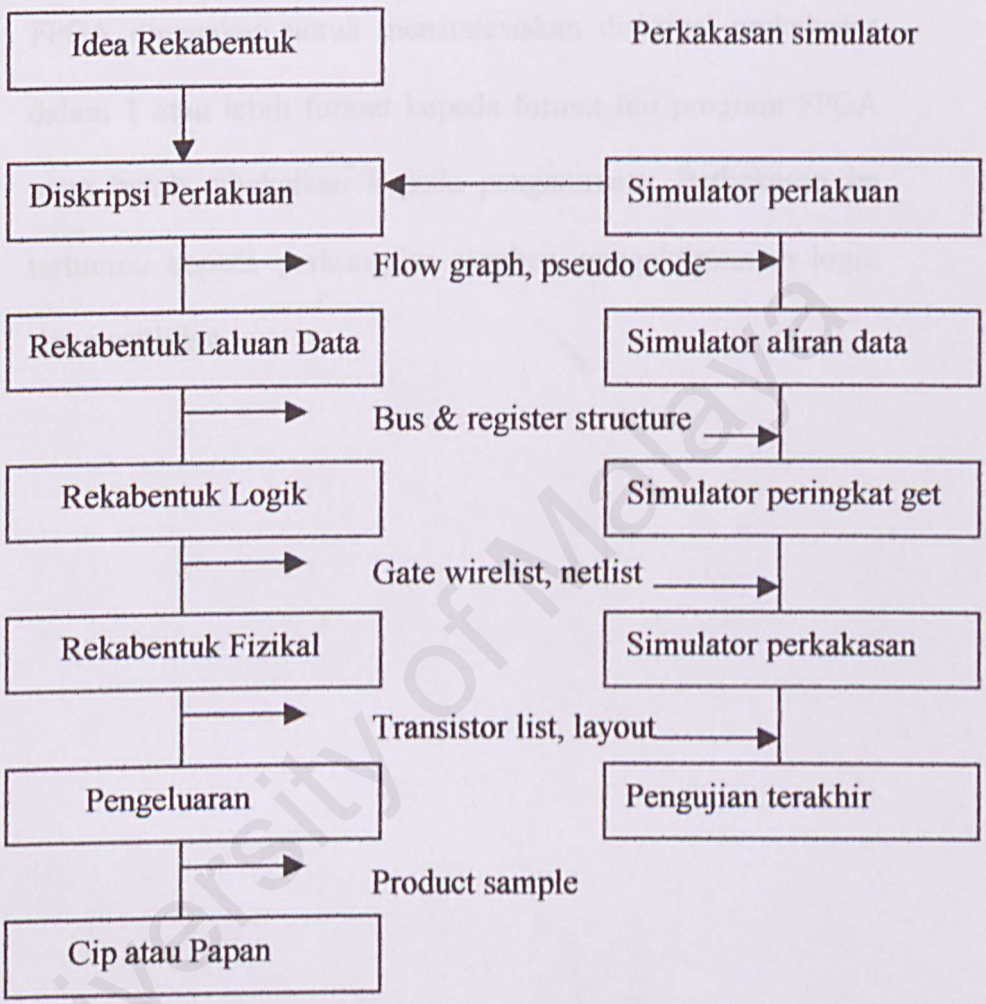
2.4.2 SIMULASI PERKAKASAN

Bahasa huraian perkakasan adalah perkakasan permodelan untuk membentuk model perkakasan. Simulasi adalah perlakuan melaksanakan model komponen sebenar untuk mengkaji kelakuannya dibawah set keadaan dan stimuli yang ditetapkan. Dengan difinisi ini, larian simulasi memerlukan model bagi objek di simulasikan dan satu set stimuli untuk mengaktifkan atau mensimulasikan kelakuan objek dikaji.

Dalam persekitaran rekabentuk perkakasan, huraian perkakasan menggunakan model komponen dan huraian daripada perpustakaan simulasi untuk membentuk model perkakasan yang boleh disimulasikan.

Simulator boleh digunakan pada mana-mana peringkat seperti yang ditunjukkan dalam **Rajah 2.3**. Pada peringkat

lebih tinggi dalam proses rekabentuk, simulasi menyediakan maklumat tentang kefungsian bagi sistem yang sedang direka.



Rajah 2.3: Menentukan setiap Peringkat Rekabentuk Dengan Mensimulasikan Outputnya

2.4.3 SINTESIS PERKAKASAN

Alatan rekabentuk yang secara automatik menukarkan diskripsi rekabentuk dari 1 bentuk kepada bentuk yang lain dinamakan perkakasan sintesis. HDL adalah medium yan gberguna untuk input dan output bagi pensintesis perkakasan.

Kebanyakan perkakasan sintesis yang boleh didapati dipasaran menggunakan peringkat rekabentuk laluan data sebagai input dan menghasilkan netlist untuk litar. Dalam tesis ini, perkakasan sintesis FPGA digunakan. Perkakasan sintesis FPGA digunakan untuk mensintesiskan diskripsi perkakasan dalam 1 atau lebih format kepada format fail program FPGA yang boleh dibekalkan kepada pengaturcara. Perkakasan ini tertumpu kepada perkongsian sumber, pengoktimuman logik dan penjilidan.

ANALISA SISTEM

3.0 ANALISA SISTEM

3.1 KEPERLUAN PERKAKASAN

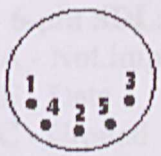
3.1.1 SAMBUNGAN FIZIKAL

Port fizikal PS/2 mungkin salah satu daripada 2 jenis sambungan: 5-pin DIN atau 6-pin mimi-DIN. Kedua-dua sambungan ini adalah sama secara elektrikal, apa yang berbeza antara keduanya adalah susunan pin. Ini bermakna kedua-dua jenis sambungan ini boleh ditukar ganti dengan mudah meggunakan adaptor hard-wired. Piawai DIN dicipta oleh German Standardization Organization (Deutshes Institute fuer Norm).

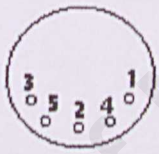
Papan kekunci komputer menggunakan sama ada sambungan 6-pin mini-DIN atau 5-pin DIN. Sekiranya papan kekunci menggunakan 6-pin mini-DIN dan komputer mempunyai sambungan 5-pin DIN atau sebaliknya, kedua-duanya boleh disepadankan dengan menggunakan adaptor seperti yang dijelaskan tadi. Papan kekunci 6-pin mini-DIN biasanya dirujuk sebagai papan kekunci PS/2 manakala papan kekunci yang mempunyai 5-pin DIN dipanggil perkakasan AT. Semua papan kekunci moden dibina untuk komputer adalah dari jenis PS/2, AT atau USB.

Kabel yang menyambungkan papan kekunci ke komputer biasanya 6 kaki panjang dan terdiri daripada 4 hingga 6 wayar 26 AWG disalut dengan lapisan nipis kerajang mylar.

Terdapat satu jenis lagi penyambung yang boleh digunakan pada papan kekunci. Jika kebanyakan kabel papan kekunci adalah hard-wired kepada papan kekunci, terdapat juga jenis boleh tanggal dan datang dalam komponen berasingan (terpisah). Kabel ini mempunyai sambungan DIN pada satu hujung (disambungkan ke komputer) dan sambungan SDL (Shielded Data Link) pada hujung satu lagi. SDL ini hampir serupa dengan sambungan pada telefon iaitu menggunakan wayar dan spring berbanding pin dan klip yang memegangnya pada tempatnya.



MALE
(PLUG)



FEMALE
(SOCKET)

5-pin DIN (AT/XT):
1 - Clock
2 - Data
3 - Not Implemented
4 - Ground
5 - Vcc (+5V)

Rajah 3.1: 5-pin DIN

MALE
(PLUG)

FEMALE
(SOCKET)

6-pin Mini-DIN (PS/2):
1 - Data
2 - Not Implemented
3 - Ground
4 - Vcc (+5V)
5 - Clock
6 - Not Implemented
6-pin SDL

Rajah 3.2: 6-pin mini-DIN

6-pin SDL:
A - Not Implemented
B - Data
C - Ground
D - Clock
E - Vcc (+5V)
F - Not Implemented

Rajah 3.3: 6 pin SDL

3.1.2 SAMBUNGAN ELEKTRIKAL

Dalam bahagian ini saya akan menggunakan istilah host untuk merujuk kepada komputer dan peranti untuk merujuk kepada papan kekunci.

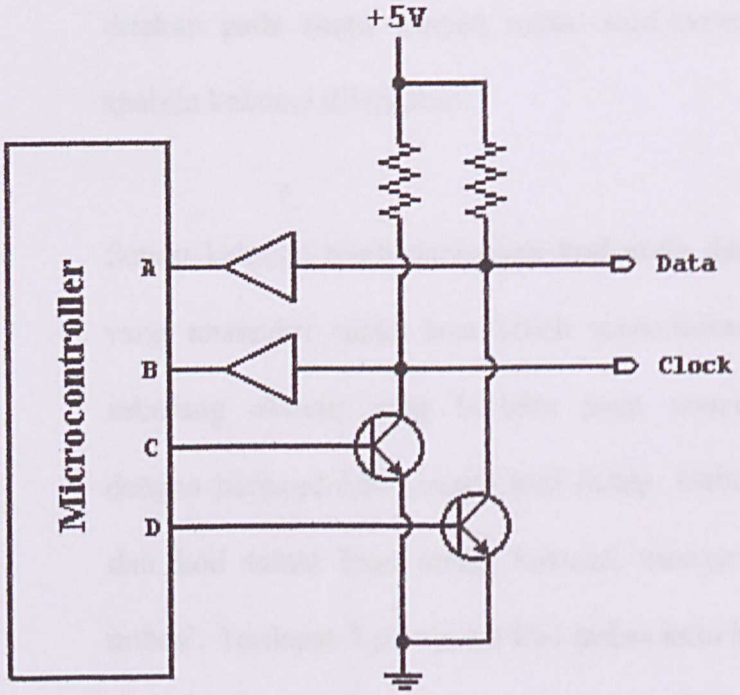
Vcc/Ground membekalkan kuasa kepada papan kekunci. Papan kekunci tidak sepatutnya menerima lebih daripada 100mA dari host dan langkah-langkah keselamatan perlu diambil untuk mengelakkan berlakunya 'transient surges'. 'Surges' boleh disebabkan oleh perbuatan 'hot-plugging' papan kekunci contohnya menyambung atau memutuskan sambungan peranti semasa komputer

sedang terpasang. Papan ibu sebelum ini mempunyai fius 'surface mounted' untuk melindungi port papan kekunci. Apabila fius ini terbakar, papan ibu itu tidak dapat digunakan lagi bahkan tidak boleh diperbaiki. Kebanyakan papan ibu hari ini meggunakan fius 'Poly' untuk mengatasi masalah ini. Walaubagaimanapun, ini bukanlah piawai kerana masih banyak papan ibu jenis lama digunakan.

Ringkasan: Spesifikasi Kuasa

Vcc = +5V
Kuasa maksima: 100mA

Line Data dan Clock adalah 'open-collector' dengan rintangan tekan-tutup (pull-off) +5V. Antaramuka 'open-collector' mempunyai 2 keadaan yang mungkin: impedance rendah atau tinggi. Pada keadaan rendah, transistor menekan line ke paras bumi (ground). Pada impedance tinggi, antaramuka bertindak sebagai litar terbuka dan tidak mengubah line kepada tinggi atau rendah. Tambahan pula, rintangan tekan-tutup disambungkan diantara bas dan Vcc maka bas akan ditekan tinggi jika tidak terdapat peranti semasa bas sedang menekannya rendah. Nilai sebenar rintangan ini tidaklah begitu penting (1~10kOhm); semakin besar rintangan menyebabkan kurang kuasa dan rintangan rendah menyebabkan masa peningkatan lebih cepat. Antaramuka 'open-collector' umum adalah seperti yang ditunjukkan dibawah.



Rajah 3.4: Antaramuka 'open-collector' umum. Data dan Clock dibaca pada pin A dan B mikrokawalan. Kedua-dua line biasanya ditetapkan pada +5V, tetapi boleh ditekan ke bumi dengan memasukkan logik '1' pada C dan D. hasilnya, Data bersamaan dengan D, dan Clock bersamaan dengan C.

3.1.3 KOD IMBAS (SCAN CODE)

Kawalan papan kekunci memperuntukkan hampir keseluruhan masanya untuk mengimbas atau mengawasi matrik kekunci. Sekiranya ia mendapati bahawa mana-mana kekunci ditekan, dilepaskan atau ditekan pada suatu tempoh masa, papan kekunci akan menghantar paket pemberitahuan (informasi) yang dikenali sebagai 'kod imbas' ke komputer. Terdapat dua jenis kod imbas: 'kod mula' (make codes) dan 'kod tamat' (break codes). Kod mula dihantar apabila kekunci ditekan atau

ditekan pada suatu tempoh masa. Kod tamat pula dihantar apabila kekunci dilepaskan.

Setiap kekunci telah ditetapkan kod mula dan kod tamatnya yang tersendiri maka host boleh menentukan dengan tepat sebarang aktiviti yang berlaku pada mana-mana kekunci dengan berpandukan sesuatu kod imbas. Gabungan kod mula dan kod tamat bagi setiap kekunci mengandungi 'set kod imbas'. Terdapat 3 piawai set kod imbas iaitu Kod Imbas Set 1 (SCS1), Kod Imbas Set 2 (SCS2) dan Kod Imbas Set 3 (SCS3). Semua papan kekunci moden menggunakan SCS2.

3.1.3.1 Kod Mula

Sebaik sahaja kekunci ditekan, kod mula bagi kekunci berkenaan dihantar ke komputer. Kod mula hanya mewakili kekunci pada papan kekunci, bukan karektor yang tercetak diatas kekunci tersebut. Ini bermakna tidak terdapat hubungan antara kod mula dan kod ASCII. Ia bergantung kepada host untuk menterjemahkan kod imbas kepada apa-apa karektor atau arahan.

Walaupun semua SCS2 hanya bersaiz 1 byte, terdapat banyak kekunci lanjutan (extended keys) kod mula bersaiz 2 atau 4

byte. Kod mula ini boleh dikenalpasti dengan byte pertama bernilai E0h.

3.1.3.2 Kod Tamat

Sebaik sahaja kod mula dihantar kepada komputer setelah kekunci ditekan, kod tamat pula akan dihantar kepada komputer apabila kekunci dilepaskan. Seperti juga kod mula, setiap kekunci mempunyai kod tamatnya tersendiri. Terdapat hubungkait antara kod tamat dan kod mula bagi setiap kekunci. Semua kod tamat set 2 adalah bersaiz 2 byte dimana byte yang pertama adalah F0h dan byte kedua merupakan kod mula bagi kekunci tersebut.

Kod tamat bagi kekunci lanjutan biasanya bersaiz 3 byte dimana E0h dan F0h bagi 2 byte pertama dan byte terakhir adalah kod mula kekunci tersebut.

3.1.3.3 Kadar Typematic

Sekali kekunci ditekan, kod mulanya akan dihantar ke komputer. Apabila sesuatu kekunci ditekan pada suatu tempoh masa, kekunci itu menjadi 'typematik' yang bermaksud papan

kekunci akan terus menghantar kod mula kekunci tersebut sehinggalah ianya dilepaskan atau kekunci lain ditekan.

Dalam satu keadaan dimana kekunci A ditekan pada suatu tempoh masa, karektor A akan muncul di skrin. Selepas lengahan singkat, A lain akan muncul diikuti dengan jujukan huruf A sehingga kekunci dilepaskan. Terdapat 2 parameter penting disini: lengahan typematik→ lengahan pendek diantara 'A' yang pertama dan kedua dan kadar typematik→ bilangan karektor per saat yang muncul di skrin selepas lengahan typematik. Lengahan typematik berkadar antara 0.25 saat hingga 1.00 saat dan kadar typematik berkadar 2.0cps (karektor per saat) ke 30.0cps. Nilai kadar dan lengahan typematik ini boleh ditukar menggunakan arahan "Set Typematic Rate/Delay" (0xF3).

Data typematic tidak di'buffer'kan bersama papan kekunci. Dalam kes dimana lebih daripada 1 kekunci ditekan sekaligus pada suatu tempoh masa, hanya kekunci terakhir yang ditekan sahaja akan menjadi typematik. Ulangan typematik (typematic repeat) akan berhenti apabila kekunci tersebut dilepaskan walaupun kekunci lain masih ditekan.

3.1.3.4 Reset

Pada 'power-on' atau 'reset perisian', papan kekunci melaksanakan diagnostik 'self-test' yang dirujuk sebagai BAT (Basic Assurance Test) dan memuatkan nilai-nilai berikut:

- lengahan typematik 500ms
- kadar typematik 10.9cps
- SCS2
- Set semua typematik/mula/tamat kekunci

Apabila memulakan BAT, papan kekunci akan menghidupkan 3 penanda LED dan mematikannya apabila BAT selesai. Ketika ini, kod pelengkap BAT mesti dihantar 500~750 milisaat selepas kuasa dihidupkan.

3.1.3.4 Set Arahan

Beberapa nota berkenaan arahan yang boleh dihantar oleh host kepada papan kekunci.

- Papan kekunci mengosongkan buffer keluarannya apabila ia menerima apa-apa arahan.
- Jika papan kekunci menerima arahan tidak sah atau percanggahan, ia mesti bertindak balas dengan "resend" (0xFE).
- Papan kekunci tidak boleh menghantar mana-mana kod imbas semasa pemprosesan arahan dilakukan.

- Jika papan kekunci sedang menunggu byte percanggahan dan menerima arahan yang lain, ia harus mengabaikan arahan sebelumnya dan proses arahan terbaru.

Arahan-arahan yang mungkin dihantar oleh host kepada papan kekunci:

- 0xFF (reset) — papan kekunci bertindak balas dengan "ack" (0xFA) kemudian memasuki mod 'Reset'.
- 0xFE (resend)—papan kekunci bertindak balas dengan menghantar semula byte terakhir. Pengecualian jika byte terakhir adalah 'resend' (0xFE). Dalam keadaan ini papan kekunci perlu menghantar semula byte terakhir yang bukan 0xFE. Arahan ini digunakan oleh host untuk menunjukan ralat penerimaan.

6 arahan berikut boleh digunakan semasa papan kekunci berada pada mana-mana mod, tetapi ia hanya memberi kesan kepada papan kekunci apabila berada dalam mod 3.

- *0xFD (set jenis kekunci mula)—mempasifkan kod tamat dan ulangan typematik untuk kekunci tertentu. Papan kekunci bertindak balas dengan 'ack' (0xFA), kemudian mempasifkan pengimbasan (jika aktif) dan membaca senarai kekunci dari host. Kekunci-kekunci ini ditentukan dengan kod mula set 3nya. Papan kekunci bertindak balas kepada setiap kod mula dengan 'ack'. Host memusnahkan senarai ini dengan menghantar kod mula set 3 yang tidak

sah. Papan kekunci kemuadia akan mengaktifkan pengimbasan (jika sebelum ini tidak aktif).

- *0xFC (set jenis kekunci mula/tamat)—sama seperti arahan sebelum ini tetapi hanya mempasifkan ulangan typematik.
- *0xFB (set jenis kekunci typematik)—sama seperti arahan sebelum ini tetapi hanya mempasifkan kod tamat.
- *0xFA (set semua kekunci typematik/mula/tamat)—papan kekunci bertindak balas dengan 'ack' (0xFA). Setkan semua kekunci kepada set normal (menjana kod imbas pada mula, tamat dan ulangan typematik)
- *0xF9 (set semua kekunci mula)—papan kekunci bertindak balas dengan 'ack' (0xFA). Sama seperti 0xFD tetapi diterapkan kepada semua kekunci.
- *0xF8 (set semua kekunci mula/tamat)—papan kekunci bertindak balas dengan 'ack' (0xFA). Sama seperti 0xFC, tetapi diterapkan kepada semua kekunci.
- *0xF7 (set semua kekunci typematik)—papan kekunci bertindak balas dengan 'ack' (0xFA). Sama seperti 0xFB, tetapi diterapkan kepada semua kekunci.
- 0xF6 (set default)—muat default kadar/lengahan typematik (10.9cps/500ms) jenis kekunci (semua kekunci typematik/mula/tamat) dan SCS2.
- 0xF5 (Disable)—papan kekunci mengimbas, muat nilai default (lihat arahan set Default) dan tunggu arahan berikutnya.

- 0xF4 (Enable)—aktifkan semula papan kekunci selepas dipasifkan menggunakan arahan sebelumnya.
- 0xF3 (set kadar/lengahan typematik)—host mengikut arahan ini dengan 1 byte bit percanggahan yang menetapkan kadar dan lengahan typematik seperti berikut.
- *0xF2 (baca ID)—papan kekeunci bertindakbalas dengan menghantar 2 byte ID peranti bagi 0xAB, 0x83 (0xAB dihantar dahulu diikuti dengan 0x83).
- *0xF0 (set SCS)—papan kekunci bertindak balas dengan 'ack', kemudian membaca bit percanggahan dari host. Percanggahan ini boleh bernilai 0x01, 0x02 atau 0x03 untuk memilih SCS1,SCS2 atau SCS3.papan kekunci bertindak balas dengan 'ack' diikuti dengan set kod imbas semasa.
- 0xEE (Echo)—papan kekunci bertindak bals dengan 'Echo' (0xEE).
- 0xED (set/reset LED)—host mengikut arahan ini dengan 1 byte percanggahan yang menentukan keadaan LED bagi NumLock,CapsLock dan ScrollLock papan kekunci.

*Terdapat pada papan kekunci PS/2 sahaja.

3.1.4 FPGA SPARTAN II

3.1.4.1 Pengenalan

Keluarga FPGA Spartan II menjanjikan kepada pengguna pencapaian yang tinggi, bantuan logic yang banyak, dan kaya dengan set cirian

Keenam-enam ahli ini menawarkan kepadatan dalam julat get sistem dari 15000 hingga 200000. Pencapaian sistem yang disokong mencapai 200MHz.

Perkakasan Spartan II membekalkan lebih banyak get, I/O, ciri per wang berbanding FPGA lain dengan menggabungkan teknologi proses lanjutan dengan senibina baris alir (streamlined) berasaskan Virtex. Ciri-cirinya termasuklah blok RAM (sehingga 54K bit), RAM teragih (sehingga 75264 bit), 16 piawai I/O boleh pilih, dan 4 DLL. Laju, sambungan yang 'boleh diramal' dimana pengulangan semula rekabentuk berturut-turut untuk penyesuaian dengan keperluan masa.

Keluarga Spartan II adalah pilihan lebih baik untuk ASIC 'mask-programmed'. FPGA ini mengelakkan pertambahan kos, kitar pembangunan yang panjang, dan risiko semulajadi seperti yang terdapat pada ASIC biasa. Kebolehaturcara FPGA membenarkan penaiktarafan rekabentuk tanpa memerlukan penggantian perkakasan (mustahil dilakukan dengan ASIC).

3.1.4.2 Ciri-ciri

i. Teknologi penggantian ASIC generasi kedua.

- Kepadatan sehingga 5292 sel logic dengan sistem get sehingga 200000.
- Ciri baris alir berdasarkan senibina Virtex
- Boleh diaturcara semula tanpa had
- Kos yang sangat rendah

ii. Ciri-ciri peringkat sistem

- Ingatan hierarki SelectRAM+
  - RAM teragih 16 bit/LUT
  - Blok RAM 4K bit yang boleh dikonfigurasikan
  - Antaramuka pantas ke RAM luar (external)
- Patuh PCI sepenuhnya
- Senibina laluan tersegmen berkuasa rendah
- Kebolehan bacabalik (readback) sepenuhnya untuk pengesahan/ketelusan
- Logic hawaan (carry) tertuju untuk aritmetik kelajuan tinggi
- Sokongan multiplier tertuju
- Rantaian untuk fungsi input meluas
- Daftar/selak tambahan dengan enable, set dan reset
- 4 DLL tertuju untuk kawalan jam lanjutan
- 4 jaring pengagihan jam global 'low-skew' prima
- logik imbas sempadan sesuai IEEE 1149.1

iii. I/O dan pembungkusan serbaboleh

- Pakej berkos rendah boleh didapati dalam semua kepadatan
- Kesesuaian jejak keluarga dalam pakej biasa
- 16 piawai antaramuka berpencapaian tinggi
- Mesra PCI Compact
- Masa pegangan sifar memudahkan sistem permasaan

iv. Disokong sepenuhnya oleh sistem Pembangunan Xilinx

- Siri Pertubuhan ISE: perisian teragih sepenuhnya
- Siri Persekutuan: untuk digunakan dengan alatan pihak ketiga
- Pemetaan, menempatan dan laluan automatic sepenuhnya

3.1.5 PAPAN XSTEND

3.1.5.1 Gambaran Umum XStend

Papan XS40 dan XS95 menawarkan kaedah pemprototaipan rekabentuk FPGA dan CPLD yang fleksibel dan murah. Walaubagaimanapun, saiz fizikalnya yang kecil menghadkan bilangan perlitaran sokongan yang boleh dipegang. Papan XStend menghapuskan had ini dengan menyediakan perlitaran sokongan tambahan yang boleh dicapai oleh papan XS40 dan XS95 melalui antaramuka breadboard mereka.

Papan XStend mengandungi sumber-sumber yang melanjutkan had aplikasi bagi papan XS kepada 3 kawasan:

i. butang tekan, switch DIP, LED, dan kawasan prototaipan sangat berguna untuk eksperimen asas makmal. Ciri ini bergabung dengan Papan XS mereplikakan kefungsian HW/UW FPGABOARD.

ii. Antaramuka monitor VGA, antaramuka papan kekunci/tetikus PS/2, dan RAM static membenarkan Papan XS digunakan dalam video dan eksperimen pengkomputeran.

iii. Codec stereo dan perlitaran input/output analog 2-saluran sangat berguna untuk pemprosesan isyarat audio dalam kombinasi dengan litar DSP disentesiskan dengan generasi perisian teras XILINX.

3.1.5.2 Ciri-ciri Papan XStend

Papan XStend melanjutkan keupayaan bagi Papan XS40 dan XS95 dengan meyediakan:

i. bekas bertindih untuk kedua-dua Papan XS40 dan XS95

- Papan XS40 dan XS95 ditindihkan diatas Papan XStend menggunakan bekas bertindih J1 atau J2. Bekas ini bergabung dengan antaramuka 'breadboard' bagi Papan XS untuk memberikan mereka capaian kepada semua sumber Papan XStend. Papan XStend juga membekalkan kuasa kepada Papan XStend melalui bekas ini.
- Untuk menggunakan Papan XS40 dengan Papan XStend, XS40 perlu dimasukkan kedalam kolum paling kanan pada bekas bertindih. Apabila menggunakan Papan XS95, ia perlu dimasukkan kedalam kolum paling kiri. Ia ditandakan pada Papan XStend untuk menunjukkan kolum mana didudukki oleh setiap jenis Papan XS.

ii. LED

- Papan XStend menyediakan tambahan 8 LED (D1 – D8) dan 2 ingi LED display (U1 dan U2) untuk kegunaan Papan XS. Kesemua

LED ini adalah 'active-low' bermakna LED atau segmen LED display akan menyala apabila logic rendah dikenakan kepadanya.

- LED 'disable' dengan mengalihkan simpang pada pelompat berikut:

Pelompat	Setting
J8	Mengalihkan simpang pada pelompat ini memutuskan kuasa dari LED D1 – D8.
J4	Mengalihkan simpang pada pelompat ini memutuskan kuasa pada LED sebelah kiri display U1.
J7	Mengalihkan simpang pada pelompat ini memutuskan kuasa pada LED sebelah kanan display U2.

Jadual 3.1: Pelompat dan Penetapannya.

iii. Switch

- XStend mempunyai simpanan 8 switch DIP dan 2 butang tekan (dilabelkan SPARE dan RESET) yang boleh dicapai dari Papan XS. (terdapat butang tekan ketiga dilabelkan sebagai PROGRAM yang digunakan untuk memulakan pengaturcaraan Papan XS40. Ia tidak dicadangkan sebagai input kegunaan umum).
- Apabila ditutup, setiap switch DIP menarik pin Papan XS yang bersambung ke bumi. Bila switch DIP dibuka, pin ditarik tinggi melalui perintang 10K.

- Apabila ditekan, setiap butang tekan menarik pin Papan XS yang bersambung ke bumi. Jika sebaliknya, pin ditarik tinggi melalui perintang 10K.

iv. Antaramuka VGA

- Papan XStend menyediakan Papan XS dengan antaramuka kepada monitor VGA melalui sambungan J5. Papan XS boleh mengawal isyarat segerak aktif-rendah(active-low) mendatar(horizontal) atau menegak(vertical) dan mengawal lebar dan tinggi bingkai video. Papan XS juga mempunyai capaian kepada setiap 2 bit bagi isyarat warna merah, hijau dan biru supaya ia dapat menjanakan pixel dalam mana-mana $2^2x2^2x2^2=64$ warna yang berbeza.

v. Antaramuka Papan Kekunci PS/2

- Papan XStend menyediakan Papan XS dengan antaramuka jenis PS/2 (mini-DIN connector J6) samada untuk papan kekunci atau tetikus. Papan XS menerima 2 isyarat daripada antaramuka PS/2: isyarat jam dan aliran data sesiri yang mana segerak dengan pinggir menurun pada isyarat jam.

vi. RAM

- Papan XStend menambahkan tambahan RAM 64Kbytes kepada 32Kbytes yang sedia ada pada Papan XS. RAM XStend bersambung pada pin yang sama seperti RAM bagi Papan XS untuk bus alamat, bus data, 'write-enable' dan 'output-enable'.

Pilihan chip bagi Papan XStend disambungkan pada pin yang berbeza supaya RAM boleh dipilih sendiri.

- RAM XStend akan 'disable' dengan mengalihkan simpang pada pelompat berikut:

Pelompat	Setting
J16	Mengalihkan simpang pada pelompat ini akan 'disable' RAM U5 sebelah kiri dengan menarik pin chip-select tinggi
J17	Mengalihkan simpang pada pelompat ini akan 'disable' RAM U6 sebelah kanan dengan menarik pin chip-select tinggi

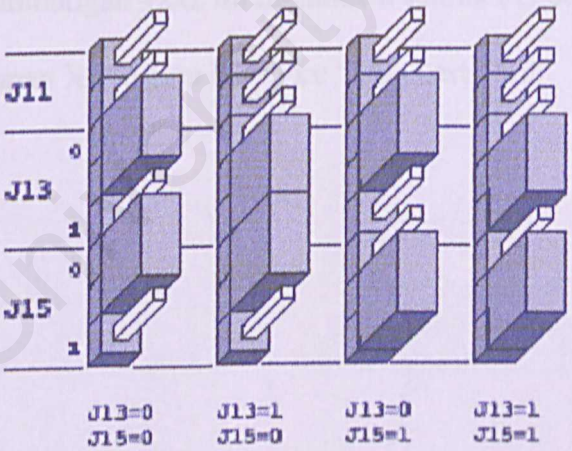
Jadual 3.2: Pelompat dan Penetapannya

vii. Codec Stereo

- Papan XStend mempunyai codec stereo yang menerima 2 analog saluran input dari jack J9, mendigitalkan nilai analog, dan menghantar nilai digital kepada Papan XS sebagai aliran bit sesiri. Codec juga menerima aliran bir sesiri dari Papan XS dan menukarkannya kepada isyarat output 2 analog yang keluar dari Papan XStend melalui jack J10.
- Codec dikonfigurasikan dengan menempatkan simpang pada pelompat berikut:

Pelompat	Setting
J11	Menempatkan simpang pada pelompat ini 'disable' codec dengan memegangnya dalam state reset.
J13, J15	Menempatkan simpang melintangi 2 daripada 3 pin pada pelompat ini memilih de-emphasis digital bagi kadar pengsampelan yang berbeza:
0 0	De-emphasis untuk 32KHz
0 1	De-emphasis untuk 44.1KHz
1 0	De-emphasis untuk 48KHz
1 1	De-emphasis 'off'

Jadual 3.3: Pelompat dan penetapannya



Rajah 3.1: Setting pelompat untuk kadar pengsampelan codec de-emphasis

viii. Kawasan Pemprototaipan

- Papan XStend mempunyai kawasan pemprototaipan berupa komponen 'through-holes' diatas grid 0.1"x0.1" diletakkan dengan rangkaian +5V dan bus GND.
- Sambungan dari Papan XS ke kawasan pemprototaipan dibuat melalui sambungan J3. Susunan pin bagi sambungan ini bersesuaian dengan susunan pin pada Papan XS40. Sebagai contoh, pin yang berada pada bawah-kiri J3 pada Papan XStend disambungkan kepada pin 68 berhampiran dengan atas-kiri pada Papan XS95.

ix. Sambungan Daughterboard

- 'Daughterboard' dengan perlitaran terkhususboleh disambungkan kepada Papan XStend melalui sambungan J18. Sambungan 42x2 ini membawa semua I/O dan kuasa/GND dari Papan XS40 atau XS95 ke 'daughterboard'.

SaS

XS40 Pins J1/J3	XS95 Pins J1/J3	Power/GND	DIP Switch	Pushbuttons	LEDs	VGA Interface	PS/2 Interface	RAMs	Stereo Codec	8051 uC	PC Parallel Port	Oscillator	Function	XS-FPGA Pin
1	77												Uncommitted XS95 I/O pin	
2	78	+5V											+5V power source	
3	1				LS_0			A0					LED segment; address line	P35
4	2				LS_1			A1					LED segment; address line	P36
5	3				LS_2			A2					LED segment; address line	P29
6	5		DIPSW4						LRCK	P13			DIP switch; codec control; uC I/O	P24
7	6		DIPSW1					LCE_		P10			DIP switch; RAM chip-enable, uC I/O port	P19
8	7		DIPSW2					RCE_		P11			DIP switch; RAM chip-enable, uC I/O port	P20
9	11		DIPSW3						MCLK	P12			DIP switch; codec clock; uC I/O port	P23
10	35				D_8			AD7		P07			LED; RAM data line; uC muxed address/data line	P61
11	31												Uncommitted XS95 I/O pin	
12	69												Uncommitted XS95 I/O pin	
13	9											CLK	XS40/95 oscillator output	
14	13									PSEN_			uC program storage-enable	
15	28												JTAG TDI; DIN	
16	30												JTAG TCK; CCLK	
17	29												JTAG TMS	
18	14				S5	RED1							XS Board LED segment; VGA color signal	
19	15				S6	HSYNC_							XS Board LED segment; VGA horiz. sync.	
20	17				S3	GREEN1							XS Board LED segment; VGA color signal	
21	68												Uncommitted XS95 I/O pin	
22	33												Uncommitted XS95 I/O pin	
23	18				S4	RED0							XS Board LED segment; VGA color signal	
24	19				S2	GREEN0							XS Board LED segment; VGA color signal	
25	21				S0	BLUE0							XS Board LED segment; VGA color signal	
26	23				S1	BLUE1							XS Board LED segment; VGA color signal	
27	32									RD_			uC read line	
28	34				RDP_			A15		P27			LED decimal-point; address line; uC I/O port	P41
29	20									ALE_			uC address-latch-enable	
30													Serial EEPROM chip-enable	
31	12												Uncommitted XS95 I/O pin	
32	81										PC_D6		PC parallel port output	
33	25												Uncommitted XS95 I/O pin	
34	80										PC_D7		PC parallel port output	
35	39				D_5			AD4		P04			LED; RAM data line; uC muxed address/data line	P66
36	45									RST			uC reset line	
37	10			RESET_						XTAL1			Pushbutton; uC clock line	P56
38	40				D_4			AD3		P03			LED; RAM data line; uC muxed address/data line	P57
39	41				D_3			AD2		P02			LED; RAM data line; uC muxed address/data line	P58
40	43				D_2			AD1		P01			LED; RAM data line; uC muxed address/data line	P59
41	44				D_1			AD0		P00			LED; RAM data line; uC muxed address/data line	P60
42	4												Uncommitted XS95 I/O pin	
43													Unconnected	
44	46								CCLK		PC_D0		Codec control line; PC parallel port output	
45	47								CDIN		PC_D1		Codec control line; PC parallel port output	
46	48								CS_		PC_D2		Codec control line; PC parallel port output	
47	50										PC_D3		PC parallel port output	
48	51										PC_D4		PC parallel port output	
49	52										PC_D5		PC parallel port output	
50	53				RS_4			A12		P24			LED segment; address line; uC I/O port	P48
51	54				RS_2			A10		P22			LED segment; address line; uC I/O port	P45
52	49	GND											Power supply ground	
53													Unconnected	
54		+3.3V											+3.3V/+5V power supply (4000E/4000XL)	
55				PROGRAM									XS40 configuration control	P55
56	55				RS_3			A11		P23			LED segment; address line; uC I/O port	P51
57	56				RS_1			A9		P21			LED segment; address line; uC I/O port	P47
58	57				RS_5			A13		P25			LED segment; address line; uC I/O port	P50
59	58				RS_0			A8		P20			LED segment; address line; uC I/O port	P46
60	61				RS_6			A14		P26			LED segment; address line; uC I/O port	P49
61	62							OE_					RAM output-enable	
62	63							WR_		P36			RAM write-enable; uC I/O port	
63													Unconnected	
64													Unconnected	
65	65							CE_					XS Board RAM chip-enable	
66	66		DIPSW7						SDOUT	P16			DIP switch; codec output data; uC I/O port	P27
67	24,67			SPARE_		VSYNC_				P17			Pushbutton; VGA horiz. sync.; uC I/O port	P18
68	26						KB_CLK			P34			PS/2 keyboard clock; uC I/O port	
69	70		DIPSW8				KB_DATA			P31			DIP switch; keyboard serial data; uC I/O port	P26
70	71		DIPSW6						SDIN	P15			DIP switch; codec input data; uC I/O port	P28
71	28												JTAG TDI; DIN	
72	59												JTAG TDO; DOUT	
73	30												JTAG TCK; CCLK	
74	74												Uncommitted XS95 I/O pin	
75	59												JTAG TDO; DOUT	
76	78												Uncommitted XS95 I/O pin	
77	72		DIPSW5						SCLK	P14			DIP switch; codec serial clock; uC I/O port	P25
78	75				LS_3			A3					LED segment; address line	P44
79	79				LS_4			A4					LED segment; address line	P38
80	36				D_7			AD6		P06			LED; RAM data line; uC muxed address/data line	P62
81	37				D_6			AD5		P05			LED; RAM data line; uC muxed address/data line	P65
82	82				LS_5			A5					LED segment; address line	P40
83	83				LS_6			A6					LED segment; address line	P39
84	84				LDP_			A7					LED decimal-point; address line	P37

Jadual 3.4: Sambungan antara Papan XS40 dengan Papan XStend

XS95 Pins (J2)	XS40 Bus (J1, J3, J18)	Power/ GND	DIP Switch	Push-buttons	LEDs	VGA Interface	PS/2 Interface	RAMs	Stereo Codec	8051 Uc	PC Parallel Port	Oscillator	Function	UW-FPGA BOARD Pin
1	3				LS_0			A0					LED segment; address line	P35
2	4				LS_1			A1					LED segment; address line	P38
3	5				LS_2			A2					LED segment; address line	P29
4	42												Uncommitted XS95 I/O pin	
5	6		DIPSW4						LRCK	P13			DIP switch; codec control; uC I/O	P24
6	7		DIPSW1					LCE_		P10			DIP switch; RAM chip-enable; uC I/O port	P19
7	8		DIPSW2					RCE_		P11			DIP switch; RAM chip-enable; uC I/O port	P20
9	13											CLK	XS40/95 oscillator output	
10	37			RESET_						XTAL1			Pushbutton; uC clock line	P56
11	9		DIPSW3						MCLK	P12			DIP switch; codec clock; uC I/O port	P23
12	31												Uncommitted XS95 I/O pin	
13	14									PSEN_			uC program storage-enable	
14	18				S5	RED1							XS Board LED segment; VGA color signal	
15	19				S6	HSYNC_							XS Board LED segment; VGA horiz. sync.	
17	20				S3	GREEN1							XS Board LED segment; VGA color signal	
18	23				S4	RED0							XS Board LED segment; VGA color signal	
19	24				S2	GREEN0							XS Board LED segment; VGA color signal	
20	29									ALE_			uC address-latch-enable	
21	25				S0	BLUE0							XS Board LED segment; VGA color signal	
23	26				S1	BLUE1							XS Board LED segment; VGA color signal	
25	33												Uncommitted XS95 I/O pin	
26	68						KB_CLK			P34			PS/2 keyboard clock; uC I/O port	
28	15												JTAG TDI; DIN	
28	71												JTAG TDI; DIN	
29	17												JTAG TMS	
30	16												JTAG TCK; CCLK	
30	73												JTAG TCK; CCLK	
31	11												Uncommitted XS95 I/O pin	
32	27									RD_			uC read line	
33	22												Uncommitted XS95 I/O pin	
34	28				RDP_			A15		P27			LED decimal-point; address line; uC I/O port	P41
35	10				D_8			AD7		P07			LED; RAM data line; uC muxed address/data line	P61
36	80				D_7			AD6		P06			LED; RAM data line; uC muxed address/data line	P62
37	81				D_6			AD5		P05			LED; RAM data line; uC muxed address/data line	P65
39	35				D_5			AD4		P04			LED; RAM data line; uC muxed address/data line	P66
40	38				D_4			AD3		P03			LED; RAM data line; uC muxed address/data line	P57
41	39				D_3			AD2		P02			LED; RAM data line; uC muxed address/data line	P58
43	40				D_2			AD1		P01			LED; RAM data line; uC muxed address/data line	P59
44	41				D_1			AD0		P00			LED; RAM data line; uC muxed address/data line	P60
45	38									RST			uC reset line	
46	44								CCLK		PC_D0		Codec control line; PC parallel port output	
47	45								CDIN		PC_D1		Codec control line; PC parallel port output	
48	48								CS_		PC_D2		Codec control line; PC parallel port output	
49	52	GND											Power supply ground	
50	47										PC_D3		PC parallel port output	
51	48										PC_D4		PC parallel port output	
52	49										PC_D5		PC parallel port output	
53	50				RS_4			A12		P24			LED segment; address line; uC I/O port	P48
54	51				RS_2			A10		P22			LED segment; address line; uC I/O port	P45
55	56				RS_3			A11		P23			LED segment; address line; uC I/O port	P51
56	57				RS_1			A9		P21			LED segment; address line; uC I/O port	P47
57	58				RS_5			A13		P25			LED segment; address line; uC I/O port	P50
58	59				RS_0			A8		P20			LED segment; address line; uC I/O port	P46
59	72												JTAG TDO; DOUT	
59	75												JTAG TDO; DOUT	
61	60				RS_6			A14		P26			LED segment; address line; uC I/O port	P49
62	61							OE_					RAM output-enable	
63	82							WR_		P36			RAM write-enable; uC I/O port	
65	65							CE_					XS Board RAM chip-enable	
66	66		DIPSW7						SDOUT	P16			DIP switch; codec output data; uC I/O port	P27
68	21												Uncommitted XS95 I/O pin	
69	12												Uncommitted XS95 I/O pin	
70	69		DIPSW8				KB_DATA			P31			DIP switch; keyboard serial data; uC I/O port	P28
71	70		DIPSW6						SDIN	P15			DIP switch; codec input data; uC I/O port	P26
72	77		DIPSW5						SCLK	P14			DIP switch; codec serial clock; uC I/O port	P25
74	74												Uncommitted XS95 I/O pin	
75	78				LS_3			A3					LED segment; address line	P44
76	76												Uncommitted XS95 I/O pin	
77	1												Uncommitted XS95 I/O pin	
78	2	+5V											+5V power source	
79	79				LS_4			A4					LED segment; address line	P38
80	34										PC_D7		PC parallel port output	
81	32										PC_D6		PC parallel port output	
82	82				LS_5			A5					LED segment; address line	P40
83	83				LS_6			A6					LED segment; address line	P39
84	84				LDP_			A7					LED decimal-point; address line	P37
24,67	67			SPARE_		VSYNC_				P17			Pushbutton; VGA horiz. sync.; uC I/O port	P18
	30												Serial EEPROM chip-enable	
	43												Unconnected	
	53												Unconnected	
	54	+3.3V											+3.3V/+5V power supply (400XL/4000XL)	
	55			PROGRAM									XS40 configuration control	P55
	63												Unconnected	
	64												Unconnected	

Jadual 3.5 :Sambungan antara Papan XS95 dan Papan XStend

3.2 KEPERLUAN PERISIAN

3.2.1 WEBPACK 4.2

Perisian yang digunakan untuk menjana kod-kod VHDL adalah perisian Xilinx WEBPACK 4.2. terdapat beberapa modul yang terdapat didalam perisian bergantung kepada jenis kerja yang hendak dibuat iaitu:

i) Design entry

"design entry and synthesis tools" adalah bahagian utama dalam perisian ini. Ia digunakan untuk hampir keseluruhan proses. Memilih salah satu modul keluarga iaitu Spartan, Virtex atau CPLD untuk menggunakan perisian dengan mudah. Antara kandungan yang terdapat dalam modul "Design Entry" termasuk:

- Skematik ECS untuk rekabentuk aras tertinggi (top level design) berasaskan HDL
- Sintesis XST untuk menyokong VHDL dan rekabentuk verilog
- Project Navigator, antaramuka yang terdapat didalam WEBPACK 4.2
- Simulasi VHDL dan Verilog

ii) CPLD Filter

Ia menyesuaikan pemuatan peranti dan perisian pengujian untuk merekabentuk mana-mana rekabentuk

Xilinx yang berasaskan keluarga CPLD seperti Xilinx XC9500, XC9500XL, XC9500XV dan sebagainya. Modul ini boleh digunakan untuk melaksanakan rekabentuk dari 2 sumber iaitu:

- Rekbentuk menggunakan bahasa VHDL, Verilog atau ABEL yang disokong modul 'Design Entry'.
- Netlist EDIF atau XNF yang disediakan menyediakan peralatan sokongan pihak ketiga 'Alliance Eda Tool'

Komponen yang terdapat dalam CPLD Fitter ialah:

- Penganalisa masa
- Perisian pemuat CPLD (cpldfit command yang menyokong semua keluarga CPLD)
- Model penjana simulasi masa untuk CPLD untuk menghasilkan model VHDL atau Verilog
- Penyunting kekangan

iii) Perlaksanaan Spartan

Menyediakan semua perlaksanaan peranti dan pengujian perisian yang diperlukan untuk memetakan rekabentuk kedalam keluarga Spartan II atau Spartan-IIE.

Perlaksanaan Spartan termasuk:

- Model simulasi pemasaan untuk FPGA untukmenghasilkan model VHDL atau Verilog
- Penyunting kekangan
- Penganalisa pemasaan

iv) Perlaksanaan Virtex

Menyediakan implementasi dan perisian pengujian yang diperlukan untuk memetakan rekabentuk kepada peranti Virtex-E atau Virtex-II. Perlaksanaan Vertex ini boleh diperolehi daripada 2 sumber iaitu:

- Rekabentuk menggunakan bahasa VHDL, Verilog atau ABEL yang disokong modul 'Design Entry'
- Netlist EDIF atau XNF yang disediakan menyediakan peralatan sokongan pihak ketiga 'Alliance Eda Tool'

Implementasi Vietex termasuk:

- Model simulasi pemasaan untuk FPGA untuk menghasilkan model VHDL atau Verilog
- Penyunting kekangan
- Penganalisa pemasaan

v) CPLD Programmer

Digunakan untuk keluarga CPLD seperti Xilinx XC9500, XC9500XL dan sebagainya.

Modul yang terdapat dalam CPLD Programmer termasuk:

- Perisian pengaturcara Xilinx impact
- Fail BSDL untuk peranti CPLD

vi) Pengaturcara FPGA

Digunakan untuk semua program yang menggunakan peranti FPGA dan Serial Proms.

Modul yang terdapat dalam FPGA Programmer adalah:

- Pengformat fail PROM
- Fail BSDL untuk semua peranti Xilinx dan Serial PROM

vii) StateCAD

Peralatan Xilinx StateCAD akan menjana rekabentuk VHDL, Verilog dan ABEL FSM

viii) HDL Bencher

Peralatan Xilinx HDL Bencher menjana test bench untukVHDL dan Verilog. Ia membantu merekabentuk dan menyunting stimulasi bentuk gelombang fizikal dan akan dihantar kedalam test bench untuk VHDL dan Verilog dan dimasukkan kedalam simulasi Project Navigator.

ix) ChipViewer

Ia adalah antaramuka grafik untuk melihat kesesuaian pin dan logik dan untuk masuk ke lokasi pin I/O dan untuk melihat semua perkara yang terdapat dalam CPLD untuk Xilinx.

x) CPLD Schematic Capture Libraries

Dalam perpustakaan komponen skematiknya menyediakan simbol termasuk rekabentuk skematik ECS yang disediakan menyediakan modul 'Design Entry'

xi) FPGA Schematic Capture Libraries

Dalam perpustakaan komponen skematiknya menyediakan simbol termasuk rekabentuk skematik ECS yang disediakan menyediakan modul 'Design Entry'

xii) XPower

Ia adalah peralatan untuk menganalisis yang menggunakan data rekabentuk dan peranti. Maklumat dipersembahkan dalam bentuk grafik dan format laporan ASCII. Ia juga menyokong Cool Runner XPLA3 dari keluarga CPLD dan Virtex II FPGA.

xiii) Peranti Sokongan

WEBPACK menyokong perlaksanaan untuk keluarga Xilinx yang berikut:

- XC9500XL(3.3V CPLDs)
- XC9500XV (2.5 CPLDs)
- XCR3000XL (CoolRunner XPLA3 3.3V zero-standby CPLDs)
- XC2C00 (CoolRunner-II 1.8V zero-standby CPLDs)
- XC2S00 (Spartan-II FPGAs)
- XC2S00E (Spartan-IIE FPGAs)
- XCV00E (Virtex-E FPGAs up to XCV300E)
- XCV00 (Virtex-II FPGAs up to XC2V250)
- XC9500 (5V CPLDs)

3.2.2 KEPERLUAN SISTEM

Sistem operasian:

- Windows 98, 98 SE / 2000
- Windows NE 4.0 (dengan service pack 4.6a)
- Windows Millenium

Perkakasan:

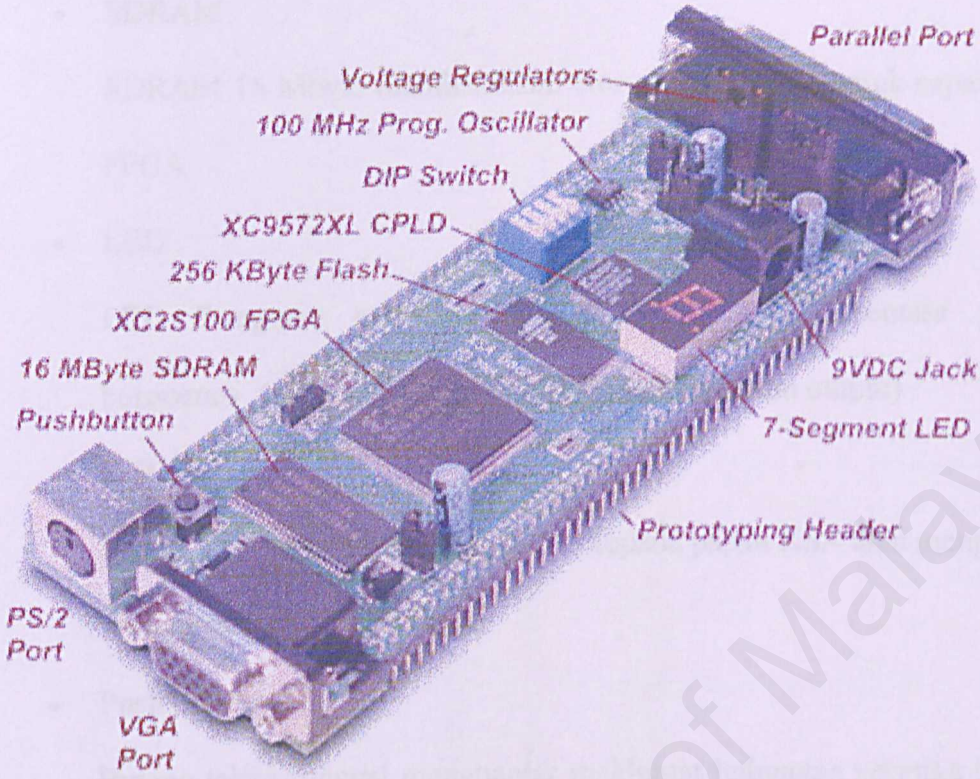
- mesin kelas IBM Pentium
- 1024 x 768 VGA Color Monitor (minimum)
- pemandu CD-ROM mematuhi ISO 9660

- port USB atau parallel untuk pengaturcaraan peranti

REKABENTUK

4.0 REKABENTUK

4.1 REKABENTUK FIZIKAL



Rajah 4.1: Rekabentuk fizikal XSA100

Ciri-ciri:

Seperti yang dapat dilihat didalam Rajah XSA-100 terdiri daripada komponen-komponen berikut:

- FPGA Spartan-II XC2S50 atau XC2S100

  Ini merupakan storan utama logik boleh program bagi Papan XSA.

- XC9572XL CLPD

  CLPD ini berfungsi untuk menguruskan antaramuka antara port selari PC dengan keseluruhan Papan XSA

- Osc

  Oscilator boleh program menjanakan jam 'master' bagi papan XSA

- Flash

Peranti Flash 256 Kbyte membekalkan storan stabil/x berubah untuk data aliran bit konfigurasi

- SDRAM

  SDRAM 16 Mbyte membekalkan storan boleh ubah untuk capaian data oleh FPGA

- LED

  LED 7-segmen membenarkan tindakbalas nyata semasa papan XSA beroperasi. (LED akan menyala untuk menunjukkan output)

- DIP Switch

  Switch DIP 4 posisi menghantar set kepada papan XSA atau mengawal alamat bit teratas bagi peranti Flash

- Push button

  Butang tekan tunggal menghantar maklumat hubungan seketika (momentary) ke FPGA

- Parallel port

  Antaramuka utama untuk menghantar aliran bit data dan konfigurasi sama ada dari atau ke papan XSA

- PS/2 port

  Papan kekunci atau tetikus boleh berantaramuka dengan papan XSA melalui port ini

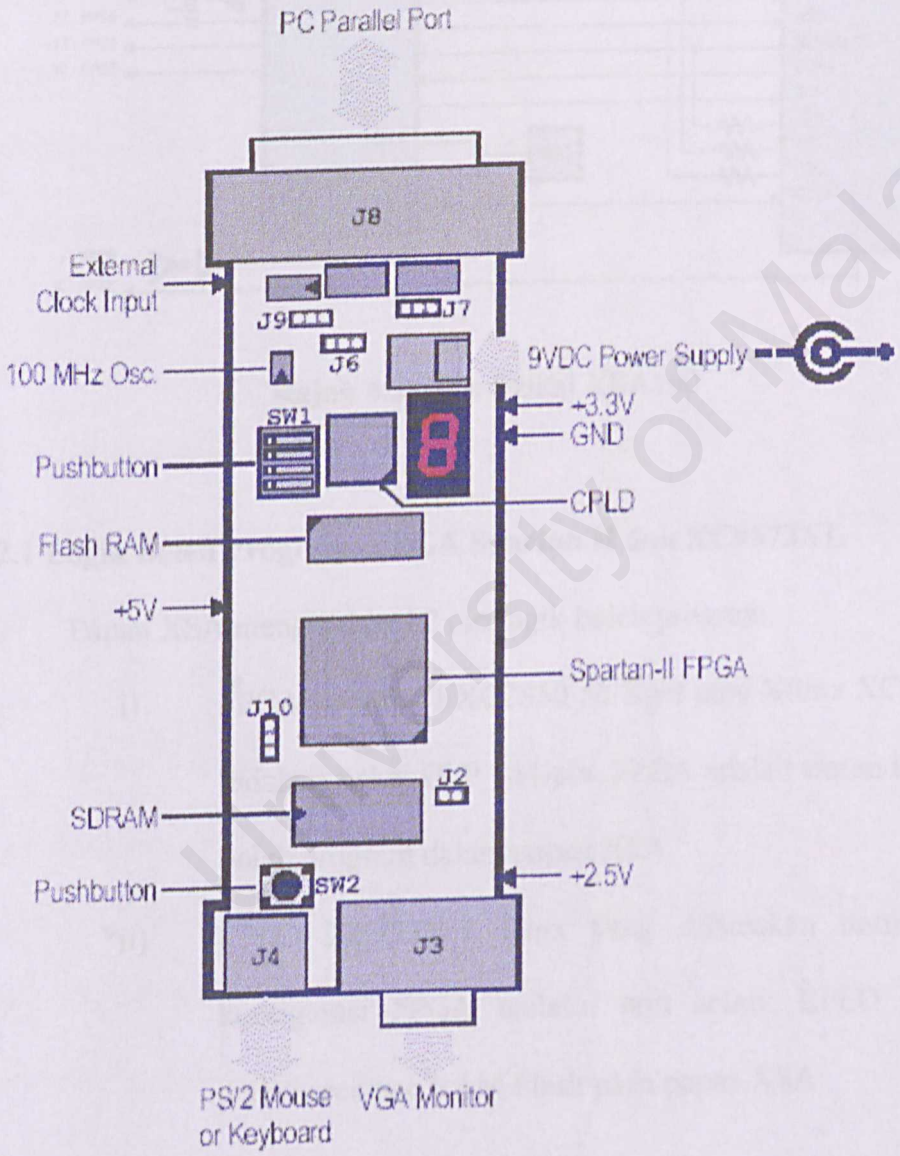
- VGA port

  Papan XSA boleh menghantar isyarat untuk gambaran grafik pada monitor VGA melalui port ini

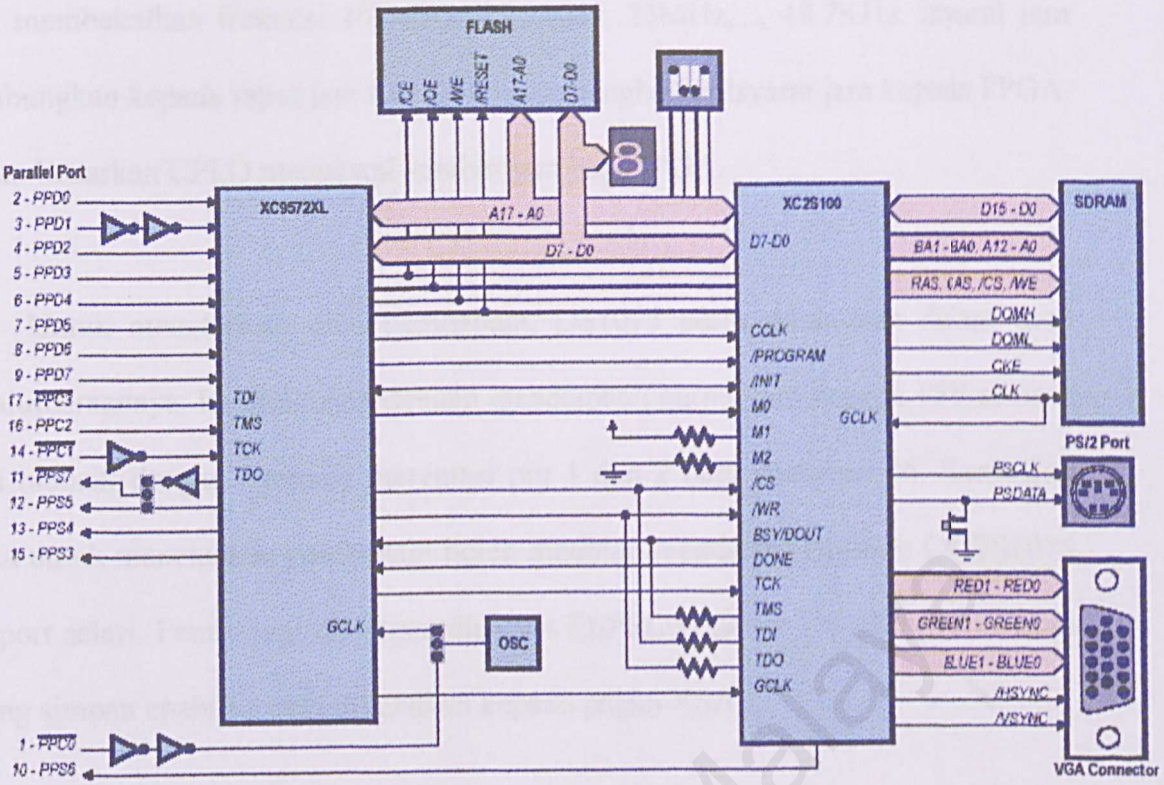
- Prototyping header

Kebanyakan daripada pin I/O FPGA disambungkan kepada 84 pin dibawah papan XSA yang dibina untuk disambungkan dengan 'breadboard' tak terpateri.

4.2 PLAN FIZIKAL



Rajah 4.2: Plan fizikal XSA100



Rajah 4.3: Plan fizikal XSA100

4.2.1 Logik Boleh Program: FPGA Spartan II dan XC9572XL

Papan XSA mengandungi 2 cip logik boleh program:

i) FPGA Spartan II XC2S50 50 Kget atau Xilinx XC2S100 100-Kget didalam pekej QFP 144-pin. FPGA adalah storan utama bagi logik boleh program dalam papan XSA.

ii) CPLD XC9572XL ilinx yang digunakan untuk menguruskan konfigurasi FPGA melalui port selari. CPLD juga mengawal pengaturcaraan RAM Flash pada papan XSA

4.2.2 Oscillator Boleh Program 100 MHz

Oscillator boleh program DS1075 Dallas menyediakan isyarat jam bagi FPGA dan CPLD. DS1075 mempunyai frekunsi maksimum 100MHz yang dibahagikan

untuk membekalkan frekunsi 100MHz, 33.3MHz, 25MHz,..., 48.7KHz. isyarat jam disambungkan kepada input jam CPLD. CPLD menghantar isyarat jam kepada FPGA. ini membenarkan CPLD mengawal sumber jam bagi FPGA.

Untuk menetapkan nilai pembahagi, DS1075 perlu diletakkan dalam mod pengaturcaraannya. Ini dilakukan dengan menetapkan output jam kepada +5V semasa kuasa menaik dengan simpang merentasi pin 1 dan 2 bagi pelompat J6. Kemudian arahan untuk menetapkan pembahagi boleh dihantar kepada pin kawalan C0 DS1075 bagi port selari. Pembahagi disimpan didalam EEPROM dalam DS1075 jadi ia akan diulang simpan apabila kuasa dikenakan kepada papan XSA.

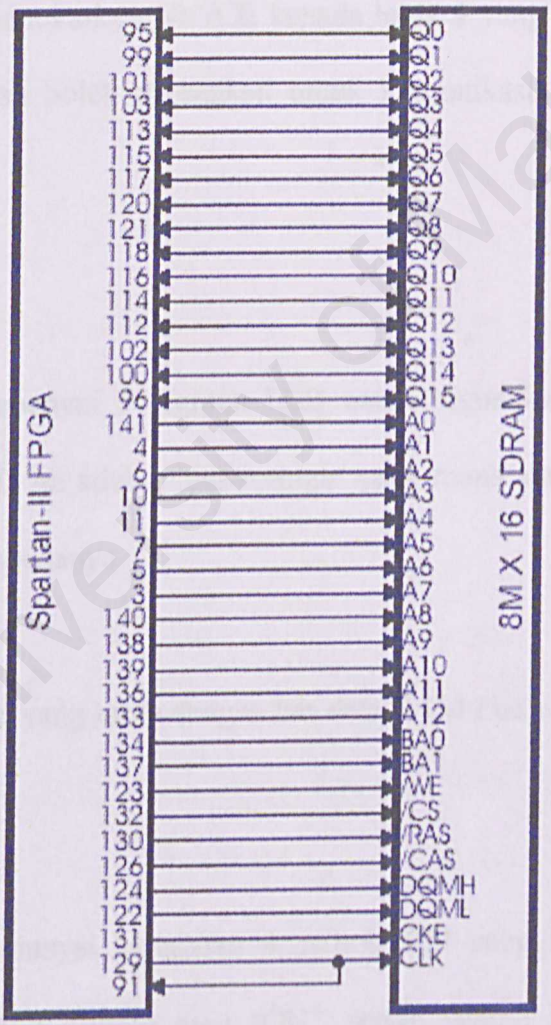
Pemirauan (shunt) pada pelompat J6 mestilah merentasi pin 2 dan 3 supaya oscillator mengeluarkan isyarat jam semasa 'power-up'. Isyarat jam memasuki input jam CPLD yang ditetapkan. Kemudian CPLD boleh mengeluarkan isyarat jam kepada input jam FPGA yang dihubunginya.

Untuk memperolehi nilai frekunsi yang tepat atau menyegerakkan perlitaran XSA dengan sistem dalaman, isyarat jam boleh dimasukkan melalui pin 64 bagi kepala pemprototaipan. Jam luaran ini menggantikan sumber jam dalaman 100MHz dalam oscillator DS1075. namun begitu perisian GXSSETCLK diperlukan untuk membolehkan masukan jam luaran DS1075.

Isyarat jam juga boleh dikenakan secara terus kepada 2 masukan jam bersambung pada FPGA melalui pin pada kepala pemprototaipan.

4.2.3 DRAM Segerak 16 MByte

SDRAM HY57V281620AT-H Hynix dengan storan 16 MByte (8M x 16) disambungkan ke FPGA seperti yang ditunjukkan dibawah. Perhatikan bahawa isyarat jam ke SDRAM juga disambungkan semula kepada masukan jam FPGA. Ini menjadikannya lebih mudah untuk segerakkan operasi dalaman FPGA dengan operasi SDRAM.



Rajah 4.4: Sambungan SDRAM dan FPGA

4.2.4 RAM Flash 256 KByte

RAM Flash AT49F002 Atmel dengan storan 256 KByte (256K x 8) disambungkan kepada FPGA dan CPLD.

CPLD dan FPGA mempunyai capaian kepada RAM Flash. Biasanya CPLD akan memprogramkan Flash dengan data merentasi port selari. Jika data tersebut adalah aliran bit konfigurasi FPGA, maka CPLD boleh dikonfigurasikan untuk memprogramkan FPGA dengan aliran bit daro Flash apabila XSA dihidupkan. Selepas kuasa dibekalkan, FPGA boleh baca dan/atau tulis pada Flash. Flash boleh di'disable'kan dengan menukarkan pin /CE kepada logik 1 yang mana line I/O yang bersambung kepada Flash boleh digunakan untuk komunikasi tujuan am diantara FPGA dan CPLD.

4.2.5 LED 7 Segmen

Papan XSA mempunyai 7 segmen LED untuk digunakan oleh FPGA atau CPLD. Segmen bagi LED ini adalah 'active-high' yang mana akan menyala apabila logik 'high' dikenakan kepadanya.

LED berkongsi pin yang sama dengan bas data RAM Flash 8 bit.

4.2.6 Switch DIP 4 Posisi

Papan XSA mampunyai simpanan 4 switch DIP yang boleh dicapai dari CPLD dan FPGA. Apabila ditutup atau 'ON', setiap switch menolak pin yang bersambungan pada FPGA dan CPLD ke bumi. Jika tidak, pin akan ditinggikan melalui rintangan apabila switch dibuka atau 'OFF".

Switch DIP juga berkongsi pin yang sama untuk 4 bit teratas RAM Flash alamat bas. Jika RAM Flash diprogramkan dengan beberapa aliran bit FPGA, DIP switch boleh digunakan untuk memilih mana-mana aliran bit yang akan dimuatkan kedalam FPGA oleh CPLD semasa kuasa dibekalkan.

4.2.7 Port PS/2

Papan XSA menyediakan antaramuka jenis PS/2 (sambungan mini-DIN J4) untuk papan kekunci atau tetikus. FPGA menerima 2 isyarat daripada antaramuka PS/2 iaitu isyarat jam dan aliran data sesiri yang disegerakkan dengan pinggir menurun pada isyarat jam.

4.2.8 Butang Tekan

Papan XSA mempunyai butang tekan tunggal yang berkongsi pin FPGA bersambung ke line data pada port PS/2. Butang tekan mengenakan aras rendah kepada FPGA apabila ditekan dan perintang menekan pin ke aras tinggi apabila butang tidak ditekan.

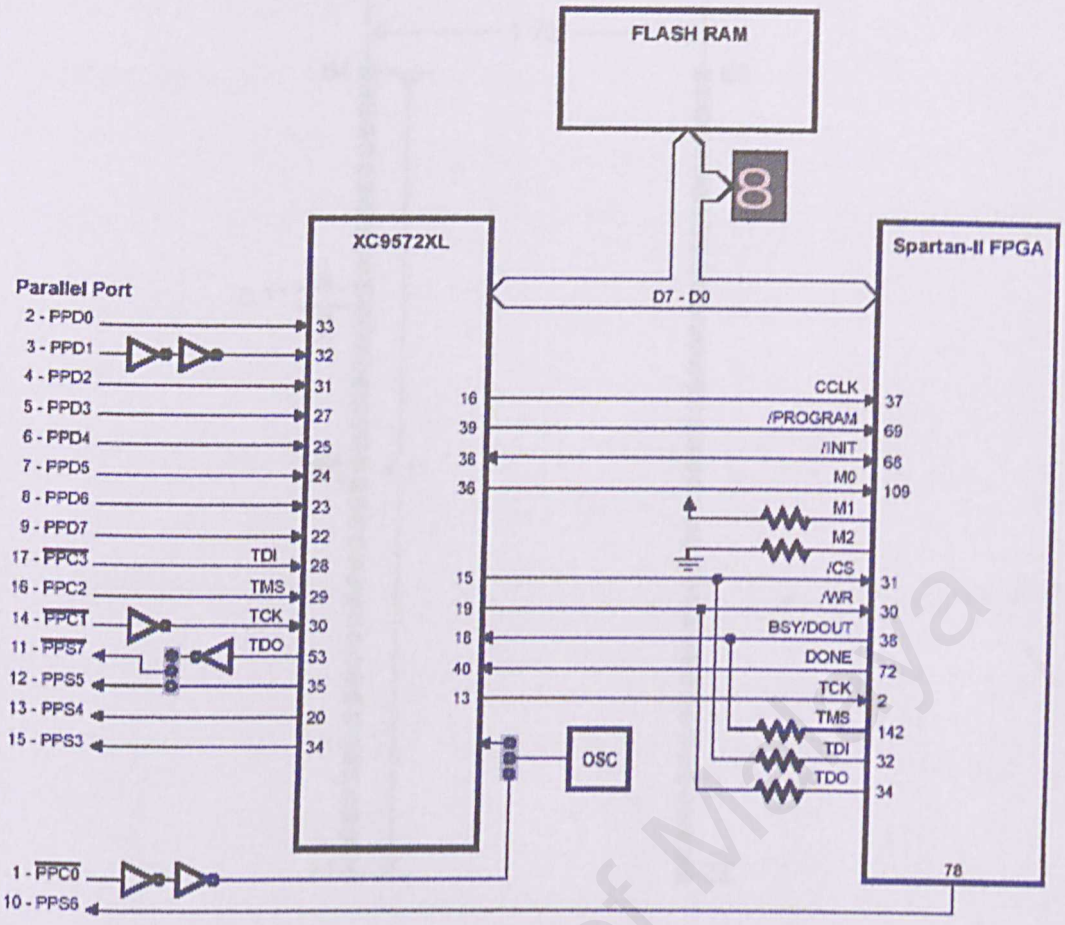
4.2.9 Antaramuka Monitor VGA

FPGA boleh menjanakan isyarat video untuk dipaparkan pada monitor VGA. Semasa FPGA menjanakan isyarat VGA, FPGA mengeluarkan maklumat 2 bit bagi warna merah, hijau dan biru ke DAC 'resistor-ladder' ringkas. Output DAC dihantarkan ke input RGB bagi monitor VGA bersama dengan denyutan segerak horizontal dan vertical (/HSYNC, /VSYNC) dari FPGA.

4.2.10 Antaramuka Port Selari

Port selari adalah antaramuka utama untuk komunikasi dengan papan XSA. Line kawalan C0 bersambung terus ke oscillator DS1075 dan digunakan untuk menetapkan pembahagi seperti yang dijelaskan sebelum ini manakala line status S6 disambungkan terus ke FPGA untuk digunakan sebagai line komunikasi dari FPGA balik ke PC. CPLD mengawal lebihan 15 line aktif antaramuka ke port selari. 11 daripada line aktif port selari dihubungkan kepada pin I/O tujuan am pada CPLD.

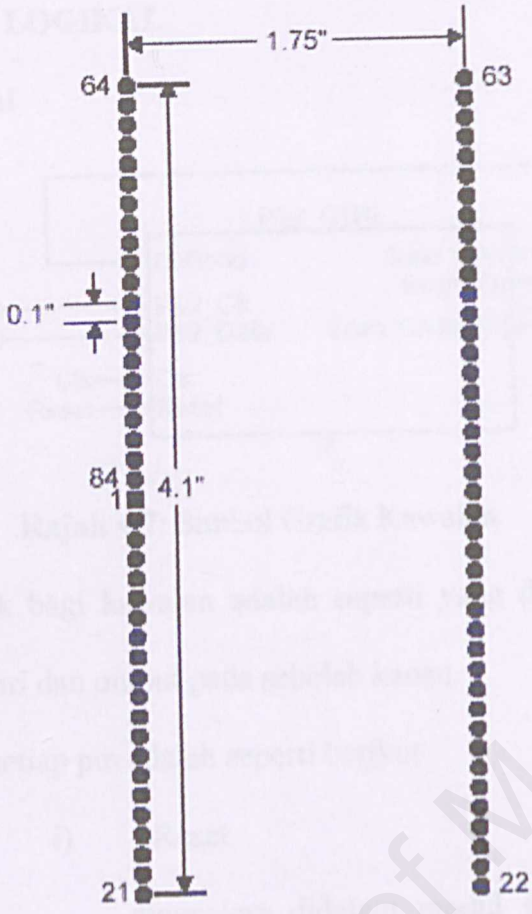
Setelah FPGA Spartan II dikonfigurasikan denganaliran bit dan pin DONE naik, CPLD dipesongkan kedalam mod yang menghubungkan data port selari dan pin status ke FPGA. Ini membolehkan data dihantar ke FPGA melalui line data port selari sambil menerima data dari FPGA melalui line status. Sambungan antara FPGA dan port selari adalah seperti yang ditunjukkan disebelah:



Rajah 4.5: Sambungan FPGA dan port selari

4.2.11 Kepala Pemprototaipan

Pin FPGA boleh dicapai melalui pin 84 kepala pemprototaipan yang terletak dibawah papan XSA. Pin 1 bagi kepala (ditandakan dengan segiempat) terletak ditengah-tengah pinggir sebelah kiri papan dan lebihan 83 pin lagi disusun melawan arah jam disekelilingnya. Dimensi fizikal kepala pemprototaipan dan susunan pin ditunjukkan disebelah:

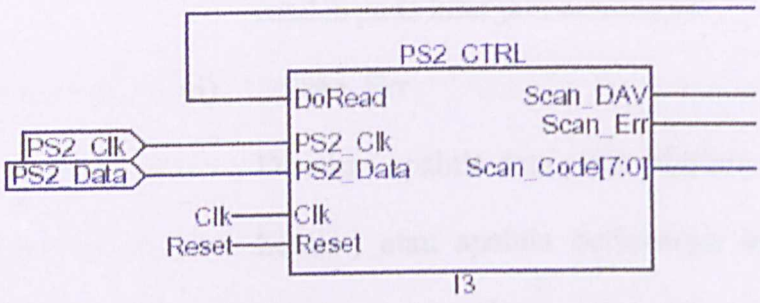


Rajah 4.6: Susunan pin kepala pemprototaipan

Subset bagi 144 pin FPGA bersambung dengan kepala pemprototaipan. Bilangan pin FPGA yang bersambung kepada pin kepala yang diberikan dicetak disebelah pin kepala pada papan. ini memudahkan untuk pencarian pin FPGA yang diberikan apabila sambungan ke sistem luar dilakukan. Semasa sebahagian daripada pin FPGA digunakan untuk menyokong fungsian papan XSA, ia juga boleh digunakan untuk antaramuka kepada sistem luar melalui kepala pemprototaipan. Pin FPGA boleh dikelaskan kepada banyak kategori.

4.3 REKABENTUK LOGIKAL

4.3.1 Diskripsi Modul



Rajah 4.7: Simbol Grafik Kawalan

Simbol grafik bagi kawalan adalah seperti yang ditunjukkan diatas dengan input pada sebelah kiri dan output pada sebelah kanan.

Penerangan setiap pin adalah seperti berikut:

i) Reset

digunakan didalam modul sebagai asynchronous dan reset global

ii) Clk

tapak sumber pemasaan untuk semua flip flop (rekabentuk segerak sepenuhnya)

iii) PS2_Clk dan PS2_Data

2 line isyarat bagi antaramuka PS/2. Merupakan input dalam modul ini.

iv) DoRead

input yang dimulakan oleh pengguna apabila kod keluar Data dibaca.

v) Scan_DAV

Menjadi '1' apabila word diterima. Kekal dengan nilai tersebut sehingga DoRead ditegaskan dan kemudian rendah pada kitar jam berikutnya.

vi) Scan_Err

Disetkan apabila data yang diterima tidak betul (pariti, henti...) atau apabila berlakunya overflow (pengguna tidak membaca kod sebelumnya dalam masa yang ditetapkan). Bendera ini akan dihapuskan secara automatik apabila penerimaan karektor berikutnya bermula.

vii) Scan_Code

(8 bit) jumlah data word yang diterima. Nilainya kekal stabil sehingga word berikutnya diterima.

4.3.2 Implementasi

Apabila host hanya membaca dan tidak menghantar data, fungsi asas yang perlu disempurnakan adalah mengsampelkan line Data selepas setiap pinggir menurun line. Ini menentukan word yang dihasilkan oleh 11 bit termasuklah bit Mula (=0), 8 bit kod (dimulakan dengan LSB), pariti bit ganjil dan bit Tamat (=1).

Kita telah melaksanakan turas kecil pada line PS2_Clk: disampelkan pada kadar jam sistem dan tunggu sehingga 8 sampel pengesahan berturut-turut sebelum

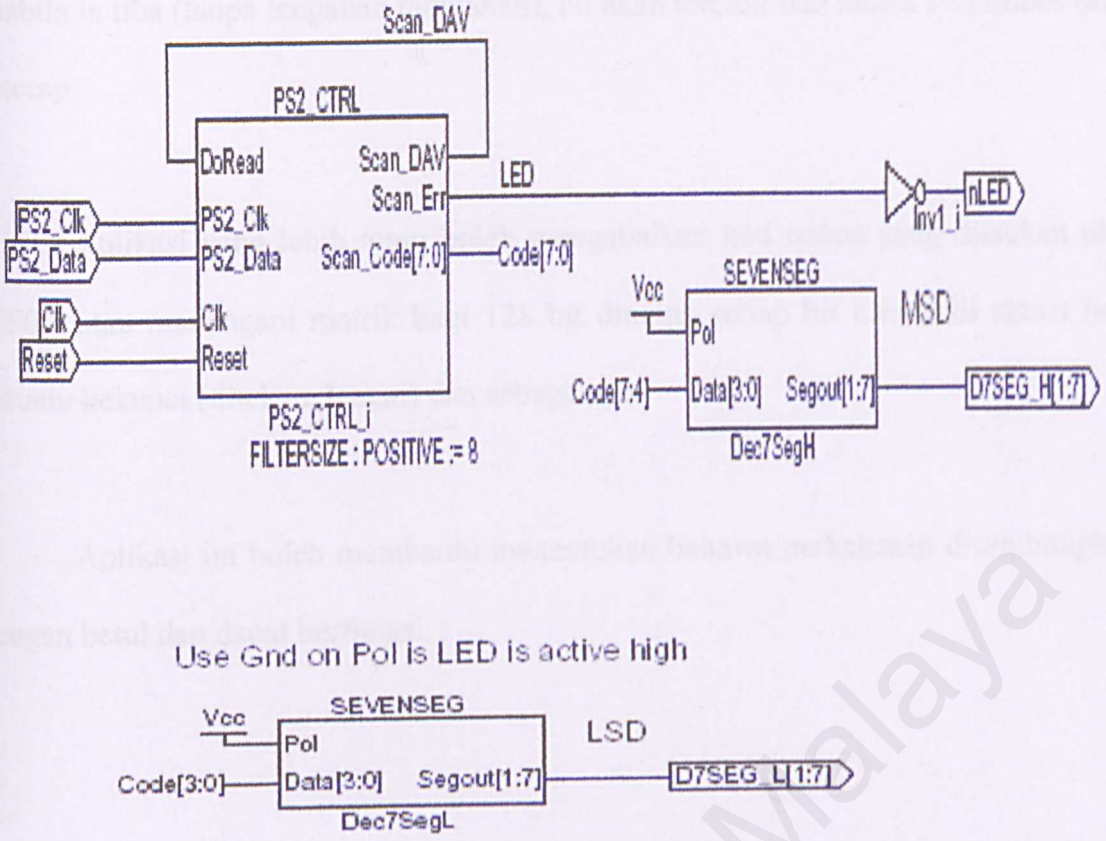
menerima nilai tersebut. Untuk nilai yang melebihi 8, implementasi yang berbeza dan lebih berkesan untuk penurasan ini (berdasarkan pembilang) patut digunakan.

Apabila perkakasan disambungkan, adalah penting untuk memastikan bahawa terdapat perintang Pull-Up pada mana-mana bahagian line PS/2 (Clock dan Data) dan nilai rintangan tersebut adalah mencukupi. Nilai rintangan yang terlalu besar boleh membentuk pinggir menaik yang lemah yang akan menmberikan masalah kepada FPGA. Didapati bahawa papan kekunci telahpun dilengkapi dengan perintang pull-up.

4.3.3 Aplikasi Ringkas

Untuk aplikasi yang paling ringkas, atau apabila modul pengguna mengutip Data yang tiba tanpa lengahan, adalah mungkin untuk menyambungkan Scan_DAV kepada DoRead.

Oleh kerana kod output kekal stabil, adalah tidak mustahil untuk membina aplikasi ringkas seperti berikut:



Rajah 4.8: Aplikasi Ringkas

Jika papan kekunci disambungkan ke PS2_Clk dan PS2_Data dan dibekalkan kuasa sebanyak +5V, penekan atau melepaskan kekunci sepatutnya memaparkan kod imbas yang terlibat. Pengesahan isyarat ini boleh dilakukan menggunakan oscilloscope untuk menetukan slop bagi pinggir menaik.

Penyalaan LED penunjuk papan kekunci (CapsLock, NumLock dll) memerlukan kawalan PS/2 untuk menghantar data kepada papan kekunci. Ciri ini (komunikasi 'host to device') tidak terdapat pada kawalan ringkas ini. Ia memerlukan I/O 2 arah bagi PS2_Data dan PS2_Clk dan juga logik kawalan yang tepat.

Apabila kekunci ditekan, papan kekunci akan mengeluarkan 2 kod berturutan: x"F0" diikuti dengan kod imbas kekunci. Oleh kerana kod diterjemahkan ke LED

apabila ia tiba (tanpa lengahan tambahan), F0 akan tercicir dan hanya kod imbas akan diserap.

Aplikasi yang lebih maju boleh mengabaikan kod imbas yang diisukan oleh x"F0", atau menangani matrik bagi 128 bit dimana setiap bit mewakili status bagi sesuatu kekunci (ditekan, lengai) dan sebagainya.

Aplikasi ini boleh membantu menentukan bahawa perkakasan disambungkan dengan betul dan dapat berfungsi.

PERLAKSANAAN

5.0 PERLAKSANAAN

Secara umumnya kod pengaturcaraan menggunakan perpustakaan ieee 1164.all kerana perpustakaannya menyokong kod pengaturcaraan yang digunakan.

i. GENERIC

Digunakan dalam pengaturcaraan ini untuk menetapkan saiz Filter.

ii. PEMETAAN PORT (PORT MAP)

Kod pemetaan ini bertujuan supaya data input yang dimasukkan akan melalui semua langkah pengaturcaraan pada komponen yang ingin diuji dan menghasilkan output yang sepatutnya. Dalam pengaturcaraan ini akan terdapat 8 port dimana 5 port masukan dan 3 port keluaran.

iii. CLK (UNTUK KITAR MASA)

Digunakan untuk mengatur ambilan bit secara sesiri untuk langkah yang seterusnya.

5.1 MODUL PENGKODAN

```
Entity ctrl1 is
    generic (FilterSize : positive := 8);
    port (  Reset : in std_logic;
            Clk : in std_logic;
            PS2_Clk: in std_logic;
            PS2_Data: in std_logic;
            DoRead: in std_logic;

            Scan_DAV: out std_logic;
            Scan_Err: out std_logic;
```

```
        Scan_Code: out std_logic_vector (7 downto 0);
    end ctrl1;
```

Port-port yang terlibat dalam penghantaran dan penerimaan isyarat.

```
    architecture Behavioral of ctrl1 is
        signal PS2_Datr : std_logic;
        - isyarat yang dihantar semasa kekunci pada papan kekunci ditekan
        subtype Filter_t is std_logic_vector (FilterSize-1 downto 0);
        signal Filter : Filter_t;
        signal Fall_Clk : std_logic;
        signal Bit_Cnt : unsigned (3 downto 0);
        - isyarat yang digunakan untuk mengira jumlah bit yang telah dihantar
```

dan juga berguna untuk mengesan jika berlaku 'framming error'.

```
        signal Parity : std_logic;
        - boleh bernilai '1' atau '0' bergantung kepada nilai yang disetkan.
```

Berfungsi untuk mengesan kesahan data yang diterima dan memaklumkan ralat parity.

```
        signal Scan_DAVi : std_logic;
        signal S_Reg : std_logic_vector (8 downto 0);
        signal PS2_Clk_f : std_logic;
        type State_t is (Idle, Shifting);
        - state yang terlibat semasa aturcara dilarikan. Data hanya akan
```

dihantar apabila system berada dalam state Shifting.

```
        signal State : State_t;
```

Isyarat-isyarat yang dihantar dan diterima semasa pengaturcaraan dilaksanakan.

```
begin
Scan_DAV <= Scan_DAVi;
process (Clk, Reset)
begin
        if Reset = '1' then
          PS2_Datr <= '0';
          PS2_Clk_f <= '0';
          Filter <= (others => '0');
          Fall_Clk <= '0';
        elsif rising_edge (Clk) then
          PS2_Datr <= PS2_Data and PS2_Data;
          Fall_Clk <= '0';
          Filter <= (PS2_Clk and PS2_Clk) & Filter (Filter'high downto 1);
          if Filter = Filter_t'(others => '0') then
            PS2_Clk_f <=1;
          elsif Filter = Filter_t'(others => '0') then
              PS2_Clk_f <= '0';
          if PS2_Clk_f <= '1' then
           Fall_Clk <= '1';
          end if;
        end if;
        end if;
```

```
end process;

process (Clk, Reset)
begin
        if Reset = '1' then
          State <= Idle;
          Bit_Cnt <= (others => '0');
          S_Reg <= (others => '0');
          Scan_Code <= (others => '0');
          Parity <= '0';
          Scan_DAVi <= '0';
          Scan_Err <= '0';
```

Bit data tidak akan dihantar semasa Reset bernilai '1' kerana pada masa tersebut state berada dalam keadaan Idle.

```
        elsif rising_edge (Clk) then
            if DoRead = '1' then
              Scan_DAVi <= '0';
            end if;

        case State is
            when Idle =>
            Parity <= '0';
            Bit_Cnt <= (others => '0');
            if Fall_Clk = '1' and PS2_Datr ='0' then
              Scan_Err <= '0';
```

```
            State <= Shifting;
        end if;
```

State akan berubah dari Idle ke Shifting apabila syarat-syarat berikut dipenuhi:

Clk = '1'

PS2_Clk = '1'

PS2_Data = '0'.

```
        elsif Fall_Clk = '1' then
            Bit_Cnt <= Bit_Cnt + 1;
            S_Reg <= PS2_Datr & S_Reg (S_Reg'high downto 1);
            Parity <= Parity xor PS2_Datr;
        end if;
```

Aturcara ini akan berulang sehingga Bit_Cnt bernilai 9 menunjukkan 1 frame bit telah selesai dihantar. Ralat parity akan diuji disini.

```
        when others =>
            State <= Idle;
    end case;
  end if;
end process;
```

Setelah suatu siri bit telah diterima, state akan kembali ke state Idle dan menunggu sehingga syarat untuk menghantar bit seterusnya dipenuhi.

```
end Behavioral;
```

PENGUJIAN SISTEM

6.0 PENGUJIAN SISTEM

Dalam bahagian ini proses pengujian system dilakukan keatas modul. 'Testbench' adalah satu bahagian kod pengaturcaraan yang berbeza yang digunakan dengan memasukkan nilai-nilai data input kedalam komponen yang ingin diuji.

Nilai data input yang akan dimasukkan ke dalam 'testbench' akan dipetakan kedalam komponen yang dikehandaki. Output untuk setiap komponen atau modul yang diuji akan dikeluarkan melalui proses simulasi. Nilai output ini akan dibandingkan dengan nilai teori untuk melihat adakah pengaturcaraan yang dihasilkan menepati nilai-nilai teori tersebut.

6.1 PENGUJIAN MODUL

6.1.1 PENGUJIAN MODUL TANPA RALAT

Modul diuji dengan nilai data input seperti yang terdapat dalam kod pengaturcaraan dibawah.

```
architecture stimulus of TESTBNCH is

component CTRL1 is
    port ( Reset : in std_logic;
           Clk : in std_logic;
           PS2_Clk : in std_logic;
           PS2_Data : in std_logic;
           DoRead : in std_logic;
```

```
        Scan_DAV : out std_logic;
        Scan_Err : out std_logic;
        Scan_Code : out std_logic_vector (7 downto 0);
end component;

constant PERIOD: time := 25ns;
signal Reset : in std_logic;
signal Clk : in std_logic;
signal PS2_Clk : in std_logic;
signal PS2_Data : in std_logic;
signal DoRead : in std_logic;
signal Scan_DAV : out std_logic;
signal Scan_Err : out std_logic;
signal Scan_Code : out std_logic_vector (7 downto 0);
signal done: Boolean := false;
signal Clock_cycle : natural := 0;

begin
        DUT: CTRL1 port map (
        Reset, Clk, PS2_Clk, PS2_Data, DoRead, Scan_DAV,Scan_Err,
Scan_Code);

CLOCK1 : process
        variable clktmp : std_ulogic := '0';
begin
```

```
        wait for (PERIOD/2);
        clktmp := not clktmp;
        clk <= clktmp;
    end process CLOCK1;

CLOCK : process
    begin
        Clock_cycle <= Clock_cycle + 1;
        PS2_Clk <= '0';
        wait for 200ns;
        PS2_Clk <= '1';
        wait for 200ns;
    end process;

STIMULUS1: process
    begin
        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;
```

```
Reset <= '1';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data yang dihantar setakat ini tidak akan dibaca oleh system kerana system berada dalam keadaan Idle.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*32;
```

Syarat untuk menukarkan state dari Idle ke Shifting dipenuhi, daya yang masuk berikutnya akan dikira sebagai input.

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;

Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang pertama, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kedua, bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*32;
```

Data bit yang ketiga dan keempat, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kelima bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*48;
```

Data bit yang keenam, ketujuh dan kelapan bernilai '1'. Saiz frame 8 bit telah dipenuhi, input yang berikutnya adalah untuk mengenalpasti parity.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

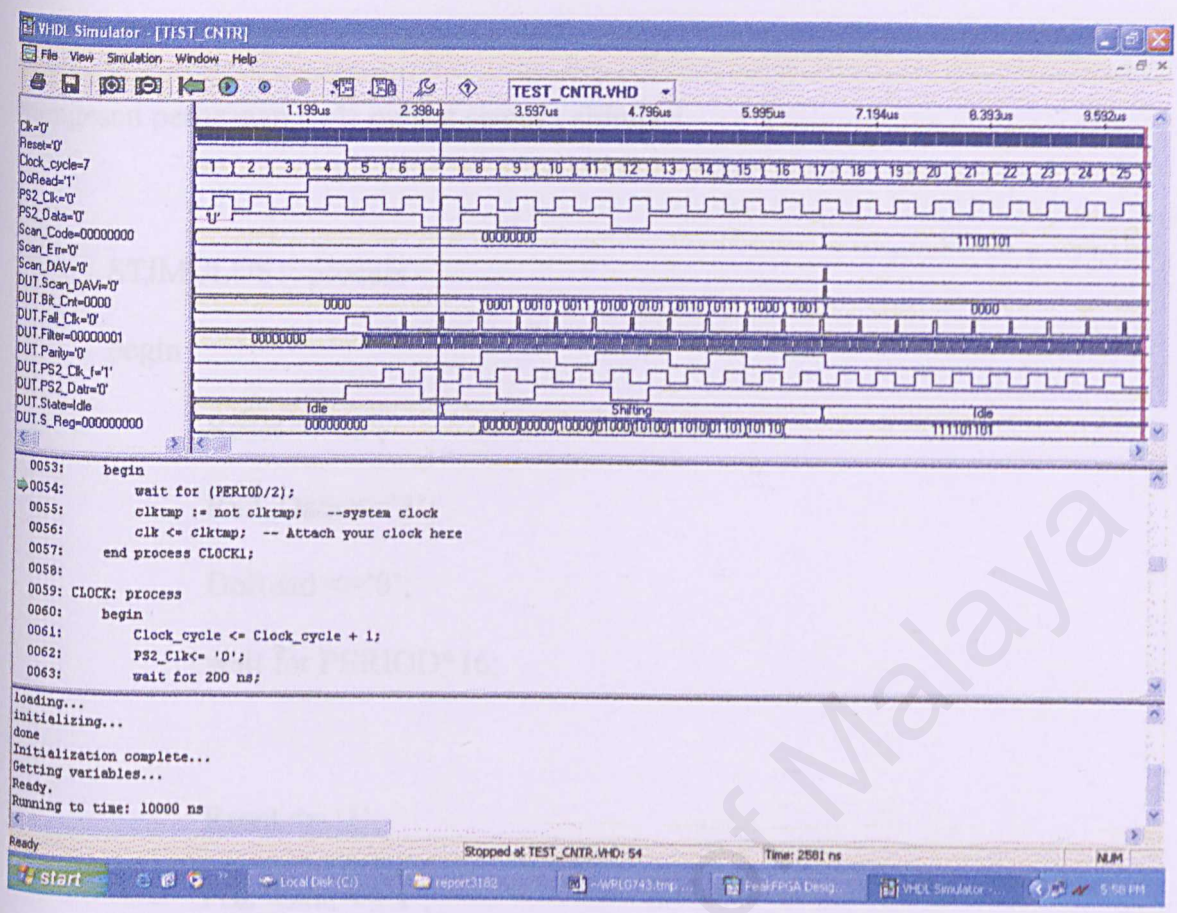
Bit parity ditetapkan sebagai '1' bermakna output akhir mestilah mempunyai bilangan bit '1' yang ganjil.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Bit tamat.

```
done <= true;
wait;
end process STIMULUS1;
end stimulus;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan modul kemudian disimulasikan.



Rajah 6.1 Simulasi modul tanpa ralat

6.1.2 PENGUJIAN MODUL DENGAN RALAT PARITI

Pengujian untuk modul dengan memasukkan ralat parity bagi mengesan perbezaan pada output semasa simulasi.

```
STIMULUS1: process
begin
        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='1';
        wait for PERIOD*16;
```

Data yang dihantar setakat ini tidak akan dibaca oleh system kerana system berada dalam keadaan Idle.

```
        Reset <= '0';
        PS2_Data <='1';
```

DoRead <='1';

wait for PERIOD*32;

Syarat untuk menukarkan state dari Idle ke Shifting dipenuhi, daya yang masuk berikutnya akan dikira sebagai input.

Reset <= '0';

PS2_Data <='0';

DoRead <='1';

wait for PERIOD*16;

Reset <= '0';

PS2_Data <='1';

DoRead <='1';

wait for PERIOD*16;

Data bit yang pertama, bernilai '1'

Reset <= '0';

PS2_Data <='0';

DoRead <='1';

wait for PERIOD*16;

Data bit yang kedua, bernilai '0'

Reset <= '0';

PS2_Data <='1';

DoRead <='1';

wait for PERIOD*32;

Data bit yang ketiga dan keempat, bernilai '1'

Reset <= '0';

PS2_Data <='0';

DoRead <='1';

wait for PERIOD*16;

Data bit yang kelima bernilai '0'

Reset <= '0';

PS2_Data <='1';

DoRead <='1';

wait for PERIOD*48;

Data bit yang keenam, ketujuh dan kelapan bernilai '1'. Saiz frame 8 bit telah dipenuhi, input yang berikutnya adalah untuk mengenalpasti parity.

Reset <= '0';

PS2_Data <='0'; ← perbezaan disini

DoRead <='1';

wait for PERIOD*16;

Bit parity ditetapkan sebagai '1' bermakna output akhir mestilah mempunyai bilangan bit '1' yang ganjil.

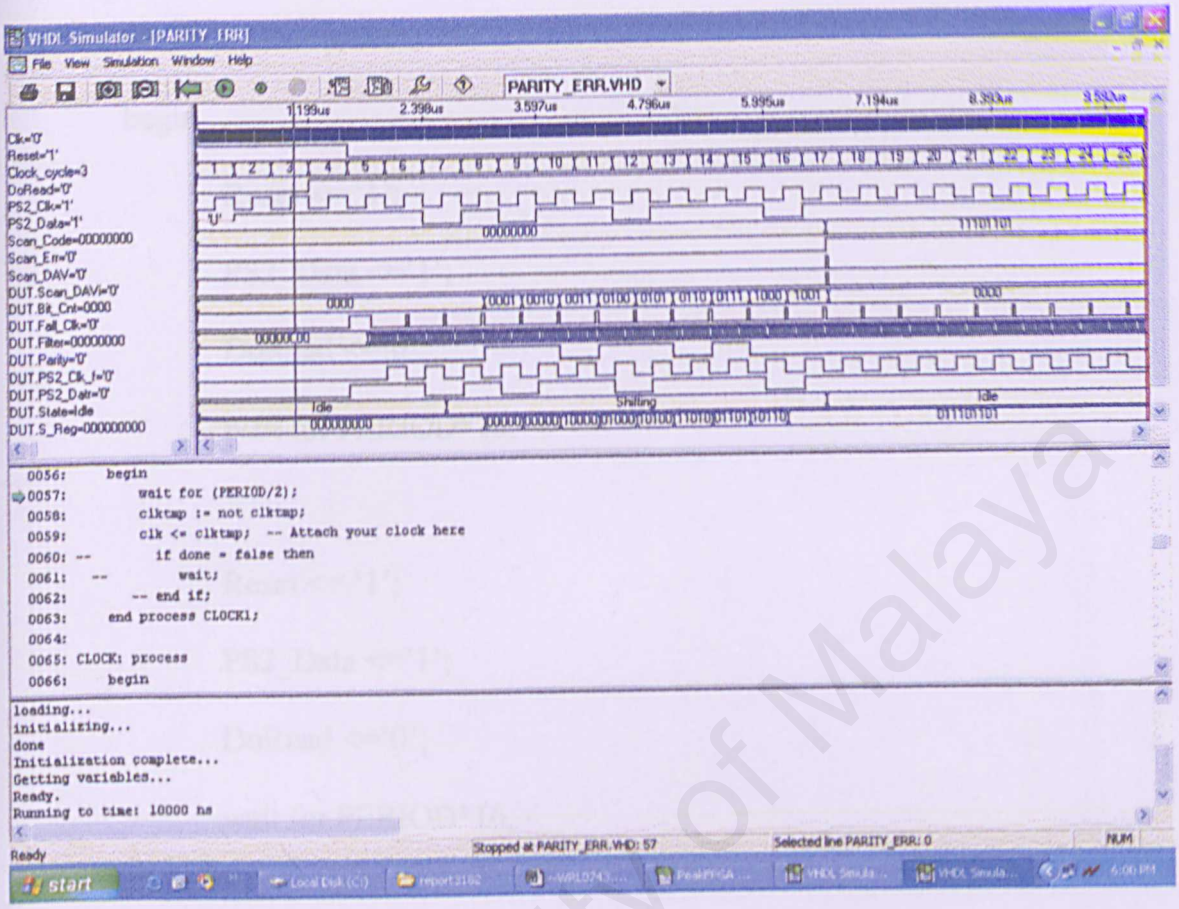
Reset <= '0';

PS2_Data <='1';

```
        DoRead <='1';
        wait for PERIOD*16;
    Bit tamat.

    done <= true;
    wait;
end process STIMULUS1;
end stimulus;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan modul kemudian disimulasikan.



Rajah 6.2 Simulasi modul dengan ralat pariti

6.1.3 PENGUJIAN MODUL DENGAN RALAT FRAME

```
STIMULUS1: process
begin
        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='1';
        wait for PERIOD*16;
```

Data yang dihantar setakat ini tidak akan dibaca oleh system kerana system berada dalam keadaan Idle.

```
        Reset <= '0';
        PS2_Data <='1';
        DoRead <='1';
        wait for PERIOD*32;
```

Syarat untuk menukarkan state dari Idle ke Shifting dipenuhi, daya yang masuk berikutnya akan dikira sebagai input.

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang pertama, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kedua, bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*32;
```

Data bit yang ketiga dan keempat, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kelima bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*48;
```

Data bit yang keenam, ketujuh dan kelapan bernilai '1'. Saiz frame 8 bit telah dipenuhi, input yang berikutnya adalah untuk mengenalpasti parity.

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

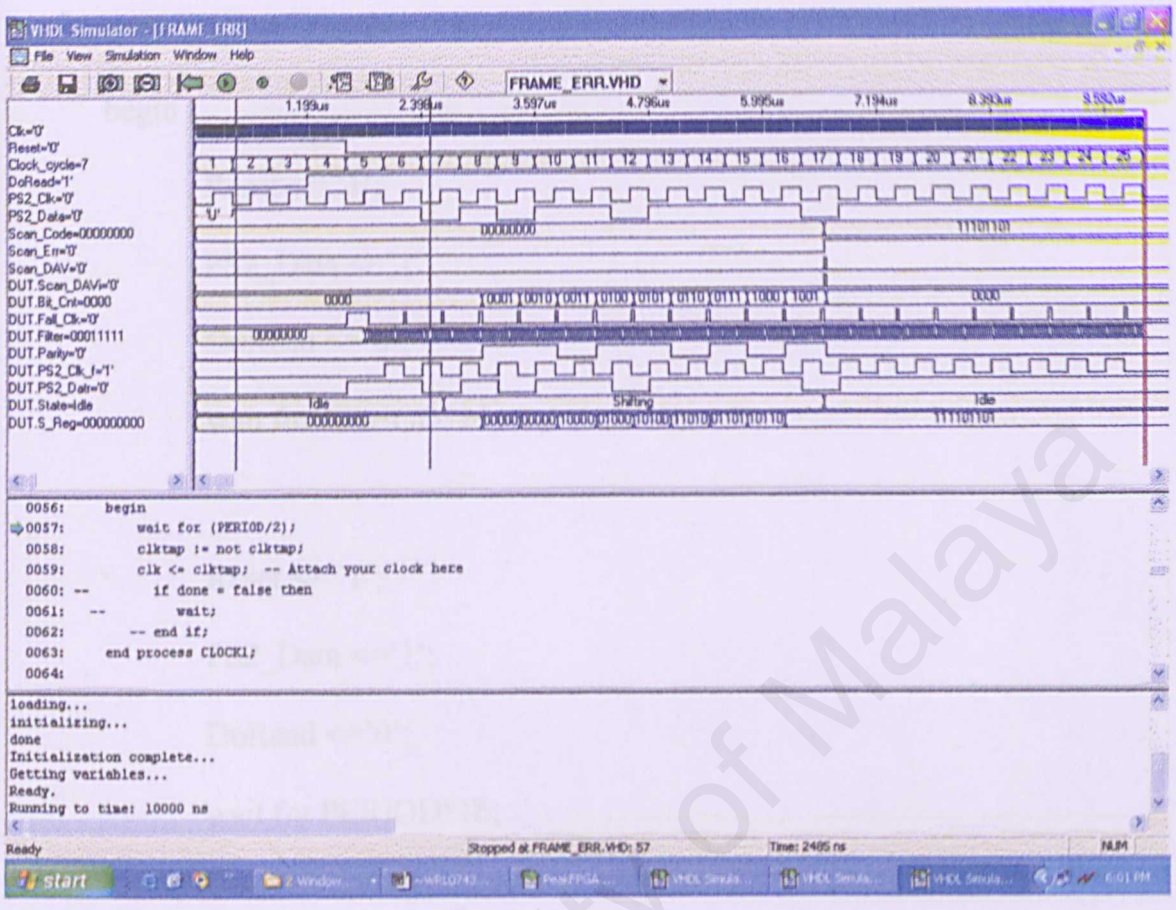
Bit parity ditetapkan sebagai '1' bermakna output akhir mestilah mempunyai bilangan bit '1' yang ganjil.

```
Reset <= '0';
PS2_Data <='0';          ← perbezaan disini
DoRead <='1';
wait for PERIOD*16;
```

Bit tamat.

```
        done <= true;
        wait;
end process STIMULUS1;
end stimulus;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan modul kemudian disimulasikan.



Rajah 6.3 Simulasi modul dengan ralat frame

6.1.4 PENGUJIAN MODUL DENGAN BANYAK DATA

```
STIMULUS1: process
begin
        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='0';
        wait for PERIOD*16;

        Reset <= '1';
        PS2_Data <='1';
        DoRead <='1';
        wait for PERIOD*16;
```

Data yang dihantar setakat ini tidak akan dibaca oleh system kerana system berada dalam keadaan Idle.

```
        Reset <= '0';
        PS2_Data <='1';
        DoRead <='1';
        wait for PERIOD*32;
```

Syarat untuk menukarkan state dari Idle ke Shifting dipenuhi, daya yang masuk berikutnya akan dikira sebagai input.

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang pertama, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kedua, bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*32;
```

Data bit yang ketiga dan keempat, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kelima bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*48;
```

Data bit yang keenam, ketujuh dan kelapan bernilai '1'. Saiz frame 8 bit telah dipenuhi, input yang berikutnya adalah untuk mengenal pasti parity.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Bit parity ditetapkan sebagai '1' bermakna output akhir mestilah mempunyai bilangan bit '1' yang ganjil.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Bit data pertama tamat.

Bit data baru dimulakan.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang pertama, bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*32;
```

Data bit yang kedua dan ketiga bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang keempat bernilai '1'

```
Reset <= '0';
PS2_Data <='0';
DoRead <='1';
wait for PERIOD*16;
```

Data bit yang kelima bernilai '0'

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*48;
```

Data bit yang keenam, ketujuh dan kelapan bernilai '1'. Saiz frame 8 bit telah dipenuhi, input yang berikutnya adalah untuk mengenal pasti parity.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

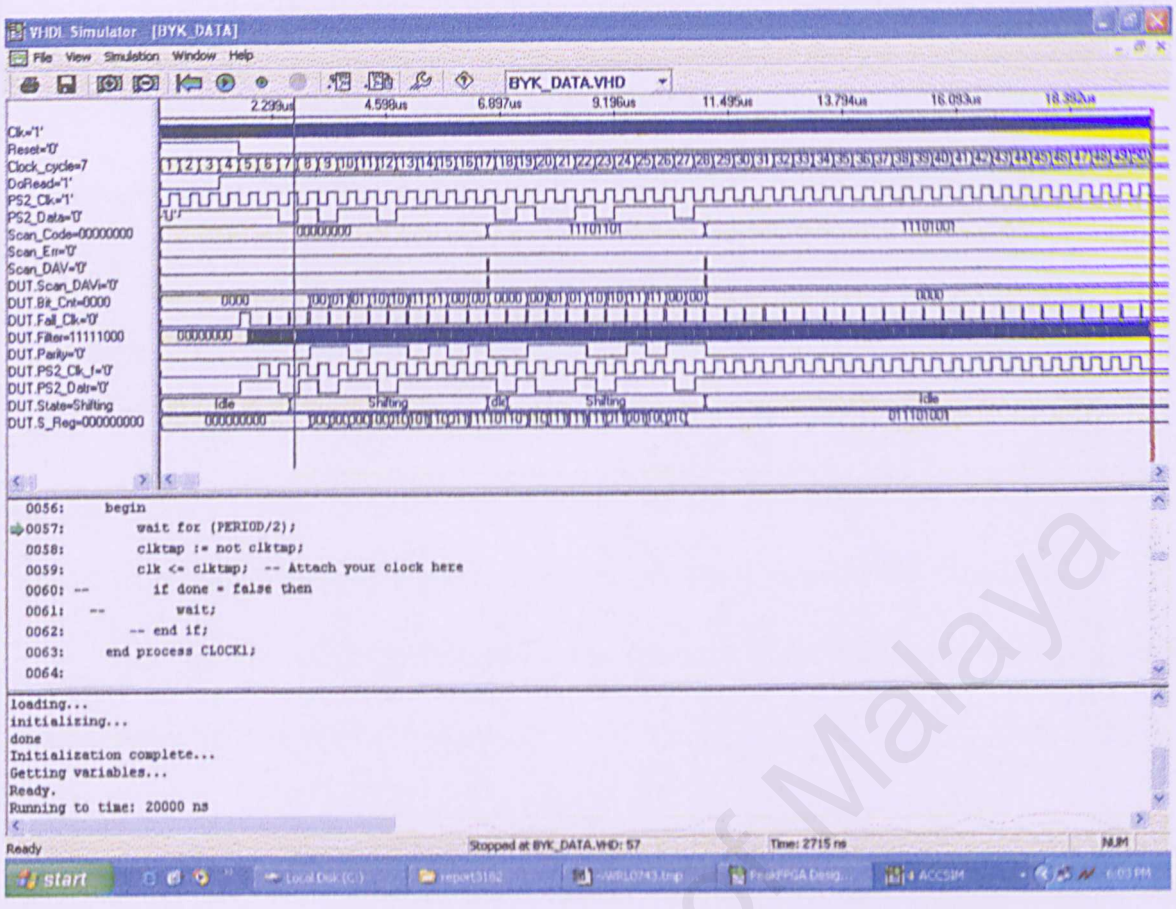
Bit parity ditetapkan sebagai '1' bermakna output akhir mestilah mempunyai bilangan bit '1' yang ganjil.

```
Reset <= '0';
PS2_Data <='1';
DoRead <='1';
wait for PERIOD*16;
```

Bit data yang kedua tamat.

```
done <= true;
wait;
end process STIMULUS1;
end stimulus;
```

Dari kod pengaturcaraan ini, ia akan dihubungkan dengan kod pengaturcaraan modul kemudian disimulasikan.



Rajah 6.2 Simulasi modul dengan banyak data

PERBINCANGAN

7.0 PERBINCANGAN

Dalam bab ini, saya akan membincangkan masalah-masalah yang dihadapi semasa melaksanakan aturcara ini.

Perisian yang hendak digunakan sebagai platform pembnaan system ini telah diubah daripada penggunaan perisian WEB PACK 4.20 kepada Xilinx ISE 6. Ini adalah kerana perisian terdahulu tidak menyediakan kemudahan mensimulasikan kod pengaturcaraan yang telah dibuat. Namun begitu Xilinx ISE 6 didapati terlalu kompleks dan agak sukar difahami. Akhirnya perisian PEAK FPGA dipilih.

Bab ini juga akan membincangkan tentang perbezaan yang terdapat pada output dan bagaimana membacanya.

7.1 OUTPUT BAGI MODUL TANPA RALAT

Apa yang perlu ditekankan disini adalah bit tidak akan dihantar selagi system berada dalam keadaan Idle. Apabila state Shifting dicapai, iaitu apabila Clk = 1, PS2_Clk = 1 dan PS2_Data = 0 bit akan mula dihantar. Oleh kerana pengaturcaraan ini menggunakan protocol UART, jujukan bit dihantar secara sesiri.

Sepanjang penghantaran bit dilakukan, nilai Bit_Cnt akan meningkat dari 0 ke 9 menunjukkan satu siri bit telah selesai dihantarkan. Ini akan mengaktifkan Scan_DAV kepada '1' bagi menunjukkan satu siri bit telah diterima. Ini juga akan mengaktifkan Reset dan menukarkan state kembali ke Idle apabila DoRead bernilai '1'.

7.2 OUTPUT BAGI MODUL DENGAN RALAT PARITI

Seperti modul sebelum ini, bit hanya akan dihantar pada state Shifting. Apa yang akan saya bincangkan disini adalah bagaimana membaca ralat parity daripada graf output.

Dapat diperhatikan pergerakan bit adalah sama seperti output sebelum ini, apa yang berbeza disini adalah Scan_Err diaktifkan sebaik saja Scan_DAV aktif. Scan_Err berubah kepada '1' kerana system telah mengesan terdapatnya ralat pariti semasa bit dihantarkan.

Daripada pengatucaraan yang ditulis sebelum ini, nilai bit parity telah ditetapkan sebagai '1' bermakna siri bit yang dihantarkan mestilah mempunyai bilangan '1' yang ganjil. Namun begitu didapati pada S_Reg bilangan '1' yang terhasil adalah genap menunjukkan berlakunya ralat parity.

7.3 OUTPUT BAGI MODUL DENGAN RALAT FRAME

Bagi output ini Scan_Err diaktifkan kerana ralat frame dikesan. Ralat frame berlaku apabila kadar penghantaran bit tidak sama dengan kadar bacaan.

7.4 OUTPUT BAGI MODUL DENGAN BANYAK DATA

Output ini ditunjukkan bertujuan untuk memberi gambaran bagaimana bentuk graf sumulasi apabila lebih dari 1 siri bit dihantarkan.

Seperti sebelum ini, data dihantarkan semasa dalam state Shifting. Apabila data tersebut selesai dihantarkan, Scan_DAV akan aktif menunjukkan siri bit tersebut telah diterima dan Reset akan menukarkan state kepada Idle apabila DoRead bernilai '1'. Oleh kerana masih terdapat data lain yang perlu dihantar, system perlu menunggu 1 kitar jam untuk state kembali ke Shifting dan memulakan penghantaran data yang baru. Scan_DAV akan diaktifkan sebaik sahaja data tersebut selesai dihantar.

RUJUKAN

RUJUKAN

1. Digital Fundamentals With VHDL, Floyd,Thomas L, Prentice Hall
2. Analysis And Design of Digital System With VHDL, Dewey, Allen M, Boston: RWS Publication CO.
3. VHDL: Analysis and Modelling of Digital Systems, Navabi, Zainalabedin, McGraw Hill
4. A VHDL Primer, Bhasker, Jayaram, Prentice Hall
5. VHDL, Douglas L. Perry, McGraw Hill
6. www.xess.com
7. www.xilinx.com
8. www.mcamafia.de
9. www.ssv-embedded.de
10. www.cs.bris_ac.uk
11. www.vhdl-online.de
12. www.alse-fr.com
13. www.usb.org
14. www.ee.cedcc.psu.edu
15. http://b2.cs.kent.edu
16. http://govschl.ndsu.nodak.edu
17. http://panda.cs.ndsu.nodak.edu
18. http://networktechinc.com