

MODUL I

Pengenalan Teknologi FPGA

I. Tujuan

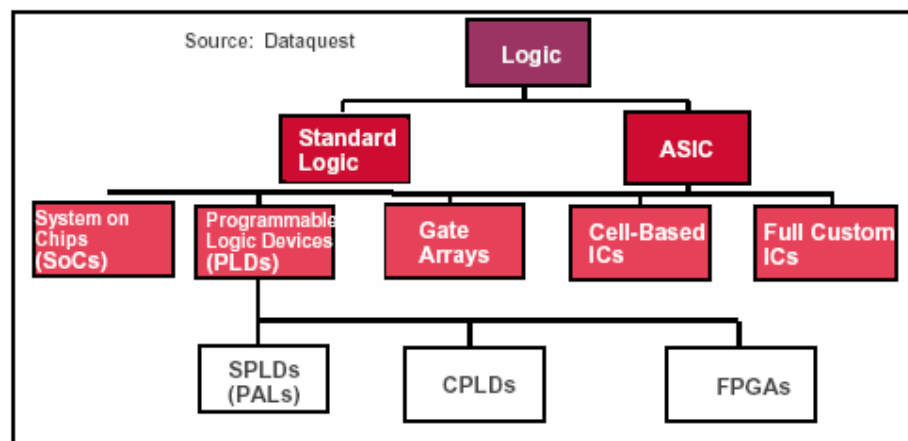
Pada Percobaan ini praktikan akan dikenalkan dengan salah satu teknologi IC VLSI yaitu Teknologi FPGA (Field Programmable gate Array) disini praktikan akan mempelajari tentang bagaimana cara mengembangkan sistem pada IC FPGA dan memahami keuntungan penggunaan teknologi FPGA . Karena percobaan kali ini adalah percobaan pertama maka praktikum akan difokuskan pada pengenalan terhadap proses yang terjadi didalam internal IC FPGA ketika diprogram serta penggunaan Development board XSA-50 dan XSTEND BOARD sekaligus pemrogramannya menggunakan software ISE Webpack.

II. Kompetensi

Setelah menyelesaikan percobaan ini diharapkan praktikan mempunyai kemampuan untuk :

- Menjelaskan arsitektur dasar IC FPGA
- Memahami keuntungan penggunaan teknologi FPGA
- Mengembangkan sistem sederhana pada FPGA dengan menggunakan software ISE Webpack dengan pendekatan berbasis schematic
- Membuat rangkaian sederhana secara schematic , mensynthesis, mensimulasi, implementasi dan mengupload file *.bit pada Development Board XSA-50.

III. Teori Dasar



FPGA merupakan kepanjangan dari *Field Programmable Gate Array* yaitu programmable device besar yang tersusun atas modul-modul logika independen yang

dapat di konfigurasi oleh pemakai yang di hubungkan melalui kanal-kanal routing yang dapat di program.

Field Programmable Gate Array (FPGA) mempunyai kelebihan sebagai berikut :

- Dikonfigurasi oleh *End User*
- Tidak memerlukan proses Fabrikasi
- Tersedia solusi yang mendukung chip customized VLSI
 - Mampu menimplementasikan logic circuit, instant manufacturing, very-low cost prototype
 - Pemrograman yang singkat untuk fungsi dan kemampuan yang setara dengan ASIC

FPGA ini sendiri juga merupakan IC tipe HDL yang mana pemakai dapat mendesain hardware sesuai yang diinginkan di dalam IC FPGA. Sedangkan perkembangan FPGA pada saat ini berlangsung dengan cepat dan dewasa ini terdapat bermacam-macam keluarga *FPGA* dengan kebutuhan perancangan dan perangkat perancangan (design tools) yang berbeda. Xilinx merupakan salah satu perusahaan yang memproduksi *FPGA* disamping perusahaan lain yang juga memproduksi *FPGA* seperti ACTEL dan PLESSEY Semiconductor. Xilinx sendiri memproduksi beberapa jenis *FPGA* yaitu VIRTEX, SPARTAN, XC3000, XC4000 dan XC5000.

Arsitektur FPGA

Secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga elemen utama yaitu *Input/Output Blok (IOB)*, *Configurable Logic Block (CLB)* dan *Interkoneksi*.

Configurable Logic Blocks :

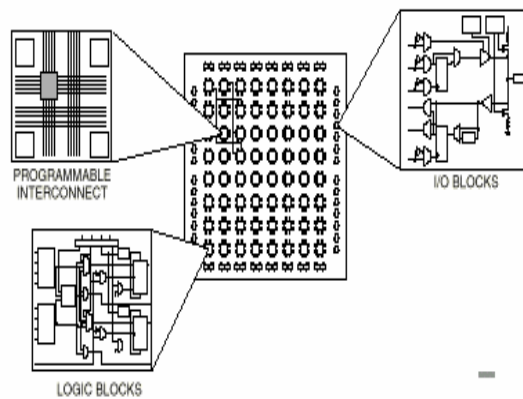
- Look up table based complex structure
- Implement the sequential circuit

Programmable Interconnect :

- Berisi wire segments dan programmable switches
- Menghubungkan anata Configurable Logic Blocks yang berbeda

Input/output block :

- Sebagai interface antara external package pin dari device dan internal user logic

Gambar I.3.1 Arsitektur *FPGA*

Fungsi logika dan interkoneksi *FPGA* ditentukan oleh data yang tersimpan pada sel memori statik internal. Ada beberapa cara untuk membuat data konfigurasi ke dalam sel memori ini, baik yang dilakukan secara otomatis pada waktu catu daya diberikan maupun dengan membaca konfigurasi data dari *eksternal Serial* atau *byte Paralel PROM (mode master)* atau Data dapat dituliskan pada *FPGA* dari *eksternal device (mode slave dan peripheral)*.

Proses Implementasi *FPGA*

FPGAs diimplementasikan dengan *efficient CAD systems*

Design Entry dapat dilakukan dengan cara

Schematic capture program,

VHDL / Verilog

Logic Optimization

Memodifikasi Boolean Expression untuk mengoptimalkan penggunaan

Logic berbasis area dan kecepatan atau kombinasi keduanya

Technology Mapping

Transformasi dari Boolean Expression atau VHDL kedalam bentuk circuit pada *FPGA* logic blocks

Area optimization – meminimalkan penggunaan block

Delay optimization – meminimalkan jumlag stage pada critical path

Placement

After mapping the circuit into logic blocks placement algorithms are used to place each block in the FPGA's array

Typically these minimize the total length of interconnect required for placement

Routing software

Mengatur FPGA's wire segments dan menentukan programmable switches untuk menghubungkan antar FPGA blocks

Memastikan 100% connections telah terbentuk

Meminimalkan propagation delay pada time-critical connections

Programming unit

Mengkonfigurasi chip setelah placement & routing step tuntas

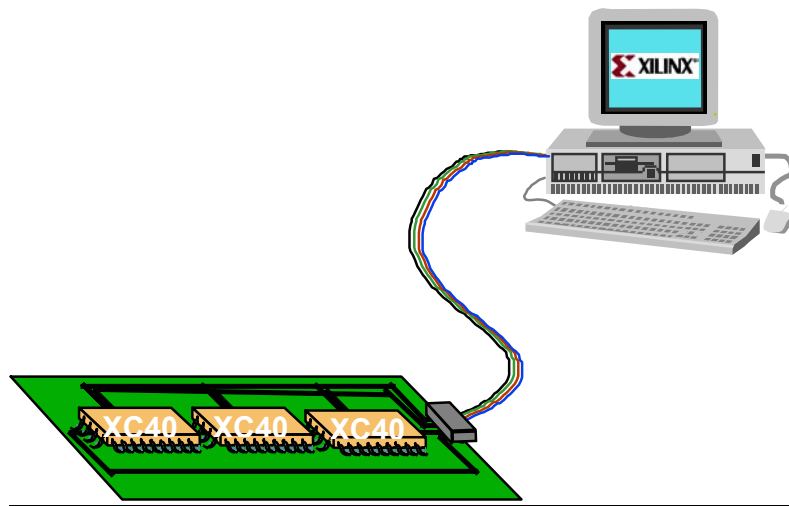
Keseluruhan proses memakan waktu antara beberapa menit sampai beberapa jam

Programmable Logic - Design Flow

Design Entry dengan cara schematic, ABEL, VHDL, dan/atau Verilog.

Implementation yaitu Placement & Routing dan bitstream generation. Juga, analyze timing, view layout,

Download meng-upload bitstream ke Hardware (FPGA chip)



Aplikasi FPGA

Prototyping

- Ideal untuk aplikasi pembuatan prototype

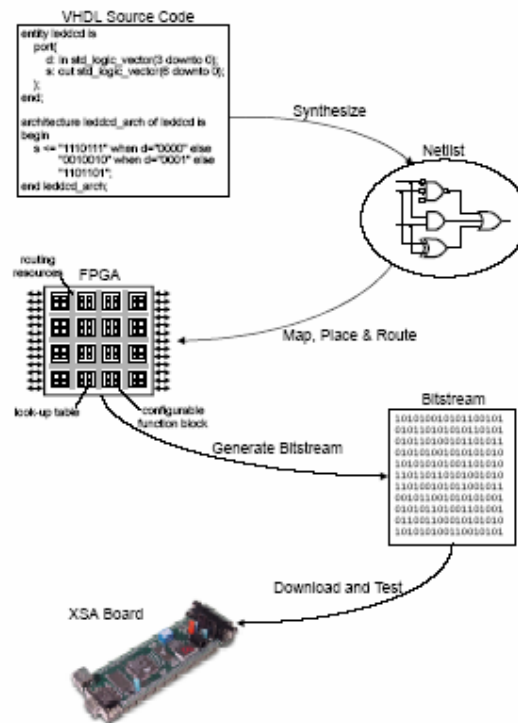
On-site re-configuration of hardware

- Untuk aplikasi hardware yang bisa dikonfigurasi ulang secara cepat

FPGA-based computation engines

FPGA sebagai processor aplikasi komputasi

Pemrograman IC FPGA



Gambar I.3.2 Proses Pemrograman IC FPGA

IV. Peralatan


1. 1 set PC yang dilengkapi dengan software ISE WebPack versi 6.1 atau lebih serta software ModelSim.
2. 1 development board XSA-50
3. 1 power-supply +9V
4. 1 kabel data

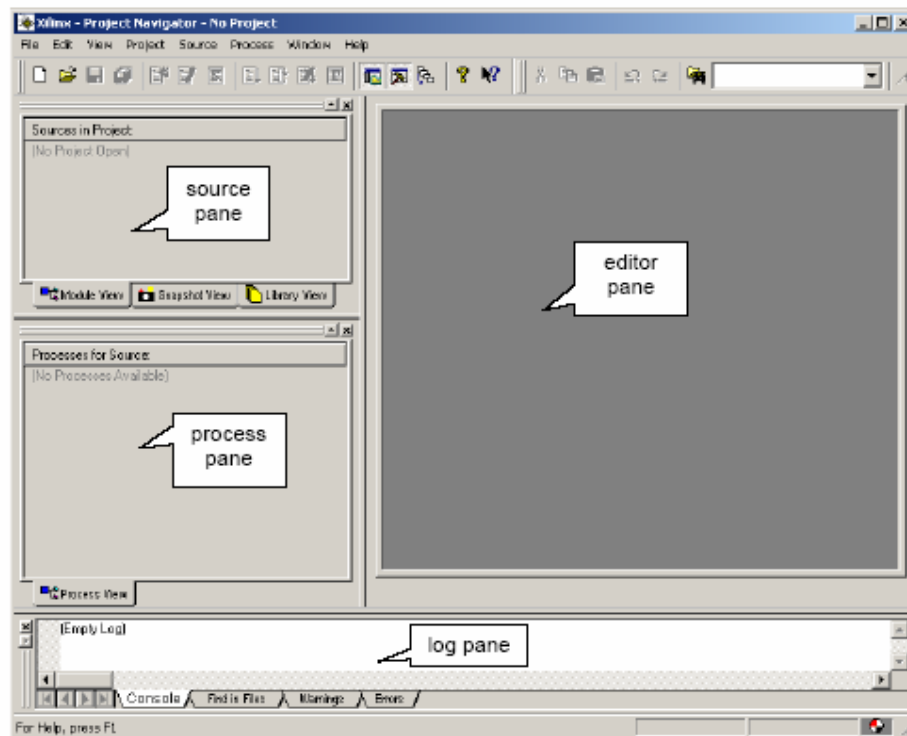
V. Prosedur Praktikum

5.0 Gambaran Disain

Pada percobaan ini sebagai contoh akan dibuat dua macam rancangan rangkaian, yang pertama adalah membuat counter sinkron 3 bit dari beberapa Flip-Flop dengan input clock dari push button serta menampilkan output counter pada bar led pada XSTEND Board. Rangkaian yang kedua adalah dengan membuat symbol baru dari rangkaian pertama kemudian memanfaatkannya bersama decoder 3to8.

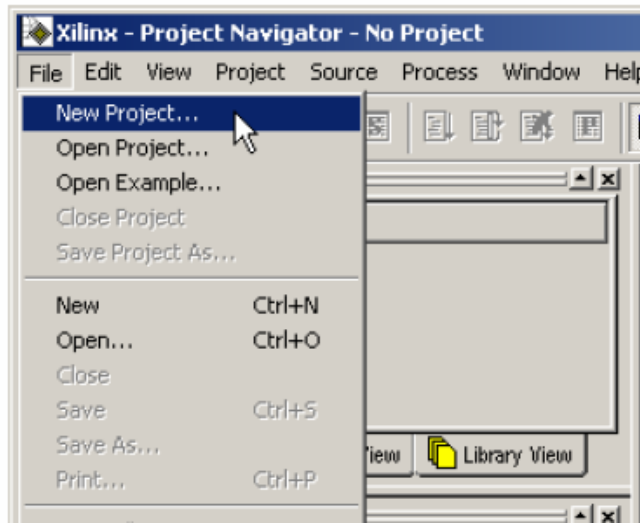
5.1 Rangkaian Counter sinkron

1. Jalankan software ISE WebPACK dengan mengklik ganda icon  pada desktop. Maka akan muncul jendela kosong seperti gambar I.5.1



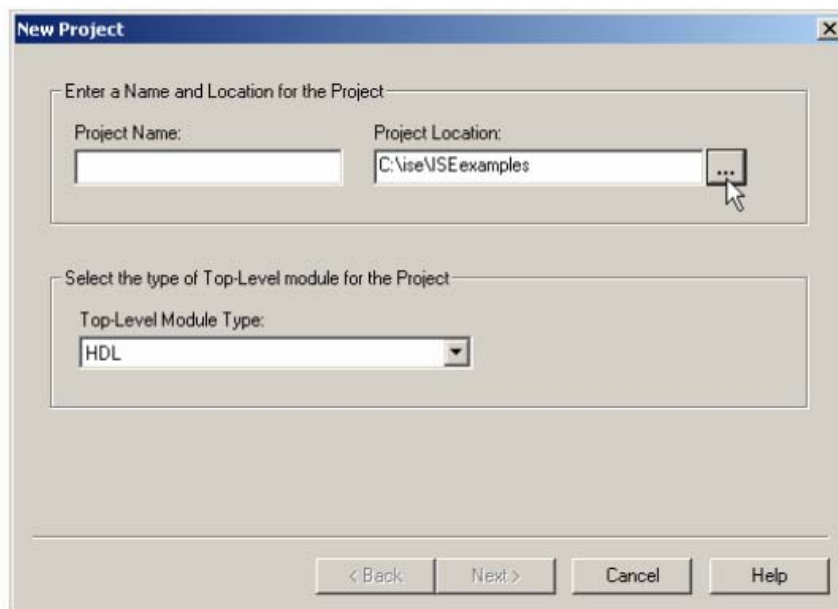
Gambar I.5.1 Jendela ISE WebPACK Project navigator

2. Buat Project baru dengan memilih **File** → **New Project** pada menu bar



Gambar I.5.2 Membuat Project baru

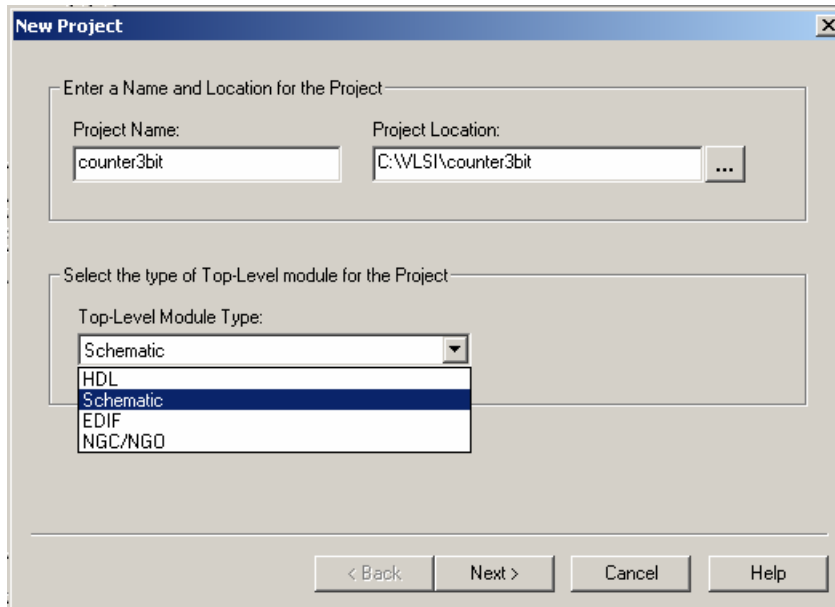
3. Klik tombol dengan tiga titik untuk memilih folder tempat untuk menyimpan project yang akan dibuat.



Gambar I.5.3 Mengatur lokasi project

4. Letakkan project pada direktori **E:\VLSI** kemudian klik tombol OK.

5. Beri nama project dengan nama **counter3bit**, secara otomatis pada Project Location akan ditambahkan nama direktori yang sama. Pada list Top-Level Module Type pilih Schematic. Kemudian klik tombol NEXT.



Gambar I.5.4 Memberi nama project dan mengatur jenis Top Level Module

6. Masukkan parameter seperti gambar I.5.5. Kemudian klik tombol NEXT sampai tiga kali, konfigurasi yang ada tidak perlu diubah, kemudian klik tombol FINISH. Maka akan tampak seperti gambar 11. Persiapan project **counter3bit** telah selesai.

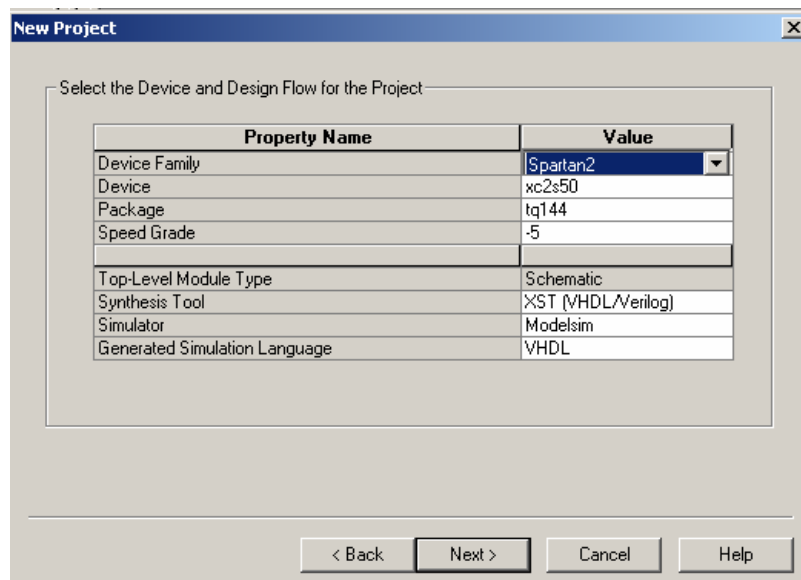
Device Family: **Spartan2**

Device: **Xc2s50**

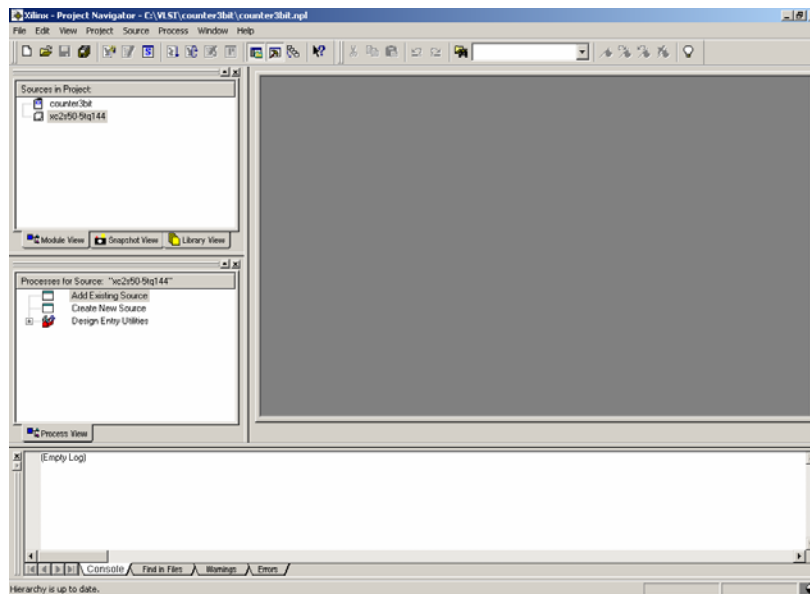
Package: **tq144**

Speed Grade: **-5**

Top-Level Modul Type: **Schematic**

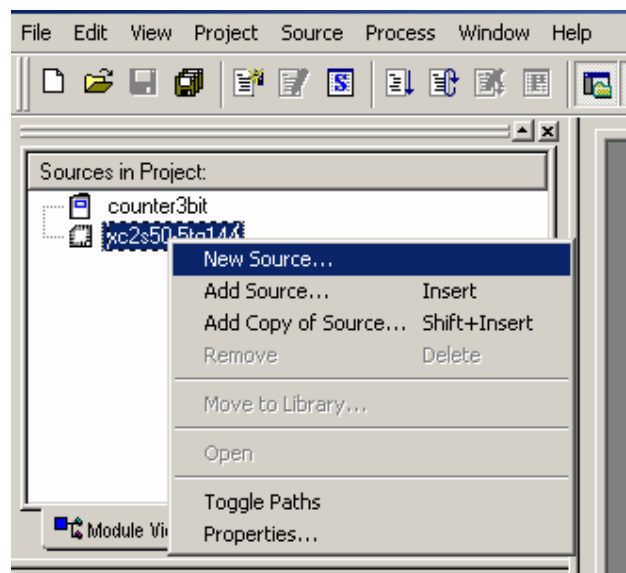


Gambar I.5.5 Menentukan Jenis Device yang akan diprogram



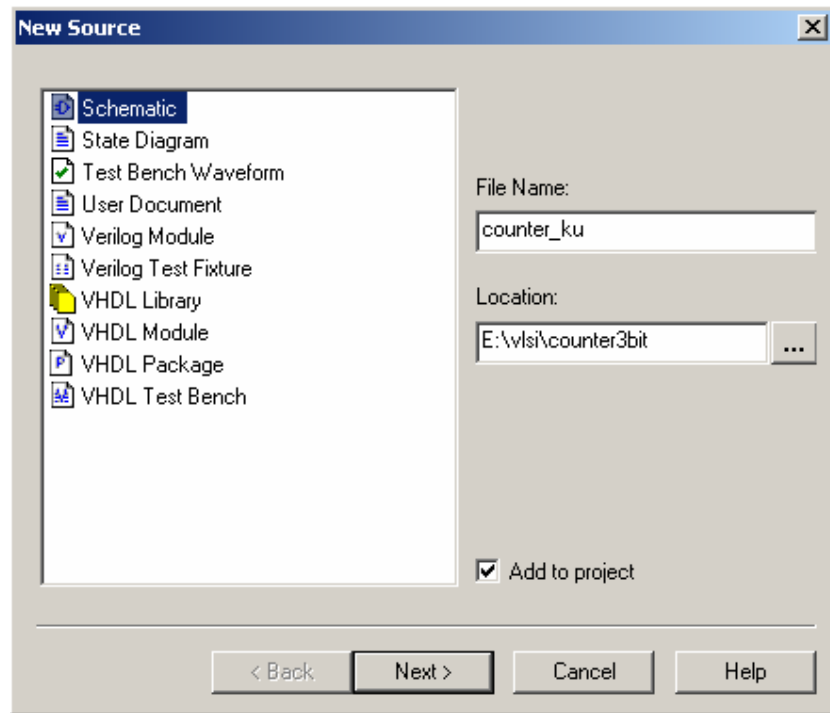
Gambar I.5.6. Hasil pembuatan project baru, masih kosong

7. Sekarang akan dibuat disain rangkaian counter sinkron secara schematic. Dimulai dengan menambahkan file Schematic kedalam project **counter3bit**. Klik kanan obyek XC2S50-5TQ144 sehingga muncul pop-up window dan pilih **New Source**.



Gambar I.5.7 Membuat source baru

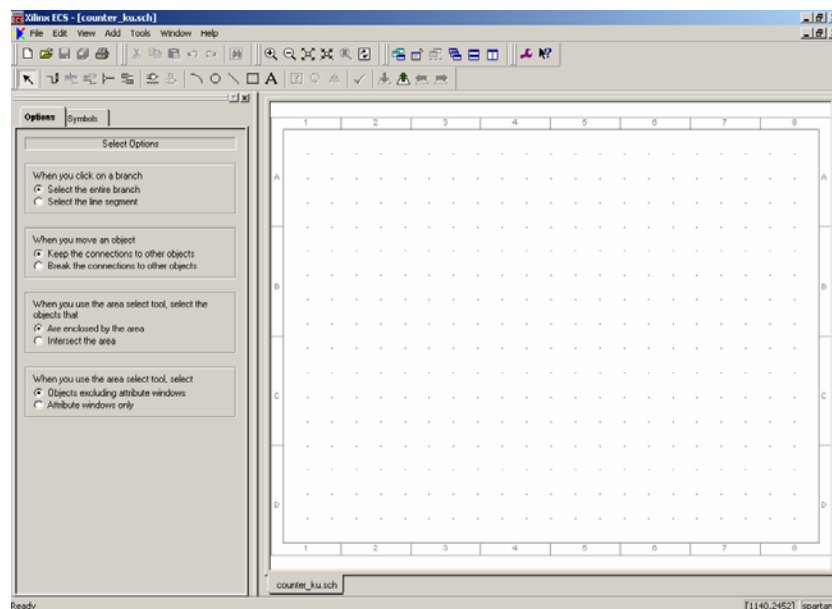
8. Beri nama file dengan nama **counter_ku** dan pilih **Schematic module**. Kemudian klik tombol next.



Gambar I.5.8 Menentukan parameter source baru yang akan dibuat

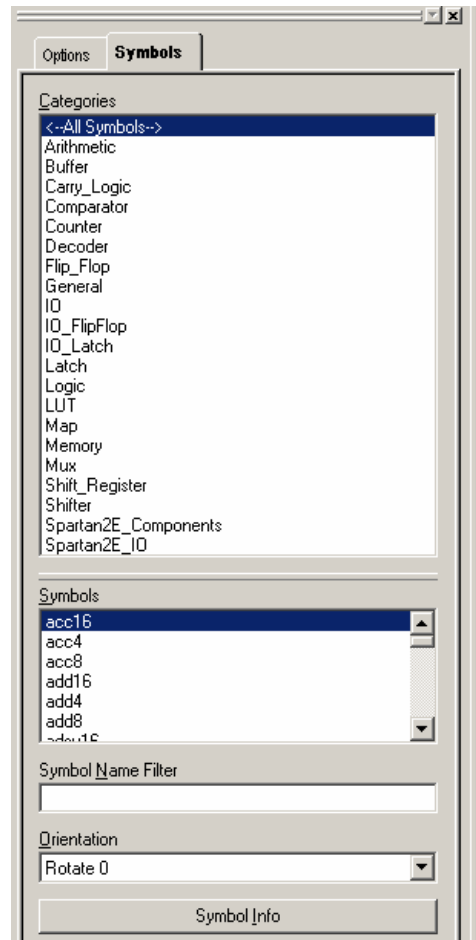
9. Selanjutnya anda akan mendapatkan jendela konfirmasi **NEW SOURCE INFORMATION** klik **FINISH** untuk melanjutkan.

10. Maka anda akan mendapatkan window Xilinx ECS kosong untuk membuat rancangan schematic.



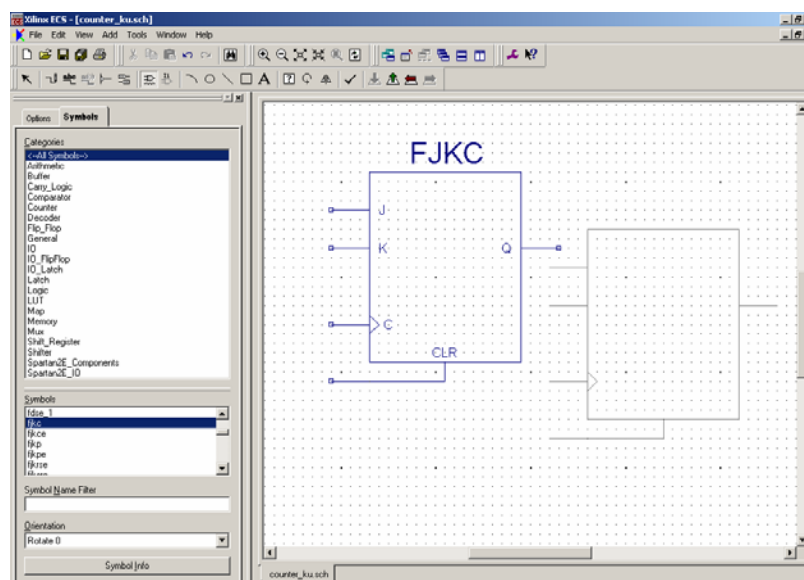
Gambar I.5.9 Jendela Xilinx ECS tempat membuat module schematic

11. Klik **Symbol** untuk mendapatkan Schematic Library seperti gambar... berikut



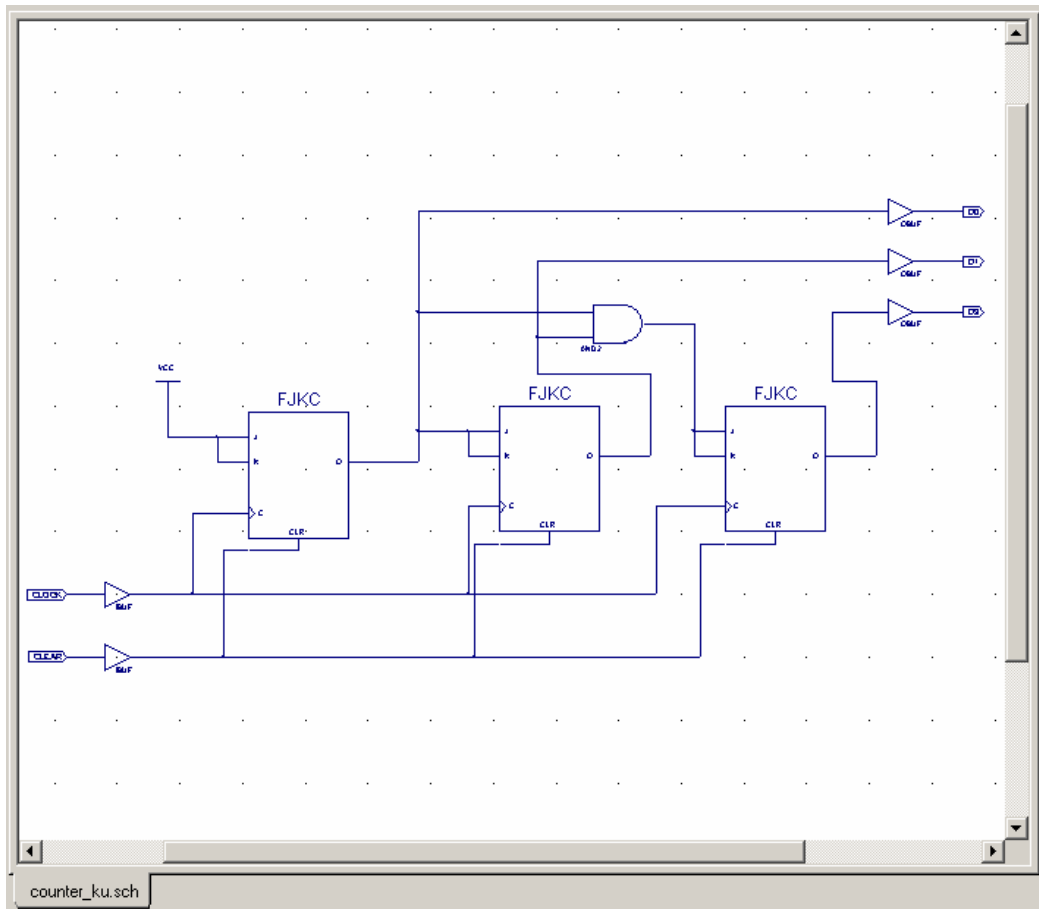
Gambar I.5.10 Schematic Library

12. Pilih symbol FJKC untuk merancang Counter




Gambar I.5.11 Mengambil Symbol FJKC (JK Flip-Flop dengan fasilitas Clear)

13. Dengan memanfaatkan symbol-symbol yang tersedia buatlah rangkaian counter sinkron 3-bit dari 3 buah flip-flop seperti gambar berikut :



Gambar I.5.11 rangkaian counter sinkron 3-bit dari 3 buah flip-flop

14. Lakukan **check schematic** dengan menekan tombol  , apabila anda berhasil mendapatkan pesan seperti dibawah ini maka schematic anda tidak ada kesalahan. Close pesan dan **save** schematik. Jika ada error maka perbaiki dahulu dan check lagi sampai tidak ada kesalahan.



Gambar I.5.12 Konfirmasi hasil check schematic

15. Langkah selanjutnya adalah menentukan lokasi pin-pin FPGA yang digunakan dengan cara mengedit pada **Edit Constrains (Text)** pada bagian **user constraints** di jendela process view. Double klik pada **Edit Constrains** maka anda akan mendapatkan pesan yang menyarankan untuk membuat *file constraint* , pilih **YES**:



Gambar I.5.13 Edit Constraint untuk menentukan lokasi pin-pin FPGA

setelah itu maka anda secara otomatis akan masuk ke jendela baru berupa teks editor yang secara otomatis file ini nantinya akan disimpan sebagai **counter_ku.ucf** yaitu file yang mendiskripsikan lokasi pin-pin yang kita gunakan.

16. Selanjutnya edit/isi file **counter_ku.ucf** ini sesuai dengan lokasi pin yang kita gunakan yaitu :

```
net CLOCK loc=p23; #pin CLOCK berada pada pin ke-23 pada FPGA
```

```
net CLEAR loc=p79; #pin CLEAR berada pada pin ke-79 pada FPGA
```

```
net D0 loc=p68; #pin D0 berada pada pin ke-68 pada FPGA
```

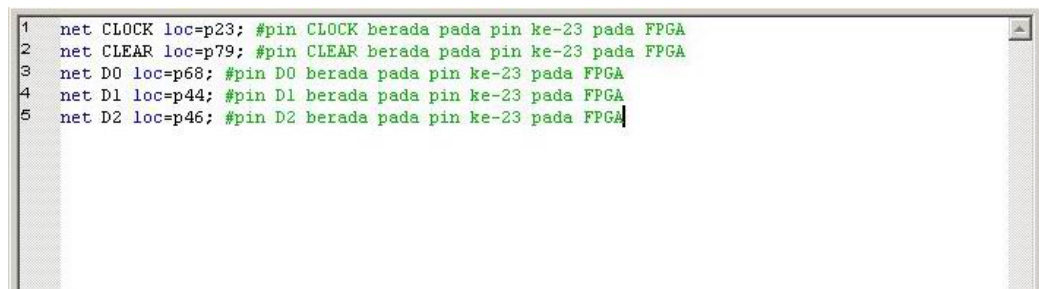
```
net D1 loc=p44; #pin D1 berada pada pin ke-44 pada FPGA
```

```
net D2 loc=p46; #pin D2 berada pada pin ke-46 pada FPGA
```

CATATAN : nama NET harus sama persis dengan nama pin pada file schematic

(Penentuan pin berdasarkan datasheet XSTENDBOARD yang digunakan)

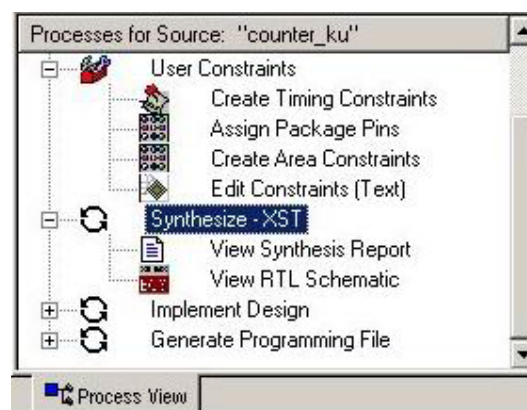
Kemudian save file



```
1 net CLOCK loc=p23; #pin CLOCK berada pada pin ke-23 pada FPGA
2 net CLEAR loc=p79; #pin CLEAR berada pada pin ke-23 pada FPGA
3 net D0 loc=p68; #pin D0 berada pada pin ke-23 pada FPGA
4 net D1 loc=p44; #pin D1 berada pada pin ke-23 pada FPGA
5 net D2 loc=p46; #pin D2 berada pada pin ke-23 pada FPGA
```

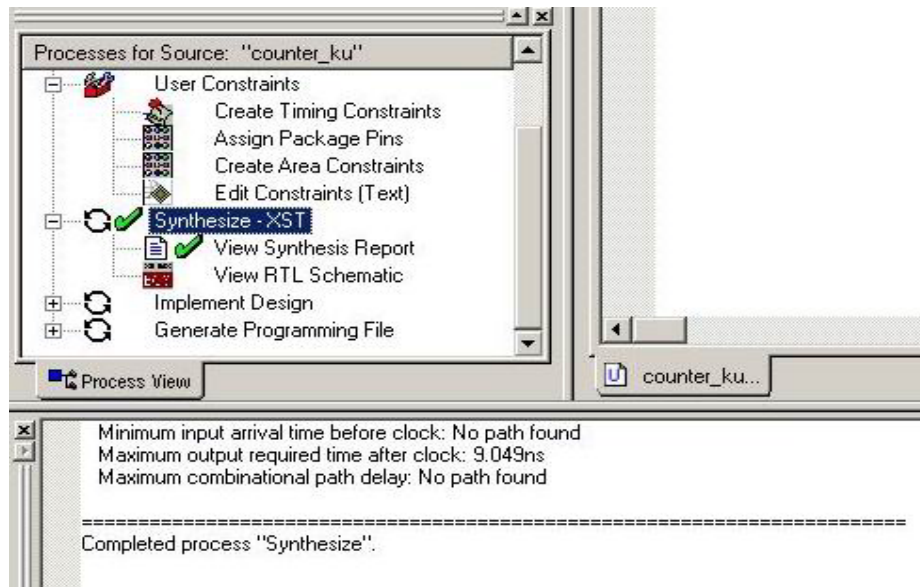
Gambar I.5.14 Edit Constraint file **counter_ku.ucf** ini sesuai dengan lokasi pin yang kita gunakan

17. Selanjutnya melakukan synthesis rancangan yang telah dibuat dengan cara melakukan double klik pada **synthesize XST**



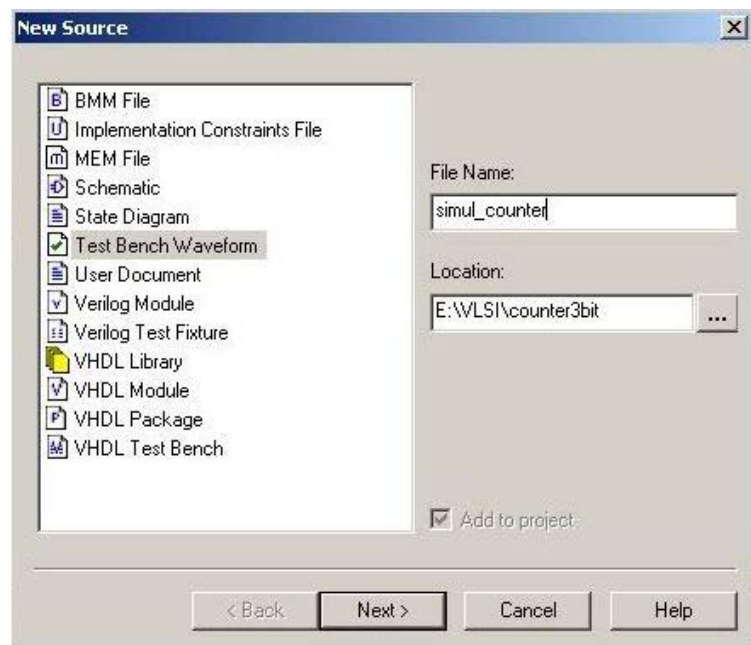
Gambar I.5.15 Melakukan synthesis rancangan yang telah dibuat

Jika proses synthesis berhasil maka anda akan mendapatkan pesan bahwa proses synthesis berhasil.



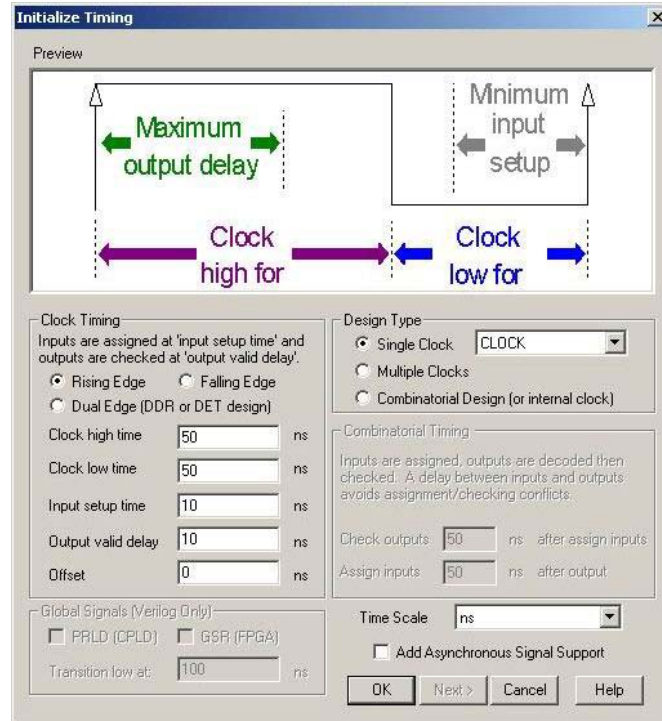
Gambar I.5.16 Proses synthesis telah berhasil

18. Langkah selanjutnya adalah melakukan simulasi , sebelumnya harus dibuat terlebih dahulu file **Test Bench Waveform**, beri nama file **counter_ku** selanjutnya pilih NEXT dan pastikan source filenya **counter_ku** dan NEXT sampai FINISH



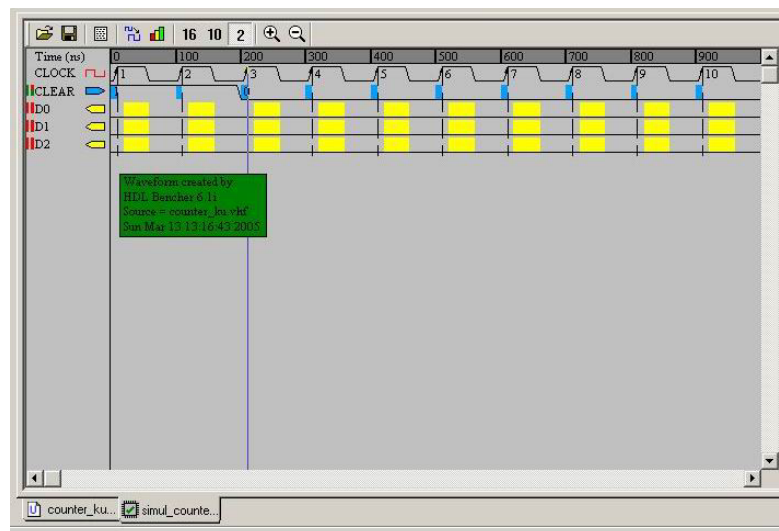
Gambar I.5.17 Membuat file *Test bench* untuk melakukan simulasi

Apabila berhasil maka anda mendapatkan gambar berikut, pilih **OK**



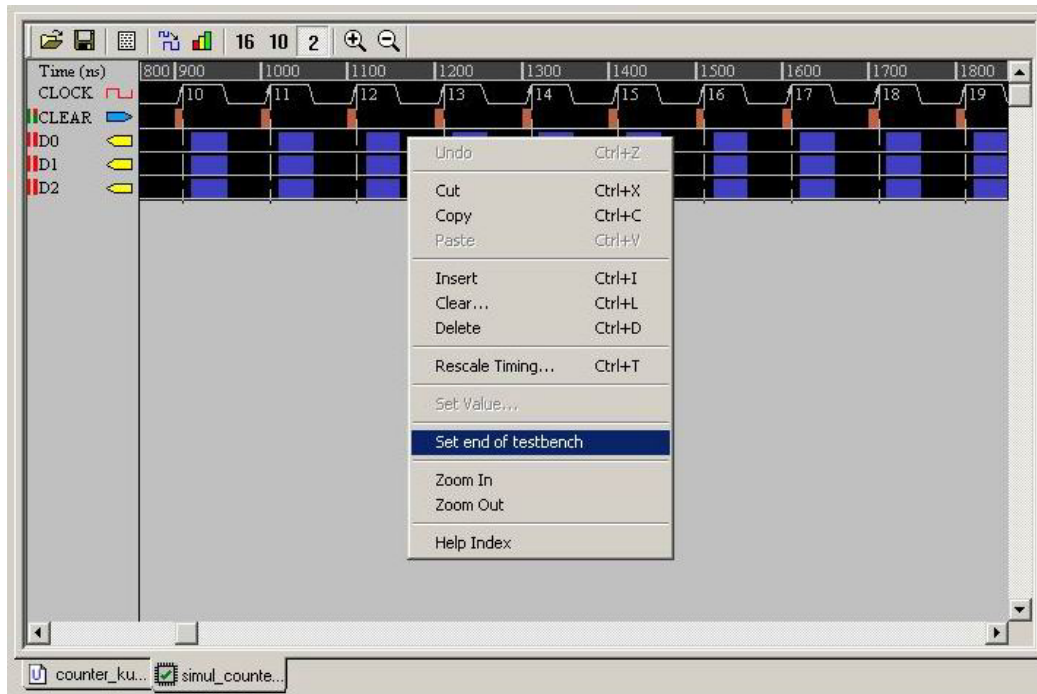
Gambar I.5.18 Mengatur pulsa clock

19. Sekarang anda telah mendapatkan source file untuk simulasi, Selanjutnya untuk mengedit file, lakukan dengan cara meng-klik pada gambar sinyal. Kemudian **SAVE**



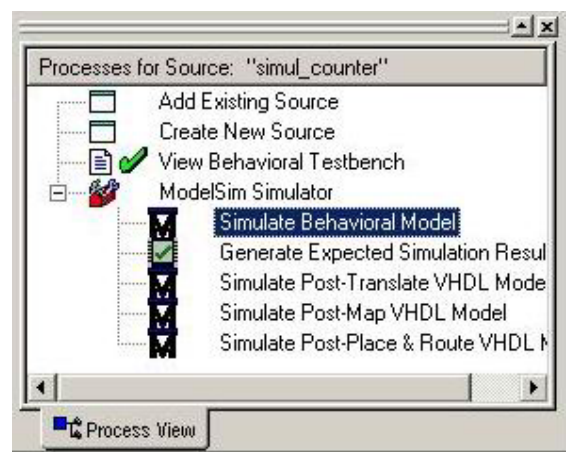
Gambar I.5.19 Contoh file *Test bench* untuk melakukan simulasi

Selanjutnya tentukan batas akhir simulasi dengan mengblok semua sinyal kemudian geser sampai batas waktu simulasi yang diinginkan dan klik kanan lalu pilih **Set End of Testbench**.
(Untuk versi ISE WEBPACK 7 cukup isi batas akhir waktu simulasi)



Gambar I.5.20 Mengatur batas akhir simulasi

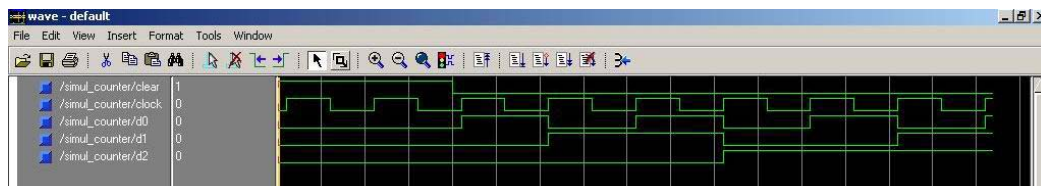
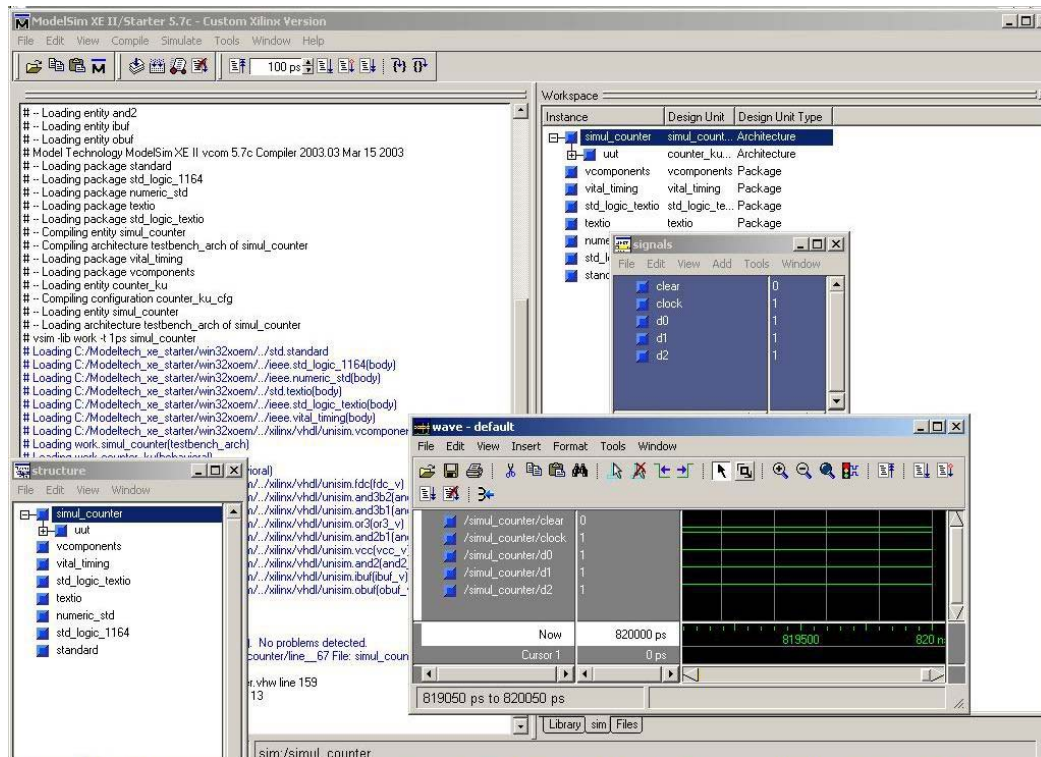
20. Untuk memulai simulasi double klik pada **Simulate Behavioral Model**



Gambar I.5.21 Memulai proses simulasi

maka secara otomatis software **MODELSIM** akan aktif dan menggambarkan hasil simulasi

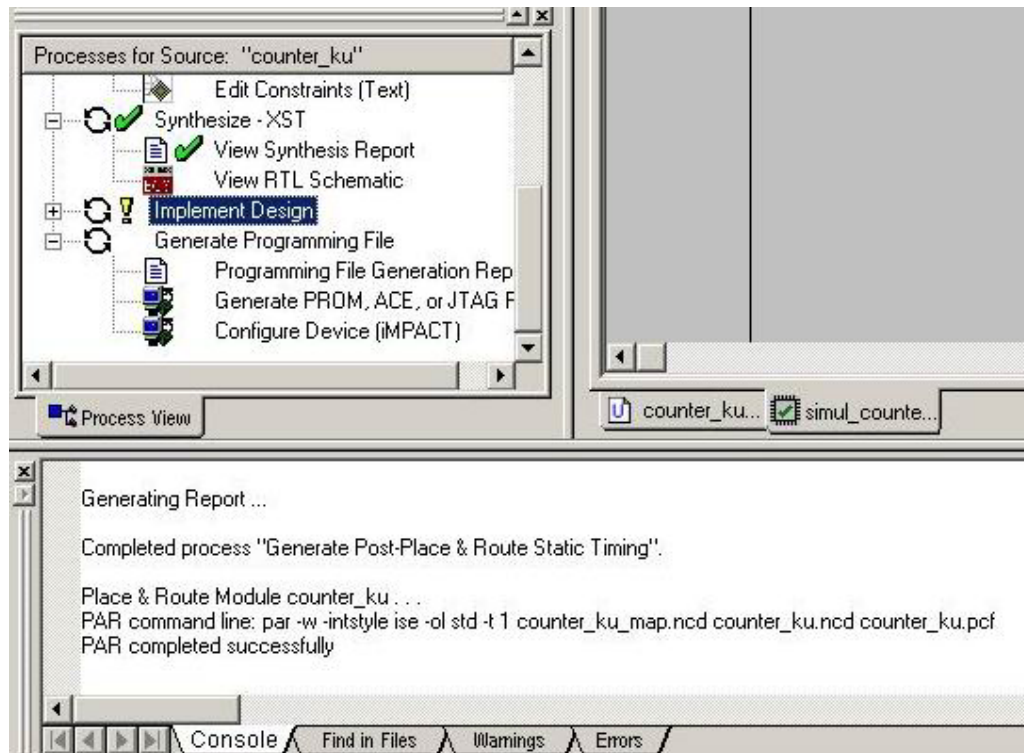
CATATAN : Simulasi yang dilakukan pada tahap ini adalah simulasi jenis **FUNCTIONAL SIMULATION**



Gambar I.5.22 Contoh hasil simulasi

Untuk melakukan simulasi lagi dengan nilai input yang berbeda maka software MODELSIM harus ditutup dahulu dan ulangi kembali ke langkah 19.

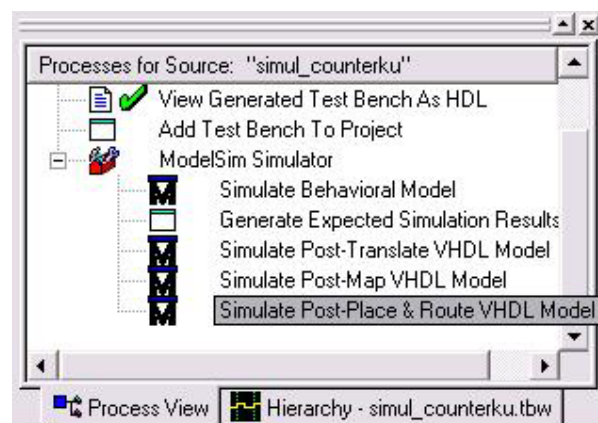
21. Apabila hasil simulasi telah sesuai dengan yang diharapkan maka langkah selanjutnya adalah mengimplementasikan rancangan pada IC FPGA dengan cara melakukan double-klik pada **Implement Design**. Pada tahap ini software melakukan proses MAP, PLACE dan ROUTE secara otomatis mengatur penggunaan resource pada FPGA.



Gambar I.5.23 Melakukan Implement Dsign (Implementasi hasil perancangan)

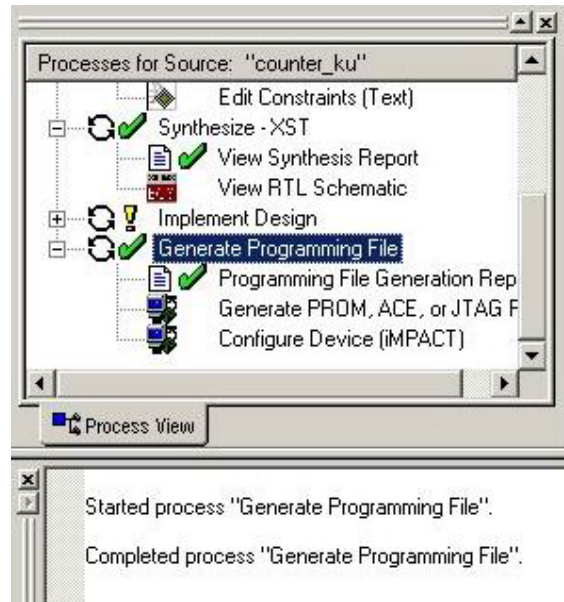
22. Lakukan simulasi sebagaimana **langkah 20**, kali ini pilih **simulate Post-Place & Route VHDL Module**

CATATAN : Simulasi yang dilakukan pada tahap ini adalah simulasi jenis **TIMING SIMULATION**




Gambar I.5.24 Melakukan simulasi TIMING

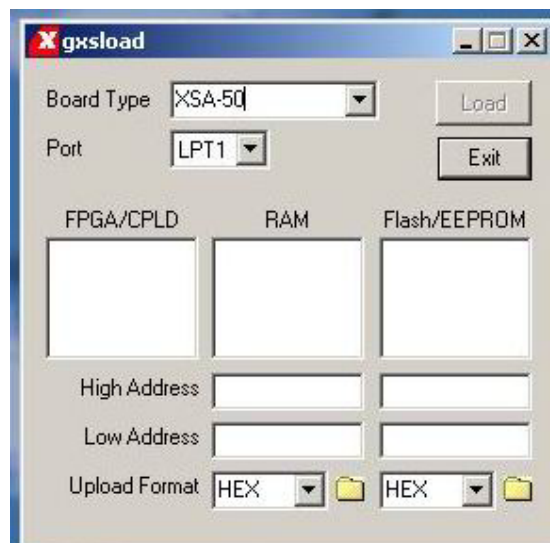
23. Langkah selanjutnya adalah membuat file *.bit yang nantinya akan diuploadkan pada FPGA. Dengan cara melakukan double-Klik pada **Generate Programming File**.



Gambar I.5.25 Membuat file *.bit

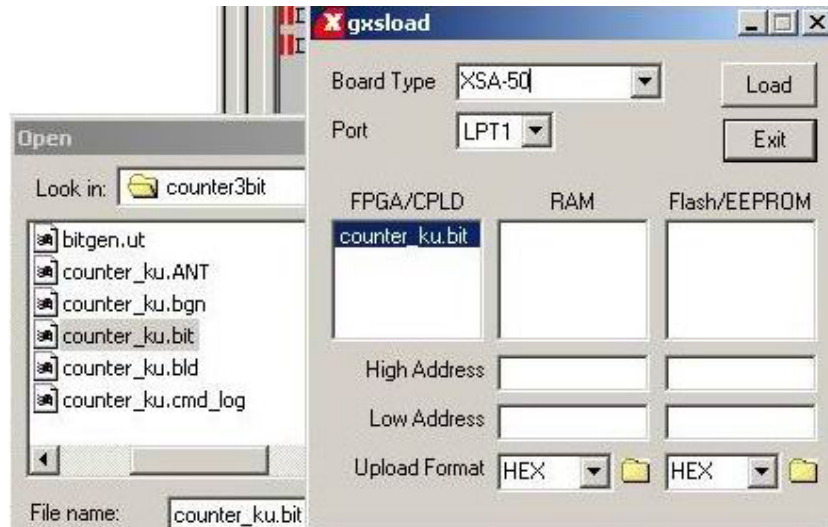
24. Langkah selanjutnya adalah meng-upload programming file (**counter_ku.bit**) ke IC FPGA dengan bantuan software **GXSLOAD**. Aktifkan terlebih dahulu software **GXSLOAD** dengan cara mengklik icon  pada desktop.

Kemudian tentukan jenis board **XSA-50** dan port yang digunakan adalah **LPT1**



Gambar I.5.26 Jendela software GXSLOAD

Kemudian **drag and drop** file `counter_ku.bit` pada jendela **FPGA/CPLD GXLOAD**



Gambar I.5.27 Proses drag and drop file *.bit ke GXLOAD

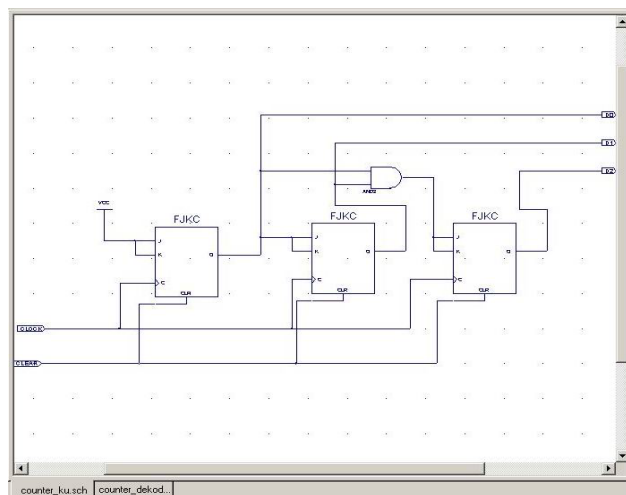
Selanjutnya klik **LOAD** untuk mengupload file `counter_ku.bit`

25. Atur posisi **Dipswitch yang ke-8** pada posisi **ON** kemudian berisi pulsa clock dengan menekan tombol **S5**. Perhatikan apa yang terjadi pada **BARLED no 1,2,3**.

5.2 Memanfaatkan rancangan sebelumnya

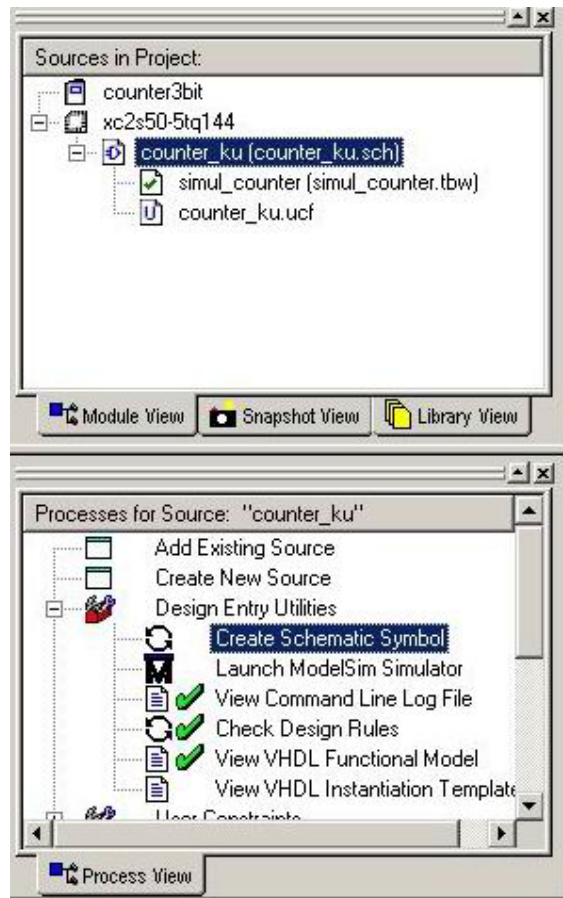
Pada percobaan selanjutnya akan dibuat rangkaian yang memanfaatkan rangkaian counter3bit yang telah dibuat sebelumnya.

26. Langkah pertama adalah dengan mengedit `counter_ku.sch` dengan menghilangkan semua buffer (IBUF maupun OBUF).



Gambar I.5.28 Hasil proses edit pada file `counter_ku.sch`

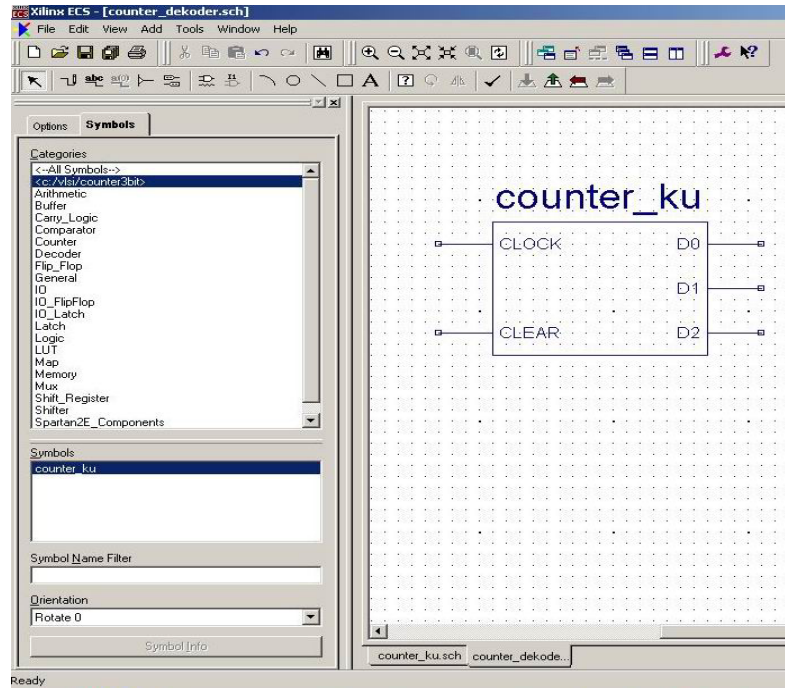
27. Membuat simbol baru dari rancangan sebelumnya dengan cara melakukan double-klik pada **Create Schematic Symbol**. Maka secara otomatis akan dihasilkan simbol baru .



Gambar I.5.29 Membuat symbol baru dari hasil rancangan sebelumnya

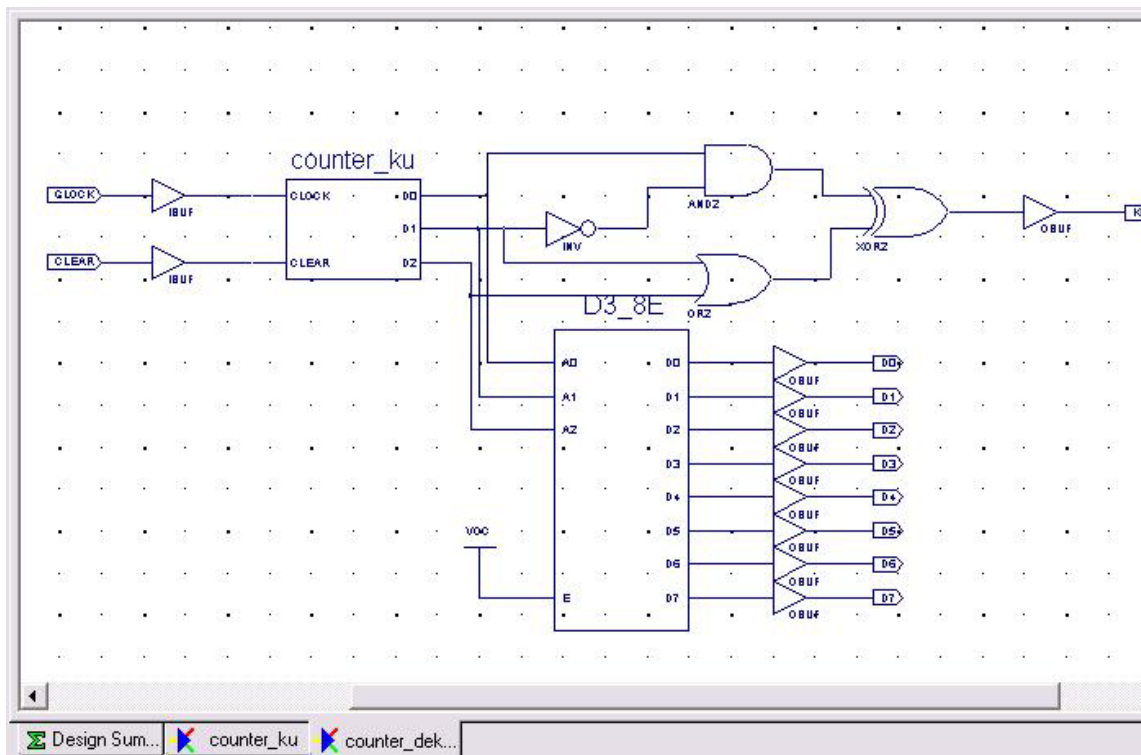
28. Langkah selanjutnya adalah membuat file Schematic baru seperti pada langkah 7, 8, 9, dan 10. Beri nama **counter_dekoder**

29. Pastikan bahwa telah terbuat simbol komponen baru pada XILINX ECS, gunakan simbol komponen ini untuk dirangkai bersama Dekoder



Gambar I.5.30 Symbol baru yang digunakan pada file schematic counter_dekoder

30. Dengan memanfaatkan symbol-symbol yang tersedia buatlah rangkaian yang menghubungkan komponen counter_ku dengan dekoderD3_8E (3 to 8) dan Rangkaian.



Gambar I.5.31 Hasil perancangan

31. **Selanjutnya lakukan langkah 14 s.d 24** sebagaimana percobaan sebelumnya.

Untuk Lokasi pin pada FPGA gunakan data sebagai berikut :

net CLOCK loc=p23; #pin CLOCK berada pada pin ke-23 pada FPGA

net CLEAR loc=p79; #pin CLEAR berada pada pin ke-79 pada FPGA

net D0 loc=p68; #pin D0 berada pada pin ke-68 pada FPGA

net D1 loc=p44; #pin D1 berada pada pin ke-44 pada FPGA

net D2 loc=p46; #pin D2 berada pada pin ke-46 pada FPGA

net D3 loc=p49; #pin D3 berada pada pin ke-49 pada FPGA

net D4 loc=p57; #pin D4 berada pada pin ke-57 pada FPGA

net D5 loc=p62; #pin D5 berada pada pin ke-62 pada FPGA

net D6 loc=p60; #pin D6 berada pada pin ke-60 pada FPGA

net D7 loc=p67; #pin D7 berada pada pin ke-67 pada FPGA

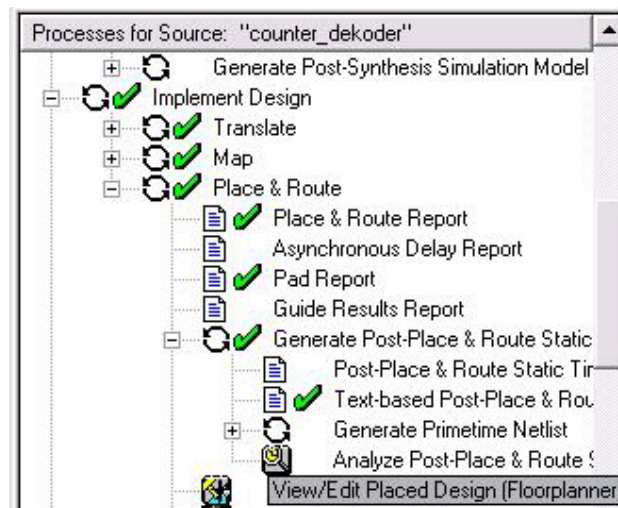
net KOMBI loc=p59; #pin COMBI berada pada pin ke-59 pada FPGA

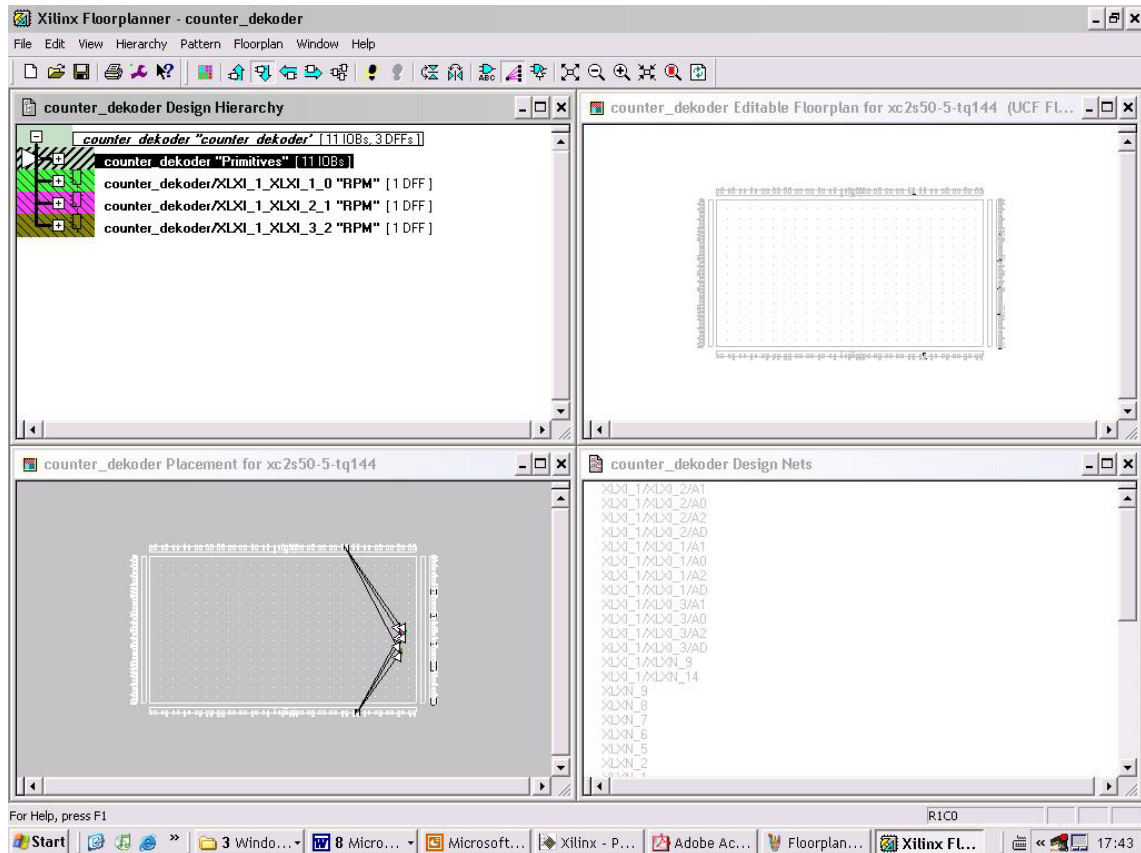
(Penentuan pin berdasarkan datasheet XSTENDBOARD yang digunakan)

32. Atur posisi **Dipswitch yang ke-8** pada posisi **ON** kemudian berisi pulsa clock dengan menekan tombol **S5**. Perhatikan apa yang terjadi pada **BARLED no 1,2,3,4,5,6,7,8 dan 10**

33. Langkah berikutnya adalah langkah untuk mengetahui apa yang terjadi didalam FPGA setelah di program.

- Klik Ganda pada **Implement Design -> View/Edit Place Design (Floorplanner)**





pada floorplanner ini akan tampak lokasi komponen yang terpakai pada FPGA, **CATAT DATA YANG DIDAPAT.**

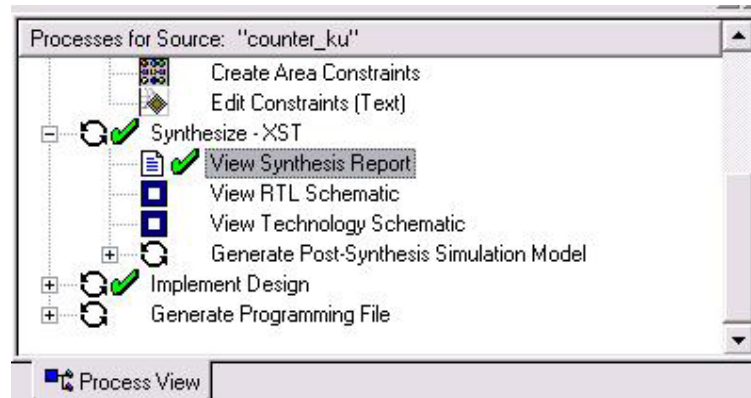
VI. Hasil Pengujian

Sebagai laporan sementara dan data pengujian maka lakukan percobaan untuk melengkapi lampiran Hasil Pengujian pada lembar yang telah tersedia (**Lampiran 1. Lembar Hasil Pengujian**)

VII. Analisa

Pada Laporan Resmi lakukan analisa kinerja dari alat yang dibuat dengan membandingkan antara hasil simulasi dengan hasil pengujian. Serta analisa perbedaan antara simulasi FUNCTIONAL dengan simulasi TIMING.

Analisa informasi yang anda dapat dari file synthesis report.



Gambar I.5.32 Bagaimana cara mendapatkan Synthesis report

VIII. Tugas

1. Apa yang dimaksud dengan proses Synthesis ?
2. Jelaskan perbedaan antara Simulasi FUNCTIONAL dan Simulasi TIMING.
3. Kenapa harus dilakukan proses EDIT CONSTRAIN?
4. Lampirkan cetakan synthesis report serta Place&Route report dan berikan penjelasan anda

IX. Daftar Pustaka

1. Kevin Skahill, "VHDL for Programmable Logic", Addison Wesley
2. M. Morris Mano, "Digital Design" (3rd Edition), Prentice Hall
3. M. Morris Mano & C. Kime, "Logic and Computer Design Fundamentals" Prentice Hall
4. Stefan Sjöholm & L. Lindh, "VHDL for Designers" Prentice Hall
5. Xilinx FPGA IseWebpack 7.0 Tutorial

Lampiran 1