

I. Tujuan Praktikum :

- Praktikan Dapat Memahami Synthesis Dalam Program FPGA.
- Praktikan Dapat Mengimplementasikan Rangkaian Digital Sederhana,

II. Dasar Teori :

- Pengenalan Synthesis Pada VHDL.
- Proses Synthesis Program VHDL Menjadi Rangkaian Digital.

III. Peralatan :

- FPGA Xilinx Artix 7 Board Nexys 4.
- Kabel Power USB.
- 1 buah PC.
- Software ISE Design Suite 14.5.

4.1 Pengertian

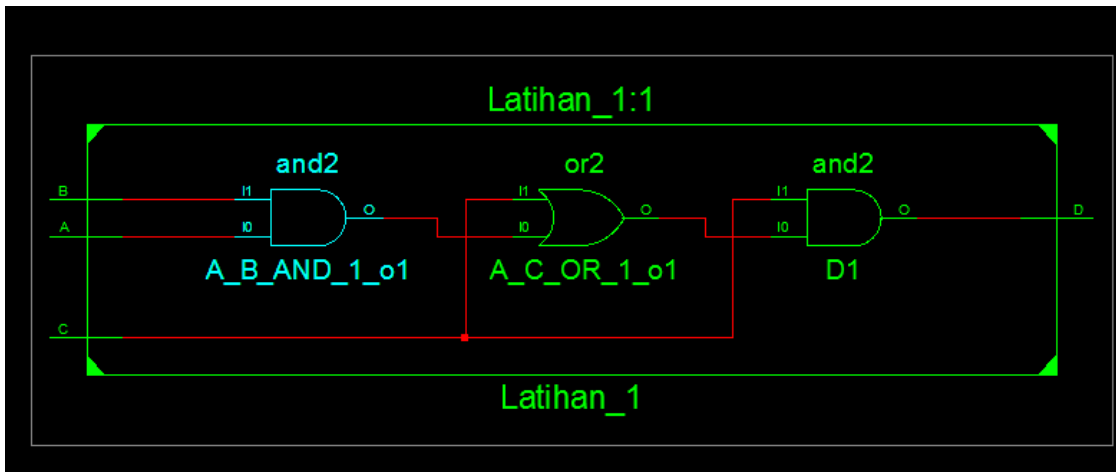
Penggunaan VHDL biasanya digunakan untuk dua tujuan secara umum, yaitu simulasi desain rangkaian elektronik dan melakukan synthesis program dari desain tersebut. Synthesis adalah sebuah proses dimana sebuah program VHDL disusun dan mengimplementasikannya menjadi sebuah rangkaian digital. Dalam kata lain, ini merupakan sebuah proses translasi atau kompilasi dari sebuah deskripsi desain program (source code) menjadi bentuk skematik desain yang nantinya akan diimplementasikan ke dalam FPGA.

Contoh :

```
entity Latihan_1 is
    Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          C : in  STD_LOGIC;
          D : out STD_LOGIC);
end Latihan_1;

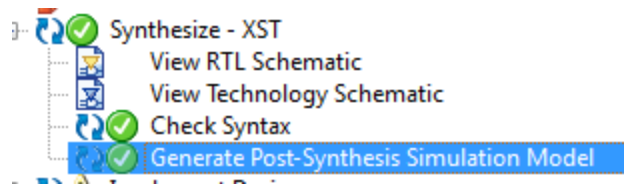
architecture Behavioral of Latihan_1 is
begin
    D <= (((A and B) or C ) and C) ; -- () and ()
end Behavioral;
```

Gambar 4.1 Contoh program.



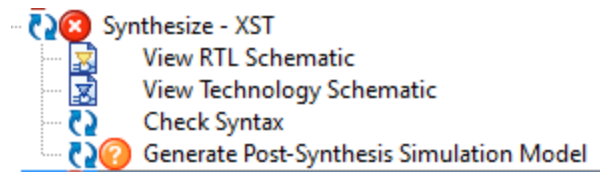
Gambar 4.2 Contoh skematik.

Dari program diatas, dapat dilihat bahwa proses sintesis pada VHDL tersebut merubah sebuah kode menjadi sebuah skematik desain. Inilah yang disebut dengan proses sintesis.

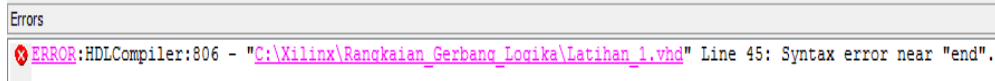


Gambar 4.3 Proses sintesis berhasil.

Pada umumnya, selain merubah kode program menjadi desain rangkaian digital, proses sintesis pada software ISE Design Suite juga mengecek syntax dari kode program yang sudah dibuat sebelumnya. Pengecekan ini meliputi apakah program yang ditulis sudah benar dengan mengikuti syntax penulisan program yang baik dan benar. Jika sudah benar, maka proses statusnya akan menjadi ceklis berwarna hijau. Namun jika terdapat error, maka statusnya akan menunjukkan symbol silang dan berwarna merah dan diikuti dibawahnya petunjuk dimana letak kesalahan dalam menuliskan program tersebut.



Gambar 4.4 Proses sintesis gagal.



Gambar 4.5 Log error pada software ISE Design Suite.

4.2 Place and Route

Setelah kode program di sintesis, maka proses selanjutnya yang perlu dilakukan adalah Place & Route. Proses ini mendeskripsikan berbagai elemen atau komponen baik itu input maupun output yang akan ditempatkan secara fisik dan dipetakan kedalam FPGA. Hasil dari proses ini yaitu sebuah file yang nantinya dapat digunakan dan diunduh kedalam FPGA tersebut. Place & Route umumnya memerlukan beberapa waktu untuk membuat program tersebut menjadi sebuah file.

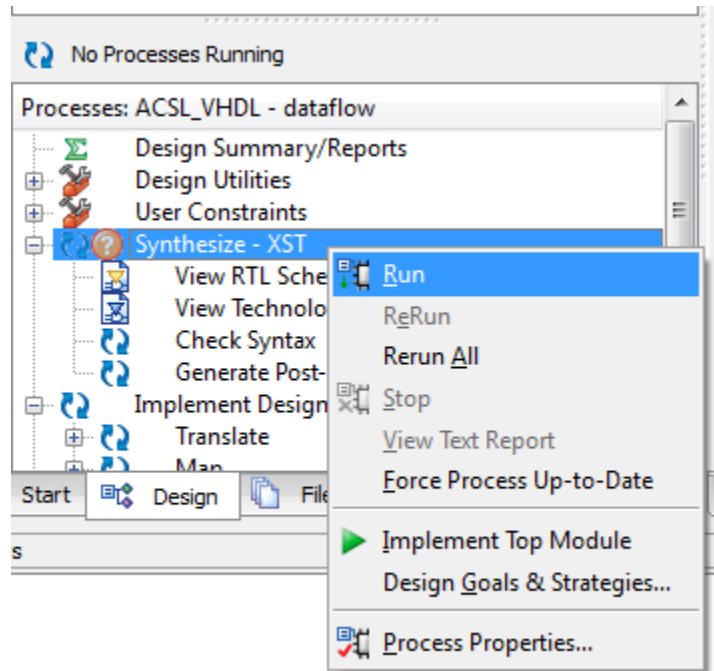
4.3 Melakukan Syntethesis Program

1. Buka aplikasi ISE Design Suite 14.5/
2. Buka menu **File > Open Project**. Cari file project sebelumnya yang sudah dibuat.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5
6  entity ACSL_VHDL is
7      Port ( A : in  STD_LOGIC;
8            B : in  STD_LOGIC;
9            C : out STD_LOGIC);
10 end ACSL_VHDL;
11
12 architecture dataflow of ACSL_VHDL is
13
14 begin
15
16 C <= A and B;
17
18 end dataflow;
19
20
```

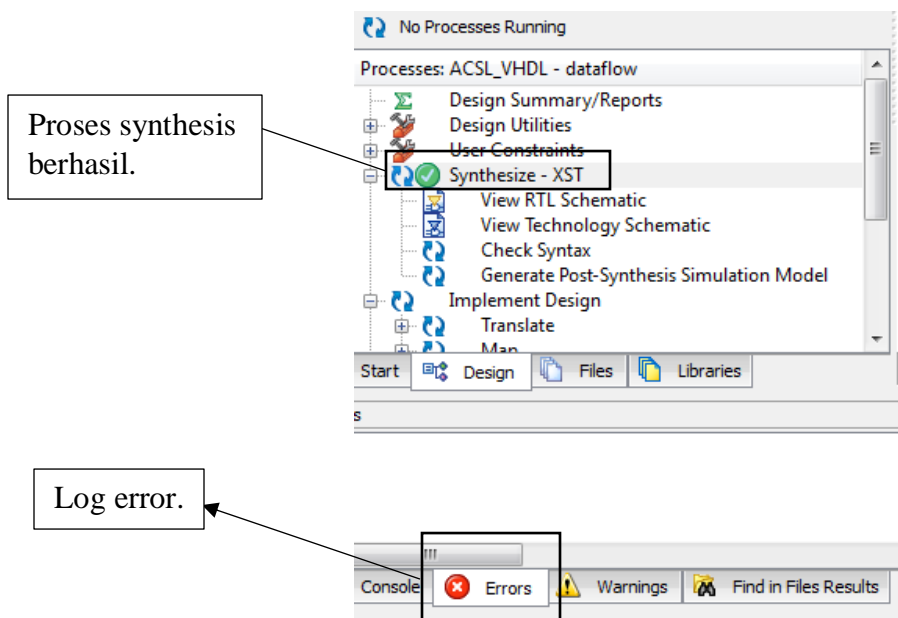
Gambar 4.6 Tampilan project sebelumnya.

3. Klik kanan, dan pilih run pada "Synthesize – XST".



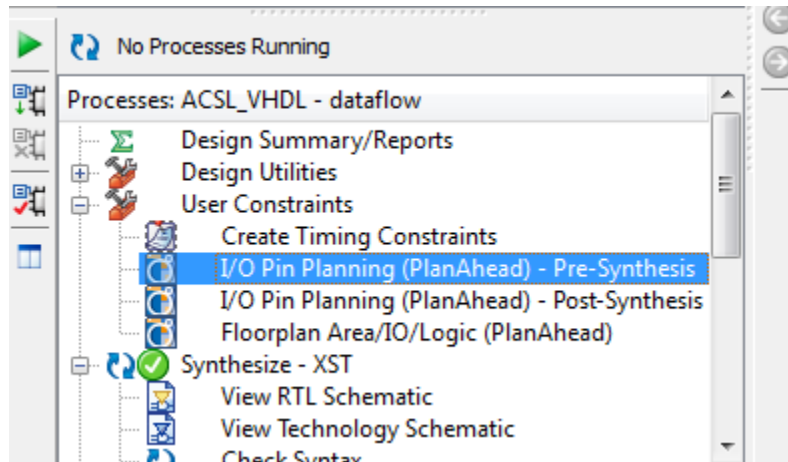
Gambar 4.7 Proses menjalankan tools synthesis.

Tunggu hingga proses selesai dan proses status menjadi ceklis berwarna hijau. Jika terdapat error, perbaiki program yang telah dibuat dengan membaca log error yang ada dibawah. Ikuti petunjuk errornya, setelah itu perbaiki.



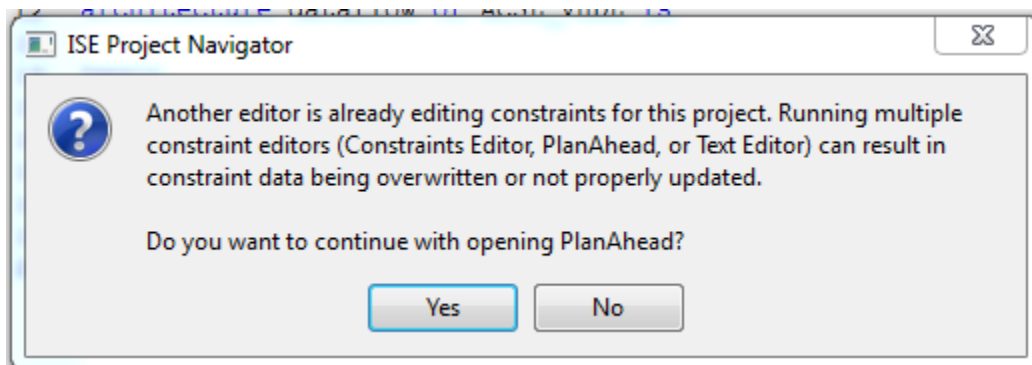
Gambar 4.8 Proses synthesis berhasil.

- Setelah berhasil, klik expand pada “**User Constraints**”, lalu pilih “**I/O Pin Planning (PlanAhead) – Pre-Synthesis**”. Jalankan dengan cara klik kanan lalu pilih run.



Gambar 4.9 Memilih User Constraints.

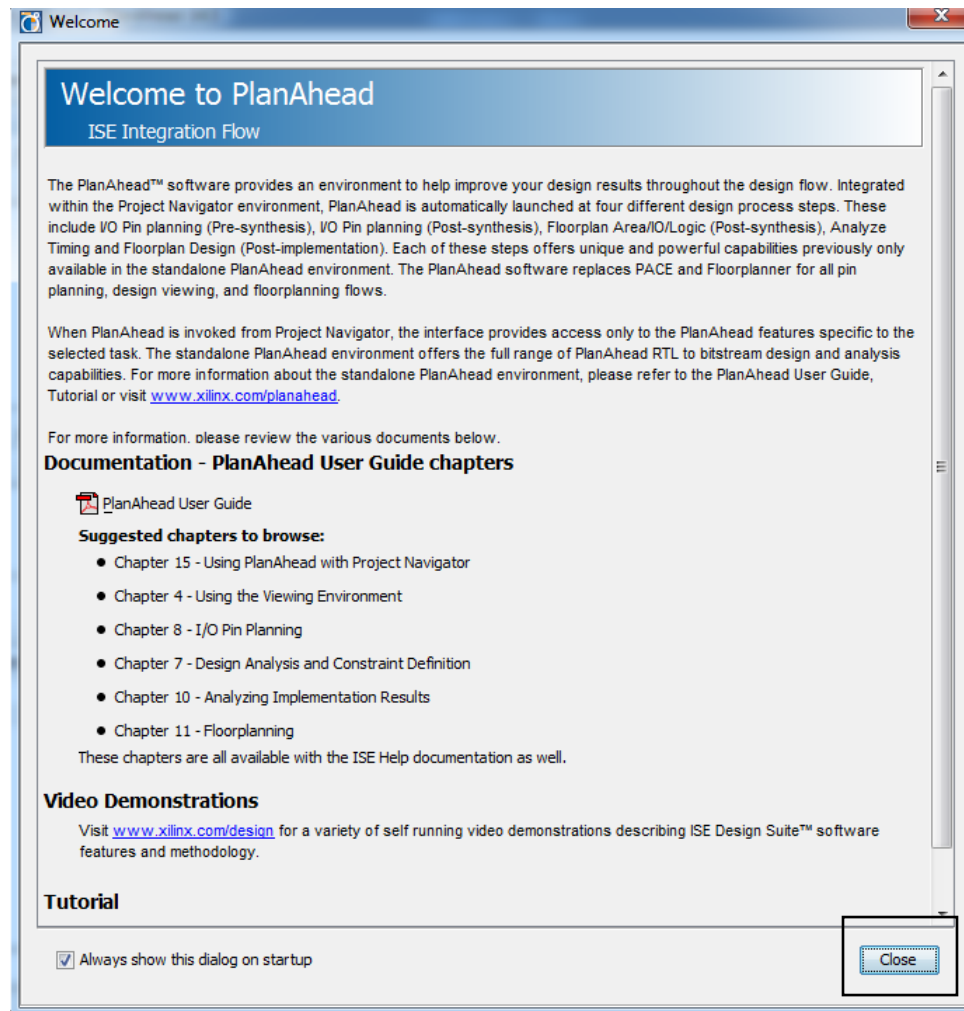
Jika muncul pesan seperti ini, pilih “**Yes**”.



Gambar 4.10 Menjalankan PlanAhead.

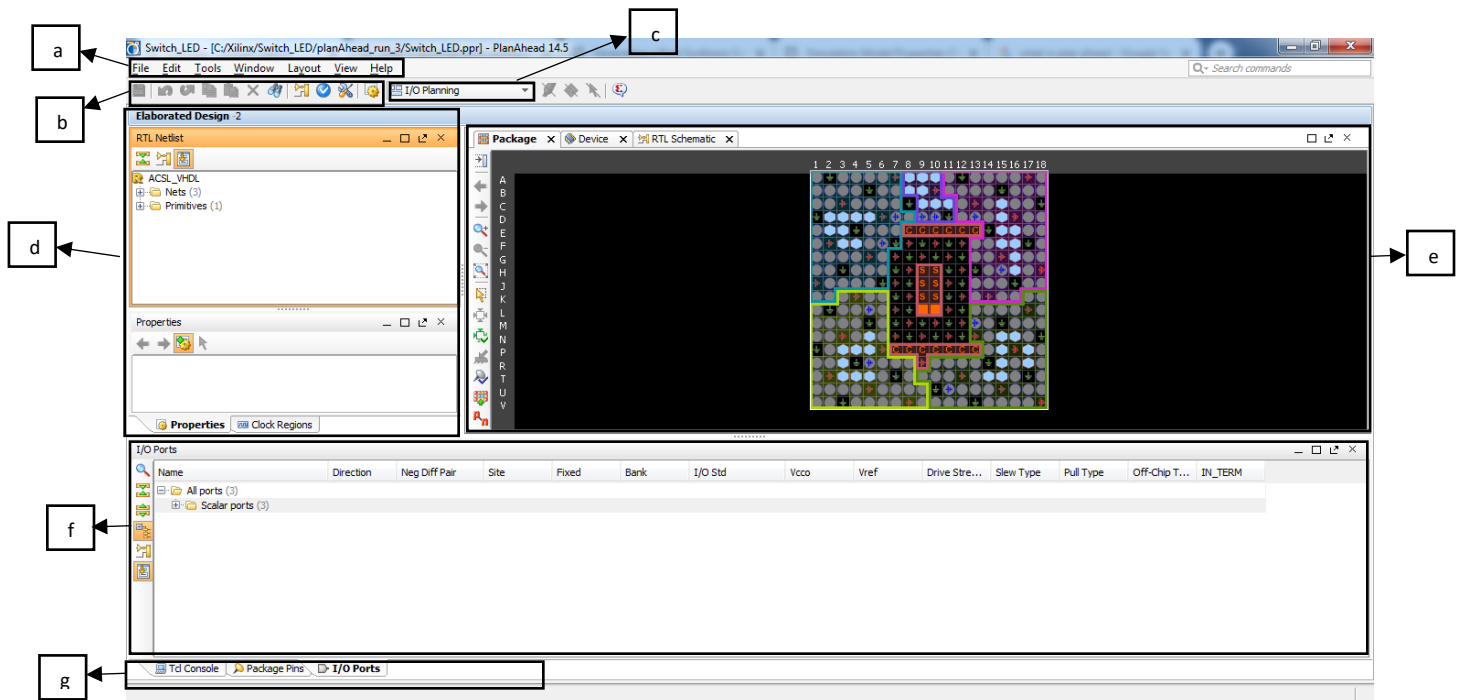
Tunggu beberapa saat agar aplikasi PlanAhead terbuka.

5. Setelah muncul tampilan PlanAhead seperti ini, klik tombol “Close” yang berada di bagian bawah untuk menutup pesan dan melanjutkan.



Gambar 4.11 PlanAhead popup.

6. PlanAhead merupakan software yang termasuk dalam paket penginstalan ISE Design Suite dimana kita dapat mengatur komponen input dan output sesuai dengan yang kita inginkan dengan memasukan pin yang ada pada board FPGA. Inilah tampilan awal pada software PlanAhead.



Gambar 4.12 Tampilan awal PlanAhead.

Keterangan :

- Main Menu, berisi perintah yang tersedia.
- Main Toolbar, berisi perintah yang umum digunakan.
- View Layout Selector, menyediakan akses ke konfigurasi tata letak tampilan yang ditentukan.
- Main Viewing Area, menampilkan ringkasan proyek dan tampilan grafis dari desain.
- Project Status Bar, menampilkan status proyek dan perintah yang aktif berjalan.
- Tcl Console and Message Area, menunjukkan status perintah, pesan aplikasi, hasil kompilasi dan laporan.
- Status Bar, menampilkan informasi tentang proyek yang sedang dibuka dan objek apa pun yang saat ini berada dibawah kursor.

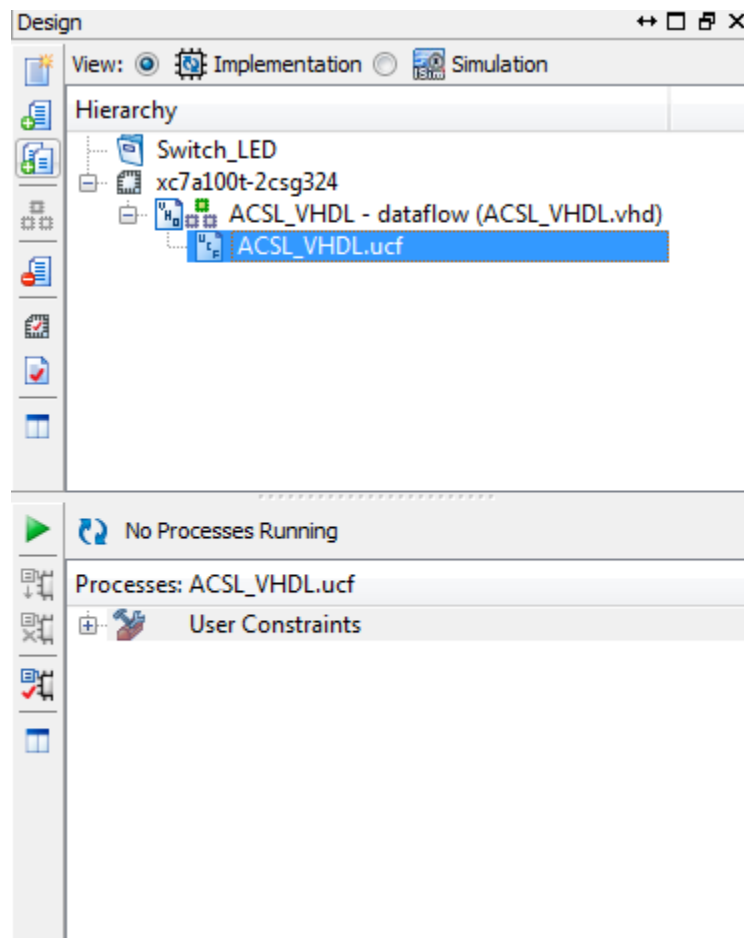
- Setelah mengetahui berbagai fungsi tools dari PlanAhead, langkah selanjutnya adalah mengkonfigurasi pin input dan output yang akan digunakan. Langkahnya yaitu klik expand pada “Scalar Ports”, dan isikan seperti dibawah ini.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew Type	Pull Type	Off-Chip T...	IN_TERM
All ports (3)													
Scalar ports (3)													
A	Input		V10	<input checked="" type="checkbox"/>		14 LVCMOS18	1.800				NONE	NONE	
B	Input		U11	<input checked="" type="checkbox"/>		14 LVCMOS18	1.800				NONE	NONE	
C	Output		V11	<input checked="" type="checkbox"/>		14 LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	

Gambar 4.13 Konfigurasi input dan output menggunakan PlanAhead.

Setelah dikonfigurasi, kemudian disimpan dengan cara klik tombol “CTRL + S” atau klik ikon save di kiri atas.

- Setelah disimpan, kembali ke aplikasi ISE Design Suite dan cek pada tab berikut ini.



Gambar 4.14 File UCF.

Terdapat sebuah file baru dengan ekstensi .ucf. File ini merupakan hasil konfigurasi yang sudah kita buat tadi.

```

1
2 # PlanAhead Generated physical constraints
3
4 NET "A" LOC = V10;
5 NET "B" LOC = U11;
6 NET "C" LOC = V11;
7 |
8
9 # PlanAhead Generated IO constraints
10
11 NET "A" IOSTANDARD = LVCMOS18;
12 NET "B" IOSTANDARD = LVCMOS18;
13 NET "C" IOSTANDARD = LVCMOS18;
14

```

Gambar 4.15 Isi file UCF.

Dapat dilihat, file ini berisikan konfigurasi pin yang telah kita buat sebelumnya. Maksud dari file yaitu :

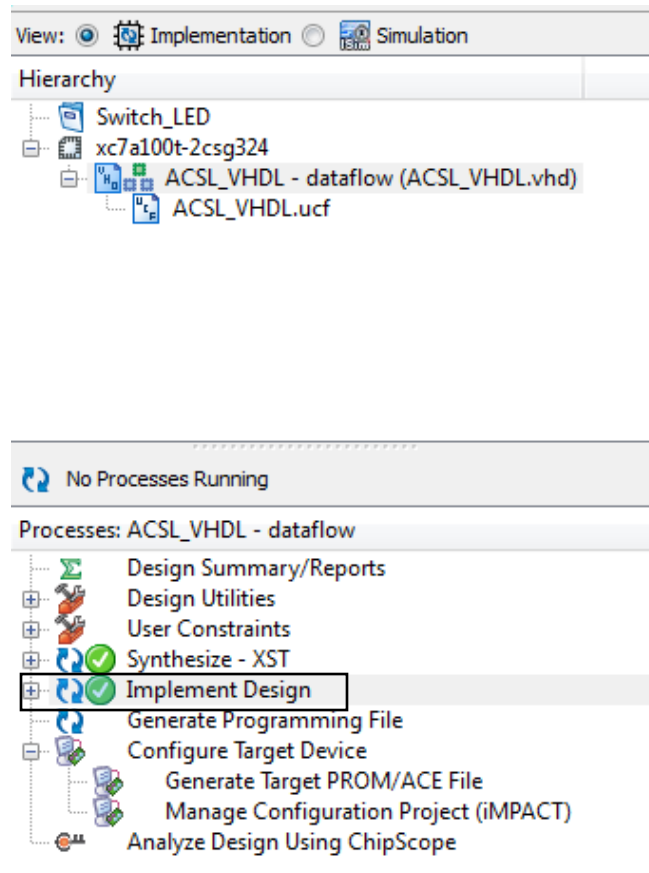
- Pin A berada dilokasi V10.
- Pin B berada dilokasi U11.
- Pin C berada dilokasi V11.

Lokasi ketiga pin tersebut dapat kita lihat pada board FPGA. Selanjutnya untuk IOSTANDARD merupakan sebuah deskripsi attribute, sedangkan LVCMOS (Low Voltage Complementary Metal Oxide Semiconductor) adalah sebuah teknologi tegangan rendah yang digunakan pada sirkuit FPGA untuk menjalankan komponen yang akan digunakan. Angka “18” sendiri menunjukkan berapa besar voltase yang digunakan, pada kasus ini angka “18” menunjukkan tegangan sebesar 1,8V. Berikut ini merupakan table dari LVCMOS tersebut.

Logic Volts	Tolerance Volts	Tolerance Percent
5.0V	+/-0.5V	+/-10.0%
3.3V	+/-0.3V	+/-9.09%
2.5V	+/-0.2V	+/-8.00%
1.8V	+/-0.15V	+/-8.33%
1.5V	+/-0.1V	+/-8.33%
1.2V	+/-0.1V	+/-8.33%
1.0V	+/-0.1V	+/-8.33%
0.9V	+/-0.045V	+/-5.00%
0.8V	+/-0.04V	+/-5.00%
0.7V	+/-0.05V	+/-7.14%

Tabel 4.1 Tabel Tegangan LVCMOS.

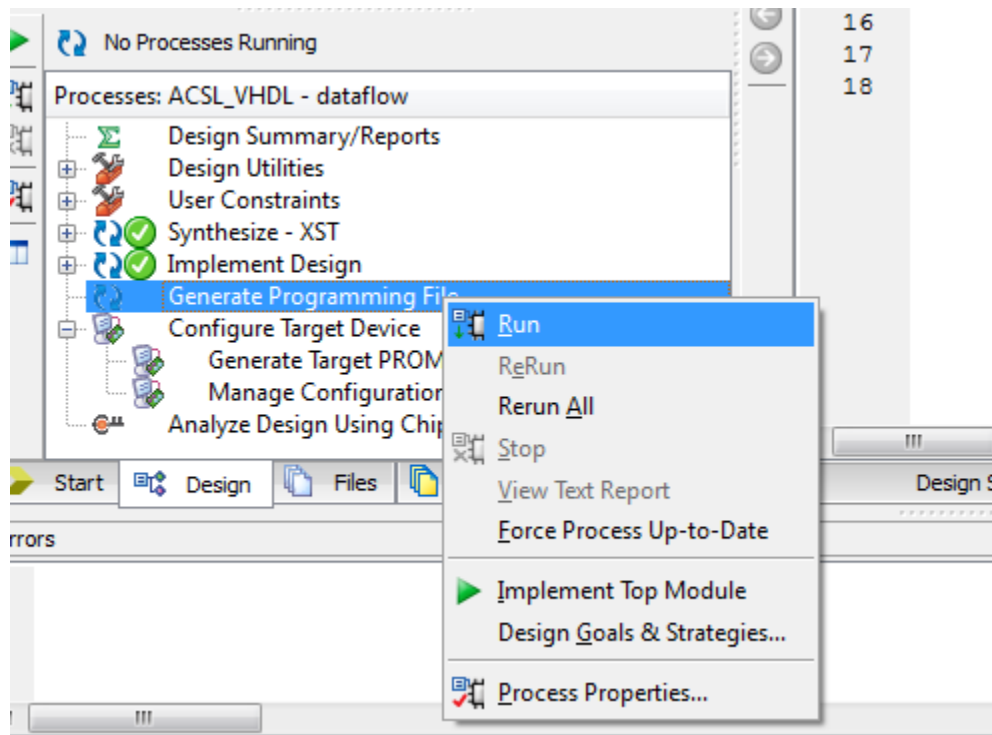
9. Setelah membuat file .ucf untuk konfigurasi input dan output, langkah selanjutnya yaitu melakukan “**Implement Design**”. Jalankan hal tersebut dengan cara klik kanan lalu pilih “run”. Tunggu hingga proses running selesai dan statusnya menjadi centang berwarna hijau.



Gambar 4.16 Proses implement desain.

Proses ini dapat diartikan dengan input dan output yang sudah dikonfigurasi sebelumnya, diimplementasikan atau digabungkan dengan source code yang sudah kita buat diawal

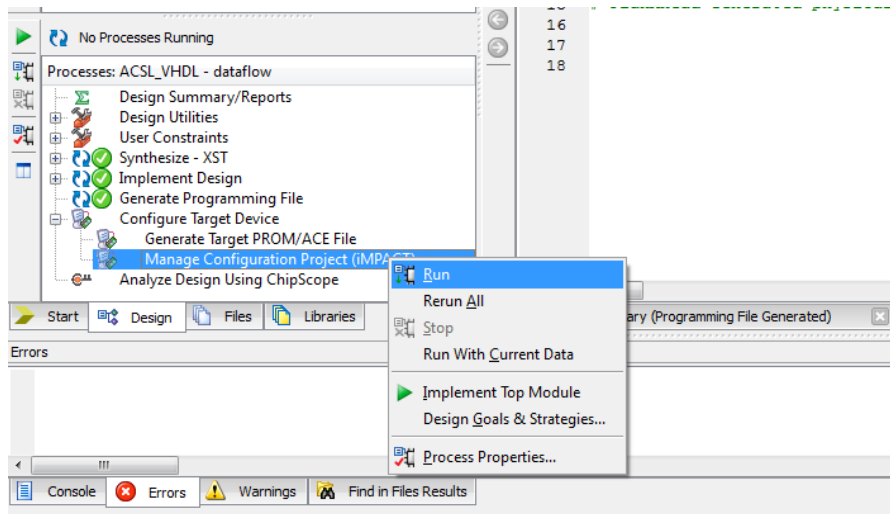
10. Setelah proses Implement Design selesai, selanjutnya yaitu menjalankan proses “Generate Programming File” dengan cara klik kanan lalu pilih “run”.



Gambar 4.17 Proses generate programming file.

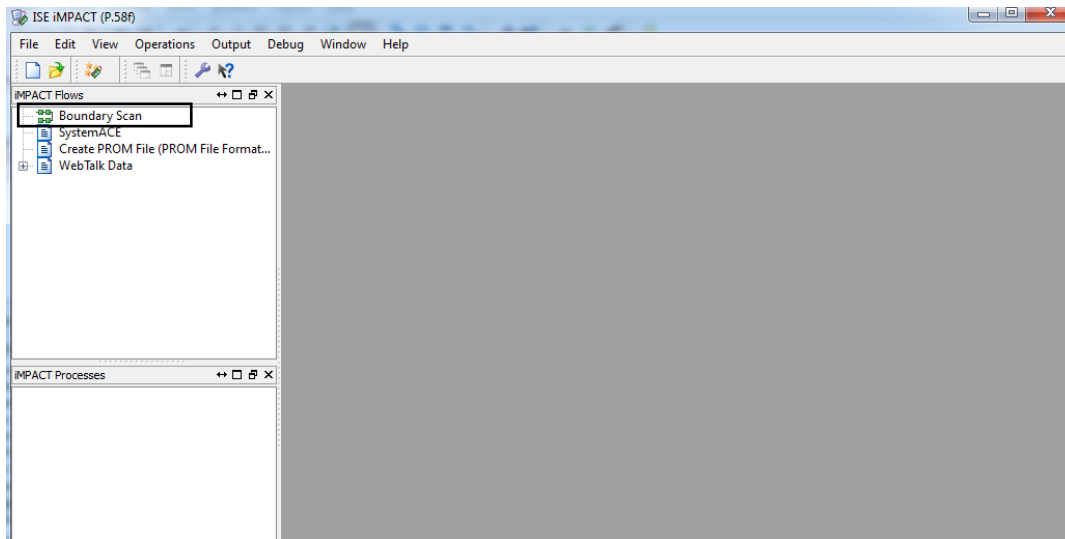
Proses ini merupakan langkah akhir untuk menjadikan file program yang dapat dimasukkan kedalam FPGA. Nantinya file inilah yang dapat dipakai berulang-ulang sesuai dengan kebutuhan dalam menggunakan FPGA. Karena seperti yang kita tahu bahwa FPGA memiliki sifat volatile, yaitu akan kehilangan fungsi atau program yang ada didalamnya ketika sumber dayanya dicabut. Namun ketika sudah memiliki file program ini, kita hanya perlu memasukkannya kembali atau flash program kedalam FPGA tersebut.

11. Langkah selanjutnya yaitu kita harus melakukan upload program yang sudah kita buat tadi kedalam FPGA. Pilih “**Manage Configuration Project (iMPACT)**”, lalu jalankan dengan cara klik kanan dan pilih “run”.



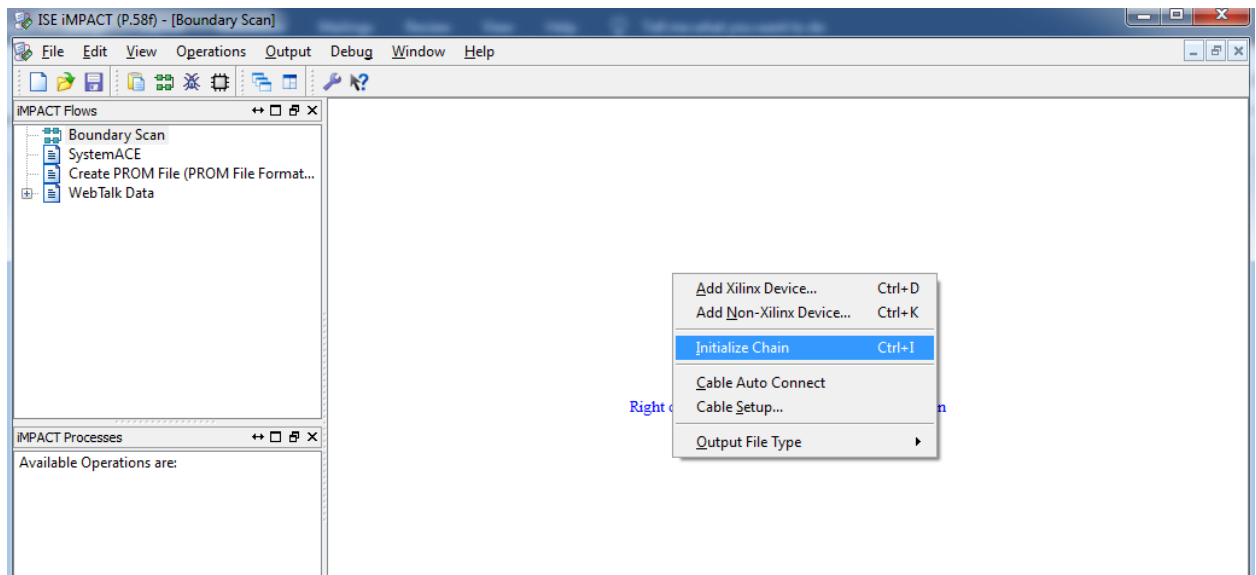
Gambar 4.17 Manage configuration project.

Nantinya akan muncul tab baru dan kita pilih “Boundary Scan”. Jalankan dengan cara klik 2 kali.



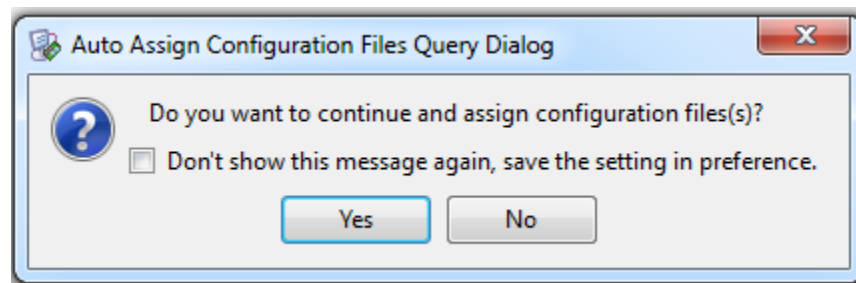
Gambar 4.18 Boundary scan.

Pasang kabel power pada FPGA. Selanjutnya klik kanan pada area kosong ditengah seperti ini dan pilih “**Initialize Chain**”.



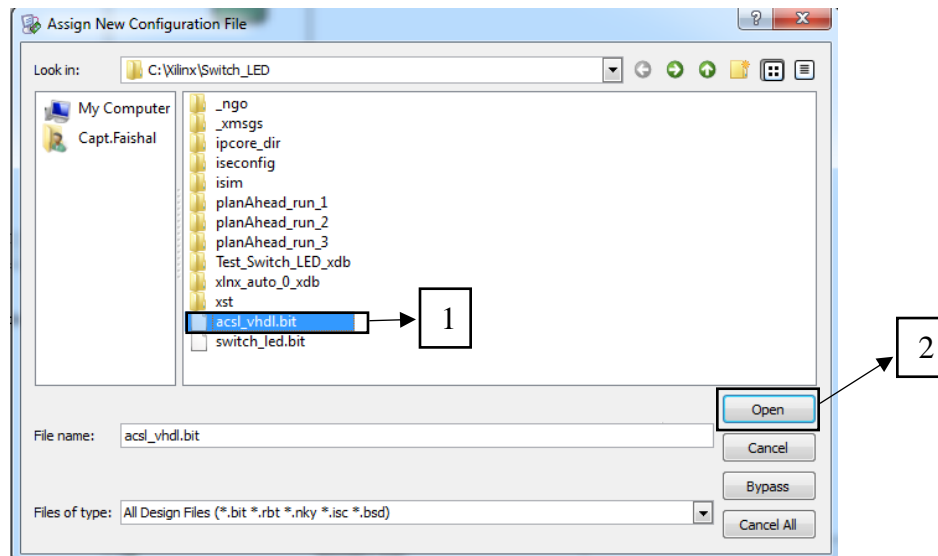
Gambar 4.19 Initialize chain.

Jika muncul pesan seperti ini, pilih “Yes”.



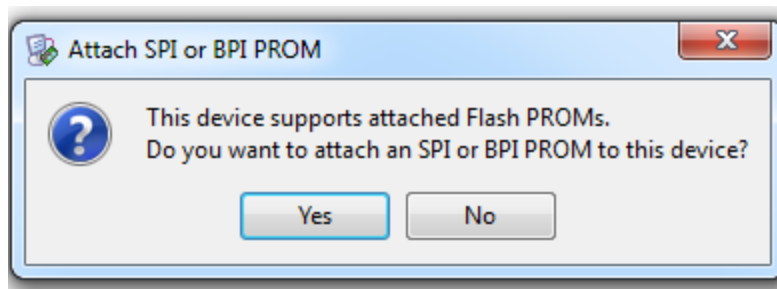
Gambar 4.20 Auto assign configuration.

Setelah itu, pilih file yang sudah kita buat lalu pilih “Open”.



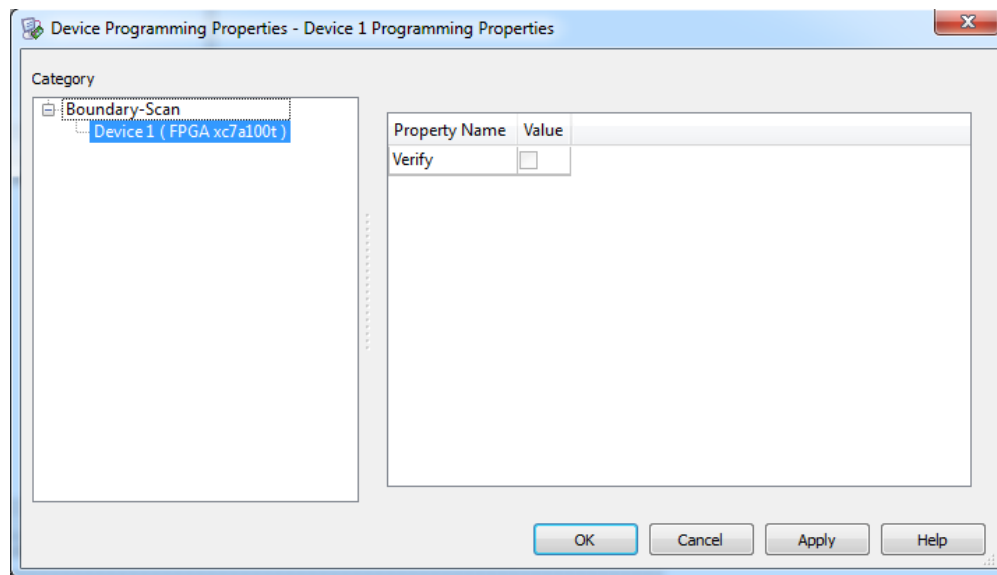
Gambar 4.21 Assign new configuration.

Jika muncul pesan seperti ini, pilih “No”.



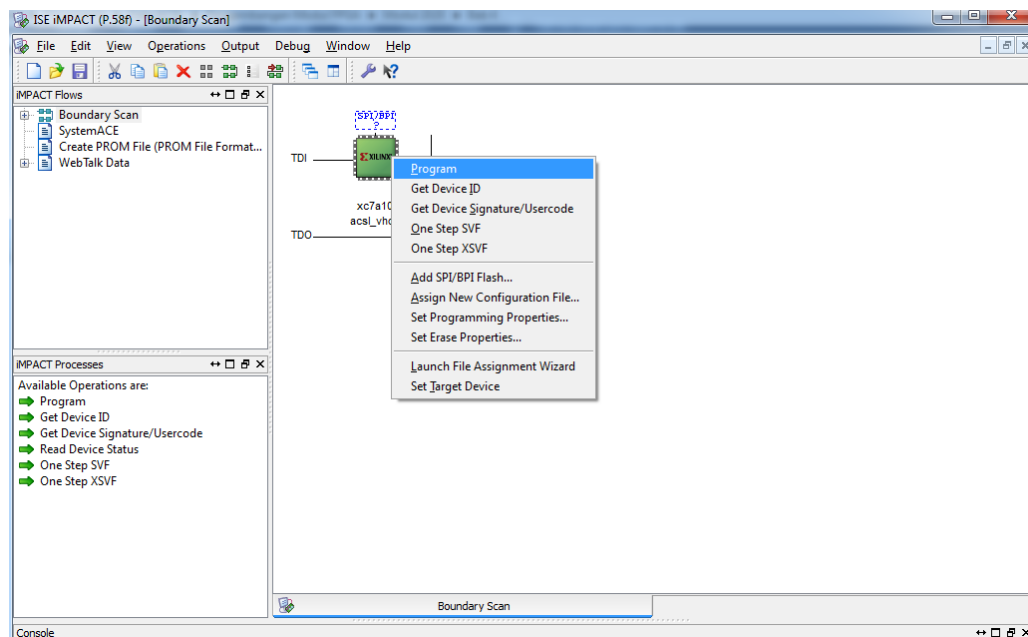
Gambar 4.22 Attach SPI or BPI PROM.

Selanjutnya pilih “Ok”.



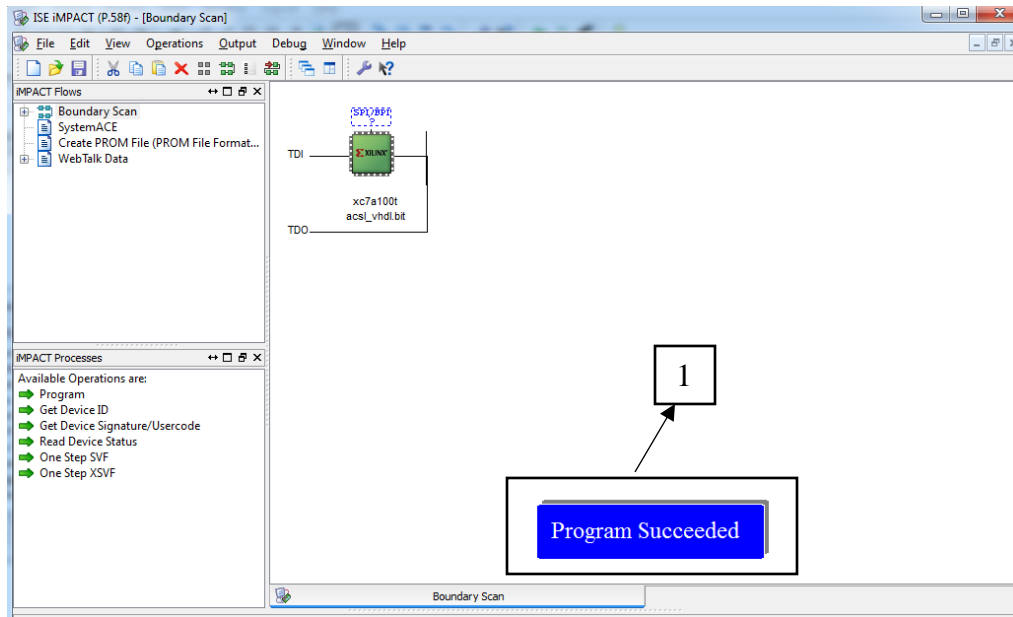
Gambar 4.23 Device programming properties.

Langkah terakhir yaitu kita melakukan upload program tersebut dengan cara klik kanan pada gambar IC yang berwarna hijau, lalu pilih “**Program**”.



Gambar 4.24 Upload program kedalam FPGA.

Tunggu beberapa saat untuk proses upload file program tersebut. Jika berhasil, maka akan muncul pesan seperti ini dan FPGA dapat digunakan sesuai dengan program yang telah dibuat sebelumnya.



Gambar 4.25 Upload program kedalam FPGA berhasil.

Note :

Simpan projek tersebut karena akan digunakan pada bab selanjutnya.