

1 D4 - TEKKOM B

Pengenalan FPGA dan ISE Design Tool Xilinx



Nama	:	Rohman Aditiya
Kelas	:	1 D4 Teknik Komputer B
NRP	:	3221600046
Dosen	:	Reni Soelistijorini B.Eng., MT.
Mata Kuliah	:	Praktikum Rangkaian Logika 2
Hari/Tgl.	:	Jumat, 4 Maret 2022



Pengenalan FPGA dan ISE Design Tool Xilinx

I. Tujuan

1. Dapat membuat *new project* menggunakan software ISE Xilinx.
2. Dapat membuat program rangkaian gerbang logika dasar dan memasukan program VHDL ke dalam *Board Xilinx* Spartan-3

II. Teori

a) FPGA dan VHDL

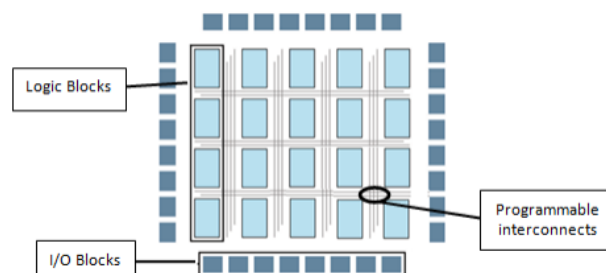
Field Programmable Gate Arrays (FPGA) adalah programmable device besar yang tersusun atas modul-modul logika independen yang dapat dikonfigurasi oleh pemakai yang di hubungkan melalui kanal-kanal routing yang dapat di program.

FPGA mempunyai kelebihan sebagai berikut:

1. Dikonfigurasi oleh *End User*
2. Tidak memerlukan proses Fabrikasi
3. Tersedia solusi yang mendukung chip customixed VLSI

FPGA ini sendiri juga merupakan IC tipe HDL yang mana pemakai dapat mendesain hardware sesuai yang diinginkan di dalam IC FPGA. Perkembangan FPGA saat ini berlangsung begitu cepat, 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.

Secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga element utama yaitu: *Input/Output Blok (IOB)*, *Configurable Logic Block (CLB)*, dan *Programmable Interconnect*. Fungsi logika dan interkoneksi FPGA ditentukan oleh 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).



Gambar 1. 1 Struktur Xilinx FPGA

Proses implementasi pada FPGA:

1. FPGAs diimplementasikan dengan *efficient CAD systems*
2. Design Entry dapat dilakukan dengan cara
 - a. Schematic capture program,
 - b. VHDL / Verilog
3. Logic Optimization
 - a. Memodifikasi Boolean Expression untuk mengcopyimalkan penggunaan
 - b. Logic berbasis area dan kecepatan atau kombinasi keduanya
4. Teknologi Mapping
 - a. Transformasi dari Boolean Expression atau VHDL kedalam bentuk circuit pada FPGA logic blocks
 - b. Area optimization – meminimalkan penggunaan block
 - c. Delay Optimixation – meminimalkan jumlah stage pada critical path
5. Placement
 - a. Setelah memetakan rangkaian ke dalam blok logika, selanjutnya algoritma penempatan digunakan untuk meletakkan masing-masing blok ke dalam FPGA array.
 - b. Meminimalkan total panjang interkoneksi yang diperlukan untuk penempatan blok logika
6. Routing Software
 - a. Mengatur FPGA's wire segment dan menentukan programmable switches untuk menghubungkan antar FPGA blocks
 - b. Memastikan 100% koneksi telah terbentuk
 - c. Meminimalkan propagation delay pada *time-critical connections*.
7. Programming unit
 - a. Mengkonfigurasi chip setelah placement dan routing step tuntas
 - b. Keseluruhan proses memakan waktu antara beberapa menit sampai beberapa jam

Secara umum proses kerja FPGA adalah sebagai berikut:

1. **Design Entry** dengan cara schematic, ABEL, VHDL, dan Verilog
2. **Implementation** yaitu Placement dan Routing serta bitsream generation. Juga analyze timing, view layout
3. **Download** meng-upload bitstream ke Hardware (FPGA chip)

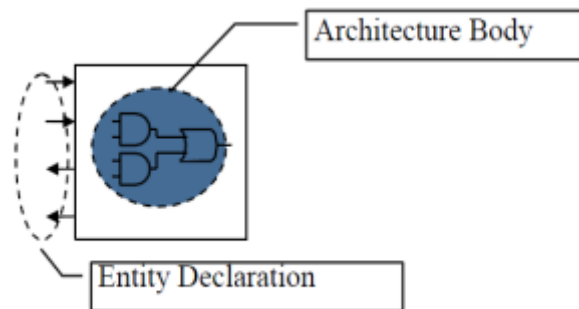
VHDL adalah singkatan dari **V** HSIC (*Very High Speed Integrated Circuits*) **Hardware Description Language**. Pada pertengahan tahun 1980-an Departemen pertahanan AS dan IEEE mensposori perkembangan bahasa deskripsi perangkat keras dengan tujuan untuk mengembangkan sirkuit terintegrasi dengan kecepatan tinggi. Hal ini telah menjadi salah satu bahasa standart industri yang digunakan untuk menggambarkan sistem

digital. Yang banyak digunakan Hardware lainnya bahasa deskripsi Verilog. Keduanya adalah bahasa yang memungkinkan anda untuk menggambarkan dan mensimulasikan sistem digital yang kompleks. Sebuah bahasa HDL ketiga adalah ABEL (Advanced Boolean Equation Language) yang khusus dirancang untuk Programmable Logic Device (PLD).

Meskipun bahasa terlihat sama seperti bahasa pemrograman konvensional terdapat beberapa perbedaan penting. Sebuah bahasa deskripsi hardware secara inhen paralel, yaitu perintah yang sesuai dengan gerbang logika yang dieksekusi (computed) secara paralel begitu masukan tiba. Sebuah program HDL meniru sistem fisik, biasanya digital. Hal ini memungkinkan penggabungan spesifikasi waktu (delay gate) serta menggambarkan sistem sebagai interkoneksi dari komponen yang berbeda.

Saat ini para desainer menggunakan *hardware description languages (HDLs)* untuk mendesain system. Bahasa yang umum dipakai adalah HDL, VHDL, dan Verilog. Kesemua bahasa membolehkan user untuk mendesain system digital dengan menuliskan behavior dari rangkaian digital. Pemrograman ini juga dapat disimulasikan operasi rangkaian dan *synthesize*-kan implementasi secara rangkai nyata dari rangkaian atau membuat aplikasi IC yang spesifik. Pada modul ini, pemrograman yang dipakai adalah pemrograman VHDL. VHDL mirip dengan bahasa pemrograman komputer seperti C/C++. Sebagai contoh, memiliki konstruksi untuk tugas variabel, pernyataan bersyarat, loop, dan fungsi. Dalam bahasa pemrograman komputer, compiler digunakan untuk menerjemahkan kode sumber tingkat tinggi ke kode mesin. Di VHDL sendiri digunakan untuk menerjemahkan kode sumber untuk deskripsi dari sirkuit hardware sebenarnya yang mengimplementasikan kode. Dari uraian ini, yang kita sebut netlist, perangkat digital fisik dapat dibuat secara otomatis. Keakurasian kode juga dapat dites menggunakan simulasi.

VHDL adalah bahasa pemrograman yang digunakan untuk mendeskripsikan *logic circuit* yang dikehendaki. Secara umum struktur dari pemrograman **VHDL** terdiri atas dua bagian **ENTITY** dan **ARCHITECTURE**.

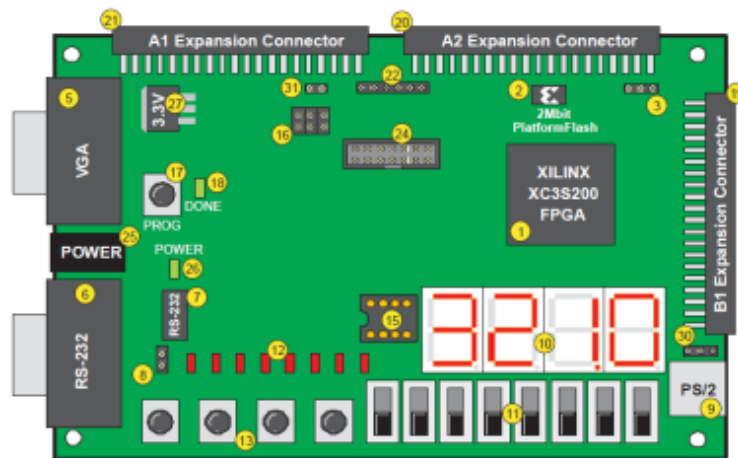


Gambar 1. 2 Struktur Pemrograman VHDL

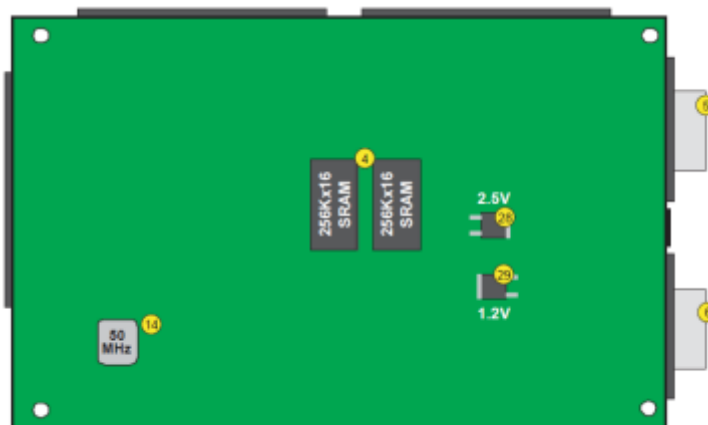
Bagian **ENTITY** menjelaskan spesifikasi pin-pin eksternal yang digunakan dari circuit atau rancangan yang akan dibuat. Bagian **ARCHITECTURE** menjelaskan atau mewakili fungsi sesungguhnya dari circuit atau rangkaian.

b) XILINX SPARTAN 3

Modul yang digunakan untuk mendesain chip adalah modul Xilinx Spartan 3. Fitur-fitur yang disediakan, seperti pada gambar 1.3 dan 1.4.



Gambar 1. 3 Modul Xilinx Spartan 3 tampak atas



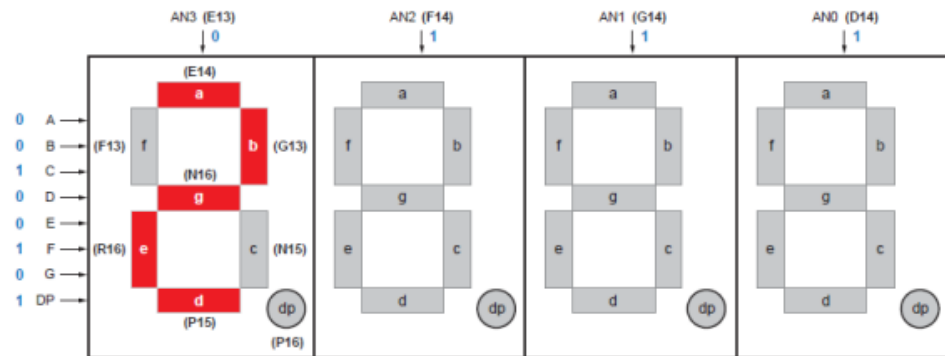
Gambar 1. 4 Modul Xilinx Spartan 3 tampak bawah

Secara ringkas fitur-fitur yang dimiliki oleh Xilinx Spartan 3 adalah:

1. 200,000-gerbang Xilinx Spartan-3XC3S200 FPGA in a 256-ball thin Ball Grid Array package (XC3S200FT256).
2. 2Mbit Xilinx XCF02S Platform Flash, in-system programmable configure PROM
3. Jumper Option allow FPGA application to read PROM data or FPGA configuration from other source
4. 1M-byte of Fast Asynchronous SRAM.
5. 3-bit, 8 color VGA display port.
6. 9-pin RS-232 Serial Port
7. RS-232 transceiver/level translator
8. Second RS-232 transmit and receive channel available on board test point
9. PS/2-style mouse/keyboard port
10. 4 karakter, seven-segment LED display
11. 8 slide switches
12. 8 LED outputs
13. 4 push button
14. 50 MHz crystal oscillator clock source
15. Socket for an auxiliary crystal oscillator clock source
16. FPGA configuration mode selected via jumper settings
17. Push button switch to force FPGA reconfiguration
18. Indikator LED ketika FPGA sukses dikonfigurasi
19. 40-pin expansion connection port to extend and enhance
20. 40-pin expansion connection port to extend and enhance
21. 40-pin expansion connection port to extend and enhance
22. JTAG port for low-cost download cable,
23. Digilent JTAG download/debugging cable connects to PC parallel port
24. JTAG download/debug port compatible with the xilinx Parallel Cable IV and MultiPRO Desktop Tool
25. AC power adapter input for included international unregulated +5V power Supply
26. Power -on indicator LED
27. On-board 3.3V regulator
28. 2.5V regulator
29. 1.2V regulator

FPGA koneksi ke Seven Segment.

Pada Seven Segment ada 8 jalur kontrol sinyal yang digunakan untuk menghidupkan LED untuk kombinasi angka tertentu. Dan setiap satu seven segment memiliki satu jalur control common anoda sebagai Enable, detail jalur tersebut seperti pada gambar 1.5.



Gambar 1. 5 Koneksi FPGA ke Seven Segment pada modul

Untuk tabel pin FPGA dengan seven segment ditunjukkan pada tabel dibawah ini.

Segment	FPGA Tabel
A	E14
B	G13
C	N15
D	P15
E	R16
F	F13
G	N16
DP	P16

Tabel 1.1 FPGA Koneksi port ke Seven Segment (aktif Low)

Anode Control	AN3	AN2	AN1	AN0
FPGA Pin	E13	F14	G14	D14

Tabel 1.2 sinyal Common Enable Seven Segment (Kontrol Anoda)

FPGA koneksi dengan Switch

Pada board Spartan 3, tombol Switch dilambangkan dengan SW. Misal Switch ke-1 ditulis dengan SW1, dimodul ini switch yang dimiliki sebanyak 8buah (SW7-SW0).

Switch	SW7	SW6	Sw5	SW4	SW3	SW2	SW1	SW0
FPGA pin	K13	K14	J13	J14	H13	H14	G12	F12

Tabel 1.3 Koneksi FPGA dengan SWITCH

Ketika SW pada posisi UP atau posisi ON, pin FPGA terhubung ke VCC/Logic High. Sedangkan jika SW pada posisi Down atau posisi OFF, pin FPGA terhubung ke Ground/Logika Low.

FPGA koneksi dengan Push Button

Pada board Spartan 3 memiliki 4 tombol push button yang ditukiskan pada board dengan BTN (BTN3-BTN0)

Push Button	BTN3	BTN2	BTN1	BTN0
FPGA Pin	L14	L13	M14	M13

Tabel 1.4 Koneksi FPGA dengan Push Button

Ketika menekan push button, pin FPGA terhubung dengan VCC

FPGA koneksi dengan LED

Pada board Spartan 3 memiliki 8 LED yang dituliskan pada board dengan LD (LD7-LD0)

Led	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
FPGA Pin	P11	P12	N12	P13	N14	L12	P14	K12

Tabel 1.5 Koneksi FPGA ke LED

Katoda dari Led terhubung ke Ground dengan resistor 270Ω.

c) Xilinx ISE Design Suite 13.1

Xilinx ISE (integrated Synthesis Environment) adalah sebuah perangkat lunak yang diproduksi oleh Xilinx untuk sintesis dan analisa desain HDL, yang memungkinkan untuk *compile* desain, analisa waktu, RTL diagram, dan menkonfigurasi perangkat target/device dengan programmer.

Xilinx ISE digunakan hanya untuk produk FPGA dari Xilinx. Xilinx ISE digunakan untuk mensintesa rangkaian dan desain sedangkan Model Sim logic Simulator digunakan untuk pengujian system level testing, juga terdapat fitur tambahan termasuk program lain seperti *Embedded Development Kit (EDK)*, *Software Developments Kit (SDK)*, dan ChipScope Pro.



Gambar 1. 6 ISE Project Navigator

III. Alat dan Bahan

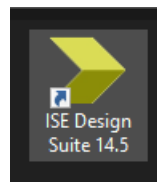
Alat dan Bahan untuk praktikum ini adalah:

1. PC beserta software ISE Design Suite 13.1
2. Modul Xilinx Spartan 3
3. Downloader JTAG
4. Power Supply DC 5V

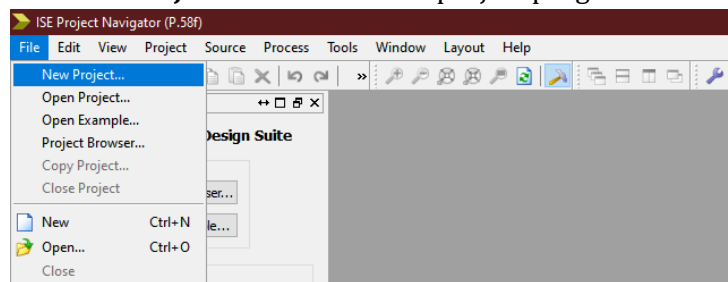
IV. Hasil Percobaan Membuat New Project pada ISE Design Suite 13.1

Pada percobaan ini akan dibuat sebuah program yang mengimplementasikan rangkaian AND.

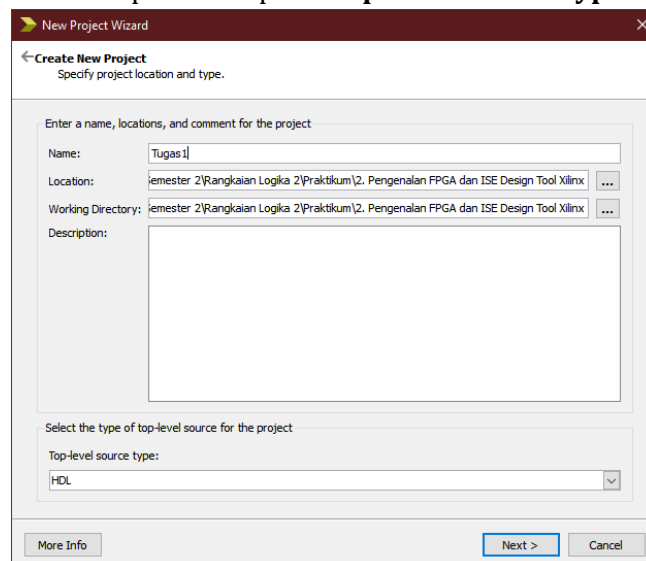
1. Buka aplikasi **ISE Design Suite 14.5** melalui shortcut yang ada di Desktop.



2. Setelah window **ISE Project Navigator** dari ISE Design Suite 14.5 muncul, klik **File -> New Project** untuk membuat project program VHDL baru.



3. Isi nama project, lokasi direktori penyimpanan project, dan direktori working. Kemudian pilih **HDL** pada **Top-Level source type**. Klik **Next**.



4. Pilih **Family= Spartan3, Device=XC3S200, Package = FT256, Speed =-4**. Pilih **Preferred Language=VHDL**. Kemudian klik **Next**.

New Project Wizard

Project Settings
Specify device and project properties.

Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3
Device	XC3S200
Package	FT256
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info < Back **Next >** Cancel

5. Setelah mengonfigurasi project, klik **Finish**.

New Project Wizard

Project Summary
Project Navigator will create a new project with the following specifications.

Project:

- Project Name: Tugas1
- Project Path: E:\PENS\CE B 2021\Semester 2\Rangkaian Logika 2\Prakti
- Working Directory: E:\PENS\CE B 2021\Semester 2\Rangkaian Logika 2\P
- Description: Top Level Source Type: HDL

Device:

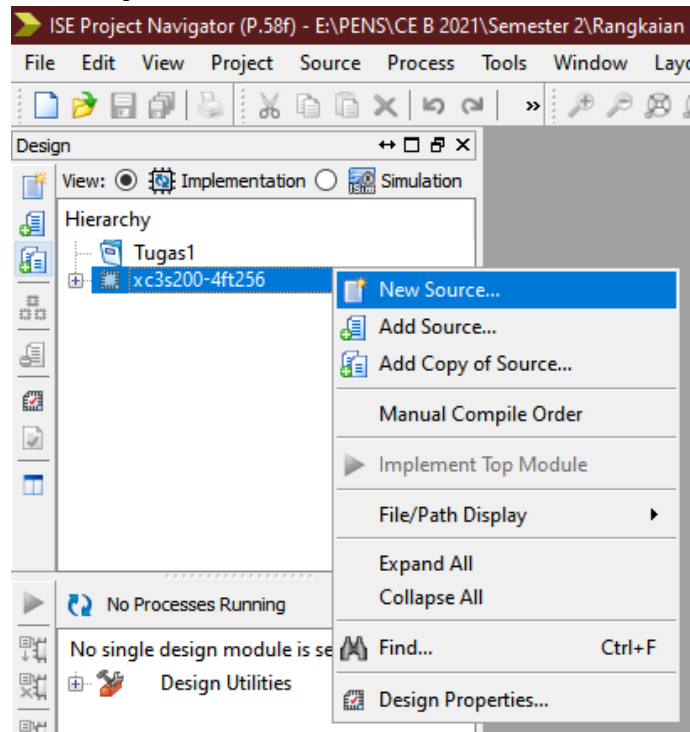
- Device Family: Spartan3
- Device: xc3s200
- Package: ft256
- Speed: -4

Top-Level Source Type: HDL
 Synthesis Tool: XST (VHDL/Verilog)
 Simulator: ISim (VHDL/Verilog)
 Preferred Language: VHDL
 Property Specification in Project File: Store all values
 Manual Compile Order: false
 VHDL Source Analysis Standard: VHDL-93

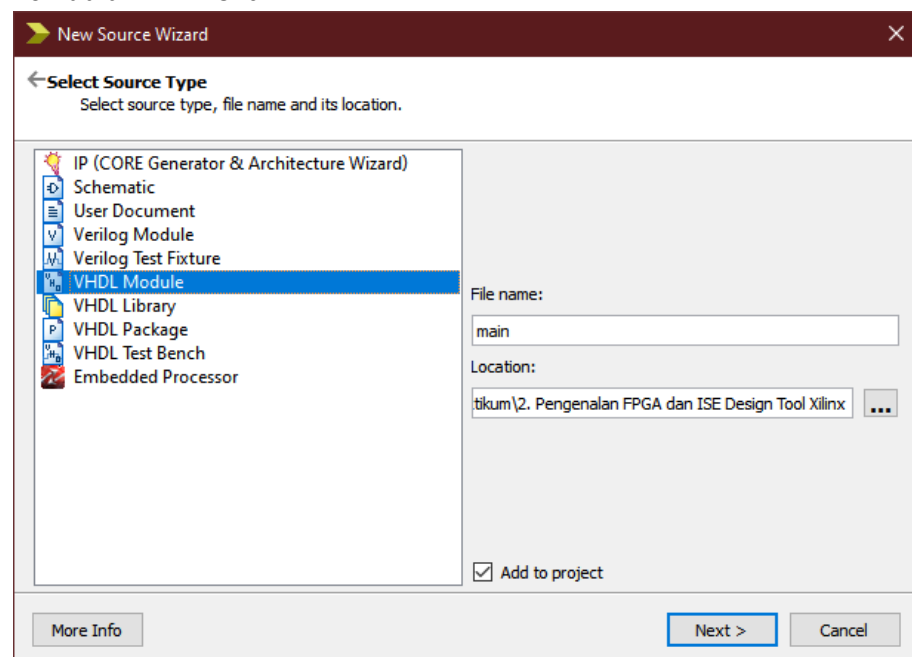
Message Filtering: disabled

More Info < Back **Finish** Cancel

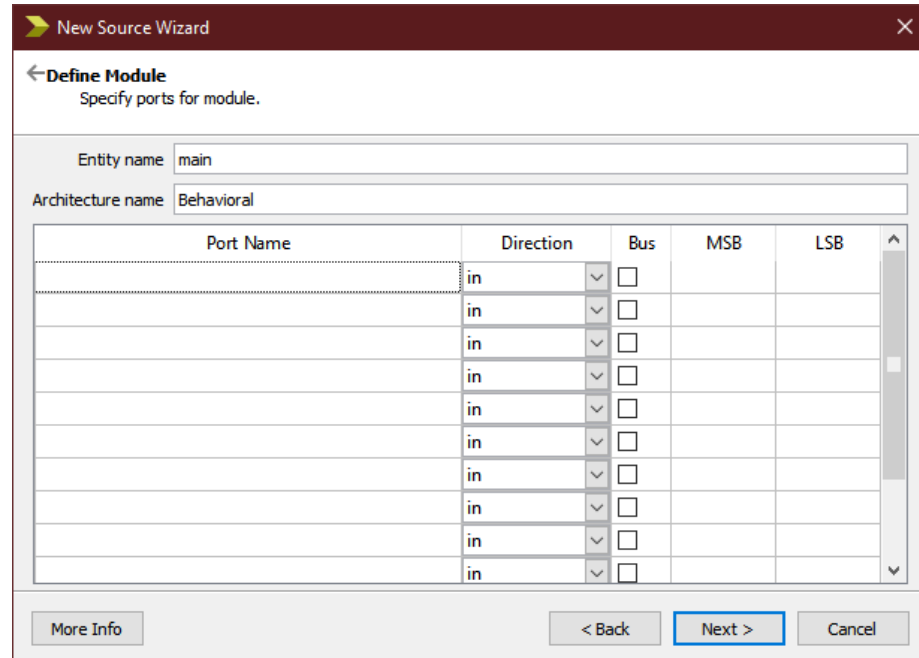
6. Setelah itu akan muncul project pada Hierarchy, klik kanan pada nama project kemudian pilih **New Source** untuk membuat file .hdl



7. Pada New source wizard, pilih **VHDL module** pada source type, isikan nama file. Location adalah lokasi tempat menyimpan file .vhdل tersebut. Kemudian klik **next**.



8. Define module untuk mendefinisikan pin-pin yang digunakan, pada bagian ini langsung dilewati saja. Karena define pin akan kita konfigurasi di akhir. Pilih **next**.



New Source Wizard

← **Define Module**
Specify ports for module.

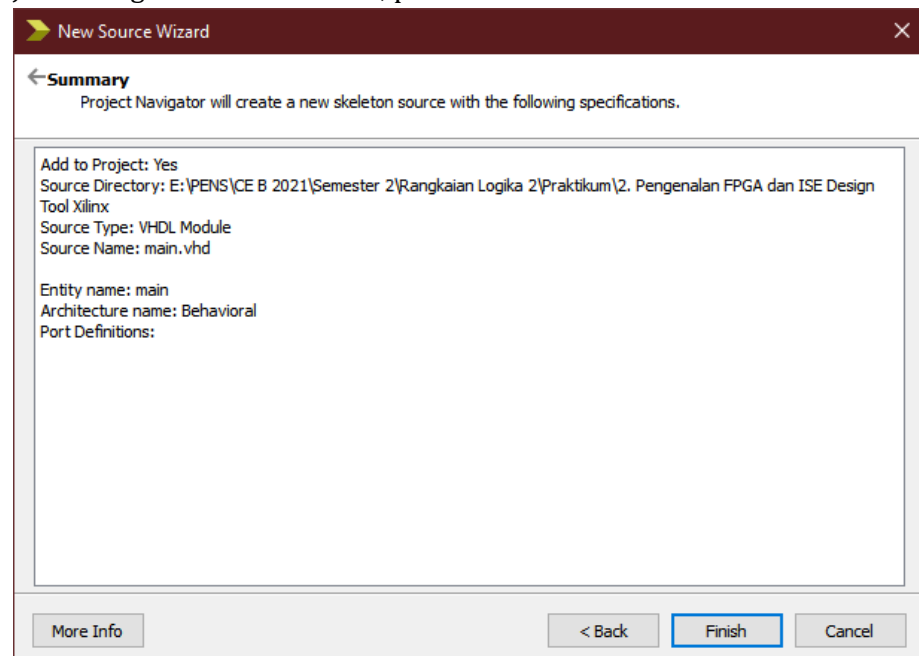
Entity name:

Architecture name:

Port Name	Direction	Bus	MSB	LSB
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

More Info < Back **Next >** Cancel

9. Jika konfigurasi sudah selesai, pilih finish



New Source Wizard

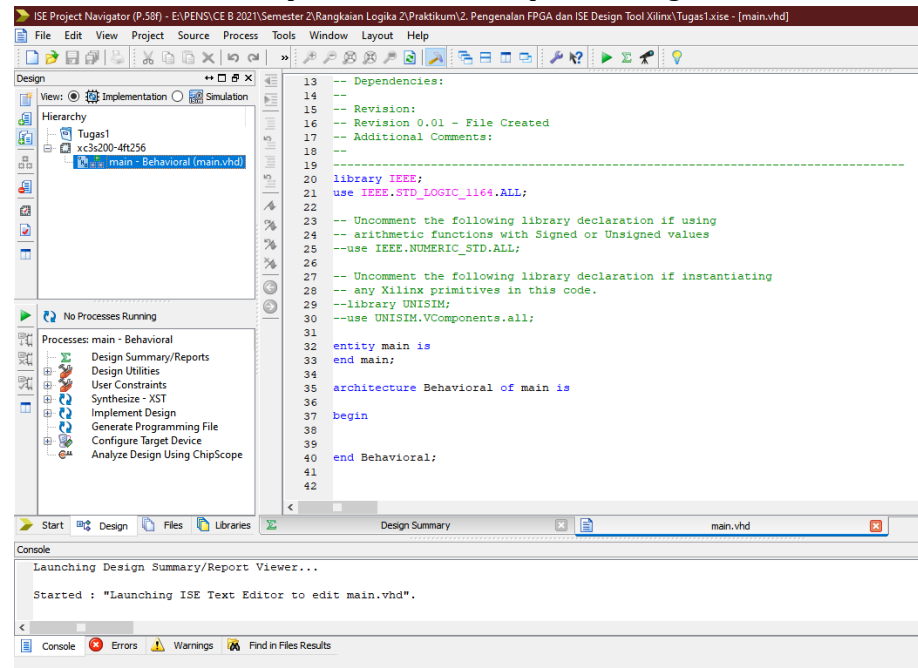
← **Summary**
Project Navigator will create a new skeleton source with the following specifications.

Add to Project: Yes
 Source Directory: E:\PENS\CE B 2021\Semester 2\Rangkaian Logika 2\Praktikum\2. Pengenalan FPGA dan ISE Design Tool Xilinx
 Source Type: VHDL Module
 Source Name: main.vhd

Entity name: main
 Architecture name: Behavioral
 Port Definitions:

More Info < Back **Finish** Cancel

10. Kemudian akan ditampilkan window template coding



11. Program dilengkapi dengan script program berikut

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity main is
    port(
        a : in STD_LOGIC;
        b : in STD_LOGIC;
        y : out STD_LOGIC
    );
end main;

architecture Behavioral of main is

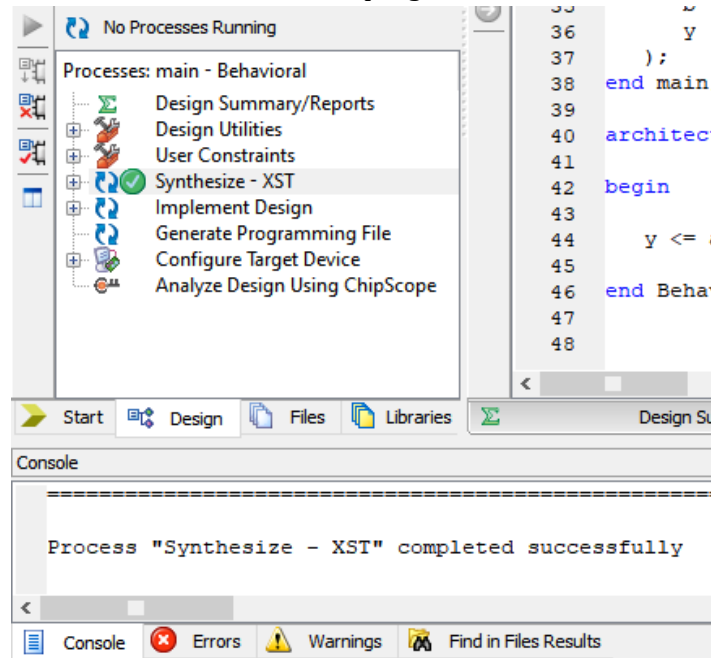
begin

    y <= a and b;

end Behavioral;

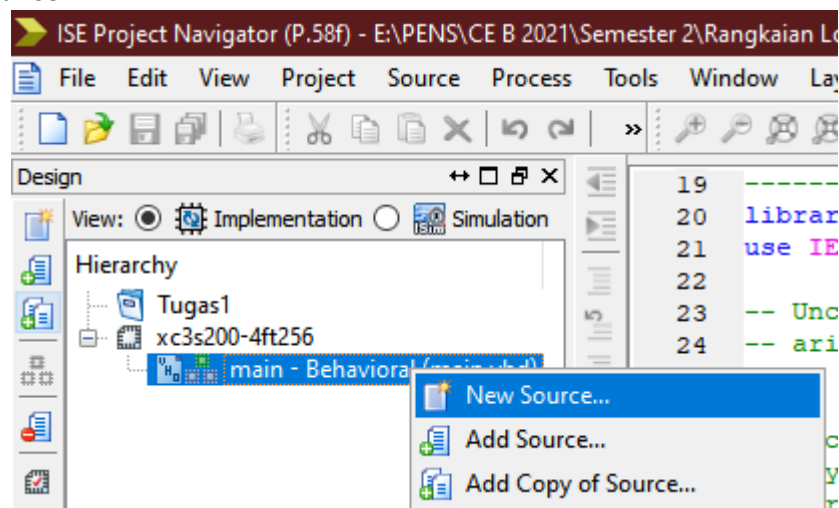
```

12. Bila selesai membuat program, save project kemudian double klik **Synthesize - XST** untuk mensistesis program.

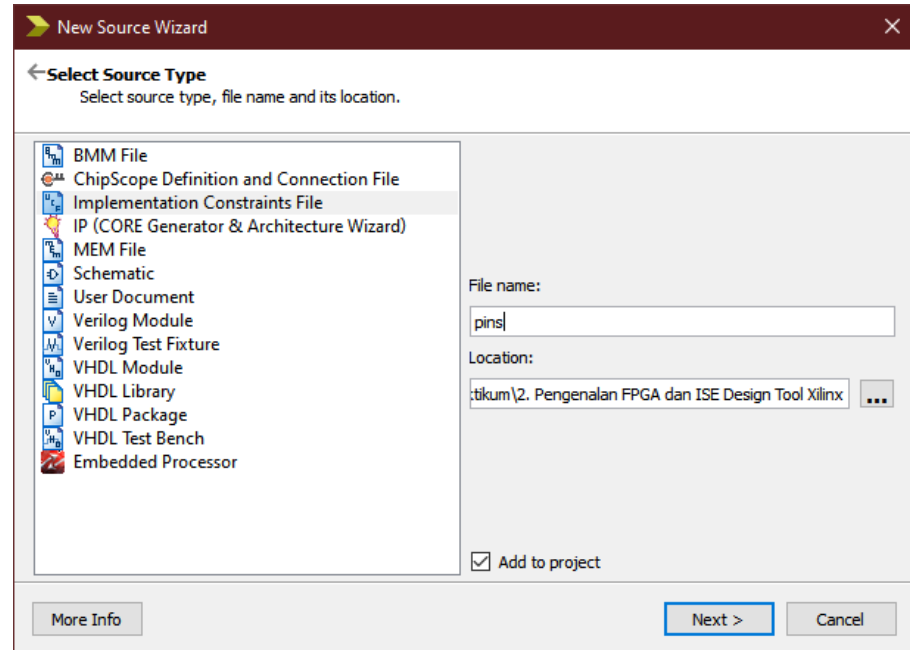


Ketika selesai mensintesis program, perhatikan pada Console window dan tanda disamping Synthesize - XST. Bila program sudah benar tanpa ada kesalahan maka akan tampil **“Process ‘Synthesize - XST’ completed successfully”** pada console window dan tanda centang pada Synthesize - XST seperti gambar diatas.

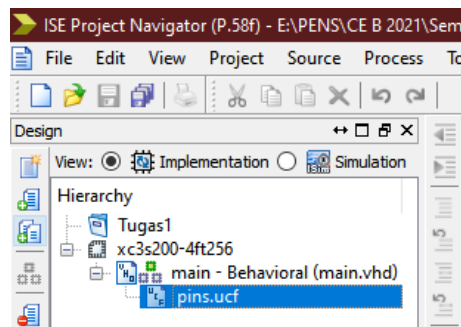
13. Kemudian klik kanan pada nama file project di kolom hierarchy, pilih **New Source**.



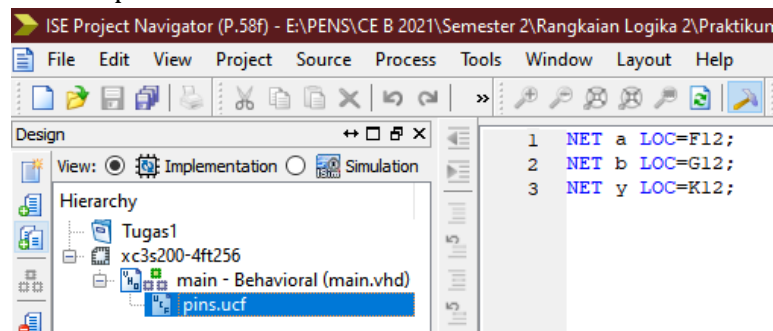
14. Pilih Implementasi Constrains File untuk mendefinisikan pin-pin dari program pada Spartan 3. Kemudian isi nama file. (File ini akan berekstensi .ucf).



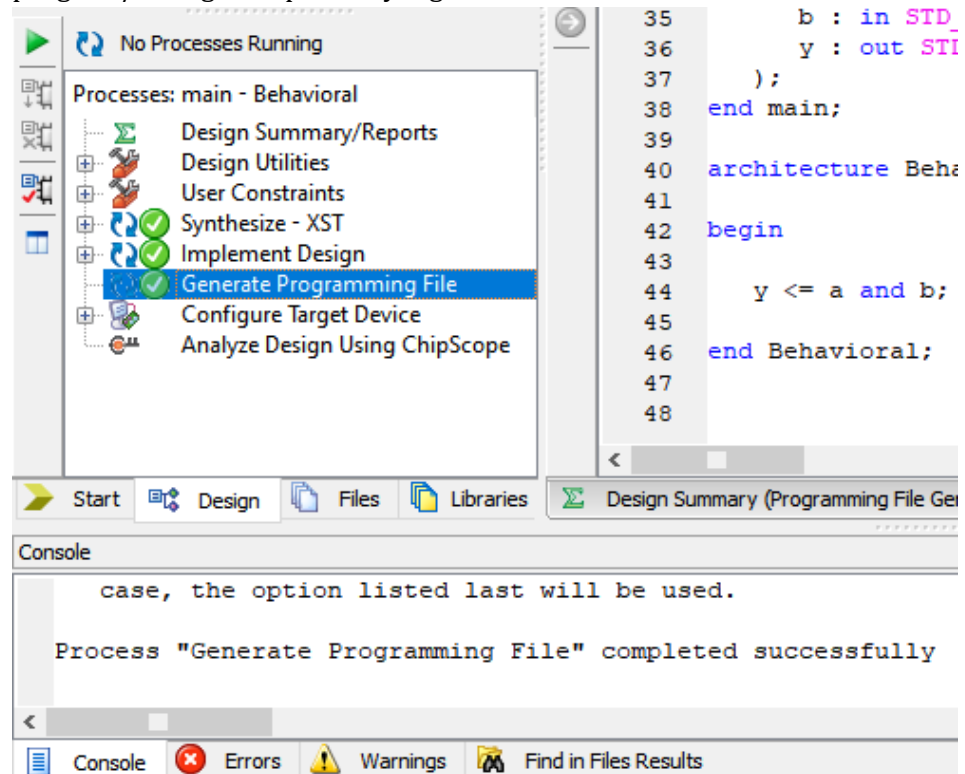
15. File .ucf yang telah kita buat berada pada sub program .vhd. Untuk membuka klik tanda panah yang berbentuk tanda plus di file project. Klik file .ucf yang sudah kita buat, akan muncul template kosong di window sebelah kanan.



16. Definisikan pin-pin sesuai dengan board Spartan 3. Lihat datasheet pada modul Xilinx Spartan3.



17. Setelah selesai, save project anda. Kemudian klik kembali file project .vhd untuk menampilkan window process desain. Setelah selesai klik Generate Programming File pada kolom Processes. Jika centang hijau maka success dan program siap di downloadkan. Jika Adatanda silang merah berarti program/konfigurasi pin ada yang salah.



V. Latihan

Buatlah sebuah program gerbang logika dasar dimana input-nya adalah tombol SW0 dan SW1 sedangkan output-nya adalah Led (LD0-LD4) dimana LD0 menampilkan hasil gerbang logika AND dari input SW0 dan SW1, LD1 untuk gerbang logika OR dari input SW0 dan SW1, LD2 untuk gerbang logika NOT dari input SW0, LD3 untuk gerbang logika NAND dari input SW0 dan SW1, LD4 untuk gerbang logika NOR dari input SW0 dan SW1.

```

y[0] <= a and b;
y[1] <= a or b;
y[2] <= not a;
y[3] <= a nand b;
y[4] <= a nor b;

```


Script Program VHDL

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity main is
    port(
        a, b : in std_logic;
        y : out std_logic_vector (4 downto 0)
    );
end main;

architecture Behavioral of main is

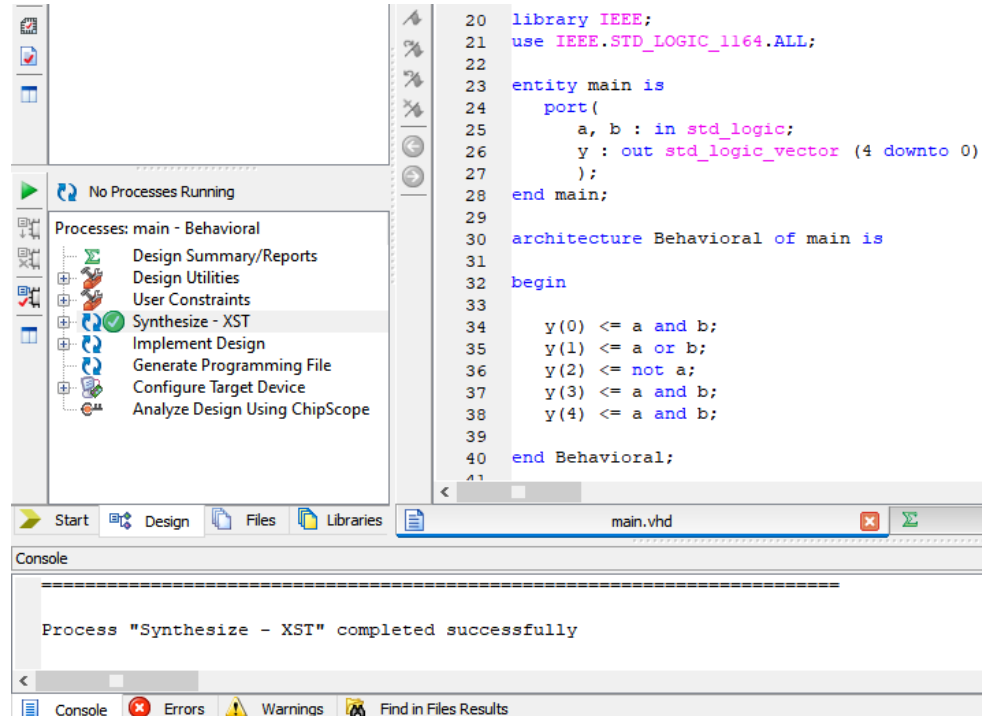
begin

    y(0) <= a and b;
    y(1) <= a or b;
    y(2) <= not a;
    y(3) <= a and b;
    y(4) <= a and b;

end Behavioral;

```

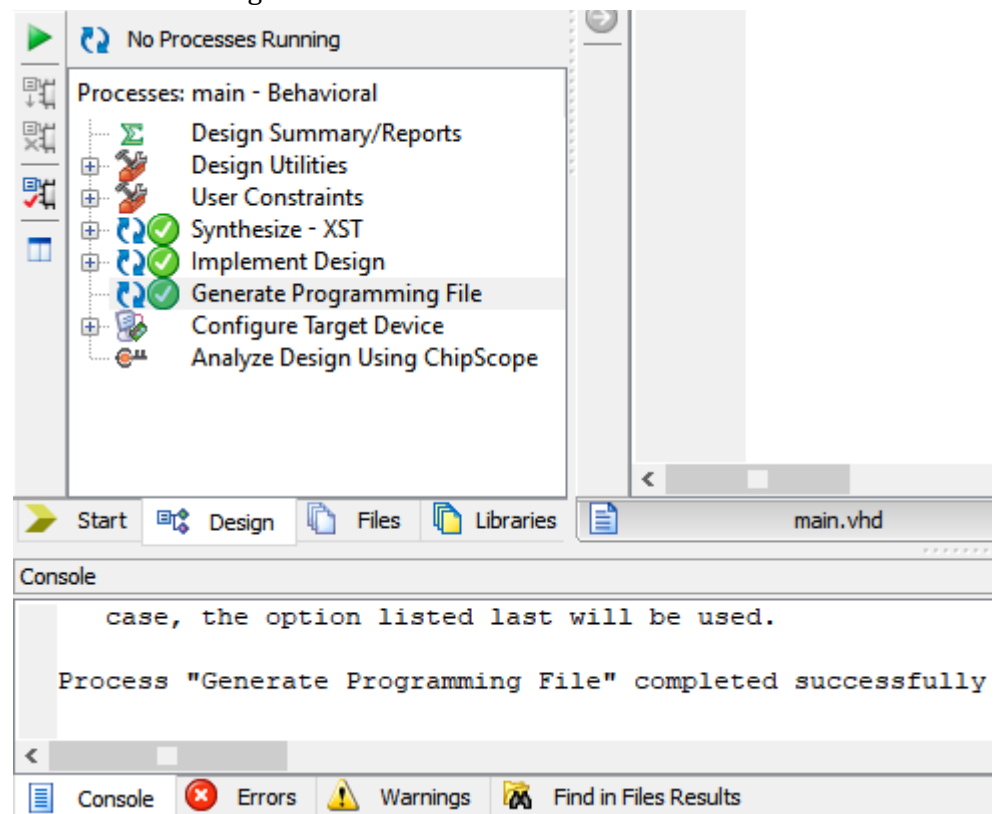
Hasil Sintesis Program



Definisi Pin – Pin yang Digunakan

```
NET a LOC = F12;  
NET b LOC = G12;  
NET y(0) LOC = K12;  
NET y(1) LOC = P14;  
NET y(2) LOC = L12;  
NET y(3) LOC = N14;  
NET y(4) LOC = P13;
```

Hasil Generate Program



VI. Analisa

VHDL memiliki syntax yang hampir sama dengan C/C++ namun memiliki kegunaan yang berbeda. Program VHDL dapat disintesis. Pada program diatas untuk menentukan pin-pin dan logic, konfigurasi program harus diletakkan pada bagian entity. Bila ingin menentukan input maka gunakan `in std_logic;` sedangkan output menggunakan `out std_logic;`. Kita juga bisa menentukan banyak input/output pada signal names. Bisa dilihat pada program diatas bahwa sinyal `y` memiliki 5 output (0-4) menggunakan `out std_logic_vector (4 downto 0)`.

Setelah menentukan input, output, pin, dan sinyal kita dapat menentukan bagaimana rangkaian beroperasi dan diterapkan dengan menambahkan program pada bagian architecture. Pada architecture ini kita bisa memberikan nilai sinyal dengan ekspresi gerbang logika. Syntax operasinya menggunakan `<=` untuk memberikan nilai sinyal. Contoh pada program diatas:

```
y(0) <= a and b;
```

Perintah diatas memberikan nilai sinyal `y` yang ke 0 dengan hasil ekspresi `a` yang di and kon dengan `b`.

Untuk menghubungkan sinyal dengan pin yang ingin digunakan kita perlu membuat Implementation Constraint File. Pada file tersebut kita isi program untuk mendefinisikan pin apa saja yang dihubungkan dengan sinyal. Pada program diatas kita ingin menghubungkan `a` dengan SW0 board Spartan 3. SW0 Spartan 3 menggunakan pin F12. Sehingga penulisannya menjadi:
`NET a LOC = F12;`

Jika semua sudah selesai, save project lalu Generate Programming File. Bila tidak ada error maka program berhasil terbuat.