

I. Tujuan Praktikum :

- Praktikan Dapat Memahami Fungsi Push Button.
- Praktikan Dapat Memahami Fungsi Keyboard
- Praktikan Dapat Mengimplementasikan Push Button dan Keyboard pada FPGA.

II. Dasar Teori :

- Pengenalan Push Button.
- Pengenalan Keyboard.
- Penerapan Push dan Keyboard pada FPGA.

III. Peralatan :

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

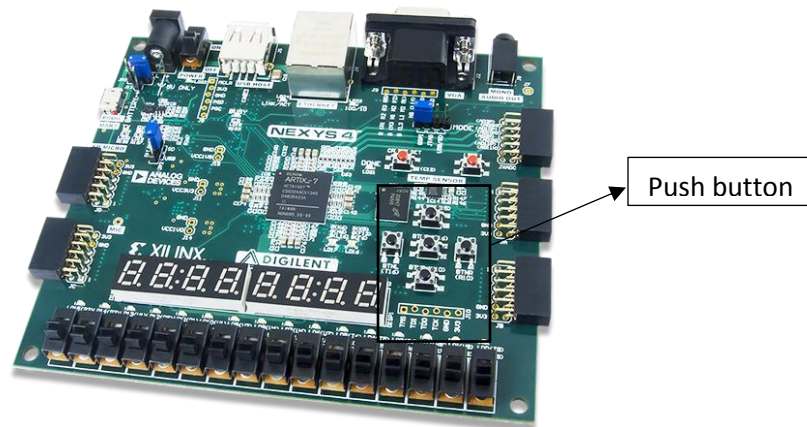
6.1 Pengertian

6.1.1 Push Button

Push button adalah saklar yang beroperasi dengan cara ditekan dan bisa melakukan dua fungsi berbeda, yaitu menutup sirkuit bila ditekan atau justru membuka sirkuit bila ditekan. Jika tekanan dilepaskan atau terjadi tekanan berikutnya maka akan menormalkan kembali tombol ke posisi semula. Secara umum, komponen ini memiliki cara kerja yang sama dengan switch, namun hanya berbeda pada cara pengoperasiannya saja.

Komponen ini juga berfungsi sebagai pemberi sinyal pada rangkaian listrik ketika atau selama bagian knopnya ditekan maka alat ini akan bekerja sehingga kontak-kontaknya akan terhubung untuk jenis Normally Open (NO) dan akan terlepas untuk jenis Normally Close (NC).

Pada FPGA Nexys 4 terdapat total 7 push button, namun yang dapat kita gunakan hanya 5 saja. Karena dua push button yang berwarna merah sudah memiliki fungsi sendiri yaitu untuk CPU reset.

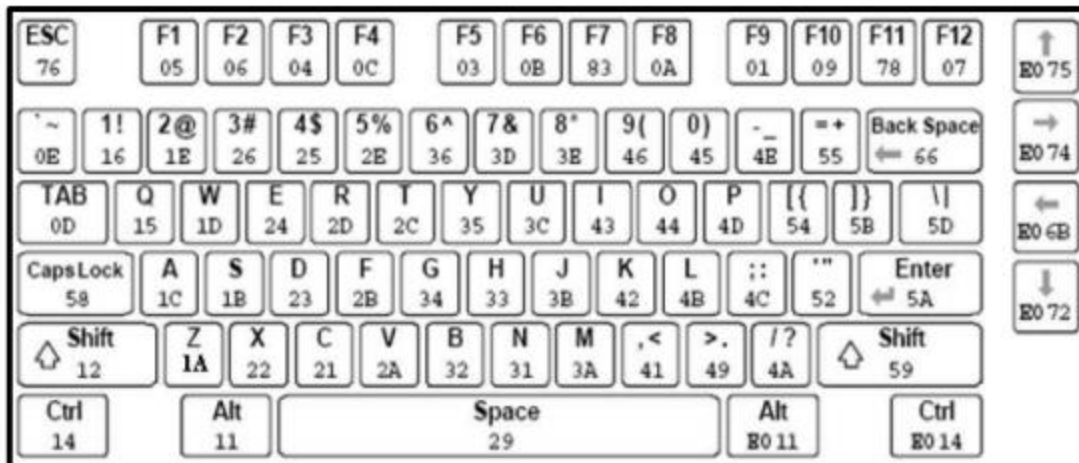


Gambar 6.1 Push button.

6.1.2 Keyboard

Keyboard adalah sebuah perangkat keras (*hardware*) pada komputer yang berfungsi sebagai alat untuk input data yang berupa huruf, angka dan simbol. Atau definisi keyboard yaitu suatu perangkat keras pada komputer yang berbentuk papan dan memiliki berbagai macam tombol yang dimana tombol tersebut fungsinya berbeda tergantung pada penekanannya yang bisa menghasilkan proses yang sesuai dengan keinginan penggunaanya. Selain sebagai alat untuk men-inputkan data keyboard juga memiliki berbagai macam fungsi tombol dengan kombinasi yang khusus. Umumnya port keyboard pada komputer ada 2 macam diantaranya PS2 dan USB akan tetapi seiring berkembangnya teknologi, sekarang keyboard sudah ada yang memakai *wireless* atau tanpa menggunakan kabel.

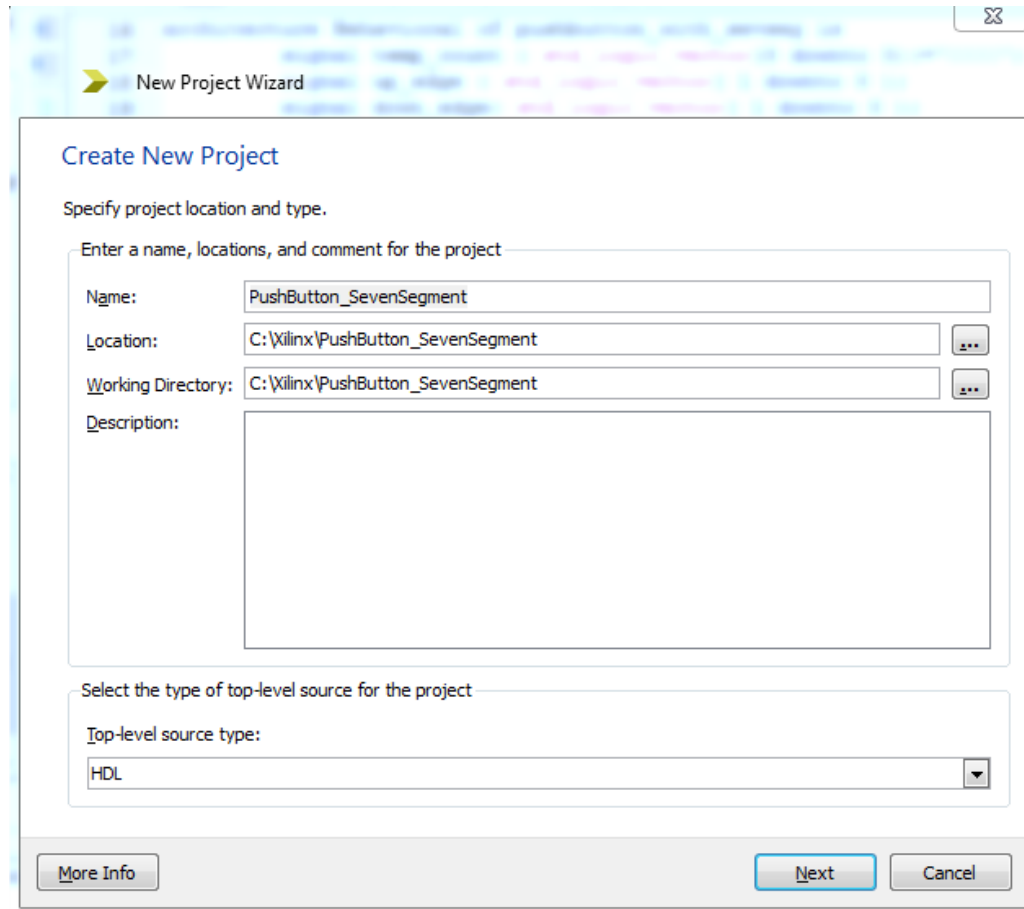
Pada keyboard, setiap kali salah satu tombol ditekan atau dilepas keyboard akan mengirimkan kode ke host. Kode yang dikirimkan ke host tersebut dinamakan *Scan Code*. Sebagai contoh, bila scan code tombol “i” adalah 43H (0100 0011), maka ketika tombol “i” ditekan, keyboard akan mengirimkan kode 43H untuk dapat diterjemahkan menjadi sesuai apa yang telah diinput. Berikut merupakan masing-masing scan code dari keyboard :



Gambar 6.2 Scan code.

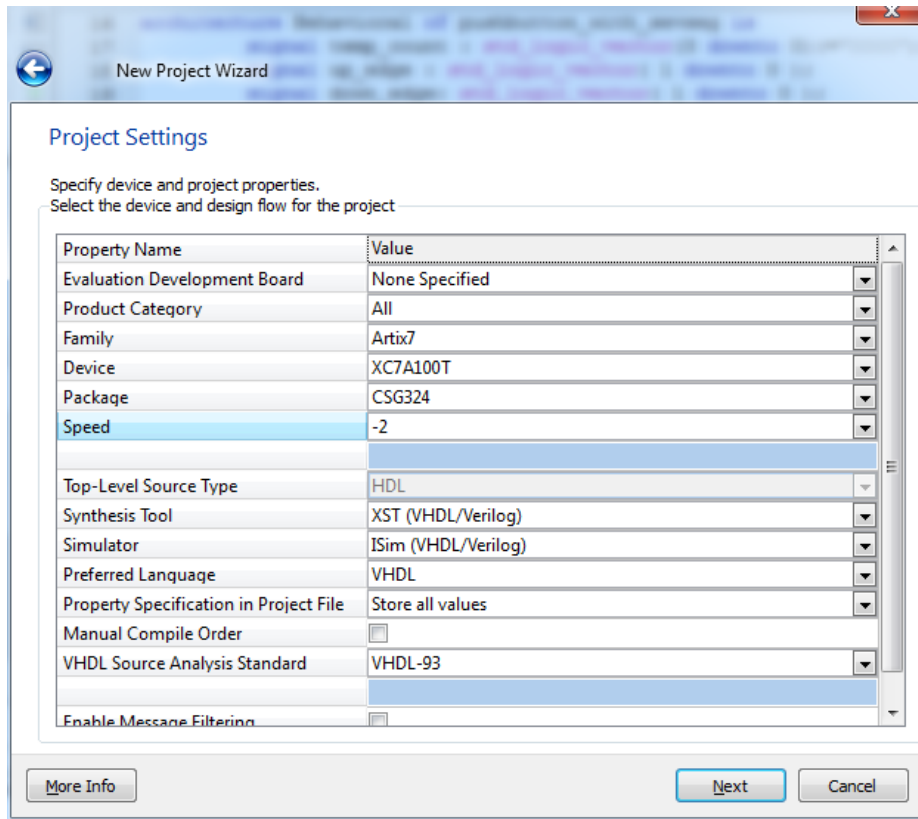
6.2 Membuat Program Dengan Seven Segment dan Push Button

1. Buka aplikasi Xilinx ISE Design Suite 14.5
2. Klik **“File”**, lalu pilih **“New Project”**. Isikan nama project tersebut dengan **“PushButton_SevenSegment”**. Setelah itu klik **“Next”**.



Gambar 6.3 New project.

3. Isikan project settingnya seperti berikut.



New Project Wizard

Project Settings

Specify device and project properties.
Select the device and design flow for the project

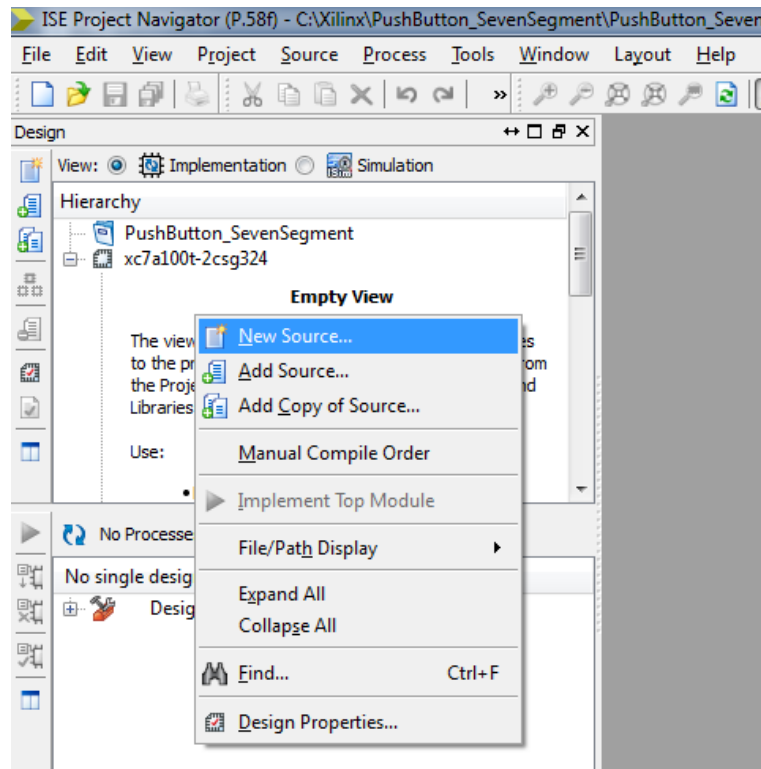
Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
Device	XC7A100T
Package	CSG324
Speed	-2
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info Next Cancel

Setelah itu klik “Next”, lalu “Finish”.

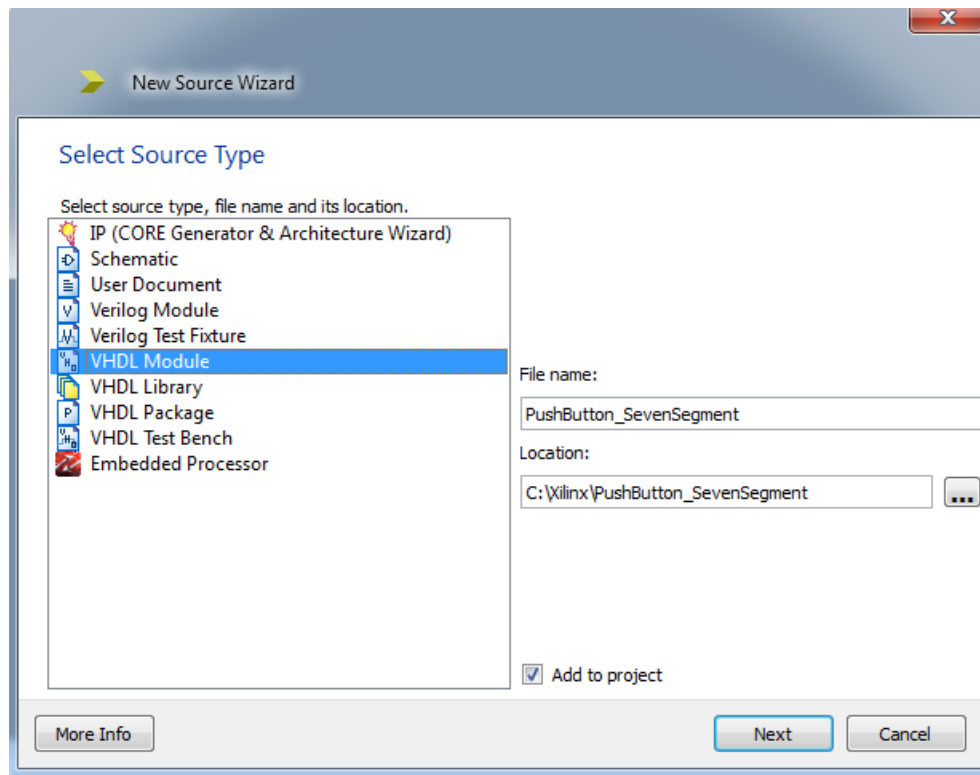
Gambar 6.4 Project setting.

4. Setelah itu akan muncul halaman form untuk project yang akan kita buat. Klik kanan pada “**PushButton_SevenSegment**” lalu pilih “New Source” seperti gambar berikut.



Gambar 6.5 New source.

5. Akan muncul tampilan baru seperti ini. Pilih “VHDL Module” lalu isikan nama filenya menjadi “**PushButton_SevenSegment**”. Setelah itu klik “Next” dan “Finish”.



Gambar 6.6 VHDL module.

6. Setelah muncul tampilan project yang akan kita buat, isikan program seperti berikut ini :

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity PushButton_SevenSegment is
7
8  port (
9      clk : in std_logic;
10     count_up : in std_logic;
11     count_down : in std_logic;
12     seg : out std_logic_vector(6 downto 0));
13
14 end PushButton_SevenSegment;
15
16 architecture Behavioral of PushButton_SevenSegment is
17
18     signal temp_count : std_logic_vector(3 downto 0) := "0000";
19     signal up_edge : std_logic_vector(1 downto 0);
20     signal down_edge : std_logic_vector(1 downto 0);
21
22 begin
23
24     process(clk, temp_count)
25     begin
26         if rising_edge(clk) then
27             up_edge <= up_edge(0) & count_up;
28             down_edge <= down_edge(0) & count_down;
29             if (up_edge = "01") then --rising
30                 if temp_count < 9 then
31                     temp_count <= temp_count + 1;
32                 else
33                     temp_count <= "0000";
34                 end if;
35             else
36                 if (down_edge = "10") then --falling
37                     if temp_count > 0 then
38                         temp_count <= temp_count - 1;
39                     else
40                         temp_count <= "1001";
41                     end if;
42                 end if;
43             end if;
44         end if;
45     end process;
```



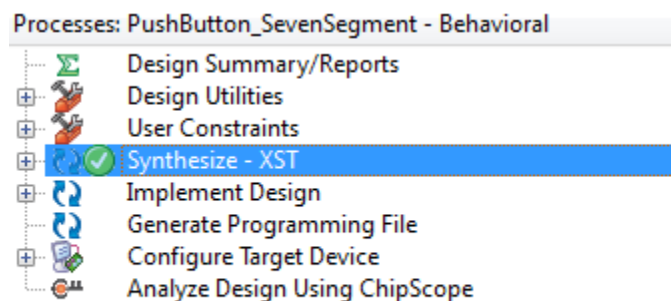
```

46
47 process(temp_count)
48 begin
49     case temp_count is
50         when "0000" => seg <= "1000000";
51         when "0001" => seg <= "1111001";
52         when "0010" => seg <= "0100100";
53         when "0011" => seg <= "0110000";
54         when "0100" => seg <= "0011001";
55         when "0101" => seg <= "0010010";
56         when "0110" => seg <= "0000010";
57         when "0111" => seg <= "1111000";
58         when "1000" => seg <= "0000000";
59         when "1001" => seg <= "0010000";
60         when others => seg <= "1111111";
61     end case;
62 end process;
63 end Behavioral;
64
65

```

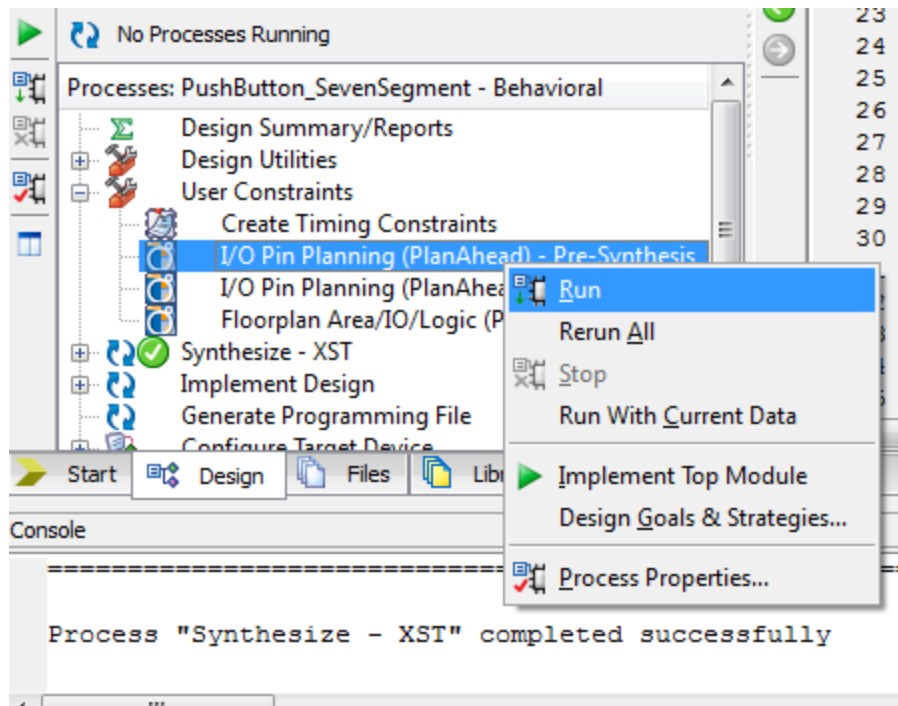
Setelah selesai menulis program, kemudian simpan program tersebut.

7. Klik kanan pada “Synthesize” dan pilih “Run” untuk mengecek apakah program tersebut sudah benar. Jika terdapat error, perbaiki error tersebut kemudian lakukan running ulang,



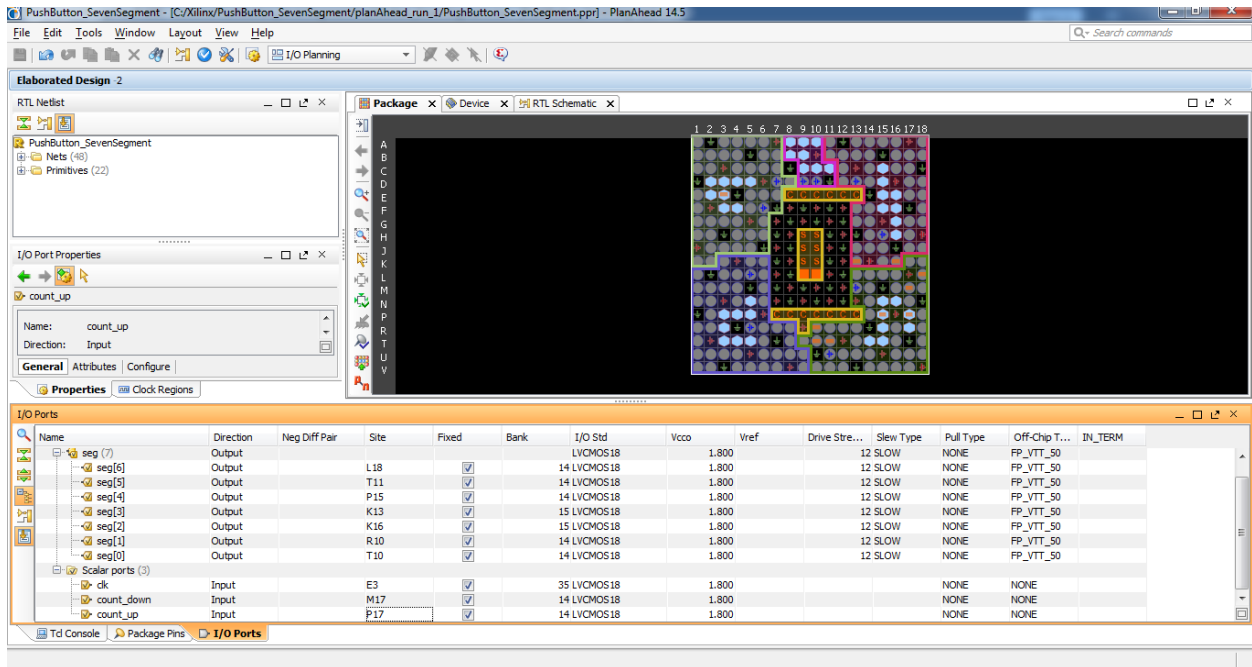
Gambar 6.7 Synthesize.

8. Jika sudah tidak ada error, klik expand pada “User Constraints” lalu pilih I/O Planning (PlanAhead) – Pre-Synthesis. Jalankan dengan cara klik kanan lalu pilih “Run”.



Gambar 6.8 User constraints – PlanAhead.

9. Setelah PlanAhead terbuka, isikan pin seperti berikut ini :



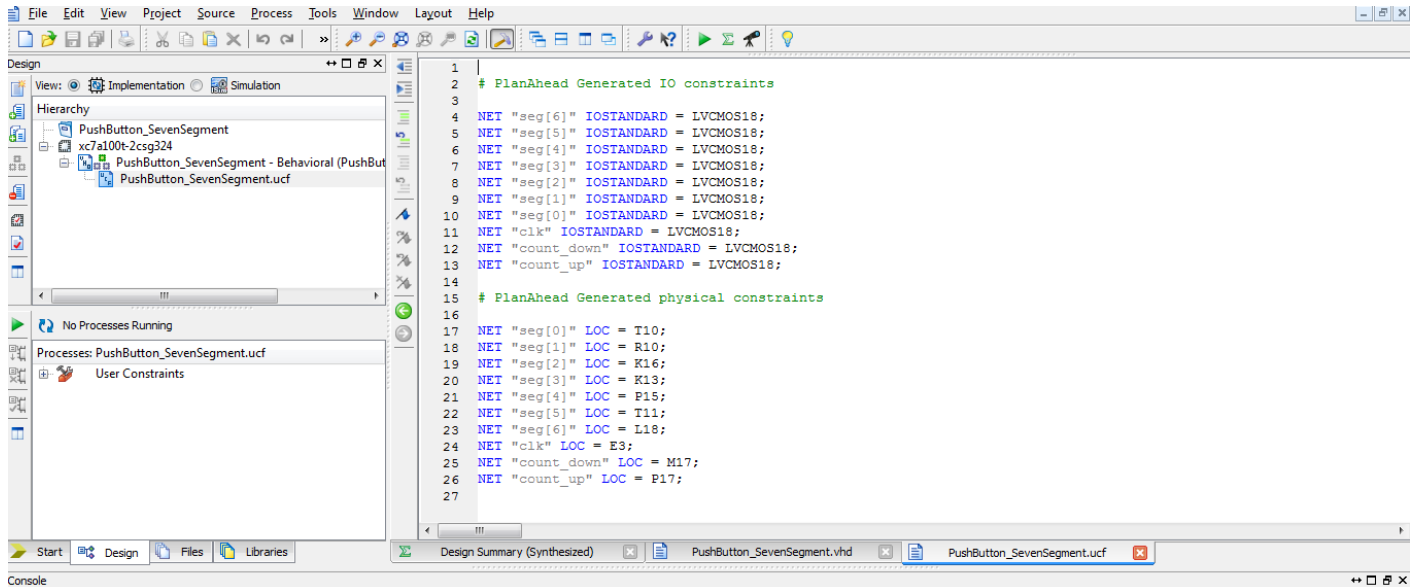
Gambar 6.9 Tampilan PlanAhead.

Keterangan :

- Seg(6) = L18
- Seg(5) = T11
- Seg(4) = P15
- Seg(3) = K13
- Seg(2) = K16
- Seg(1) = R10
- Seg(0) = T10
- Clk = E3
- Count_down = M17
- Count_up = P17

Setelah selesai, klik tombol “Save” yang ada pada kiri atas tampilan.

10. Kembali ke software Xilinx, lalu klik expand pada menu berikut. Terdapat sebuah file dengan ekstensi .ucf, buka file tersebut dengan klik dua kali lalu pilih “Yes”. Pastikan tampilannya sudah lengkap seperti berikut ini :

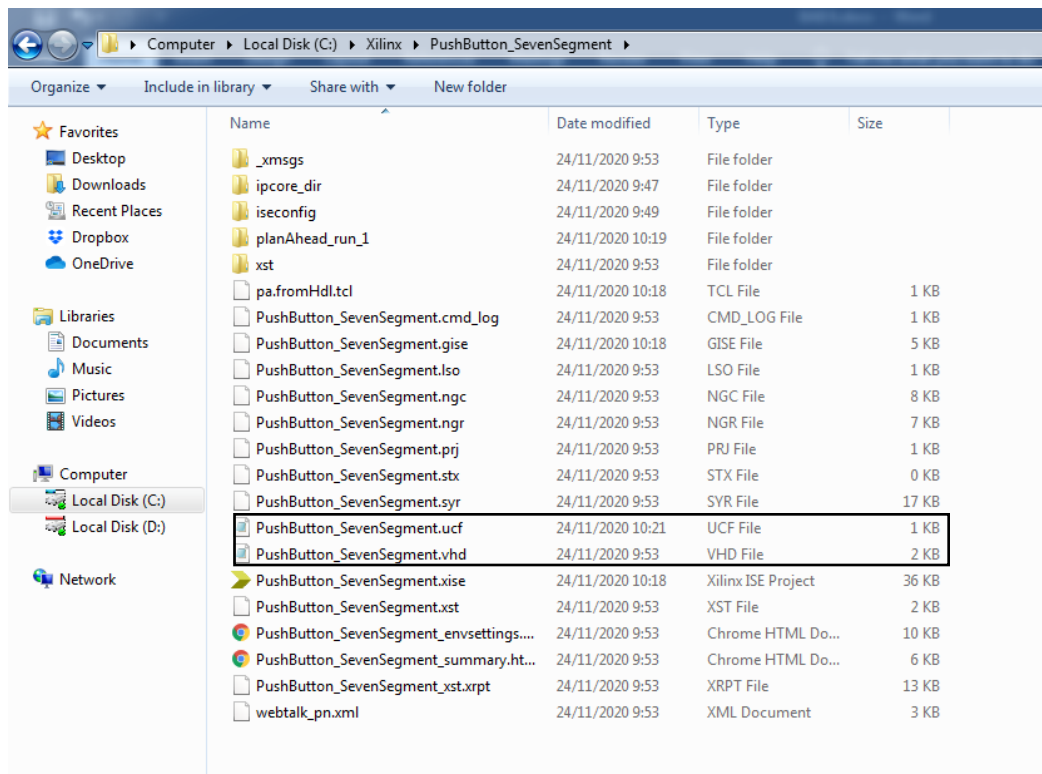


Gambar 6.10 Tampilan file UCF.

Jika sudah sesuai, kemudian simpan project tersebut.

NOTE :

1. Kirim dua buah file kepada PJ Shift masing-masing yaitu file dengan ekstensi .vhd dan .ucf. PJ Shift akan menjalankan project yang sudah kalian buat dan membuat sebuah video penjelasan dari project tersebut.
2. File dengan ekstensi .vhd dan .ucf dapat ditemukan pada local disk C:\Xilinx lalu pilih nama file yang sudah dibuat.



Gambar 6.11 Contoh lokasi file.