# **MODUL II**

Perancangan FPGA untuk Implementasi Rangkaian Sequential dan Kombinational

### I. Tujuan

Pada Percobaan ini praktikan akan mempelajari tentang bagaimana cara mengembangkan rangkaian logika sequential dan kombinational pada FPGA. Metode yang digunakan untuk dengan pemrograman VHDL.

## II. Kompetensi

Setelah meneyelesaikan percobaan ini diharapkan praktikan mempunyai kemampuan untuk :

- Mengembangkan rangakaian logika sequential dan komninational pada FPGA
- Menjelaskan struktur pemrograman VHDL
- Mampu mendiskripsikan circuit secara Behavioural, Dataflow atau Structural

# III. Dasar Teori

VHDL adalah kependekan dari Very High Speed Integrated Circuit Hardware Description Language, yaitu bahasa pemrograman yang digunakan untuk mendeskripsikan logic circuit yang dikehendaki

Secara umum struktur dari pemrograman VHDL terdiri atas dua bagian yaitu bagian ENTITY dan bagian ARCHITECTURE.



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.

Contoh :



# 8-bit Comparator



### Ada tiga cara untuk mendiskripsikan Circuit dalam VHDL:

1. Behavioural

Didesain berdasarkan Algorithma

2. Dataflow (RTL)

Didesain berdasarkan alur register data

3. Structural

Didesain berdasarkan perkomponen dan "merangkai komponen tersebut"

#### **Contoh Deskripsi behavioural**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL
ENTITY eqcomp4 IS PORT(
    a, b:
           IN std_logic_vector(3 DOWNTO 0);
   equals: OUT std logic);
END eqcomp4;
ARCHITETURE behavioral OF eqcomp4 IS
BEGIN
comp: PROCESS (a, b)
   BEGIN
      IF a = b then
         equals <= '1';
      ELSE
         equals \leq 0';
      END IF;
    END PROCESS comp;
END behavioral;
```

### **Contoh Deskripsi DataFlow**

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL

ENTITY eqcomp4 IS PORT(

a, b: IN std_logic_vector(3 DOWNTO 0);

equals: OUT std_logic);

END eqcomp4;

ARCHITETURE dataflow OF eqcomp4 IS

BEGIN

equals <= '1' WHEN (a=b) ELSE '0';

END dataflow;

Contoh Deskripsi Structural

LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL
ENTITY eqcomp4 IS PORT(
    a, b:
            IN std_logic_vector(3 DOWNTO 0);
   equals: OUT std_logic);
END eqcomp4;
USE work.gatespkg.ALL;
ARCHITETURE struct OF eqcomp4 IS
    SIGNAL x: std logic vector(0 TO 3);
BEGIN
   u0: xnor2 PORT MAP (a(0),b(0),x(0));
    u1: xnor2 PORT MAP (a(1),b(1),x(1));
    u2: xnor2 PORT MAP (a(2),b(2),x(2));
    u3: xnor2 PORT MAP (a(3),b(3),x(3));
    u4: and4 PORT MAP (x(0),x(1),x(2),x(3),equals);
END struct;
```

Tipe data yang ada dalam pemrograman VHDL yaitu :

Data	Value	Contoh
Bit Bit_Vector Boolean Integer Real Time Character String	<ul> <li>'1','0'</li> <li>(array of bits)</li> <li>True, False</li> <li>-2,-1,0,1,2,3</li> <li>1.0, -1.0E5</li> <li>1us,7ns,100ps</li> <li>'a','b','c',etc.</li> <li>(Array of Char)</li> </ul>	Q <= '1'; DataOut <= "00010101"; EQ <= True; Count <= Count + 2; V1 = V2 / 5.3; Q <= '1' after 6 ns; CharData <= 'X'; Msg <= "MEM:" & Addr

#### **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 + XSTENDBOARD
- 3. 1 power-supply +9V
- 4. 1 kabel data

V. Prosedur Percobaan Gambaran Disain

Pada percobaan ini akan dibuat sebuah counter up\_down dan decoder seven segmen dengan menggunakan VHDL editor.



1. Jalankan software ISE WebPACK dengan mengklik ganda icon

pada desktop.

2. Buat new project dengan nama "updown\_counterku" simpan di direktori

### E:/VLSI/PERC2/

<sup>y</sup> roject Name:	Project Location:
updown_counterku	E:\VLSI\PERC2\updown_counterku
lect the tune of Ton-Level m	adule for the Project
according to be a sub-restoring	
Lon Louel Module Tupe:	
Top-Level Module Type:	<b>x</b>
Top-Level Module Type: HDL	
Top-Level Module Type: HDL	
Top-Level Module Type: HDL	

3. 6. Masukkan parameter seperti gambar berikut. Kemudian klik tombol NEXT sampai tiga kali, konfigurasi yang ada tidak perlu diubah, kemudian klik tombol FINISH.

Device Family: Spartan2 , Device: Xc2s50, Package: tq144

	erty Name	Value
Device Family		Spartan2
Device		xc2s50
Package		tq144
Speed Grade		-5
Top-Level Module Type	2	HDL
Synthesis Tool		XST (VHDL/Verilog)
Simulator		Modelsim
Generated Simulation Lar	nguage	VHDL
Lisenerated Simulation Lar	nguage	VHDL

4. Buat new source jenis VHDL Module dengan nama updown\_ku

<ul> <li>O Schematic</li> <li>State Diagram</li> <li>✓ Test Bench Waveform</li> <li>✓ Lize Degrament</li> </ul>	File Name:								
Ser Document	updown_ku	_							
🗓 Verilog Test Fixture 🎦 VHDL Libraro	Location:								
VHDL Module	E:\vlsi\perc2								

Selanjutnya klik NEXT sampai FINISH

5. Maka anda akan mendapatkan jendela HDL editor

```
📚 Xilinx - Project Navigator - C:\VLSI\updown_counterku.npl - [updown_ku.vhd]
                                                                                                                                                                       _ 8 ×
Pile Edit View Project Source Process Window Help
                                                                                                                                                                   » _ & ×
  8 🖬 🖻 의 🗠 🙀
                                                                                                                          · * * * * % 9
                                                                N?
                                                         × ×
                                                                        library IEEE;
  Sources in Project:
                                                                        use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
     updown_counterku
                                                                   3456789
   - 🛄 xc2s50-5tq144
        vpdown_ku-behavioral (c:\vlsi\updown_ku.vhd)
                                                                        -- Uncomment the following lines to use the declarations that are
                                                                        -- provided for instantiating Xilinx primitive components.
--library UNISIM;
                                                                         --use UNISIM. VComponents.all;
                                                                   10
11
                                                                        entity updown_ku is
end updown_ku;
                                                                   11
12
13
14
   📲 Module View 🗈 Snapshot View 🗈 Library View
                                                                        architecture Behavioral of updown_ku is
                                                                   15
16
                                                                        begin
                                                          * | x
  Processes for Source: "updown_ku-behavioral"
                                                          -
                                                                   18
       Add Existing Source
Create New Source
                                                                   19
20
                                                                         end Behavioral;
      1
            Design Entry Utilities
                  Create Schematic Symbol
Launch ModelSim Simulator
            M
                 View Command Line Log File
View VHDL Instantiation Template
            8
             User Constraints
             Synthesize · XST
            •
                                                                   •
                  16---- DTL C-I
                                                                  🗹 updown_ku...
   Process View
      (Empty Log)
×
  Console / Find in Files / Warnings / Errors /
```

Selanjutnya ketikkan program berikut pada jendela HDL editor :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD LOGIC ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;
ENTITY counters IS
  PORT(
         : IN std_logic_vector(3 downto 0);
    d
    pulsa : IN std_logic;
    clear : IN std_logic;
    load : IN std logic;
    up_down : IN std_logic;
         : OUT std_logic_vector(3 downto 0)
    qd
                );
END counters:
ARCHITECTURE Behavioral OF counters IS
BEGIN
  -- An up/down counter
  PROCESS (pulsa, up down)
     VARIABLE cnt
                      : std logic vector(3 downto 0);
     VARIABLE direction : std logic vector(3 downto 0);
  BEGIN
    IF (up down = '1') THEN --Generate up/down counter
       direction := "0001";
       ELSE
         direction := "1111";
```

```
END IF;
      IF (pulsa 'EVENT AND pulsa = '1') THEN
         IF (load = '1') THEN
                                --Generate loadable
                       --counter. Take these
           cnt := d;
           ELSE
                           --lines out to increase performance.
           cnt := cnt + direction;
         END IF;
         --The following lines will produce a synchronous
         --clear on the counter
          IF (clear = '0') THEN
            cnt := "0000";
          END IF;
       END IF;
  qd <= cnt; --Generate outputs
  END PROCESS;
END Behavioral;
```

6. Kemudian SAVE file dan periksa penulisan program VHDL dengan cara melakukan klik ganda pada check syntax

····· 🗹	ounters-behavioral		
		(updown_ku.vhd)	
<b>T</b> Module	liew 📩 📩 Snapshot	View 🚺 Library View	
			10
Processes fo	r Source: "counter	s-behavioral"	
Processes fo	r Source: "counter Launch Moo	s-behavioral" delSim Simulator	
Processes fo	r Source: "counter Launch Moo View Comm	s-behavioral" delSim Simulator and Line Log File	ļ
Processes fo	r Source: "counter Launch Moo View Comm View VHDL	s-behavioral" delSim Simulator and Line Log File Instantiation Template	
Processes fr	r Source: "counter Launch Moo View Comm View VHDL User Constraints	s-behavioral" delSim Simulator and Line Log File Instantiation Template	
Processes fo	r Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST	s-behavioral" delSim Simulator and Line Log File Instantiation Template	
Processes fo	r Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST View Synthe	s-behavioral" delSim Simulator and Line Log File Instantiation Template esis Report	
Processes fo	r Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST View Synthe	s-behavioral" delSim Simulator and Line Log File Instantiation Template esis Report chematic	
Processes fo	Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST View Synthe View RTL S Check Synthesize	s-behavioral" delSim Simulator and Line Log File Instantiation Template esis Report chematic	
Processes fo	Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST View Synthe View RTL S Check Synthesize	s-behavioral" delSim Simulator and Line Log File Instantiation Template esis Report chematic	
	Source: "counter Launch Moo View Comm View VHDL User Constraints Synthesize - XST View Synthe View RTL S Check Synt Implement Design	s-behavioral" delSim Simulator and Line Log File Instantiation Template esis Report chematic	

Ulangi langkah ini sampai tidak ada kesalahan, apabila sudah tidak ada error maka ikutilah langkah selanjutnya.

7. Kemudian synthesize modul dengan cara mengklik ganda pada Synthesize-XST serta Implementasi Design dengan cara mengklik ganda pada Implement Design.



8. Buat **new source** untuk simulasi modul VHDL hasil perancangan , berilah nama file **simul\_updown** 

9. Berikanlah nilai logic pada parameter input *(yang berwarna biru)* sebelum melakukan simulasi. Setelah itu lakukan simulasi dan gambarkan hasil simulasi pada lembar laporan sementara, **Jangan lupa simpan file hasil simulasi**.



10. Selanjutnya lakukan simulasi FUNCTIONAL (Simulate Behavioral Model) dan simulasi TIMING (Simulate Post Place&Route VHDL Model).CATATAN : Perhatikan perbedaan hasilnya



11. Setelah itu lanjutkan dengan membuat VHDL modul baru dengan cara membuat new

source, beri nama modul ini Dekoder\_Sev

12. Ketikkan listing program berikut pada jendela VHDL editor

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- komentar dapat dihapus
entity Dekoder_Sev is
port(
d : in std_logic_vector(3 downto 0);
s : out std_logic_vector(6 downto 0)
);
end Dekoder_Sev;
architecture DataFlow of Dekoder_Sev is
begin
       s <= "1110111"
                             when d="0000" else --0
              "0010010"
                             when d="0001" else --1
              "1011101"
                             when d="0010" else --2
                             when d="0011" else --3
              "1011011"
              "0111010"
                             when d="0100" else --4
              "1101011"
                             when d="0101" else --5
              "1101111"
                             when d="0110" else --6
              "1010010"
                             when d="0111" else --7
              "1111111"
                             when d="1000" else --8
              "1111011"
                             when d="1001" else --9
              "1111110"
                             when d="1010" else --A
              "0101111"
                             when d="1011" else --B
              "1100101"
                             when d="1100" else --C
              "0011111"
                             when d="1101" else --D
              "1101101"
                             when d="1110" else --E
              "1101100";
                                            --F
```

end DataFlow;

Xilinx - Project Navigator - C:\VLSI\updown_counterku.npl - [Deko	der_Sev.vhd]	
	2 * * * * * * * *	2 <u>- 19 -</u>
Sources in Project: □ updown_counterku □ 2 xo250-5kq144 □ counters behavioral (updown_ku vhd) □ counters behavioral (updown_ku vhd) □ counters behavioral (Dekoder_Sev.vhd)	<pre>1 library IEEE; 2 use IEEE.STD_LOGIC_lI64.ALL; 3 use IEEE.STD_LOGIC_ARITH.ALL; 4 use IEEE.STD_LOGIC_INSIGNED.ALL; 6 6 8 komentar dapat dihapus 7 entity Dekoder_Sev is 8 port( 9 d : in std_logic_vector(3 downto 0); 10 s: out std_logic_vector(7 downto 0) 11 ); 12 end Dekoder_Sev; 13</pre>	*
Module View     Snapshot View     Create New Source     Create New Source     Create New Source     Create New Source     Surce     Surce Source     Surce Source     Surce Source     Surce Source     Generate Programming File	<pre>14 architecture Behavioral of Dekoder_Sev is 15 begin 18 s &lt;= "00000110" when d="0001" else1 17 "0101101" when d="0010" else2 18 "0100110" when d="0100" else3 19 "01100110" when d="0101" else3 19 "01101110" when d="0101" else5 11 "0111110" when d="0101" else6 22 "00000111" when d="0101" else6 22 "00000111" when d="0101" else7 23 "0111111" when d="1010" else8 24 "0110111" when d="1010" else8 24 "0110111" when d="1010" else8 25 "0111011" when d="1010" else9 26 "0111010" when d="1010" else5 27 "0011001" when d="1010" else5 28 "0011100" when d="1010" else6 29 "0111001" when d="1010" else5 30 "0111001" when d="1010" else5 31 "0011111", -0 32 end Behavioral;</pre>	
T the name of View	▼ ■ DekoderSe ♥ DekoderSe.	× •

13. Dengan cara yang sama dengan langkah 6 s.d 10 sebelumnya, CHECK SYNTAX,

SYNTHESIZE, IMPLEMENT DESIGN dan SIMULASIKAN module dekoder ini.

Berilah nama file simulasi **simul\_dekoder.** 

End Time: 3000 ns		50 ns 250 ns 450 ns 650 ns 850 ns 1050 ns 1250 n
표 🔀 d[3:0]	15	
표 <mark>4X</mark> s(6:0)	0	0
7		

14. Setelah itu lanjutkan dengan membuat VHDL modul baru dengan cara membuat **new source,** beri nama modul ini **debouncing** 

15. Ketikkan listing program berikut pada jendela VHDL editor

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL; use IEEE.STD LOGIC UNSIGNED.ALL; entity debouncing is port( clk : in std logic; enter : in std logic; enter\_db : out std\_logic ); end debouncing; architecture Behavioral of debouncing is signal A : std logic vector(2 downto 0) := "000"; begin process(clk,enter) begin if (clk'event and clk='1') then A <= A(1 downto 0) & enter; -- shift left end if; end process;

enter\_db <= not(A(2)) and A(1) and A(0); end Behavioral;

16. Dengan cara yang sama dengan langkah 6 s.d 10 sebelumnya, CHECK SYNTAX,

#### SYNTHESIZE, IMPLEMENT DESIGN dan SIMULASIKAN module debouncing ini.

Berilah nama file simulasi **simul\_debouncing** 

10000 ns		 100 n1100 ns 2100 ns 3100 ns 4100 ns 5100 ns 6100 ns 7100 ns 8100 ns 91
CIK	0	הההההההההההההההההההההההההההההההההההההה
<u> I</u> enter	0	
<b>∛</b> ¶ enter_db	0	

17. Buat Source VHDL baru yang berfungsi menghubungkan tiga komponen/blok diatas, beri

#### nama chip\_updown

18. Ketikkan listing program berikut pada jendela VHDL editor

library IEEE; use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL; use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

```
entity chip_updown is
port(
                        : IN std_logic;
    clock
                        : IN std_logic_vector(3 downto 0);
    pin d
    pin pulsa : IN std logic;
    pin_clear : IN std logic;
    pin_load : IN std_logic;
    pin_up_down : IN std_logic;
                pin_s : out std_logic_vector(6 downto 0)
);
end chip_updown;
architecture Structural of chip_updown is
       component counters
               port(
                d
                         : IN std_logic_vector(3 downto 0);
    pulsa : IN std_logic;
    clear : IN std_logic;
    load : IN std_logic;
    up_down : IN std_logic;
        : OUT std_logic_vector(3 downto 0)
    qd
               );
       end component;
       signal out_counter : std_logic_vector(3 downto 0);
       component Dekoder_Sev
               port(
                d : in std_logic_vector(3 downto 0);
                s : out std_logic_vector(6 downto 0)
               );
       end component;
       component debouncing
               port(
                clk : in std logic;
                enter : in std logic;
                enter db : out std logic
               );
       end component;
       signal s_pulsa : std_logic;
begin
counter_ku:
counters port map
                       (
                       d => pin d, pulsa => s pulsa, clear => pin clear,
                       load => pin_load, up_down => pin_up_down, qd => out_counter
                       );
decoder_ku:
Dekoder_Sev port map
                       d=> out counter, s=> pin s
                       );
debouncing ku:
debouncing port map
                       (
```

```
clk => clock, enter => pin_pulsa , enter_db => s_pulsa
);
```

end Structural;

19. Selanjutnya Simpan, Check Syntax dan Synthesized



20. Kemudian tentukan pin-pin pada FPGA yang akan kita gunakan dengan mengedit file constraint dengan cara melakukan klik ganda pada Edit Constraint (Text)



Pilih pin-pin yang digunakan pada FPGA sesuai dengan datasheet dari XSTENDBoard untuk

dapat menggunakan pushbutton, Barled, Switch maupun SevenSegment.

File ucf untuk mendefinisikan pin-pin IC

Nomor-nomor pin yang dipakai untuk seven segmen, barled, push-button dan dip-switch.

```
net ledtwo<0> loc=p47; # rightmost 7-segment LED
net ledtwo<1> loc=p40;
net ledtwo<2> loc=p28;
net ledtwo<3> loc=p29;
net ledtwo<4> loc=p27;
net ledtwo<5> loc=p42;
```

```
net ledtwo<6> loc=p48;
net ledtwo<7> loc=p38;
net ledone<0> loc=p64; # leftmost 7-segment LED
net ledone<1> loc=p65;
net ledone<2> loc=p76;
net ledone<3> loc=p50:
net ledone<4> loc=p51;
net ledone<5> loc=p54:
net ledone<6> loc=p56;
net ledone<7> loc=p63;
net barled<1> loc=p68; # bargraph LED
net barled<2> loc=p44;
net barled<3> loc=p46;
net barled<4> loc=p49;
net barled<5> loc=p57;
net barled<6> loc=p62;
net barled<7> loc=p60;
net barled<8> loc=p67;
net barled<9> loc=p39;
net barled<10> loc=p59;
net pushsw<3> loc=p78; # pushbuttons
net pushsw<4> loc=p26;
net pushsw<5> loc=p23;
net dipsw<1> loc=p30; # DIP switches
net dipsw<2> loc=p58;
net dipsw<3> loc=p74;
net dipsw<4> loc=p75;
net dipsw<5> loc=p66;
net dipsw<6> loc=p77;
net dipsw<7> loc=p80;
net dipsw<8> loc=p79;
```

```
21. Edit Constraint file sebagai berikut:
```

```
# INPUT
net pin pulsa loc=p23; #pin CLOCK berada pada pin ke-23 pada FPGA atau tombol S5 pada
XSTENBoard
net clock loc=p88;
net pin_clear loc=p79;
net pin load loc=p77;
net pin up down loc=p80;
net pin d<0> loc=p30;
net pin d<0> loc=p74;
net pin_d<0> loc=p75;
net pin_d<0> loc=p66;
# OUTPUT
net pin s<0> loc=p64; # leftmost 7-segment LED
net pin_s<1> loc=p65;
net pin s<2> loc=p76;
net pin s<3> loc=p50;
net pin s<4> loc=p51;
net pin s<5> loc=p54;
net pin_s<6> loc=p56;
```

#### 22. Kemudian Implement Design

23. Buat **new source** untuk simulasi modul VHDL hasil perancangan , berilah nama file **simul\_chip** 

10000 ns		10	0 r	111	00	) n	s 2	10	0 r	is:	310	00	ns	411	00	ns	5	10	0 r	IS	61	00	) n	IS	71	00	n	38	100	) ns	; 9
			1			1	1	1		1	-			1		1	1		_	1			1	1	1		1	L		1	1
pin_clear 1	8																														
🛺 pin_load 🛛 🛛 🕻	)							1															1								
💦 pin_up_down 🛛 1	Ē.						Π			I									T	T						T					Ι
🔊 clock 🛛	)	Л			Π	П	Π	П	Π	П	Π	П	ΠΠ	Π	П	Π	Л	Л	l	П	Γ		Γ	П	Π	ľ	П	ΠΓ		11	IJ
🔰 pin_pulsa 🛛 🕻	)			1000			22.3		5000										300	30		255	10.53	2712	0.024						
🗉 💦 pin_d(3:0) 👘 (	)	$\left( \right)$																0													
표 秋 pin_s(6:0) 🛛 🛛 🛈	)	$\langle \$																0													

24. Selanjutnya lakukan **SIMULASI** FUNCTIONAL (Simulate Behavioral Model) dan simulasi TIMING (Simulate Post Place&Route VHDL Model).dan **ambil data simulasi** 

ModelSim XE III/Starte	r 6.0a - Custom 2	(ilinx Versi	on								
File Edit View Format Com	pile Simulate Add	Tools Wind	dow Help								
0 🚅 🖬 🚭   🐰 🖻 🛍	3 <u>22</u>   <b>M</b> 🛱	. <b>8</b> 5 <b>M</b>	۵ 🗰 🎝	X   🕇	<b>≧₹</b> 100		Et   79 7	Contai	ns.	□ <i>2</i>  ] J	₽ 🕅 Te :
K 🖪 I 🍳 🔍 👫	3⊷										
wave - default											
Isimul_chip/clock	1	JUUU		JULUU	mm	JUUUU	hunn	mm	JUUUU	JUUUU	JUUUU
⊞–� /simul_chip/pin_d	0000	0000							_		
/simul_chip/pin_pulsa	0									<u>م الم الم الم الم الم الم الم الم الم ال</u>	
/simul_chip/pin_clear	1										
/simul_chip/pin_ioau	1										
	5D	77		X12					)(5D		

### 25. Generate Programming File

- 26 Upload file \*.bit yang dihasilkan dengan menggunakan program GXSLOAD
- 27. Ambil data hasil percobaan

### 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.



## VIII. Tugas

- 1. Rubahlah module VHDL yang ada sehingga counter naik atau turun sebesar 2 desimal.
- 2. Dari synthesis report TOPLEVEL, berapakah jumlah slice yang dipakai serta berapa kecepatan frekuensi maksimum?
- 3. Jelaskan kenapa terdapat perbedaan hasil yang sangat besar antara simulasi Functional dengan simulasi timing saat simulasi :
  - (1) simul\_updown
  - (2) simul\_chip
- 4. Kenapa modul VHDL chip\_updown dikategorikan menggunakan deskripsi structural?

# 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 Sjoholm & L. Lindh, "VHDL for Designers" Prentice Hall
- 5. Xilinx FPGA IseWebpack 7.0 Tutorial

# Lampiran 1