

MODUL V

Implementasi Rangkaian Digital dengan FPGA

1 Tujuan

2 Alat dan Bahan

3 Dasar Teori

4 Praktikum

4.1 Pengenalan IO

Perlu gambar board FPGA yang akan digunakan

Beberapa daftar port dari board FPGA yang digunakan:

Tabel 1: Beberapa port yang digunakan pada board FPGA

Clock	PIN_23
Reset Push Button	PIN_25
Buzzer	PIN_110
IR	PIN_100
Push Button	
key1	PIN_88
key2	PIN_89
key3	PIN_90
key4	PIN_91
LED	
LED1	PIN_87
LED2	PIN_86
LED3	PIN_85
LED4	PIN_84
Seven-segment LED	
DIG1	PIN_133
DIG2	PIN_135
DIG3	PIN_136
DIG4	PIN_137
SEG0	PIN_128
SEG1	PIN_121
SEG2	PIN_125
SEG3	PIN_129
SEG4	PIN_132
SEG5	PIN_126
SEG6	PIN_124
SEG7	PIN_127

Untuk PIN assignment, buat skematik atau file Verilog terlebih dahulu, kemudian compile. Buka PIN assignment, set pin yang diperlukan.

Prosedur ini diberikan pada contoh menyalakan LED.

4.1.1 LED

Buat project baru dengan nama led_light, misalnya. Kemudian tambahkan file Verilog berikut ke project. Kode Verilog ini memberikan nilai logika ke tiap LED (hardwired, tanpa ada input).

```
module led_light(led);
    output[3:0] led;
    assign led = 4'b0000; // coba ubah-ubah nilai ini untuk tiap LED
endmodule
```

Compile file ini dengan cara klik icon Compile atau dengan menu Processing -> Start Compilation atau menggunakan shortcut Ctrl + L.

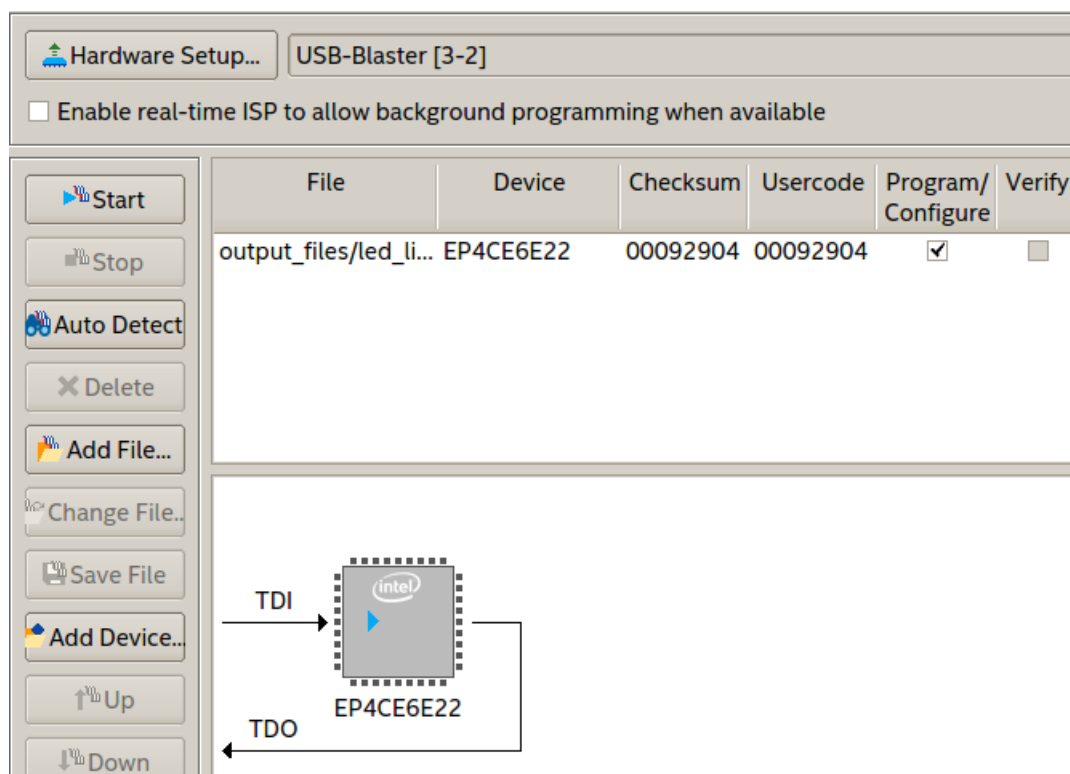
Jika tidak ada kesalahan pada saat proses kompilasi, maka langkah selanjutnya adalah melakukan PIN assignment, yang dapat dilakukan dengan memilih menu Assignment -> Pin Planner atau menggunakan shortcut Ctrl + N. Atur PIN assignment sesuai dengan Tabel 1.

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
led[3]	Output	PIN_87	5	B5_NO	PIN_87
led[2]	Output	PIN_86	5	B5_NO	PIN_86
led[1]	Output	PIN_85	5	B5_NO	PIN_85
led[0]	Output	PIN_84	5	B5_NO	PIN_84
<<new node>>					

Gambar 1: PIN Assignment untuk 4 LED

Compile lagi file tersebut.

Jika tidak ada pesan error, langkah selanjutnya adalah mendownload program ini ke FPGA. Proses ini dapat dilakukan dengan cara memilih menu Tools -> Programmer. Klik button Add File untuk menambahkan file led_light.sof. File ini biasanya ada di dalam subdirektori output dari direktori project. Pastikan juga hardware terdeteksi. Jika belum terdeteksi, tambahkan melalui dengan mengklik button Hardware Setup.



Gambar 2: Tampilan tool Programmer

Catatan

Pada board FPGA yang digunakan urutan LED dari kiri ke kanan adalah LED1, LED2, LED3, dan LED4. Misalkan memberikan assignment sebagai berikut.

- LED1 diwakili dengan `led[0]`
- LED2 diwakili dengan `led[1]`

- LED3 diwakili dengan `led[2]`
- LED4 diwakili dengan `led[3]`

Misalkan juga kita memberikan nilai logika pada `led` dengan kode Verilog berikut.

```
led = 4'b1010;
```

Maka nilai 0 (nilai bit paling kanan atau LSB) diberikan pada `led[0]` atau LED1. Nilai pada bit kedua dari kanan diberikan untuk `led[1]`, bit ketiga untuk `led[2]`, dan bit keempat (paling kiri atau MSB) untuk `led[3]`.

Bagian output `[3:0]` `led` pada kode di atas dapat diganti dengan output `[1:4]` `led` untuk memudahkan assignment nilai logika sesuai dengan urutan LED di board yang digunakan. Sehingga kita dapat melakukan assignment sebagai berikut.

- LED1 diwakili dengan `led[1]`
- LED2 diwakili dengan `led[2]`
- LED3 diwakili dengan `led[3]`
- LED4 diwakili dengan `led[4]`

Cobalah bereksprimen dengan cara mengganti-ganti nilai logika dari `led`, kemudian isilah tabel berikut.

Nilai logika	Keadaan LED (on/off)
0	
1	

4.1.2 Push buttons

Buat project baru, dan buat file Verilog dengan mendefinisikan satu modul dengan input dari push button dan output ke LED.

```
module test_buttons( buttons, led );
    input  [3:0] buttons;
    output [3:0] led;

    assign led = buttons;
endmodule
```

Bisa juga menggunakan potongan kode berikut.

```
assign led[0] = buttons[0];
assign led[1] = buttons[1];
assign led[2] = buttons[2];
assign led[3] = buttons[3];
```

Cobalah bereksperimen dengan kode Verilog yang ada dan juga menggunakan operator Verilog seperti

Nilai logika	Keadaan PB
0	
1	

4.1.3 Seven segments

Lihat Modul 1.

4.1.4 Clock

Berapa frekuensi clock yang digunakan ? 50 MHz ?

LED blinking (sudah menggunakan counter, implementasinya mudah pada Verilog)

4.1.5 IR

Menggunakan protokol NEC.

Ubah kode Verilog yang sudah ada menjadi blok yang menampilkan

4.2 Rangkaian kombinasional

- rangkaian pendeteksi genap ganjil, input 4 PB, output 1 LED
- Input PB -> BCD -> seven segment
- IR -> BCD -> seven segment
- Implementasi XOR
- half adder dan full adder
- Menyalakan satu atau beberapa LED dengan kombinasi input 4 push button yang diberikan.

- LED1 menyala jika button1 dan button3 ditekan atau button1 dan button 4 ditekan
- LED2 menyala jika button2 dan button4 ditekan atau button1 ditekan

- Input BCD (dari) ke output seven segment. Buat tabel kebenaran dan rangkaian (dalam skematik atau Verilog struktural).

Input				Output							
PB1	PB2	PB3	PB4	a	b	c	d	e	f	g	dp
0	0	0	0								
0	0	0	1								
0	0	1	0								
...								
1	1	1	1								

4.3 Rangkaian sekuensial

- Implementasi D flip-flop
- Implementasi J-K flip-flop dan

- Implementasi T flip-flip. Toggle operation: buzzer dan LED
- register, counter
- Kalkulator sederhana, input dari remote IR ?
- Rangkaian multiplexing LED
- Menampilkan LED perdigit

5 Tugas Pendahuluan