

## BAB IV

### ANALISA DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai keseluruhan pengujian sistem. Materi pengujian sistem meliputi dua bagian yakni dengan simulasi dan pengujian pada monitor. Simulasi dilakukan untuk melihat proses yang terjadi saat dilakukan pemberian input melalui *keypad* serta *output* dari sinyal R, G, B, Vs dan Hs.

#### 4.1 Hasil Simulasi Sistem

Setelah program untuk mengkonfigurasi FPGA selesai dibuat, kemudian untuk memastikan kinerja dari program apakah sudah sesuai dengan yang diinginkan maka ada baiknya program tersebut disimulasikan terlebih dahulu. Simulasi dilakukan perbagian dari setiap bagian pada sistem menggunakan Active-HDL. Hasil dari simulasi adalah berupa *waveform* atau *timing* diagram.

##### 4.1.1 Clock 25Mhz

Untuk membuat *clock* 25MHz dapat dilihat pada penggalan *source code* VHDL berikut ini :

```
process (clk50_in)
begin
    if clk50_in'event and clk50_in='1' then
        if (Clk25 = '0')then
```

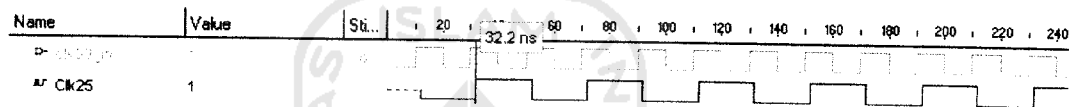
```

        Clk25 <= '1' after 2 ns;
    else
        Clk25 <= '0' after 2 ns;
    end if;
end if;

end process;

```

Dari program pembangkit sinyal 25Mhz diatas dapat dilihat hasil simulasinya sebagai berikut :



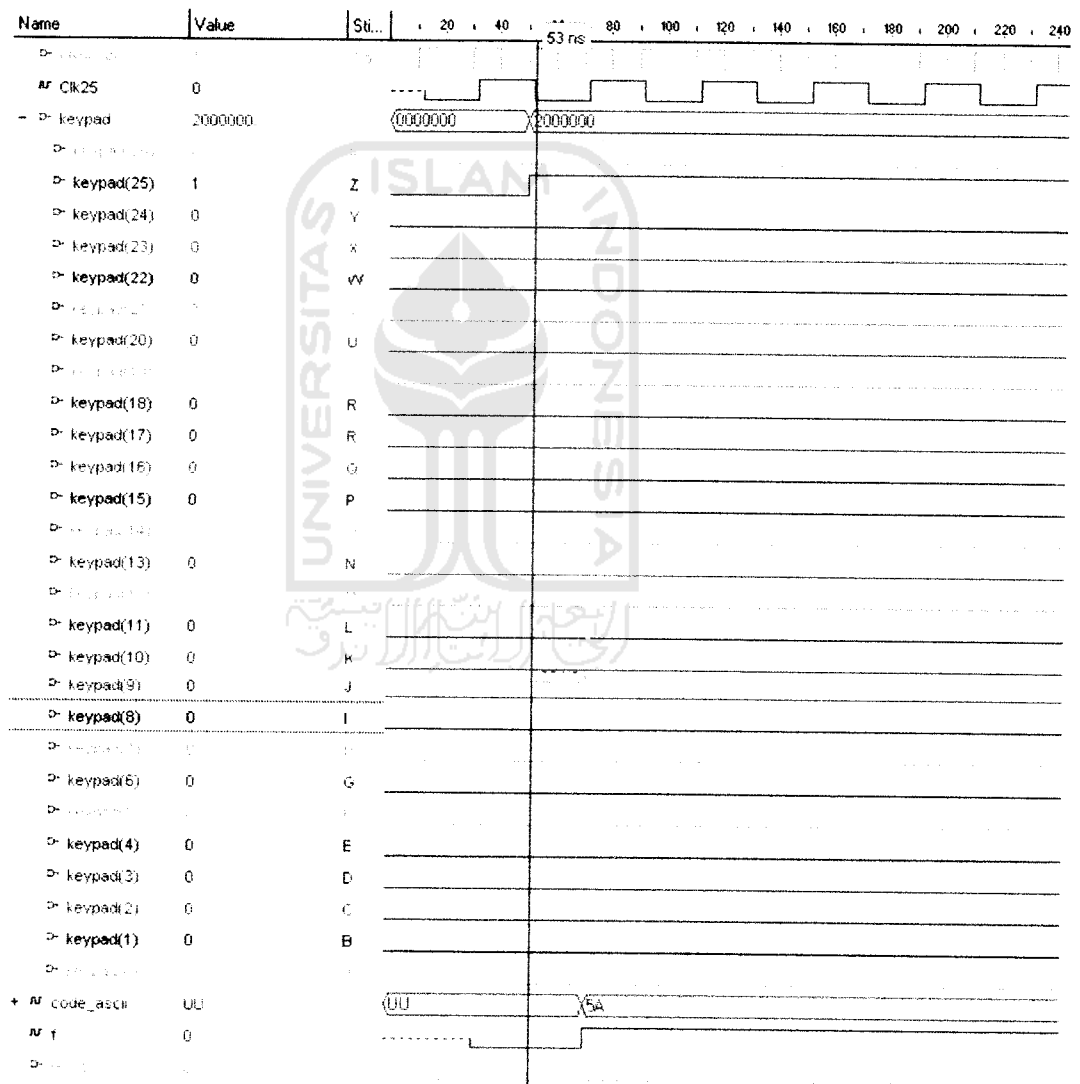
**Gambar 4.1** Hasil simulasi pembangkit sinyal 25Mhz.

Setelah 10ns *clock* 50Mhz (clk50\_in) yang merupakan *clock internal* pada FPGA mulai berdetak sepanjang 20 ns. Untuk menghasilkan 25Mhz atau *clock* sepanjang 40ns maka buat saja 2 ns setelah clk50\_in naik bernilai 1 atau *event*, clk25 bernilai 1, dan setelah *clock* clk50\_in mencapai 1 interval atau 20 ns maka 2 ns setelah *clock* clk50\_in naik clk25 dibuat bernilai 0 maka akan mendapatkan *clock* 25Mhz dengan panjang 1 interval clk25 adalah 40ns. Nilai 2 ns diatas hanya sebagai *delay* saja jika tidak menginginkan *delay* maka “*after 2 ns*” pada program dapat dihapus saja.

### 4.1.2 Keypad

Pada *keypad* terdapat 27 tombol masukan untuk huruf yakni tombol 0 sampai dengan 26 yang pada Gambar 4.1 ditunjukkan *keypad* (0) sampai dengan *keypad* (25) untuk huruf A sampai Z dan *keypad* (26) sebagai tombol reset.

Pada simulasi ini akan mencoba untuk memberi masukan pada sistem yakni huruf Z yang terlihat sebagai berikut :

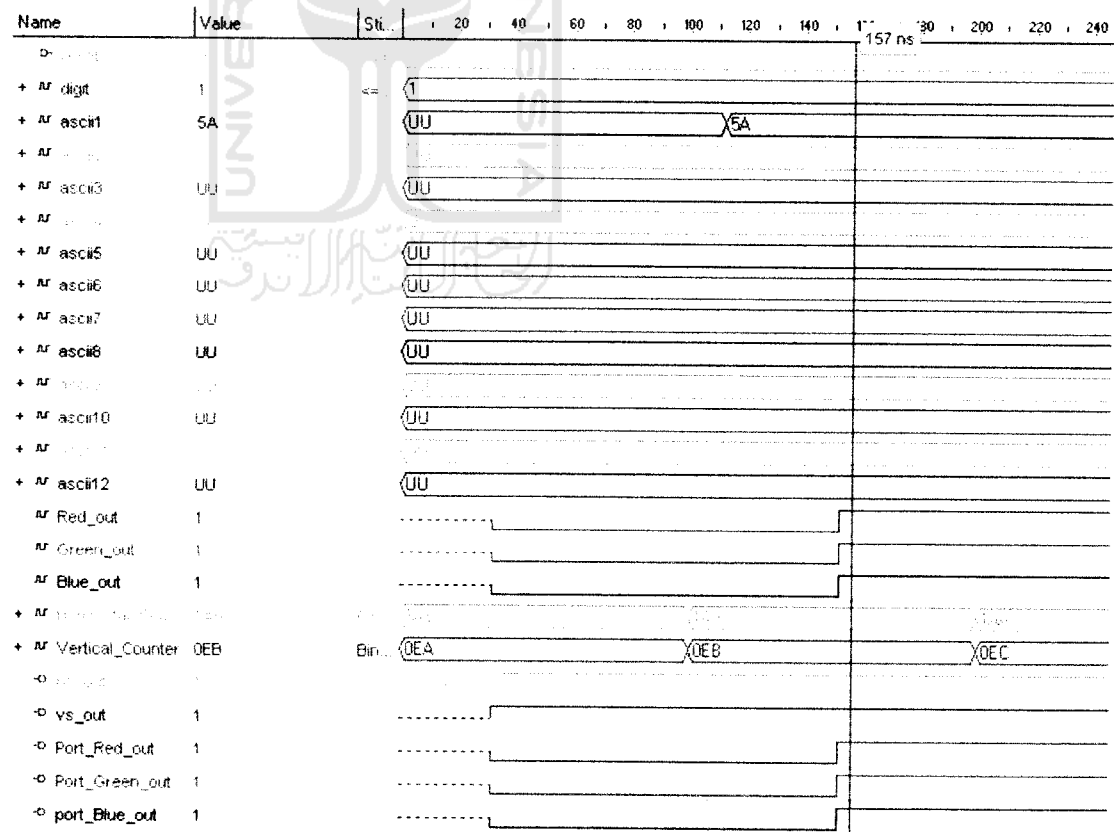


**Gambar 4.2** Hasil simulasi dari *keypad*.

Dari hasil simulasi diatas dapat dilihat jika salah satu tombol ditekan maka sinyal `code_ascii` akan memiliki suatu nilai dan sinyal `trigger (f)` akan berlogika 1. Masukkan `keypad` pada simulasi diatas kita beri stimulator `hotkey`, jadi pada hasil simulasi diatas setelah 50 ns, ditekan huruf Z pada `keyboard` sehingga `keypad(25)` yang memiliki stimulator Z nilainya berubah menjadi 1. `Case keypad` pada program dibuat 27 bit agar pengalaman `pin` pada Xilinx terdeteksi 27 keypad.

#### 4.1.3 Program Utama

Pada bagian ini dapat dilihat hasil simulasi lanjutan dari dari `clock` dan `keypad` diatas. Hasil dari simulasinya adalah sebagai berikut :



Gambar 4.3 Hasil simulasi dari output.

Gambar diatas adalah potongan dari hasil simulasi *clock* dan *keypad*. Sinyal *ascii1* mempunyai keluaran sama dengan *code\_ascii* yang terlihat pada simulasi *keypad*, karena keluaran *code\_ascii* yang pertama kali dikeluarkan akan digunakan sebagai keluaran sinyal *ascii1* dan seterusnya sampai *code\_ascii* yang kesepuluh mengisi *ascii10*.

Setelah *ascii1* mempunyai nilai maka sinyal *Red\_out*, *Green\_out*, *Blue\_out* akan bernilai sehingga *Port\_Red\_out*, *Port\_Green\_out*, *Port\_Blue\_out* akan bernilai 1 yang nantinya *Port\_Red\_out*, *Port\_Green\_out*, *Port\_Blue\_out* akan dihubungkan ke monitor.

Untuk simulasi ini sistemnya akan berbeda dengan sistem sebenarnya karena pada simulasi *Port\_Red\_out*, *Port\_Green\_out*, *Port\_Blue\_out* akan bernilai 0 jika nilai digit, *horisontal\_counter* dan *vertical\_counter* tidak disinkronkan. Pada simulasi diatas digit diberi nilai 1 sehingga *horisontal\_counter* yang diberi stimulator *counter* diawali dengan nilai 424 dan *vertical\_counter* 234 karena untuk digit 1 atau *display area* untuk karakter pertama diawali dengan nilai tersebut. Jadi untuk melakukan simulasi tiap masukan dilakukan sendiri-sendiri karena setiap masukan dan *display area* dimana masukan tersebut akan diletakkan harus diatur nilai awal *horisontal\_counter* dan *vertical\_counter* dari *display area* yang diinginkan.

Untuk sinyal *Hs* akan bernilai satu setelah *horisontal counter* mencapai nilai 93 yaitu dari *sync pulse* 92 ditambah 1 dan *Vs* juga akan bernilai satu setelah *vertical counter* mencapai 3 yaitu dari *sync pulse* 2 ditambah satu. *Source code* VHDL-nya terlihat sebagai berikut :

```

if (Horizontal_Counter > "0000000000" )
    and (Horizontal_Counter < "0001100001" ) -- 96+1
        then
            hs_out <= '0';
        else
            hs_out <= '1';
    end if;

if (Vertical_Counter > "0000000000" )
    and (Vertical_Counter < "0000000011" ) -- 2+1
        then
            vs_out <= '0';
        else
            vs_out <= '1';
    end if;

```

Beberapa detik setelah program di-*download* pada FPGA maka monitor akan aktif.

#### 4.2 Pembentuk mode 640 x 480 dan pewarnaan

Dalam pembentukan mode 640 x 480 dibuat *horizontal counter* yang mempunyai panjang 640 dan *vertical counter* yang mempunyai lebar 480. Berikut adalah penggalan *source code* VHDL-nya :

```

if Clk25'event and Clk25 = '1' then

```

```

if (Horizontal_Counter >= "0010010000" ) -- 144
and (Horizontal_Counter < "1100010000" ) -- 784
and (Vertical_Counter >= "0000 100001" ) -- 33
and (Vertical_Counter < " 1000000001" ) -- 513

then

    Red_out <= '0';
    Green_out <= '0';
    Blue_out <= '0';

```

Setelah Clk25 aktif maka langkah selanjutnya menentukan *horisontal counter* sepanjang 640 yang dimulai dari 144. Angka ini didapat dari *sync pulse* 96 dan *back porch* 48 sehingga setelah *sync pulse* dan *back porch* mencapai 144 maka akan didapat *pixel* atau kolom yang aktif sampai 784 dan setelah itu terdapat *front porch* 16 jadi untuk mode 640 beserta *frame*-nya dalam satu baris mempunyai total *pixel* atau kolom sebanyak 800. Sedangkan untuk *vertical counter* dimulai dengan nilai 33, angka ini didapat dari *back porch* 31 dan *sync pulse* 2 maka akan didapat baris yang aktif sampai 513 jadi jumlah total *line* atau baris dalam 1 kolom adalah 513 ditambah *front porch* 11 adalah 524.

Setelah pembentukan mode 640 x 480 maka selanjutnya menentukan warna dari text yang nanti akan ditampilkan. Disini berlaku *active low* jadi warna putih yang berlogika R '1', G '1' B '1' maka kita balik menjadi '0'.

### 4.3 Display Area

Untuk menentukan *display area* dimana akan meletakkan suatu karakter atau menentukan letak digit pada monitor sebagai contoh dapat dilihat dalam penggalan *source code* VHDL berikut ini :

```

if (Horizontal_Counter >= "0110101000" ) -- 424
  and (Horizontal_Counter <= "0110101111")  -- 431

  and (Vertical_Counter >=  "0011101010") -- 234
  and (Vertical_Counter <= "0011101010") then -- 245

```

maka karakter akan diletakkan pada kolom antara 424 sampai 431 dan antara baris 234 sampai 245.

