Teknologi Implementasi dan Metodologi Desain Sistem Digital
TKC305- Sistem Digital Lanjut

Eko Didik Widianto

Prodi Sistem Komputer - Universitas Diponegoro
Review Kuliah

- Sebelumnya dibahas tentang teknologi implementasi sistem digital di IC keluarga 7400, PLD (PLA, PAL, CPLD, FPGA) serta board evaluasi FPGA Xilinx dan CAD Xilinx ISE Webpack untuk mengembangkan sistem berbasis FPGA

- Berikutnya akan dibahas tentang:
  - Metodologi desain sistem digital menggunakan Xilinx ISE
  - Pengantar HDL (Verilog)

- Referensi:
  3. Verilog Tutorial (online):
     http://www.asic-world.com/verilog/veritut.html
Board Starter Kit Xilinx
Bahasan

Metodologi Desain
  Metodologi dan Alur Desain HDL
  Komponen Desain HDL
  Spesifikasi Sistem

Kegiatan Desain
  Manajemen Proyek
  Membuat Modul Baru
  Proses Kompilasi
  Memprogram Device

Simulasi HDL
Bahasan

Metodologi Desain
Metodologi dan Alur Desain HDL
Komponen Desain HDL
Spesifikasi Sistem

Kegiatan Desain
Manajemen Proyek
Membuat Modul Baru
Proses Kompilasi
Memprogram Device

Simulasi HDL
Metodologi desain:

- **Top-down**
  - Perancangan dimulai dari modul global (top) sampai elemen penyusunnya

- **Bottom-up**
  - Perancangan dimulai dari elemen-elemen penyusunnya sampai ke modul top
Perancangan secara Top-Down

- Modul top dibuat terlebih dahulu
- Modul dipecah menjadi sub-modul
- Submodul dipecah dan diimplementasikan menjadi elemen-elemen desain
Perancangan secara Bottom-Up

- Elemen-elemen dibuat terlebih dahulu
- Elemen disusun membentuk submodul
- Submodul disusun untuk membentuk modul top
Alur Desain Sistem Digital dengan ISE
Desain dan Verifikasi
Tahapan Desain HDL

1. Entry Desain
   ▶ Memprogram HDL (Hardware Description Language) untuk memenuhi kebutuhan fungsional dan konstrain sistem
     ▶ Bentuk: kode program atau skematik
     ▶ Deskripsi struktural dan perilaku sistem
     ▶ Bahasa: VHDL, Verilog dan Altera HDL
   ▶ Verifikasi: simulasi perilaku (diagram pewaktuan)

2. Sintesis
   ▶ Membangkitkan skematik RTL (Register Transfer Level) sesuai teknologi yang digunakan (misalnya Xilinx FPGA)
   ▶ Verifikasi: simulasi perilaku
3. Implementasi
   ▶ Meliputi translating, mapping/fitting dan placing&routing untuk mengimplementasikan desain ke teknologi device FPGA yang sesuai
     ▶ Optimasi agar memenuhi kebutuhan fungsional dan konstrainnya (power, area, speed)
     ▶ Hasil: berupa file konfigurasi FPGA

4. Pemrograman
   ▶ Download file konfigurasi ke device FPGA (Xilinx)
     ▶ File konfigurasi didownload melalui interface USB-JTAG
     ▶ Digunakan program iMPACT
     ▶ Verifikasi: boundary-scan/JTAG
Bahasan

Metodologi Desain
  Metodologi dan Alur Desain HDL
  Komponen Desain HDL
  Spesifikasi Sistem

Kegiatan Desain
  Manajemen Proyek
  Membuat Modul Baru
  Proses Kompilasi
  Memprogram Device

Simulasi HDL
Proyek Desain

- Satu proyek desain dapat mempunyai beberapa file desain
  - Satu file memuat satu modul desain
  - Satu desain mempunyai satu modul paling atas *(top module)*, dan modul-modul penyusun/komponen lainnya
  - Desain metodologi bisa top-down atau bottom-up
    - *top-down*: desain dari top module sistem kemudian dijabarkan komponen-komponen penyusunnya
    - *bottom-up*: desain dari komponen paling kecil yang kemudian disusun untuk membangun sistem
  - Komponen bersifat *reusable* (dapat digunakan oleh proyek lain)
Komponen Desain

Xilinx Webice ISE mengenal setidaknya 4 tipe file:

1. File teks HDL (ekstensi * .v untuk verilog, * .vhd untuk VHDL)
2. File skematik (ekstensi * .sch)
3. File definisi pin/ User Constraint File (ekstensi * .ucf)
4. File konfigurasi/programming
   - File konfigurasi bitstream FPGA (ekstensi * .bit)
   - File PROM dan konfigurasi FPGA untuk flash (ekstensi * .mcs)
1. File Desain HDL Modul (*.v, *.vhd)

- HDL (hardware description language)
  - verilog: *.v dan vhdl: *.vhd
- mendeskripsikan modul-modul desain (*composable*)
  - struktur: pin dan interkoneksi
  - perilaku
- Contoh:
  - Satu modul full-adder 1-bit (FA) akan terdiri atas 3 masukan (x, y dan cin) dan 2 keluaran (sum, cout)
  - Perilaku FA adalah \( \text{sum} = x \oplus y \oplus \text{cin} \) dan \( \text{cout} = xy + (x + y)\text{cin} \)
2. File Desain Skematik (*.sch)

- File ini merupakan alternatif masukan desain secara grafis
- Simbol-simbol modul komponen saling diinterkoneksikan untuk menghasilkan satu modul baru
3. User Constraint File (*.ucf)

- Mendefinisikan konstrains yang harus dipenuhi oleh desain
  - konstrains waktu/speed
  - daya/power
  - penempatan/placement pad logika di pin device target
- Misalnya: pin LD-7 disambungkan dengan pin F9 FPGA XC3S500E, dengan TTL low-voltage (dibahas selanjutnya)

```plaintext
NET "led<7>" LOC = "F9" IOSTANDARD = LVTTL ;
```
4. File Konfigurasi/Programming (*.bit, *.mcs)

1. File konfigurasi bitstream FPGA yang bisa diprogram langsung lewat JTAG (ekstensi *.bit)
   - File ini dibangkitkan dari ISE Project Navigator;

2. File PROM dan konfigurasi FPGA yang akan disimpan di flash secara permanen dan digunakan dalam mode Slave-Serial (ekstensi *.mcs)
   - File ini dibangkitkan dengan PROM File Formatter di program ISE iMPACT;
Bahasan

Metodologi Desain
- Metodologi dan Alur Desain HDL
- Komponen Desain HDL
- Spesifikasi Sistem

Kegiatan Desain
- Manajemen Proyek
- Membuat Modul Baru
- Proses Kompilasi
- Memprogram Device

Simulasi HDL
Struktur dan Perilaku

Spesifikasi sistem memuat spesifikasi kebutuhan desain yang diinginkan

1. Struktur
   - Struktur dinyatakan dalam diagram sistem masukan-keluaran yang akan dirancang
   - Diagram menunjukkan interkoneksi antara device masukan, FPGA dan device keluaran
   - Bisa dinyatakan dengan HDL maupun skematik

2. Perilaku
   - Perilaku menyatakan fungsi masukan-keluaran yang diinginkan
   - Dinyatakan dengan HDL
     - Dalam block always, initial
Struktur: Diagram Blok Rancangan

- Menggambarkan antarmuka masukan dan keluaran sistem
- interkoneksi antar-blok, interkoneksi sistem dengan peripheral I/O
Antarmuka dan peripheral I/O

- **Masukan:**
  - 4 buah push-button: BTN North, BTN East, BTN South, BTN West
  - 1 buah push-button: ROT-A/B
  - 4 buah saklar geser: SW0, SW1, SW2, SW3

- **Keluaran:**
  - 8 buah LED: LD0-7

- Pin antarmuka I/O didefinisikan dalam file UCF (**User Constraint File**)
  - Nama, tipe, konstrain lainnya
Tombol Tekan dan Rotary-Knob

- Keluaran sinyal BTN_* akan '1' (high) saat tombol ditekan

```
NET "BTN_EAST" LOC = "H13" | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_NORTH" LOC = "V4" | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_SOUTH" LOC = "K17" | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_WEST" LOC = "D18" | IOSTANDARD = LVTTL | PULLDOWN;
NET "ROT_CENTER" LOC = "V16" | IOSTANDARD = LVTTL | PULLDOWN;
```
Saklar Geser dan LED

\[
\begin{align*}
\text{SW3} & \quad \text{(N17)} \quad \text{SW2} \quad \text{(H18)} \quad \text{SW1} \quad \text{(L14)} \quad \text{SW0} \quad \text{(L13)} \\
\text{HIGH} & \\
\text{LOW} & \\
\text{NET "SW}<0>" & \text{ LOC = "L13" | IOSTANDARD = LVTTL | PULLUP ; } \\
\text{NET "SW}<1>" & \text{ LOC = "L14" | IOSTANDARD = LVTTL | PULLUP ; } \\
\text{NET "SW}<2>" & \text{ LOC = "H18" | IOSTANDARD = LVTTL | PULLUP ; } \\
\text{NET "SW}<3>" & \text{ LOC = "N17" | IOSTANDARD = LVTTL | PULLUP ; } \\
\text{NET "LED}<7>" & \text{ LOC = "F9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<6>" & \text{ LOC = "E9" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<5>" & \text{ LOC = "D11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<4>" & \text{ LOC = "C11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<3>" & \text{ LOC = "F11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<2>" & \text{ LOC = "E11" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<1>" & \text{ LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; } \\
\text{NET "LED}<0>" & \text{ LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ; }
\end{align*}
\]
Perilaku Sistem
Menjabarkan Hubungan Input-Output Sistem

- Saat keadaan rotary-knob UP/tidak tertekan (ROT_CENTER=0), penyalaan LED mengikuti keadaan saklar geser dan tombol tekan
  - Misalnya: saat SW0=SW2=0 maka LD0 dan LD2 akan menyala (bernilai 1). Saat tombol WEST ditekan LD6 akan menyala (bernilai 1);
- Saat keadaan rotary-knob DOWN/tertekan (ROT_CENTER=1), penyalaan LED merupakan inversi dari keadaan saklar geser dan tombol tekan
  - Misalnya: saat SW0=SW2=0 maka LD0 dan LD2 akan mati, yang lainnya menyala. Saat tombol WEST ditekan LD6 akan mati (bernilai 0);
1. Membuat dan manajemen proyek
2. Menambahkan file desain ke dalam proyek, yaitu file 
   *.v dan *.ucf
   ▶ Membuat modul baru atau
   ▶ menyertakan modul yang sudah ada
3. Mengkompilasi proyek: sintesis dan implementasi
4. Membangkitkan file programming *.bit
5. Mengkonfigurasi (memprogram) device target FPGA 
   menggunakan iMPACT;
Bahasan

Metodologi Desain
Metodologi dan Alur Desain HDL
Komponen Desain HDL
Spesifikasi Sistem

Kegiatan Desain
Manajemen Proyek
Membuat Modul Baru
Proses Kompilasi
Memprogram Device

Simulasi HDL
Membuat Proyek Desain

Urutan untuk membuat proyek desain adalah:

1. Nama proyek, lokasi dan tipenya
2. Menentukan device dan propertinya

Sebuah workspace akan dibuat dan siap untuk merancang sistem
1. Nama proyek, lokasi dan tipenya

![New Project Wizard]

- **Name:** Master-Modul1-InputOutput
- **Description:** Modul#1: Masukan Keluaran
  <Master> Ganti dengan nama kelompok praktikum
- **Top-level source type:** HDL
2. Menentukan device dan propertinya

![New Project Wizard](image)

**Device Properties**

Specify device and project properties.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Category</td>
<td>All</td>
</tr>
<tr>
<td>Family</td>
<td>Spartan3E</td>
</tr>
<tr>
<td>Device</td>
<td>XC3S500E</td>
</tr>
<tr>
<td>Package</td>
<td>FQ320</td>
</tr>
<tr>
<td>Speed</td>
<td>-4</td>
</tr>
<tr>
<td>Top-Level Source Type</td>
<td>HDL</td>
</tr>
<tr>
<td>Synthesis Tool</td>
<td>XST (VHDL/Verilog)</td>
</tr>
<tr>
<td>Simulator</td>
<td>I$Sim (VHDL/Verilog)</td>
</tr>
<tr>
<td>Preferred Language</td>
<td>Verilog</td>
</tr>
<tr>
<td>Manual Compila Order</td>
<td></td>
</tr>
</tbody>
</table>

More Info  < Back  Next >  Cancel
Workspace telah Siap
Bahasan

Metodologi Desain
  Metodologi dan Alur Desain HDL
  Komponen Desain HDL
  Spesifikasi Sistem

Kegiatan Desain
  Manajemen Proyek
  Membuat Modul Baru
  Proses Kompilasi
  Memprogram Device

Simulasi HDL
Menambah Modul HDL

Modul HDL akan dibuat:

1. Klik tombol new source untuk memulai menambahkan satu file modul baru
   1.1 Untuk modul top, file DHL dan UCF-nya perlu untuk dibuat
2. Memberikan nama port dan tipenya untuk modul
3. Memprogram perilaku (prosedural) atau struktur modul
1. Menambah File Baru

- Tipe kode sumber perlu ditambahkan

![New Source Wizard](image-url)
2. Menentukan Port untuk Modul

![Source Wizard](image)

**Define Module**

Specify ports for module.

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Direction</th>
<th>Bus</th>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTN_NORTH</td>
<td>input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN_EAST</td>
<td>input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN_SOUTH</td>
<td>input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN_WEST</td>
<td>input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW</td>
<td>input</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ROT_CENTER</td>
<td>input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED</td>
<td>output</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

[More Info] [Back] [Next] [Cancel]
File Desain Telah Dibuat

// Additional Comments:

module input_output(
    input BTN_NORTH,
    input BTN_RAST,
    input BTN_SOUTH,
    input BTN_WEST,
    input [3:0] SW,
    input ROT_CENTER,
    output [7:0] LED
);

//Elemen penyimpan untuk LED*/
reg [7:0] LED;

//Blok always dieksekusi saat sinyal dalam daftar berubah*/
always @(BTN_NORTH or BTN_RAST or BTN_SOUTH or BTN_WEST or SW or ROT_CEN
begin
    if (ROT_CENTER == 0) begin
        LED[3:0] <= SW; /*Non-blocking assignment*/
        LED[7:4] <= (BTN_NORTH, BTN_EAST, BTN_SOUTH, BTN_WEST);
module input_output(
    input BTN_NORTH,
    input BTN_EAST,
    input BTN_SOUTH,
    input BTN_WEST,
    input [3:0] SW,
    input ROT_CENTER,
    output [7:0] LED
);

/*Elemen penyimpan untuk LED*/
reg [7:0] LED_REG;

/*Blok always dieksekusi saat sinyal dalam daftar berubah*/
always @(BTN_NORTH or BTN_EAST or BTN_SOUTH or BTN_WEST or SW
or ROT_CENTER)
begin
    if (ROT_CENTER == 0) begin
        LED_REG[3:0] <= SW; /*Non-blocking assignment*/
        LED_REG[7:4] <= {BTN_NORTH, BTN_EAST, BTN_SOUTH, BTN_WEST};
    end else begin /*Invers*/
        LED_REG[3:0] <= ~SW; /*Non-blocking assignment*/
    end

    assign LED = LED_REG; /*Continuous assignment*/
endmodule
Membuat File UCF

- untuk mendefinisikan pemetaan sinyal ke pin FPGA
Bahasan

Metodologi Desain
  Metodologi dan Alur Desain HDL
  Komponen Desain HDL
  Spesifikasi Sistem

Kegiatan Desain
  Manajemen Proyek
  Membuat Modul Baru
  Proses Kompilasi
  Memprogram Device

Simulasi HDL
Sintesis dan Implementasi

- Dilakukan dari top module
- mensintesisis fungsi menjadi RTL dan membangkitkan file konfigurasi untuk FPGA
Melihat Design Summary dan Skematik RTL
Bahasan

Metodologi Desain
Metodologi dan Alur Desain HDL
Komponen Desain HDL
Spesifikasi Sistem

Kegiatan Desain
Manajemen Proyek
Membuat Modul Baru
Proses Kompilasi
Memprogram Device

Simulasi HDL
Memprogram Device Lewat JTAG

- Konfigurasi FPGA Dengan iMPACT
Chain JTAG untuk Konfigurasi

INFO:IMPACT:2257 - Startup Clock has been changed to 'JtagClk' in the bitstream stored in memory, but the original bitstream file remains unchanged.

UserID read from the bitstream file = 0xFFFFFFFF.

INFO:IMPACT:551 - '1': Added Device xc3s500e successfully.
Simulasi Fungsional

- Mensimulasikan perilaku sistem
- Membuat file testbench (test pattern)
  - Membangkitkan nilai-nilai masukan untuk menguji sistem yang didesain dan mengamati nilai keluarannya
  - Nama file biasanya diakhiri dengan _tb, contoh: <design>_tb.v
timescale 1ns/10ps
module mux_8_1s151_tb;
   // outputs
   reg [2:0] sel;
   reg [7:0] din;
   reg strobe;
   // inputs
   wire y;
   parameter cycle = 5;
   // Instantiate the Unit Under Test (UUT)
   mux_8_1s151 uut(.sel(sel), .din(din),
       .strobe(strobe), .y(y));
initial begin
   $fsdbDumpfile("mux_8_1s151.fsdb");
   $fsdbDumpvars;
   sel = 0; din = 0; strobe = 0; //time=0
   #(cycle) sel= 3'b000; din = 8'b11111111; strobe = 1; // time=5
   #(cycle) sel= 3'b010; din = 8'b00000000; strobe = 1; //time=10
   #(cycle*2) sel= 3'b000; din = 8'b01010111; strobe = 0; //time=20
   $display("%d", y);
   #1 $finish;
end
endmodule
Program Simulasi di Xilinx ISE

- Program yang digunakan adalah ISIM (terintegrasi dengan Xilinx ISE)
- Diset dari property desain (menu Project -> Design Properties...)
- Merupakan tahap simulasi perilaku
- Verifikasi desain dilakukan dengan menganalisis diagram pewaktu