Verilog HDL – 0
Introduction of Verilog HDL and Xilinx Vivado

黃元豪
Yuan-Hao Huang

國立清華大學電機工程學系
Department of Electrical Engineering
National Tsing-Hua University

Outline

- Introduction
- Sample Design
- Structural Modeling
- Logic Modeling and Simulation Using Xilinx Vivado
- An Example of Combinational Circuits

Introduction

Verilog HDL

- Verilog 硬體描述語言(Verilog Hardware Description Language)
  - 在積體電路設計（特別是超大型積體電路的計算機輔助設計）的電子設計自動化領域中，**Verilog HDL**是一種用於描述、設計電子系統（特別是數位電路的硬體描述語言）。
  - Verilog是電力電子工程師學會（IEEE）的1364號標準。

- Verilog 硬體描述語言在邏輯設計上的用途
  - 用途一: 邏輯電路設計的模擬與驗證
  - 用途二: FPGA邏輯電路(**IC電路)的設計與實作

    ** IC電路的設計與實作在【EE4292積體電路設計實驗】教授。

Verilog HDL Utilization Scenario I

- Simulation and verification of logic circuits on PC.

Specification

2-to-1 Multiplexer

Logic Design

Verilog HDL Coding

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

not U0(sel_n,sel);
and U1(t1,a,sel);
and U2(t2,b,sel_n);
or    U3(out,t1,t2);

endmodule
```

Test Pattern

a, b, sel
000
001
010
…
111

Verilog HDL **Simulator**

- Imaging what happens when the circuit is as complex as a CPU or MP3 player processor.

Verilog HDL Utilization Scenario II

- Design and Implementation of logic circuits in FPGA (IC)

Specification

2-to-1 Multiplexer

Verilog HDL Coding

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

 assign out = sel ? a:b;

endmodule
```

Verilog HDL Logic **Synthesizer**

Synthesized Logic Circuits

FPGA **Implement Design** and **Programming**

Now, it can work !

Verilog HDL – Levels of Abstraction (2/2)

- Behavioral Level (Architectural/Algorithmic Level)
  - Describes a system by the flow of data between its functional blocks
  - Defines signal values when they change
- Register Transfer Level (Dataflow)
  - Describe a system by the flow of data and control signals between and within its functional blocks
  - Defines signal values with respect to a clock
  - RTL (Register Transfer Level) is frequently used for the Verilog description with the combination of behavioral and dataflow constructs which is acceptable to logic synthesis tools.
- Gate Level (Structural)
  - A model that describes the logic gates and the interconnections between them
- Transistor/Switch/Physical Level
  - A model that describes the transistors and the interconnections between them

Verilog HDL – Levels of Abstraction (1/2)

Level of Abstraction Example

- Describe the operation of a circuit at various levels of abstraction (MP3 Player Decoder as example)
  - Behavior (MP3 Decoding, C/C++, HDL)
  - Function (Filtering, Fourier Transformation, C/C++, HDL)
  - Structure (Adder/Subtractor, Multiplier, Divider, HDL)

- Compared with C/C++ Program, verilog HDL
  - describes the timing (delay) of a circuit
  - expresses the concurrency (parallelism) of circuit operation

Behavior Level Abstraction

- Describe the design without implying any specific internal architecture
  - Use high level constructs (@, case, if, repeat, wait, while)
  - Usually use behavioral construct in test-bench
  - Synthesis tools accept only a limited subset of behavior level description
    - Case 1 : assign Z=(S) ? A: B;

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire out;

assign out = (sel) ? a : b ;

endmodule
```

Behavior Level Abstraction

- Case 2: always @ (input1 or input2 or …)

```
begin
    out1=
end
```

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
reg out;

always @(a or b or sel)
  if (sel)
    out=a;
  else
    out=b;
endmodule
```

Gate Level Abstraction

- Gate-level abstraction describes a pure structural design of circuits.
  - You must derive and draw the circuit schematics first before writing Verilog code.

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

not U0(sel_n,sel);
and U1(t1,a,sel);
and U2(t2,b,sel_n);
or   U3(out,t1,t2);

endmodule
```

VLSI Design Flow

Verilog Simulation

Sample Design

Scenario

Device under Test (DUT)
Stimulus & Control
U0
A
B
a
b
out
sel
OUT
SEL
U1
A'
B'
a
b
out
sel
OUT '
SEL'
Response Generation & Verification
a
b
out
sel
SMUX
Module + Testbench

Verilog Module

- module module_name(port_names);
  - Port declaration
  - Data type declaration
  - Task & function declaration
  - Module functionality or structure
  - Timing Specification
- endmodule

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

not U0(sel_n,sel);
and U1(t1,a,sel);
and U2(t2,b,sel_n);
or  U3(out,t1,t2);

endmodule
```

Testbench (1/4)

- module testfixture;
  - Declare signals
  - Instantiate modules
  - Applying stimulus
  - Monitor signals
- endmodule

Testbench (2/4)

- Declare signals
  - Test pattern must be stored in storage elements first and then apply to DUT (Device under Test)
    - Use "reg" to declare the storage element
- Instantiate modules
  - Both behavioral level or gate level model can be used.

Testbench (3/4)

- Describing Stimulus
  - The testbench always be described behaviorally.
  - Procedural blocks are bases of behavioral modeling.
  - The simulator starts executing all procedure blocks at time 0 and executes them concurrently.
  - Two types of procedural blocks
    - initial
    - always

Testbench (4/4)

```
module test_SMUX;

reg          A,B,SEL;
wire    OUT;

SMUX U0(.out(OUT),.a(A),.b(B),.sel(SEL));

initial
begin
        A=0;B=0;SEL=0;
        #10     A=0;B=1;SEL=1;
        #10     A=1;B=0;
        #10     SEL=0;
        …..
        #10     SEL=1;
end

endmodule
```

Declare signals

Make an instance

Assign values to storage elements

#10 to specify 10 time unit delay

Structural Modeling

Verilog Primitives

- and : Logical AND
- or : Logical OR
- not : Inverter
- buf : Buffer
- xor : Logical exclusive OR
- nand : Logical AND inverted
- nor : Logical OR inverted
- xnor : Logical exclusive OR inverted

Structural Modeling

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

not U0(sel_n,sel);
and U1(t1,a,sel);
and U2(t2,b,sel_n);
or  U3(out,t1,t2);

endmodule
```

assign

- **assign** continuous construct
  - combinational logics

```
module SMUX (out,a,b,sel);
output out;
input a,b,sel;

  assign out = (a&sel) | (b&(~sel));

endmodule
```

This out has to be declared as “wire” or “output” data type.
This expression can not be inside always @().

always

- **always** statements

sensitivity list

```
module SMUX (out,s,b,sel);
output out;
input a,b,sel;
reg out;

always @(a or b or sel)
    out = (a&sel) | (b&(~sel));

endmodule
```

This out has to be declared as "reg" data type.

Logic Modeling and Simulation Using Xilinx Vivado

Design Flow

- General design flow
  - Design construction
  - Behavioral simulation
  - Design implementation
  - Timing simulation
- HDL-based design Flow

Important Notes

- **Draw schematic first** and then construct Verilog codes.
- Verilog RTL coding philosophy is not the same as C programming
  - Every Verilog RTL construct has its own logic mapping (for synthesis)
  - You should have the logics (draw schematic) first and then the RTL codes
  - You have to write **synthesizable** RTL codes

Open Vivado

Vivado

Vivado 2019.2
File
Flow
Tools
Window
Help
Quick Access
VIVADO
HLx Editions
XILINX
Quick Start
Create Project
Open Project
Open Example Project
Tasks
Manage IP
Open Hardware Manager
Xilinx Tcl Store
Learning Center
Documentation and Tutorials
Quick Take Videos
Release Notes Guide
Recent Projects
binary_counter
C:/Users/hp/LD/binary_counter
nsw
C:/Users/hp/LD/nsw
stopwatch
C:/Users/hp/LD/stopwatch
shift_reg
C:/Users/hp/LD/shift_reg
ssd
C:/Users/hp/LD/ssd
Tcl Console

Open New Project (1/3)

Open New Project (2/3)

New Project

Project Type

Specify the type of project to create.

- RTL Project
  You will be able to add sources, create block designs in IP Integ and analysis.
  - Do not specify sources at this time
- Post-synthesis Project: You will be able to add sources, view de
  - Do not specify sources at this time
- I/O Planning Project
  Do not specify design sources. You will be able to view part/pac
- Imported Project
  Create a Vivado project from a Synplify, XST or ISE Project File.
- Example Project
  Create a new Vivado project from a predefined template.

New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: Parts Boards

Filter

Product category: All

Family: Artix-7

Package: cpg236

Speed grade: -1

Temp grade: All Remaining

Reset All Filters

Search: xc7a35tcp (1 match)

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	Gb Transceivers	Available IOBs	LUT Elements
xc7a35tcpg236-1	236	50	90	41600	2	2	106	20800

? < Back Next > Finish Cancel

New Project

VIVADO HLx Editions

New Project Summary

A new RTL project named 'lab1' will be created.

The default part and product family for the new project:
Default Part: xc7a35tcpg236-1
Product: Artix-7
Family: Artix-7
Package: cpg236
Speed Grade: -1

To create the project, click Finish

XILINX ALL PROGRAMMABLE.

? < Back Next > Finish Cancel

Finish

Open New Project (3/3)

smux - [C:/Users/hp/LD/smux/smux.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - smux

Sources

- Design Sources
- Constraints
- Simulation Sources
- Utility Sources

Hierarchy | Libraries | Compile Order

Properties

Select an object to see properties

Project Summary

Overview | Dashboard

Settings Edit

Project name:	smux
Project location:	C:/Users/hp/LD/smux
Product family:	Artix-7
Project part:	xc7a35tcpg236-1
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Synthesis

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a35tcpg236-1
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	None

Implementation

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a35tcpg236-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

DRC Violations

Run Implementation to see DRC results

Timing

Run Implementation to see timing results

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy	Report Stra
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2019)	Vivado Syn
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Imp

New Source (1/5)

Press and Right Click

Add Sources

New Source (2/5)

New Source (3/5)

New Source (4/5)

New Source (5/5)

Add Testbench (1/5)

smux - [C:/Users/hp/LD/smux/smux.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - smux

Sources

Design Sources (1)

smux (smux.v)

Constraints

Simulation Sources (1)

sim_1 (1)

Utility Sources

Hierarchy Libraries

Press and Right Click

Properties... Ctrl+E

Hierarchy Update

Refresh Hierarchy

IP Hierarchy

Edit Constraints Sets...

Edit Simulation Sets...

Add Sources... Alt+A

Source File Properties

smux.v

Enabled

Location: C:/Users/hp/LD/smux/smux.srcs/sources_1/new

Type: Verilog

Library: xil_defaultlib

Size: 0.6 KB

General Properties

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS
synth_1	constrs_1	Not started			
impl_1	constrs_1	Not started			

Project Summary smux.v *

C:/Users/hp/LD/smux/smux.srcs/sources_1/new/smux.v

```
// Description:
//
// Dependencies:
```

Add Sources

VIVADO HLx Editions

Add Sources

This guides you through the process of adding and creating sources for your project

- ○ Add or create constraints
- ○ Add or create design sources
- ◉ Add or create simulation sources

XILINX

? < Back Next > Finish Cancel

Add Testbench (2/5)

Add Testbench (3/5)

Add Sources

Add or Create Simulation Sources

Specify simulation specific HDL files, or directories containing HDL files, to your project.

Specify simulation set: sim_1

In...	Name	Library	Location
1	test_smux.v	xil_defaultlib	<Local to Project>

Add Files | Add Directories | Create File

Scan and add RTL include files into project

Copy sources into project

Add sources from subdirectories

Include all design sources for simulation

< Back | Next > | Finish | Cancel

Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Module name: test_smux

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
	input	☐	0	0

OK | Cancel

Add Testbench (4/5)

Add Testbench (5/5)

smux - [C:/Users/hp/LD/smux/smux.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - smux

Sources

- Design Sources (1)
  - smux (smux.v)
- Constraints
- Simulation Sources (2)
  - sim_1 (2)
    - smux (smux.v)
    - test_smux (test_smux.v)
- Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

test_smux.v

Enabled

Location: C:/Users/hp/LD/smux/smux.srcs/sim_1/new

Type: Verilog

Library: xil_defaultlib

Size: 0.5 KB

General Properties

Project Summary | smux.v | test_smux.v *

C:/Users/hp/LD/smux/smux.srcs/sim_1/new/test_smux.v

```
21
22
23 module test_smux();
24 wire OUT;
25 reg A, B, SEL;
26
27 smux U0(.a(A), .b(B), .sel(SEL), .out(OUT));
28
29 initial
30 begin
31
32 A=0;B=0;SEL=0;
33 #10 A=0;B=0;SEL=1;
34 #10 A=0;B=1;SEL=0;
35 #10 A=0;B=1;SEL=1;
36 #10 A=1;B=0;SEL=0;
37 #10 A=1;B=0;SEL=1;
38 #10 A=1;B=1;SEL=0;
39 #10 A=1;B=1;SEL=1;
40 #10 A=0;B=0;SEL=0;
41 end
42
```

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy	Report Stra
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2019)	Vivado Syn
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Imp

34:19 Insert Verilog

Simulation (1/4)

smux - [C:/Users/hp/LD/smux/smux.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

PROJECT MANAGER
- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR
- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION
- Run Simulation
  - Run Behavioral Simulation
  - Run Post-Synthesis Functional Simulation
  - Run Post-Synthesis Timing Simulation
  - Run Post-Implementation Functional Simulation
  - Run Post-Implementation Timing Simulation

RTL ANALYSIS
- Open Elabo

SYNTHESIS
- Run Synthesis
- Open Synthesized Design

IMPLEMENTATION
- Run Implementation
- Open Implemented Design

PROGRAM AND DEBUG
- Generate Bitstream
- Open Hardware Manager

Press and Select

PROJECT MANAGER - smux

Sources
- Design Sources (1)
  - smux (smux.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - test_smux (test_smux.v) (1)
- Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

Size: 0.7 KB

General Properties

Project Summary | smux.v | test_smux.v

C:/Users/hp/LD/smux/smux.srcs/sim_1/new/test_smux.v

```
21
22
23 module test_smux();
24 wire OUT;
25 reg A, B, SEL;
```

```
37 #10 A=1;B=0;SEL=1;
38 #10 A=1;B=1;SEL=0;
39 #10 A=1;B=1;SEL=1;
40 #10 A=0;B=0;SEL=0;
41 end
42
```

Run Simulation

Executing simulate step...

Background Cancel

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy	Report Stra
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2019)	Vivado Syn
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2019)	Vivado Imp

Vivado Simulator

Simulation (2/4)

Simulation (3/4)

Use to zoom in or zoom out

Scroll this to the beginning

Simulation (4/4)

Press to close simulation

smux - [C:/Users/hp/LD/smux/smux.xpr] - Vivado 2019.2

File Edit Flow Tools Reports Window Layout View Run Help Quick Access Ready

10 us Default Layout

Flow Navigator

PROJECT MANAGER
- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR
- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION
- Run Simulation

RTL ANALYSIS
- Open Elaborated Design

SYNTHESIS
- Run Synthesis
- Open Synthesized Design

IMPLEMENTATION
- Run Implementation
- Open Implemented Design

PROGRAM AND DEBUG
- Generate Bitstream
- Open Hardware Manager

SIMULATION - Behavioral Simulation - Functional - sim_1 - test_smux

smux.v | test_smux.v | Untitled 1

Name	Value
OUT	0
A	0
B	0
SEL	0

0.000 ns, 10.000 ns, 20.000 ns, 30.000 ns, 40.000 ns, 50.000 ns, 60.000 ns, 70.000 ns, 80.000 ns, 90.000 ns, 100.000 ns, 110.000 ns

Tcl Console | Messages | Log

Sim Time: 1 us