Hardware Design with VHDL

Instructor:

Prof. Jim Plusquellic

Text:

RTL Harware Design Using VHDL: Coding for Efficiency, Portability, and Scalability, Pong P. Chu, ISBN-13: 978-0-471-72092-8 ISBN-10: 0-471-72092-5

Supplementary texts:

FPGA Prototyping By VHDL Examples, Xilinx Spartan-3 Version, Pong P. Chu, ISBN: 978-0-470-18531-5

Web: http://www.ece.unm.edu/~jimp/vhdl_fpgas
Web: http://www.ece.unm.edu/jimp/vhdl_fpgas

Course Goals

Simply stated, my objective is to teach you to:

Take an algorithm and/or specification, either combinational, sequential, or an FSMD, write synthesizable VHDL code for it, and download & debug the code on an FPGA

To accomplish this, you will learn:

- A set of fundamental components of digital design
- A set of synthesizable constructs in VHDL -- a hardware description language
- Xilinx Vivado software development tools

Platform Choices

- Full-custom ASIC
- Standard cell ASIC
- Gate array ASIC
- Complex field programmable logic device (FPGA), CPLD
- Off-the-shelf ICs (Commercial off-the-shelf or COTS)

Device Technologies

Standard Cell:

- Circuit made of a set of pre-defined logic, known as standard cells, with basic logic gates, NAND, NOR, D FF, etc.
- Layout of a cell is pre-determined, but layout of the complete circuit is customized
- Masks needed for all layers

Gate Arrays:

- Circuit is built from an array of a single type of cell (known as base cell)
- Base cells are pre-arranged and placed in fixed positions, aligned as one- or two-dimensional array
- More sophisticated components (macro cells) can be constructed from base cells
- Masks needed **only** for metal layers (connection wires)

Device Technologies

FPGAs/CPLDs:

Device consists of an array of generic logic cells and general interconnect structure

Logic cells and interconnect can be "programmed" by utilizing "semiconductor fuses or "switches", i.e., customization is done "in the field"

Two categories:

CPLD (Complex Programmable Logic Device)

FPGA (Field Programmable Gate Array)

No custom mask needed

Comparison of Technologies

Area (Size): silicon "real-estate"

Standard cell is the smallest since the cells and interconnect are customized

FPGA is the largest, overhead for "programmability", capacity cannot be completely utilized

Device Technologies

Comparison of Technologies (cont.)

Cost

NRE (Non-Recurrent Engineering) cost: one-time, per-design cost

Part cost: per-unit cost

Time-to-market "cost" loss of revenue

- Standard cell: high NRE, small part cost but large lead time
- FPGA: low NRE, large part cost but small lead time

SUMMARY	FPGAs	Gate Array	Standard Cell
Tailored masks	0	3 to 5	15 or more
Area			best (smallest)
Speed			best (fastest)
Power			best (minimal)
NRE cost	best (smallest)		
Per part cost			best (smallest)
Development cost	best (easiest)		
Time to market	best (shortest)		
Per unit cost		depends on volume	

System Representation

View: different perspectives of a system

- **Behavioral view**

  Describe functionalities and I/O behavior

  Treat the system as a black box

- **Structural view**

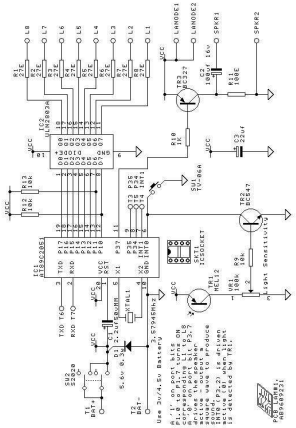
  Describe the internal implementation (components and interconnections)

  Essentially a block diagram

- **Physical view**

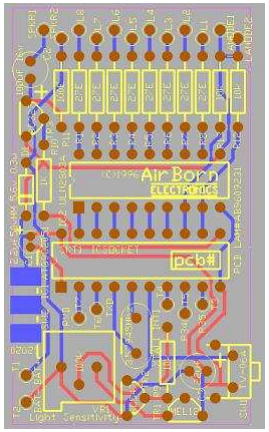
  Adds more info to structural view: component size, component locations, routing of wires, e.g., layout of a printed circuit board

Structure and physical view:



Structural

Physical



Levels of Abstraction

How does one manage complexity for a chip with 10 million transistors?

Abstraction: simplified model of a system

The model shows only the selected features and **ignores** detail

The purpose of abstraction is to reduce the amount of data to a manageable level and to enable the focus to be on only the *critical features*.

Levels of Abstraction

Four levels

- Transistor level
- Gate level (schematic)
- Register transfer level (RTL) (where we will work)
- Processor level

RTL:

A design methodology in which the system operation is described by how the data is manipulated and moved among registers

Development Tasks

The main tasks associated with the development of a custom digital circuit:

- Synthesis
- Physical design
- Verification
- Testing

Synthesis

A **refinement process** that realizes a description with components from the lower abstraction level

Type of synthesis:

- High-level synthesis (Behavioral to RT-level description)
- RT level synthesis (RTL behavioral to RT-level structural representation)
- Gate level synthesis (RT-level structural to gate level)
- Technology mapping (gate level to standard cells or CLBs in FPGAs)

Physical Design

Refines the **structural view** to a **physical view** and generates a layout

Development Tasks

Verification

Check whether a design meets the specification and performance goals

- Functionality
- Performance (timing)

Methods of Verification

Simulation

spot check: cannot verify the absence of errors

Can be computation intensive

Timing analysis

Just check delay

Formal verification

Apply formal math techniques to determine properties

E.g, equivalence checking

Hardware emulation

Development Tasks

Testing

Testing is the process of detecting physical defects in a die or a package that occurred at the time of manufacturing