# EE-4702-1—Digital Design Using Verilog

01-1

Basic Information

Call Number 1584 (Spring 2001)

URL: `http://www.ee.lsu.edu/v`

Offered by:

David M. Koppelman, Room 349 EE Building, 578-5482.

`koppel@ee.lsu.edu`, `http://www.ee.lsu.edu/koppel/koppel.html`

Tentative office hours: Monday, Wednesday 9:40-11:10; Thursday 14:30-16:30.

Teaching Assistant:

TBA





01-2

Formal Prerequisite

Credit or Registration in EE 3750 or permission of instructor.

Informal Prerequisites (What you need to know.)

C (The computer language.)

Digital hardware design: Should be able to answer these:

Is a ripple adder something to get excited about?

What does it mean to clock a register?





01-3

Graded Material

Midterm Exam, 35%

Fifty minutes, open notes.

Final Exam, 35%

Two hours, open notes.

Yes, it's cumulative.

Homework/Projects, 30%

Written and computer assignments.

Lowest grade or unsubmitted assignment dropped.





# Electronic Design Automation (EDA)

01-4

Electronic Design Automation (EDA) (Short Definition)

The use of software to automate electronic (digital and analog) design.

Electronic Design Automation (EDA) (Longer Definition)

Electronic design in which

the design is entered using *design capture tools*

or using a text editor and a *hardware description language*

possibly consisting of "parts" from a vendor's library

the functionality of the design is verified by simulation

the correctness, testability, and compliance of a design is checked by software

and the design is converted to a manufactureable form using *synthesis tools*.

## Hardware Description Languages

*Hardware Description Language*
A language used for describing the structure of hardware or how the hardware should behave.

Some Hardware Description Languages

A Hardware Programming Language (AHPL)

APL-like syntax. (APL was an early language for representing math.)

Developed at the University of Arizona.

Used in Hill and Peterson's *Digital Systems* textbook.

Not used in industry.





Verilog

Widely used in industry.

C-like syntax. (Many elements of the language are different from C.)

Developed by Gateway Design Automation in 1984, later bought by Cadence.

Became IEEE standard 1364 in 1995.

VHSIC Hardware Description Language (VHDL)

Widely used in industry.

Ada-like syntax. (Ada is a DoD-developed language for large embedded systems.)

Developed as part of U.S. Department of Defense (DoD) VHSIC program in 1983

Became IEEE standard 1076 in 1987.





Next Generation Languages

Efforts underway to extend Verilog and VHDL.

Extensions intended for building "systems on a chip."

Language Popularity

Both Verilog and VHDL are widely used in industry.

Verilog is considered easier to use.





## Design Flow

*Design Flow*
The steps used to produce a design, from initial design entry to the generation of the final manufacureable form. Describes which programs will be used, when they will be used, and how they will be used.

EDA tool vendors usually provide design flows that show how their products can be used.

Companies develop design flows that are used to produce their designs.

A simple design flow is described below.

## Simple Design Flow

### Simple Design Flow (Simple Flow, for short)

Three easy steps (not counting step zero).

Used to describe the major steps in a typical design flow.

### List of Steps in Simple Design Flow

Simple Flow Step 0: Goal Determination.

Simple Flow Step 1: Design Capture

Simple Flow Step 2: Behavioral Verification

Simple Flow Step 3: Synthesis and Timing Verification

### Simple Flow Step 0:

Start with: an idea for a new chip.

Goal: a box full of the new chips.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.




### Simple Flow Step 1: Design Capture

Using the back of an envelope or some other suitable medium ...
... develop a rough draft of the design.

Using a text editor ...
... write a *Verilog description* of the design.

Using a text editor ...
... write a *Verilog description* of a *testbench* used to test the design.

The testbench generates inputs for the design and verifies the design's outputs.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.




### Simple Flow Step 2: Behavioral Verification

Using a *simulator* and *waveform viewer* ...
... check if design passes testbench tests ...
... and if not, debug.

Waveform viewer is sort of a virtual logic analyzer, can view signals on any part of design.

Simulator output includes messages generated by behavioral code ...
... including "pass" or "fail" message produced by testbench.

Using text editor ...
... fix bugs, and tune performance.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.




### Simple Flow Step 3: Synthesis and Timing Verification

Using synthesis programs ...
... generate *design database*.

Design database has information needed to fabricate the chip ...
... and to perform simulations with accurate timing.

Re-simulate, and verify that timing is acceptable ...
... if timing is not acceptable edit the Verilog structural description and repeat steps above.

Using the Internet, E-mail design database and credit card number to fab.

After a few weeks, get parts back in mail.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.

## Topics Covered in This Course

- Coding in Verilog.
- Writing structural and synthesizable descriptions.
- Writing testbench code.
- Using simulation, waveform viewers and similar tools.
- Synthesis.
- Using synthesis tools.





## Software Used in This Course

Workstation Labs: Mentor Graphics

- Programs available for simulation (Modelsim) and synthesis (Leonardo).
- Full versions of programs used in industry.

Home Use: Simucad Silos Demo Version.

- Simulation only.
- Intended for demonstration and educational use so design size limited.





## Design Capture

*Design Capture*
Entering a design in electronic form.

- Start: Idea in engineer's head, scribbles on back of envelope.
- Finish: Design in electronic form readable by some EDA tools.

### Design Capture Methods

Schematic Capture

- Enter design using GUI (graphical user interface) schematic editor.
- Easy for beginners but tedious for all but small designs.

Finite-State Machine Editors

- Programs meant for designing FSM, to be part of larger design.





### Hardware Description Languages

- Entered using standard or specialized text editor.

## HDL Descriptive Styles

01-17

*Descriptive style* refers to a set of rules that a description adheres to.

HDL's are used to write a hardware *description* or *model* (don't call it a program).

Descriptive Styles

*Structural:* how parts are connected together, like a schematic.

*Behavioral:* what hardware is supposed to do.

*Register Transfer Language (RTL):* a form in which registers are explicit. Covered later in semester.

Intent of structural code is the description of hardware.

Intent of behavioral code can be

description of hardware, for use by a synthesis program

testbench, used to verify correctness of other descriptions

simulate a part not yet designed in detail





## Synthesis Design Target

01-18

*Design Target*
The type of device to be manufactured or programmed. Synthesis programs generate output for a particular design target.

Design Targets

*Programmable Logic Array (PLA)*

Chip that can be programmed (once) to implement a logic function.

Usually programmed at the factory.

PLAs might be used in prototypes or when only a few parts are needed.

*Application-Specific Integrated Circuit (ASIC)*

A fully custom chip.

Usually the fastest design target, can have the most components.





01-19

*Gate Array*

A chip full of gates manufactured in two steps:

First generic layers containing gates are fabricated ...
... but gates are not connected to each other.

Later, metal layers connecting gates are added.

Designer using gate arrays specifies only metal layer.

Since gates fabricated in advance time is saved.

*Field-Programmable Gate Array (FPGA)*

A chip full of logic whose connection and function can be programmed and later re-programmed.





## Typical Synthesis Steps

01-20

Start With:

Choice of Design Target

Type of target: FPGA, ASIC, etc.

Manufacturer and family.

A synthesis program or programs.

Behavioral or Structural Description

Functionality has been verified by simulation.

Behavioral description (if used) follows synthesizebility rules specified for synthesis program.

01-21

Major Synthesis Steps (Summary)

Synthesis of technology-independent gate-level description.

Map gates and modules to technology-specific versions.

*Place and route*.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.
01-21


01-22

Major Synthesis Steps (Details)

Synthesis of *technology-independent* gate-level description.

Synthesis program infers registers and minimizes logic.

Registers aren't explicitly declared (even though it will appear otherwise) ...
... so synthesis program must determine (infer) where they are needed.

Because (most) synthesis programs minimize combinational logic ...
... descriptions should be written for clarity.

Output of this step is purely structural code ...
... consisting of gates and standard modules (*e.g.*, for arithmetic), and library modules.

Based on output, designer might tweak design or give hints to synthesis program.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.
01-22


01-23

Place and Route

*Placement* is the determination of the physical location of a part.

*Routing* is the determination of paths for wires interconnecting parts.

Output of this step:

Timing information (since technology and wire lengths are known) which may be ...
... *backannotated* (written into) the original behavioral description.

Behavioral descriptions re-simulated to see if they meet timing criteria.

For FPGAs, code to program the devices.

For ASICS and gate arrays, ...
... a design database to *tape-out* and send to a fab.

Fabrication facilities apply additional steps, not covered here.


EE 4702 Lecture Transparency. Formatted 10:02, 17 January 2001 from lsli01.
01-23