EE 4755—Digital Design Using Hardware Description Languages

Final Exam Review

When / Where

Wednesday, 6 December 2017, 15:00-17:00 CST

225 Tureaud Hall (Here)

Conditions

Closed Book, Closed Notes

Bring one sheet of notes (both sides), 216 mm $\times$ 280 mm.

No use of communication devices.

Format

Several problems, short-answer questions.

Resources

Lecture "slides" and code used in class: `http://www.ece.lsu.edu/koppel/v/ln.html`

Study Guides

Synthesis: `http://www.ece.lsu.edu/koppel/v/guides/syn.pdf`

Solved tests and homework:. `http://www.ece.lsu.edu/koppel/v/prev.html`

It is important that homework solutions be studied.

Study Recommendations

Study this semester's homework assignments **and solutions**. Similar problems may appear on the exam.

Solve Old Problems—memorizing solutions **is not the same** as solving.

Following and understanding solutions **is not the same as** solving.

Use the solutions for brief hints and to check your own solutions.

Previous Exams

Be sure to look at previous midterm and final exams, but note differences in coverage.

Course Material Areas

Verilog

The System Verilog language, including structural and behavioral code.

Synthesis

How hardware is inferred, mapped, and optimized from Verilog.

Digital Design

The functioning of the circuits covered in class.

How to design digital circuits.

Sequential Circuit Design

Pipelined Circuit Design

Tools

Understand what simulation and synthesis tools do.

Specific Circuits

Combinational and Sequential Left Shifters

Multipliers

Verilog Topics

Objects

See `http://www.ece.lsu.edu/v/2017/l020-types.v.html`.

Object Types: *variable* v *net* objects.

Key difference:...
... variables are assigned, nets are driven (connected to something).

Data Types

Four-State Integer Types

Two-State Integer Types

Floating-Point Types

String Type

Integer Data Types

Four-State Integer Types: `logic`, `integer`, `time`.

Two-State Integer Types: `int`, `bit`, `byte`, `shortint`, `longint`.

Integer qualifiers: `signed`, `unsigned`.

Real Data Types

Real Types: `real`, `shortreal`.

Type punning: `$realtobits`, `$bitstoreal`, etc..

Arrays

Packed v. Unpacked Arrays

```
uwire [7:0] e_pluribus_unum;

uwire plain_array [7:0];
```

Element and bit numbering:

```
uwire [7:0] color;

uwire [0:7] colour;
```

Static, Dynamic, and Associative arrays.

Modules

Port and parameter declaration.

Module and primitive instantiation.

Object declarations.

Continuous `assign`.

Procedural code.

Generate statements.

Procedural Code

Execution of `initial`, `always`, `always_comb`, and `always_ff`.

Delays (*e.g.*, `#5`).

Event controls (*e.g.*, `@( posedge clk)`).

Blocking v. non-blocking assignment.

Elaboration and Generate Statements

`http://www.ece.lsu.edu/v/2017/l025-gen-elab.v.html`

Elaboration-time constants.

Difference between a module parameter and a port.

Emphases, Key Skills

Verilog—Key Skills

Given a design in one form, write design in another:

Explicit Structural

Implicit Structural

Synthesizable Behavioral

Logic Diagram

Use generate statements to interconnect modules.

Synthesis Key Skills

Given Verilog code:

Show inferred hardware (before optimization).

Show expected optimizations.

Logic Design Skills

Cost and Delay Computation

`http://www.ece.lsu.edu/v/2017/lsli-cost-model.pdf`

Compute Cost using Simple Model

Compute Delay using Simple Model

Synthesis Topics

Understand what is done during inference, optimization, technology mapping.

`http://www.ece.lsu.edu/v/2017/l040-syn.v.html`.

Inference of combinational logic.

`http://www.ece.lsu.edu/v/2017/l042-syn-comb.v.html`

`http://www.ece.lsu.edu/v/2017/l045-syn-comb-behav.v.html`

Inference of registers.

`http://www.ece.lsu.edu/v/2017/lsli-syn-seq.pdf`

Optimization of combinational logic.

Use of timing constraints in synthesis.

Digital Design Topics

Digital Design Topics

Common Components

- Multiplexor
- Binary Full Adder, Ripple Adder
- Integer Equality and Magnitude Comparison

Common Component Skills

- Show how to implement using basic gates.
- Know how to optimize for special cases (a constant input, etc.).

Cost and Delay Estimation

Simple Cost Model

`http://www.ece.lsu.edu/v/2017/lsli-cost-model.pdf`

Cost of n-input AND and OR gates are $n - 1$ units.

Inverters (NOT gates) are free.

Delay of n-input gate is $\lceil \lg n \rceil$ units.

Tools

Synthesis (RTL Encounter).

- `read_hdl`, `elaborate`
- `define_clock`
- `synthesize -to_generic`
- `synthesize -to_mapped`
- `report area`, `timing`