LAB 0: FPGA BASIC METHODOLOGY

James E. Stine, Jr.
Edward Joullian Endowed Chair in Engineering
Oklahoma State University
Electrical and Computer Engineering Department
Stillwater, OK 74078 USA
james.stine@okstate.edu

Objective

- The first lab has three main objectives:
  - The first objective is to learn how to design a basic digital design and verify it using a testbench.
  - The next objective is to implement it on your DSDB board and compare it to what you get from simulation.
  - The last objective is to understand the lab structure and report.
- History has indicated many people will wait – you cannot complete any lab overnight, be forewarned.
  - We will revisit lab report later today.
  - Also, you probably cannot finish this lab in one lab session.

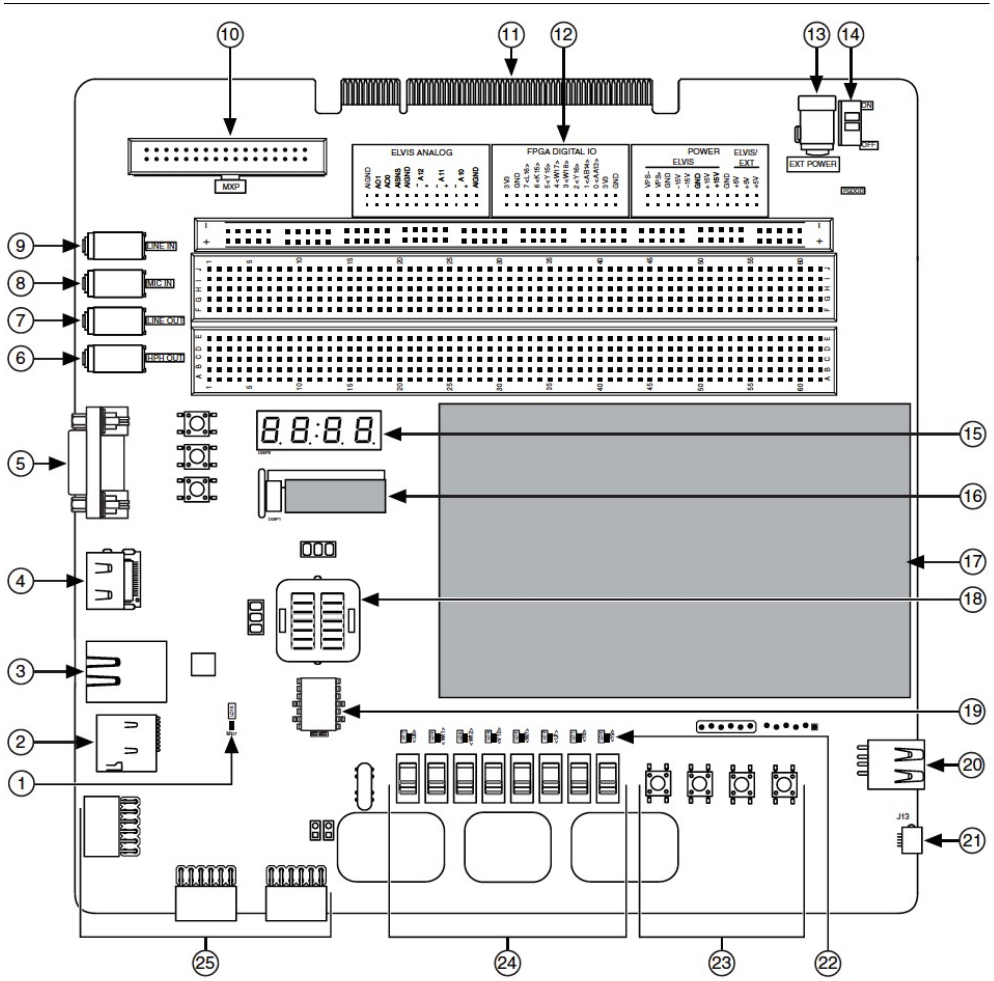
Things to keep in mind

- Start early and work with your partner to get things done quickly.
  - Communication is key to a good relationship.
  - Do not ask what I can do, just jump in and contribute even if you duplicate work.
- Labs are due one week after the lab is complete.
  - For lab 0, this Friday February 4
- Try to schedule the last week of lab for writing up your work, so you basically only have 2 weeks to complete this lab.
  - Back up often and work upwards (inverted pyramid).
  - Ask questions on slack or in person ***early***!
  - Know we are here to help – post screenshots/questions on slack or stop by.

Basics

- For this laboratory, we will be using the ELVIS III board.
- We have made most of it very simple if you follow the procedure and understand what is going on.
- I have uploaded a version of the files that include a directory with the items you need to copy over to your working area once your simulation works.
- There is also simulation that I will demonstrate how to make sure this is working **before** you go to the ELVIS board.
- Make sure you have a DSDB board at your desk before you begin.
  - Ask your TA to install it, if its not there.

DSDB

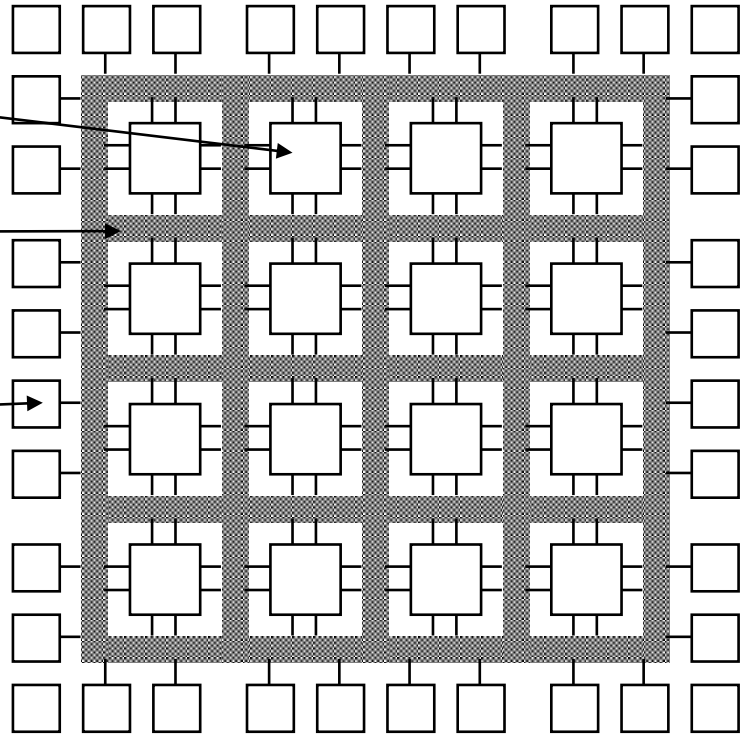


18: Zynq XC7Z020-1CLG484C with included Heat Sink

All of the FPGA pins are accessible via the DSDB board either through the schematic or attaching to the debugger

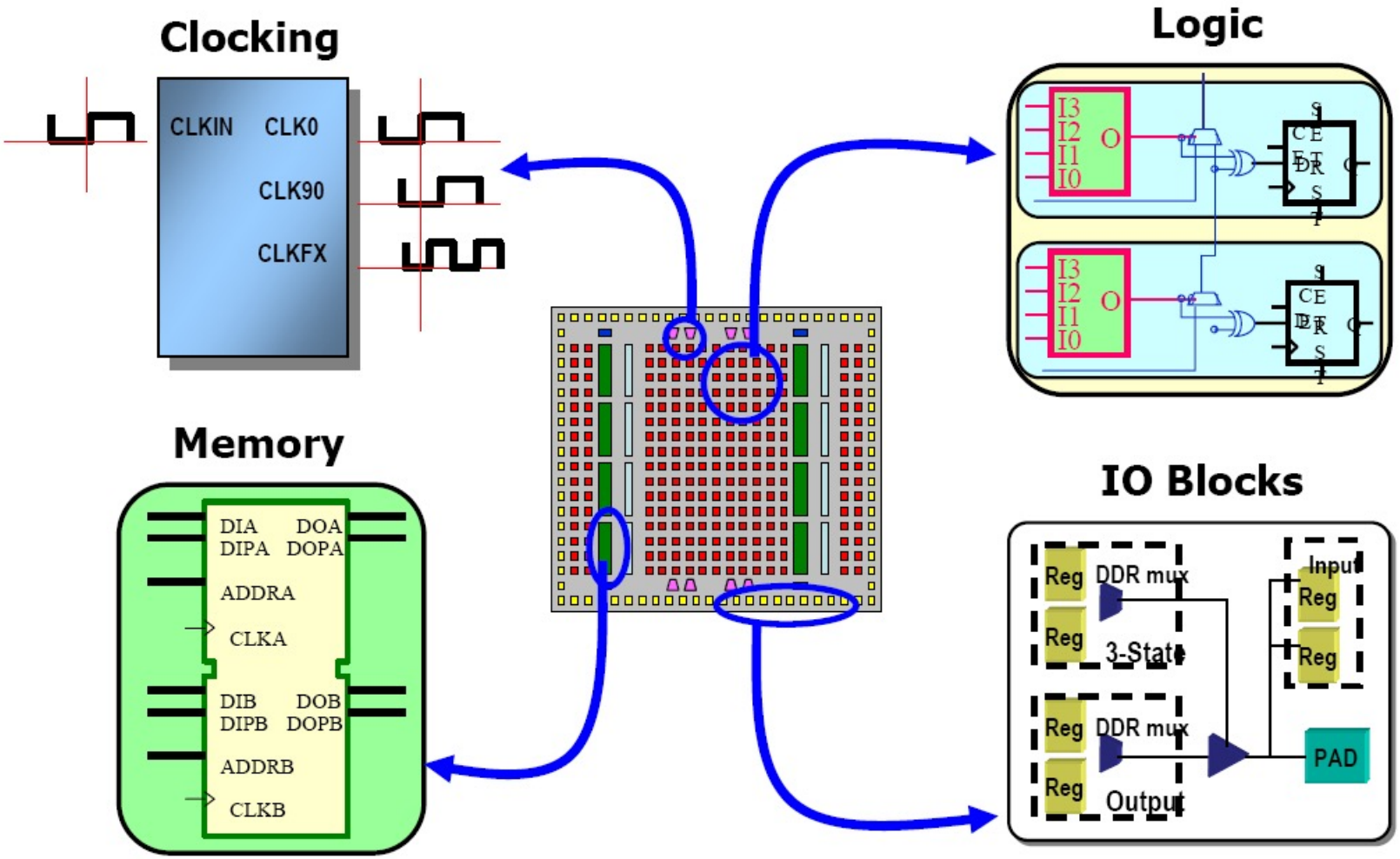
Field-Programmable Gate Arrays

- Logic blocks
  - To implement combinational and sequential logic
- Interconnect
  - Wires to connect inputs and outputs to logic blocks
- I/O blocks
  - Special logic blocks at periphery of device for external connections



- Key questions:
  - How to make logic blocks programmable?
  - How to connect the wires?
  - *After the chip has been fabbed* or fabricated!

Evolution of FPGA architectures



Tasks

- Download `labs0.zip` from Canvas
  - Unzip your files somewhere (e.g., Desktop)
- First test sillyfunction SystemVerilog with ModelSim
  - Navigate to your unzipped files.
  - Try compiling silly.sv with: `vlog silly.sv`
    - Examine testbench with text editor to see what is happening
  - Simulate and Examine with ModelSim: `vsim –do silly.do`
- Implement the full adder (3 bits in, 2 bits out) in SystemVerilog
  - Boolean Logic is given and implement new SystemVerilog file using silly.sv as a template
  - Modify the silly_tb.sv for the FA (Do not try to reinvent the wheel – just change the silly items and the vectors)
    - Test all 8 inputs!
  - Simulate with ModelSim and Check that it works
    - You might want to write down the correct truth table on paper
    - Examine that the truth table matches the waveform for every combination ($2^3 = 8$ combinations)
- Much of these tasks can be done on your own outside or inside laboratory

HDL to Gates

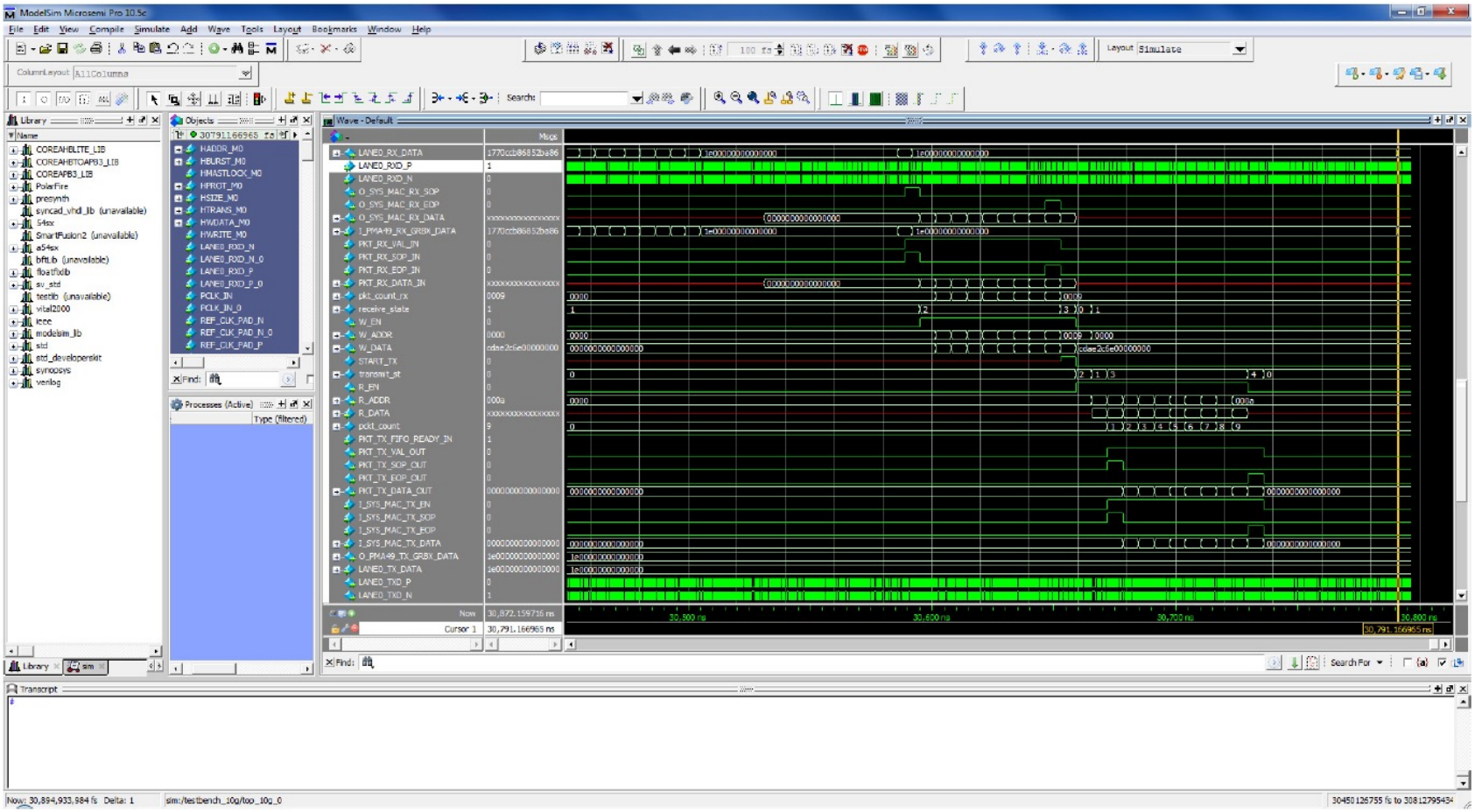
- **Simulation**
  - Inputs applied to circuit
  - Outputs checked for correctness
  - Millions of dollars saved by debugging in simulation instead of hardware

- **Synthesis**
  - Transforms HDL code into a *netlist* describing the hardware (i.e., a list of gates and the wires connecting them)
- **Verification**
  - Verifying the forward translation of synthesis is indeed what you started with

Demo of Simulation

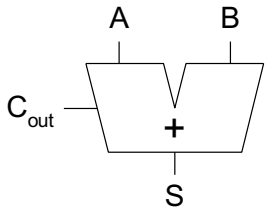
- Let's try to simulate `silly.do`

MGC ModelSim `vsim` window



Full Adder

Half Adder

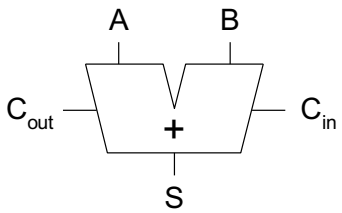


A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = A \oplus B$

$C_{out} = AB$

Full Adder



C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$S = A \oplus B \oplus C_{in}$

$C_{out} = AB + AC_{in} + BC_{in}$

- The full adder is sometimes called the basic building block of any computation.
- Specifically, the full adder is sometimes called a (3,2) counter because it counts the numbers of inputs that are 1.
- Use the basic silly module discussed in Lecture 3 for building the full adder.
- Make sure you test it thoroughly for all 8 combinations in your simulation with a testbench before you go to your DSDB board.

Instantiating designs

```
module and3(input  logic a, b, c,
            output logic y);
  assign y = a & b & c;
endmodule


module inv(input  logic a,
           output logic y);
  assign y = ~a;
endmodule


module nand3(input  logic a, b, c
             output logic y);
  logic n1;                      // internal signal

  and3 andgate(a, b, c, n1);     // instance of and3
  inv  inverter(n1, y);          // instance of inv
endmodule
```

- When creating HDL, you instantiate other designs in your higher-level HDL.
- This allows you to build bigger and better designs.
- It uses hierarchy to make more complex designs.
- `andgate` is our instantiated name

Debugging?

- You will be implementing your design on your FPGA on the National Instruments DSDB board.
  - The implementation will be inside the silicon which you cannot see.
  - So, how do you figure out if something is working? How do you tell something does not work the way you expected it to?
- Use debugging techniques!
  - This includes switches, LEDs, push buttons to help you examine what is happening with the board.
  - Try to use as many LEDs as you can to see what is on and what is off.
  - Use your brain to figure out possible problems.

Implementation on DSDB

- Once you simulate your design, you should hook up to the DSDB.
- Inside your `lab0.zip` there should be a Demo program.
  - Unzip and open with Vivado
- Modify the `top_demo.sv` to instantiate your "working" SV and hook up inputs and outputs to things you can see on the DSDB:
  - Hook your input/output of your instantiated design to debugging items (e.g., LEDs).
  - Test all 8 possible combinations of your full adder to make sure its working as expected.

Port	Type	Description
`sw[3:0]`	Input	push buttons (#23)
`btn[7:0]`	Input	SPDT slide switches (#24)
`led[7:0]`	Output	Light Emitting Diodes (LEDs) (#22)

Table 1: Ports Used for Lab 0

- Using Vivado (which is Xilinx's synthesis and place/route tool) - this should be on your desktop in ENDV 360 to run
  1. Run Synthesis
  2. Run Implementation
  3. Generate Bitstream
- Per the lab0.pdf, download to the board using the Hardware Manager

Check on DSDB board

- DO NOT TRY AN IMPLEMENTATION ON YOUR DSDB BOARD UNTIL IT WORKS IN SIMULATION!
  - It takes several minutes to go through the three steps inside Vivado (e.g., synthesis).
  - You want to hopefully do the implementation once or twice.
  - Use your debugging items and your brain to make sure things are working.
- Check on DSDB board whether things work (check against the truth table).
  - Use the Light-Emitting Diodes (LEDs) to make sure all 8 possible cases match your simulation from ModelSim and theory.
- If you like, save any output from the Vivado screen for your report
  - e.g., Schematic, Implementation Results
- Write your report!

Lab Report

- The lab report should be simple and efficient with 5 sections (much of this information is in the rubric)
  - Section 1: Introduction
  - Section 2: Baseline Design
  - Section 3: Design
  - Section 4: Testing Strategy
  - Section 5: Evaluation
  - Team Evaluation

One report for each team but both team members *must* submit a Team Evaluation to get full credit!

- Check Lab Report Formats
  - Use a 10-pt font
  - Your report should be a **maximum** of four pages not including figures.
  - Figures and Tables go in the Appendix
- Do not read too much into the rubric: just get it done to showcase what you did.

Last-minute Items

- Challenge yourself and get things done early!
  - Last semesters students waited until the second week when its too late!
  - You may have to visit the lab more than once! (i.e., you cannot always complete your work in one lab session)
- Submit all HDL, scripts, lab reports, and team evaluation for each member through Canvas.
  - Do not forget your HDL, testbenches, DO files or you will get a deduction.
  - Also, do not forget the team evaluation from each team member!!!!
- Believe in yourself – we all tend to be negative about our abilities (don't!)
  - You can do it!
  - OSU students are the best!
- Enjoy the opportunity!