2012 Fall CSE140L

Digital Systems Laboratory

by

Dr. Choon Kim

CSE Department
UCSD
Welcome to CSE140L!
CSE140L Overview
(by Choon Kim)

Three main goals
#1. Introduction to CAD tool (Altera Quartus II CAD SW)
#2. Logic Design & Testing Skills using HDL modeling, simulation and synthesis (Verilog)
#3. System Implementation & Testing on FPGA HW Board with real-time clock (Altera Cyclone II FPGA)

LAB Activities

Complexity

Comb. CKT design using Schematic Capture

Comb. & Sequential CKT. using Verilog

FSM system design: Soda Vending Machine Controller Design

A small computer system design with common & special instructions (testing done by Assembly test codes)

Closed book exam covering all

LAB Activities

LAB#1

LAB#2

LAB#3

LAB#4

Final Exam

Time

Comb. CKT design using Schematic Capture

Comb. & Sequential CKT. using Verilog

FSM system design: Soda Vending Machine Controller Design

A small computer system design with common & special instructions (testing done by Assembly test codes)

Closed book exam covering all

LAB Activities

LAB#1

LAB#2

LAB#3

LAB#4

Final Exam

Time
Preparations you need to do
(Ref: course webpage)

1. Make a LAB group and obtain your group number

2. Purchase the Cyclone II FPGA Starter Development Kit (a.k.a. DE1 kit).

3. Install CAD SW on your computer.
   A directory, C:\altera\90sp2\..., will be created on your computer
Homework #1

After installing CAD SW, study & practice the following documents and become familiar with Kit operation

During your study, it will help you a lot later if you pay close attention to

a) inputs, e.g., SW(Switches), KEY(PushButton), and

b) outputs, e.g., LEDG(LED green), LEDR(LED red), HEX(7-segment display).

- **Getting Started with Altera’s DE1 Board** (Here, make sure you install a USB-Blaster Driver properly on your PC. It should work. In case of trouble during Driver installation, you may try this new driver.)

- **DE1 User Manual** (Ch1,2,4 contain useful information needed for our LAB projects)

- **Quartus II Introduction Using Schematic Design** (Here, follow the tutorial and complete the design. Implement a light controller circuit on DE1 board by following the step-by-step instructions described in the document.)
What is CAD flow?
Figure 1. Typical CAD flow.
CAD flow

- **Design Entry** – the desired circuit is specified either by means of a schematic diagram, or by using a hardware description language, such as Verilog or VHDL.

- **Synthesis** – the entered design is synthesized into a circuit that consists of the logic elements (LEs) provided in the FPGA chip.

- **Functional Simulation** – the synthesized circuit is tested to verify its functional correctness; this simulation does not take into account any timing issues.
CAD flow

- **Fitting** – the CAD Fitter tool determines the placement of the LEs defined in the netlist into the LEs in an actual FPGA chip; it also chooses routing wires in the chip to make the required connections between specific LEs.

- **Timing Analysis** – propagation delays along the various paths in the fitted circuit are analyzed to provide an indication of the expected performance of the circuit.

- **Timing Simulation** – the fitted circuit is tested to verify both its functional correctness and timing.

- **Programming and Configuration** – the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs and establish the required wiring connections.
General Design Flow

Languages:
C, System C, Verilog, VHDL

Types:
Behavior Description
Structure Description
Algorithm Architecture

Register Transfer Level Description
Logic Synthesis

Netlist of Logic
Placement, Routing

Physical Layout

Mask Fabrication
FPGAs

1. Design Specification: Hardware Description
2. Synthesis: Logic, Physical Layout
3. Analysis: Functional, Timing Verification
Example: Combinational circuit design flow:

Design requirements

=> Truth Table

=> (minimized) Boolean equation for each output

=> CAD Design flow

=> system on HW(FPGA, Board)
## Combinational Logic

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>16 different outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0000000011111111</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0000111100001111</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0011001100110011</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0101010101010101</td>
</tr>
</tbody>
</table>

Q: What is the name of logic function for each output?
(e.g., AND, NOT, XOR, etc.)
Q: How to design N-way light controller?
Example: 2-way light controller design

Figure 11. The light controller circuit.
Combinational circuit design flow (cont'd):

Example: 2-way light controller (or N-way light controller in general)

Assuming the initial condition of “the output is OFF when input \( x_1 = x_2 = 0 \) (=Down)”, the design requirement is "Odd number of 1s in the input switches turns the output light ON, and even number of 1s in the input switches turns the output light OFF".

\[
\begin{array}{c|c|c}
 x_1 & x_2 & f \\
 0 & 0 & 0 (=OFF) \\
 0 & 1 & 1 (=ON) \\
 1 & 0 & 1 \\
 1 & 1 & 0 \\
\end{array}
\]

\[
f = (\sim x_1 \ AND \ x_2) \ OR \ (x_1 \ AND \ \sim x_2)
\]

=> QuartusCAD

=> Sim. => Program in DE1 board => Testing

=> Demo to others!
Timing behavior

- Real circuits have delays
- Gate delay – time for an output of the gate to change after its input changes
- We can simulate timing delays in Quartus II to see these delays

- \textbf{Tpd}
  Specifies the maximum acceptable \textit{input to non-registered output} delay, that is,
  the time required for a signal from an \textit{input pin} to propagate through combinatorial logic and appear at an \textit{output pin}. 
Gate delay

- Notice rise time, fall time, and gate delay:
Quartus II Timing Simulation

Notice the glitches and delay in the output
Digital Technologies

CPU (Central Processing Unit)
GPU (Graphics Processing Unit)
DSP (Digital Signal Processor)
SoC (System on Chip)
FPGA (Field Programmable Gate Array)
ASIC (Application Specific Integrated Circuit)
Custom Designs
etc.