Electronic Design Automation (EDA)

EE 260
University of Hawaii
Outline

• Design Flow
  – Hardware description languages (HDL), e.g., verilog and VHDL

• Programmable Logic
  – PALs and PLAs
  – FPGAs
Simplified Design Flow

**A description (or a model) of a circuit**

<table>
<thead>
<tr>
<th>Design Problem</th>
<th>Schematic</th>
<th>Hardware Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image.png" alt="Diagram" /></td>
<td><strong>Language (HDL)</strong></td>
</tr>
<tr>
<td>Design Circuit</td>
<td></td>
<td><strong>Verilog or VHDL</strong></td>
</tr>
<tr>
<td>Verify/Simulate</td>
<td></td>
<td><strong>module Xcircuit(a0,a1,a2,y)</strong></td>
</tr>
<tr>
<td>Functionality</td>
<td></td>
<td>input a0, a1, a2;</td>
</tr>
<tr>
<td>(Debugging)</td>
<td></td>
<td>output y;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wire w1, w2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assign w1 = a0&amp;a1;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assign w2 = ~a2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assign y = w1&amp;w2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>endmodule</td>
</tr>
</tbody>
</table>

This can be used to simulate design or to implement in hardware

We’ll focus on these, but there’s more!
Simplified Design Flow

Hardware Description Language (HDL)
Verilog or VHDL

module Xcircuit (a0,a1,a2,y)
input a0, a1, a2;
output y;
wire w1, w2;

assign w1 = a0&a1;
assign w2 = ~a2;
assign y = w1&w2;
endmodule
Design Problem

Design Circuit

Verify/Simulate Functionality (Debugging)

HDL model of a circuit (functional model, can be somewhat abstract)

Logicworks

Draw schematic

Simulate in Logicworks

HDL

Write HDL code

Simulate using tools such as Modelsim synopsis
Design Problem

Design Circuit

Verify/Simulate Functionality (Debugging)

HDL model of a circuit (functional model, can be somewhat abstract)

HDL model of a circuit

**Synthesize** model to get a gate level description

Verify/Simulate Logic and Timing

Implement in hardware

Make sure design is consistent in hardware
Computer Aided Design (CAD)
Hardware Technologies

• Programmable Logic Devices (PLDs)
  – Programmable Read Only Memory (PROM). Erasable PROMs (EPROMs)
  – Programmable Arrayed Logic (PALs) and Programmable Logic Arrays (PLAs)*

• Field Programmable Gate Arrays (FPGAs)*

• Application Specific ICs (ASICs)
Note that the next set of slides are (heavily) modified versions of slides found at http://subjects.ee.unsw.edu.au/~elec1041 by Saeid Nooshabadi. The originals were adapted from R. Katz’s *Contemporary Logic Design*
Programmable Logic Arrays (PLAs)

- Programmable technology for combinatorial logic
  *Sum of Products*

- Array of ANDs followed by an array of ORs. Prefabricated

- Programmable by deleting connections at intersections
Programmable Array Logic (PALs)

- Each OR has its own set of ANDs (product terms)
- Easier to build, faster, and most cases it isn’t much of a limitation
PLD (Programmable Logic Devices)

Registered PAL

Can implement a Mealy or Moore circuit

These things can be big: Complex PLDs (CPLDs)

Note feedback
Field-Programmable Gate Arrays

- **Logic blocks**
  - To implement small combinational and sequential circuits

- **Interconnect**
  - Wires and switches to connect logic blocks to each other and to inputs/outputs

- **I/O blocks**
  - Special logic blocks at periphery of device for external connections
Configurable Logic Block (CLB)

- 5-input, 1 output function or two 4-input, 1 output functions
- optional register on output
Can be configured to any small combinational or sequential circuit. In the case of comb circuits, the flip flops are bypassed.
Xilinx 4000 Interconnect

We can connect CLBs and IOBs by using wires and PSMs.

Figure 28: Single- and Double-Length Lines, with Programmable Switch Matrices (PSMs)
Xilinx 4000 IOB

Similar to CLB but it’s used to connect pins of the chip with the internal circuit

Pad can be programmed as an input or output
FPGA
Final Comments

• ASICs
  – Usually cheaper (in bulk) and better performance
  – Goes to foundary and takes time. Better once final design is done -- no changes

• FPGAs
  – Better for very rapid design and redesign. Good for prototyping but also end design.
  – Better for small numbers of products
  – More expensive, and less in performance