FPGA Design
EECE 277

Implementation Technology Part 2

Dr. William H. Robinson
January 26, 2005

http://eecs.vanderbilt.edu/courses/eece277/

Topics

“Nothing harder than being given your chance. At least, that's what I hear.”

– Uncle Joe
from Great Expectations (1998)

• Administrative stuff
  – Office hours cancelled today
  – Laboratory Assignment #1 (due Friday, February 4)

• ASICs

• Programming technologies

• FPGA Architecture

Guest Lecture

• Dr. Robinson will be on travel
  – Friday, January 28, 2005

• Presentations on current FPGA research
  – Jason Scott
    • Embedded System Design Tools and Xilinx FPGAs
  – Philippe Adell
    • Implementing a Rad-Hard digital power controller with an FPGA

Potential Final Project Ideas!!!

Laboratory Assignment #1

• Quartus II Software
  – Any installation issues?

• UP2 Design Laboratory Kits
  – All teams are assigned

• Laboratory Report
  – Great Expectations

• Potential Lab Session
  – Tuesday?
Programmable Logic Devices (PLDs)

Inputs (logic variables)

Logic gates and programmable switches

Outputs (logic functions)

- General-purpose chip for implementing logic circuitry
- Customizable “black box” with gates and switches

Programmable Logic Devices (PLDs)

- Programmable Logic Devices (PLDs)
  - SPLDs
  - CPLDs
  - PROMs
  - PLAs
  - PALs
  - GALs
  - etc.

Programming Technologies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Symbol</th>
<th>Predominantly associated with ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusible-link</td>
<td>(\sim) (\sim)</td>
<td>SPLDs</td>
</tr>
<tr>
<td>Antifuse</td>
<td>(\sim)</td>
<td>FPGA</td>
</tr>
<tr>
<td>PROM</td>
<td>(\sim)</td>
<td>SPLDs and CPLDs</td>
</tr>
<tr>
<td>EPROM</td>
<td>(\sim)</td>
<td>SPLDs and CPLDs</td>
</tr>
<tr>
<td>E2PROM/FLASH</td>
<td>(\sim)</td>
<td>SPLDs and CPLDs (some FPGAs)</td>
</tr>
<tr>
<td>SRAM</td>
<td>(\sim)</td>
<td>FPGA (some CPLDs)</td>
</tr>
</tbody>
</table>

Taxonomy

- Programmable Logic Devices (PLDs)
  - SPLDs
  - CPLDs
  - PROMs
  - PLAs
  - PALs
  - GALs
  - etc.

- ASICs
  - Gate Arrays
  - Structured ASICs
  - Standard Cell
  - Full Custom

Increasing complexity
Gate Array

- Part of the chip is prefabricated, part of the chip is custom fabricated to user specification
- All logic cells are identical
- Amortize cost of chip fabrication

Complete Gates

Figure 3.4. Construction of (a) NOT, (b) AND, and (c) OR gates using only NAND gates or only NOR gates.

CMOS NAND Gate

The logic function $f_1 = x_2x_3 + x_1x_3$ in the gate array

- The custom design includes the wiring connections of the gates in the array
Programming Technologies

- **SRAM**
  - Can be programmed many times
  - Must be programmed at power-up

- **Antifuse**
  - Programmed once

- **Flash**
  - Similar to SRAM but using flash memory
**FPGA Fabric (Architecture)**

- Logic
- Interconnect
- I/O pins

**Terminology**

- **Configuration:**
  - Bits that determine logic function + interconnect.
- **CLB:**
  - Combinational Logic Block = Logic Element (LE).
- **LUT:**
  - Lookup table = SRAM used for truth table.
- **I/O block (IOB):**
  - I/O pin + associated logic and electronics.

**Logic Element**

- **Programmable**
  - Input connections
  - Internal function
- **Coarser-grained than logic gates**
  - Typically 4 inputs
- **Generally includes register**
- **May provide specialized logic**
  - Adder carry chain

**Programmable Logic Block**

- 4-input LUT
- Flip-flop
- Mux
Example: Logic Element

• Lookup table:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0 1</td>
</tr>
</tbody>
</table>

Logic Synthesis

• How do we break the function into logic elements?

• How do we implement an operation within a logic element?

Placement

• Where do we put each piece of logic in the array of logic elements?

Programmable Wiring

• Organized into channels
  – Many wires per channel

• Connections between wires made at programmable interconnection points

• Must choose:
  – Channels from source to destination
  – Wires within the channels
Choosing a path

Routing Problems

- **Global routing:**
  - Which combination of channels?

- **Local routing:**
  - Which wire in each channel?

- **Routing metrics:**
  - Net length
  - Delay

Figure 3.68. Pass-transistor switches in FPGAs.
I/O Blocks

- **Fundamental selection**
  - Input, output, tri-state

- **Additional features**
  - Register
  - Voltage levels
  - Slew rate

Configuration

- **Must set control bits for**
  - LE
  - Interconnect
  - I/O blocks

- **Usually configured off-line**
  - Separate burn-in step (antifuse)
  - At power-up (SRAM)

Configuration vs. Programming

- **FPGA configuration:**
  - Bits stay at the device they program
  - A configuration bit controls a switch or a logic bit

- **CPU programming:**
  - Instructions are fetched from a memory
  - Instructions select complex operations

Reconfiguration

- **Some FPGAs are designed for fast configuration**
  - A few clock cycles, not thousands of clock cycles

- **Allows hardware to be changed on-the-fly**
FPGA Fabric Questions

- **Given limited area budget**
  - How many logic elements?
  - How much interconnect?
  - How many I/O blocks?

Logic Element Questions

- How many inputs?
- How many functions?
  - All functions of $n$ inputs or eliminate some combinations?
  - What inputs go to what pieces of the function?
- Any specialized logic?
  - Adder, etc.
- What register features?

Interconnect Questions

- How many wires in each channel?
- Uniform distribution of wiring?
- How should wires be segmented?
- How rich is interconnect between channels?
- How long is the average wire?
- How much buffering do we add to wires?

I/O Block Questions

- How many pins?
  - Maximum number of pins determined by package type.
- Are pins programmed individually or in groups?
- Can all pins perform all functions?
- How many logic families do we support?