L12: Reconfigurable Logic Architectures

Acknowledgements:


- Frank Honore

- Lecture Notes prepared by Professor Anantha Chandrakasan
History of Computational Fabrics

- Discrete devices: relays, transistors (1940s-50s)
- Discrete logic gates (1950s-60s)
- Integrated circuits (1960s-70s)
  - e.g. TTL packages: Data Book for 100’s of different parts
- Gate Arrays (IBM 1970s)
  - Transistors are pre-placed on the chip & Place and Route software puts the chip together automatically – only program the interconnect (mask programming)
- Software Based Schemes (1970’s- present)
  - Run instructions on a general purpose core
- Programmable Logic (1980’s to present)
  - A chip that be reprogrammed after it has been fabricated
  - Examples: PALs, EPROM, EEPROM, PLDs, FPGAs
  - Excellent support for mapping from Verilog
- ASIC Design (1980’s to present)
  - Turn Verilog directly into layout using a library of standard cells
  - Effective for high-volume and efficient use of silicon area
Lisp Machine
Reconfigurable Logic

- 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? (after chip has been fabbed!)
  - What should the logic granularity be?
  - How to make the wires programmable? (after chip has been fabbed!)
  - Specialized wiring structures for local vs. long distance routes?
  - How many wires per logic block?
Programmable Array Logic (PAL)

- Based on the fact that any combinational logic can be realized as a sum-of-products.
- PALs feature an array of AND-OR gates with programmable interconnect.

Diagram of PAL array with input signals, AND array, OR array, and output signals.
Each input pin (and its complement) sent to the AND array

OR gates for each output can take 8-16 product terms, depending on output pin

“Macrocell” block provides additional output flexibility...
Cypress PAL CE22V10

From Cypress

Combinational/active low
Combinational/active high

Outputs may be registered or combinational, positive or inverted

From Lattice Semiconductor

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>Output Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Registered/active low</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Registered/active high</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Combinational/active low</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Combinational/active high</td>
</tr>
</tbody>
</table>

0 = Programmed EE bit
1 = Erased (charged) EE bit
RAM Based Field Programmable Logic - Xilinx

Programmable Interconnect

Configurable Logic Blocks (CLBs)

I/O Blocks (IOBs)
Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)
Two 4-input Functions, Registered Output and a Two Input Function

Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)
5-input Function, Combinational Output

Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)
LUT Mapping

- N-LUT direct implementation of a truth table: any function of n-inputs.
- N-LUT requires $2^N$ storage elements (latches)
- N-inputs select one latch location (like a memory)

4LUT example

Why Latches and Not Registers?

Latches set by configuration bitstream
Configuring the CLB as a RAM

Memory is built using Latches not FFs

Read is same a LUT Function!

16x2
Xilinx 4000 Interconnect

Single- and Double-Length Lines, with Programmable Switch Matrices (PSMs)
Wires are not ideal!
Xilinx 4000 Flexible IOB

Adjust Transition Time

Outputs through FF or bypassed

Adjust the Sampling Edge
Add Bells & Whistles

Gigabit Serial

Hard Processor

Multiplier

I/O

Programmable Termination

Clock Mgmt (DCM)

BRAM

The Virtex II CLB (Half Slice Shown)
Adder Implementation

\[ Y = A \oplus B \oplus \text{Cin} \]

LUT: \( A \oplus B \)

Dedicated carry logic

1 half-Slice = 1-bit adder
Carry Chain

1 CLB = 4 Slices = 2, 4-bit adders

64-bit Adder: 16 CLBs


CLBs must be in same column
Virtex II Features

Double Data Rate registers

Digital Clock Manager

Embedded Multiplier

Block SelectRAM
The Latest Generation: Virtex-6

### Key Features
- **DSP with 25x18 multiplier**
- **Gigabit ethernet support**

### Comparison Table

<table>
<thead>
<tr>
<th></th>
<th>CLB</th>
<th>Dist RAM</th>
<th>Block RAM</th>
<th>Multipliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtex 2*</td>
<td>8,448</td>
<td>1,056kbit</td>
<td>2,592kbit</td>
<td>144 (18x18)</td>
</tr>
<tr>
<td>Virtex 6*</td>
<td>667,000</td>
<td>6,200kbit</td>
<td>22,752kbit</td>
<td>1,344 (25x18)</td>
</tr>
<tr>
<td>Spartan 3E</td>
<td>240</td>
<td>15kbit</td>
<td>72kbit</td>
<td>4 (18x18)</td>
</tr>
</tbody>
</table>

* Compare 2nd most performance

---

*Preliminary product information, subject to change. Please contact your Xilinx representative for the latest information.*
Technology Mapping: Schematic/HDL to Physical Logic units

Compile functions into basic LUT-based groups (function of target architecture)

```verilog
always @ (posedge clock)
begin
    if (reset) q <= 0;
    else q <= (a & b & c) | (b & d);
end
```
Design Flow – Placement & Route

- **Placement** – assign logic location on a particular device

- **Routing** – iterative process to connect CLB inputs/outputs and IOBs. Optimizes critical path delay – can take hours or days for large, dense designs

Challenge! Cannot use full chip for reasonable speeds (wires are not ideal).

Typically no more than 50% utilization.
Example: Verilog to FPGA

module adder64 (a, b, sum);
  input [63:0] a, b;
  output [63:0] sum;
  assign sum = a + b;
endmodule

64-bit Adder Example

- Synthesis
- Tech Map
- Place & Route

Virtex II – XC2V2000
How are FPGAs Used?

- **Prototyping**
  - Ensemble of gate arrays used to emulate a circuit to be manufactured
  - Get more/better/faster debugging done than with simulation

- **Reconfigurable hardware**
  - One hardware block used to implement more than one function

- **Special-purpose computation engines**
  - Hardware dedicated to solving one problem (or class of problems)
  - Accelerators attached to general-purpose computers (e.g., in a cell phone!)

FPGA-based Emulator
(courtesy of IKOS)
Personal FPGA - BASYS

- 4 pushbuttons
- 4 digit display
- VGA port
- Clock freq select
- USB prog. port & power
- 8 slide switches
- 8 LED’s
- 4 user i/o
- 16 total
- reset
- 4 digit display
- 4 pushbutton
FPGA Software

- **Xilinx ISE Web-pack**
  - A free, downloadable design environment for both Microsoft Windows and Linux - but it's a 2.25GB download.
  - All the tools and features of ISE Foundation, including the Xilinx CORE Generator™ system and FPGA Editor
  - Support for Xilinx FPGA families, including the Virtex-5 Family of platform FPGAs (does not support big FPGA's like XCV6000!)

- **Modelsim available for student download (Modelsim PE) with 6 month license.**

- **Most prototyping boards can be programmed via USB.**
Summary

- FPGA provide a flexible platform for implementing digital computing
- A rich set of macros and I/Os supported (multipliers, block RAMS, ROMS, high-speed I/O)
- A wide range of applications from prototyping (to validate a design before ASIC mapping) to high-performance spatial computing
- Interconnects are a major bottleneck (physical design and locality are important considerations)

“College students will study concurrent programming instead of “C” as their first computing experience.”

-- David B. Parlour, ISSCC 2004 Tutorial