ECE 471/571
Implementation Strategies for Digital ICs
Lecture-11
Gurjeet Singh
A growing gap between design complexity and design productivity
Generic Digital Processor

- The datapath
- The Control Module
- The memory module
- The interconnect
Generic Digital Processor
System-on-a-Chip
Impact of Implementation Choices

Energy Efficiency (in MOPS/mW)

- None
  - Hardwired custom: 100-1000
  - Configurable/Parameterizable: 10-100
  - Domain-specific processor (e.g., DSP): 1-10
  - Embedded microprocessor: 0.1-1

- Somewhat flexible
- Fully flexible
Design Methodology
From Custom to Semicustom and Structured-Array Design Approaches
Implementation Strategies

Digital Circuit Implementation Approaches

- Custom
  - Cell-based
    - Standard Cells
    - Compiled Cells
  - Macro Cells
- Semicustom
  - Array-based
    - Pre-diffused (Gate Arrays)
    - Pre-wired (FPGA's)
Custom Circuit Design
Custom Circuit Design

Advantages
● High Density and Performance

Disadvantages
● High cost
● Labor Intensive
● Long time to market

When to use:
● Reuse
● Large volumes
● Cost no bar (supercomputers etc.)
Cell-based Design Methodology
Cell-based Design Methodology

- Custom Design is too expensive and time-consuming.
- Shorter the design time, larger the penalty incurred.
- Penalty in reduced integration density and performance.
- Cell based design reduce implementation effort by reusing a limited library of cells.
- Reduces possibility of fine-tuning the design.
Standard Cell

- Provides a collection of logic gates with range of fan-in and fan-outs
- Design is captured in schematic
- Layout is generated automatically
- Cells are placed in rows, separated by routing channels.
Standard Cell Example
Historical Perspective
Historical Perspective

Every logic function can be expressed in sum-of-products format (AND-OR)

\[ f_0 = x_0 x_1 + \overline{x_2} \]

\[ f_1 = x_0 x_1 x_2 + \overline{x_2} + \overline{x_0 x_1} \]

\textit{minterm}

Inverting format (NOR-NOR) more effective

\[ \overline{f_0} = (\overline{x_0} + \overline{x_1}) + \overline{x_2} \]

\[ \overline{f_1} = (\overline{x_0} + \overline{x_1} + \overline{x_2}) + \overline{x_2} + (\overline{x_0} + \overline{x_1}) \]
Historical Perspective
Compiled Cells

- Cell libraries are expensive to make
- Have to migrate entire library to a new technology.
- Changes in minimum metal width or contact rules with new technology
- To overcome these issues. Cell layouts can generated on the fly using transistor netlists.
Automatic Cell Generation

Initial transistor geometries  Placed transistors  Routed cell  Compacted cell  Finished cell
Macrocells, Megacells and Intellectual Property

● Hard Macro
  ○ Dense layout,
  ○ Optimized
  ○ Predictable performance and power dissipation
  ○ Hard to port to newer technologies

● Soft Macro
  ○ Module represents functionality but without physical implementation
  ○ Can be ported to range of technologies and processes
  ○ Reduced design effort
  ○ Cost over wide set of designs
Intellectual Property

- Building and IC from scratch is expensive
- Third party vendors usually provide modules
- Macromodels distributed in this style are called intellectual property (IP) modules
- MCUs, DSP Processors, PCI etc.
Semicustom Design Flow
Semicustom design flow
Combining synthesis with design

- RTL (Timing) Constraints
- Physical Synthesis
- Place-and-Route Optimization
- Netlist with Place-and-Route Info
- Macromodules
  - Fixed netlists
- Artwork
Array-Based Implementation Approaches
Prediffused (or Mask-Programmable) Arrays

- **Uncommitted Cell**
  - Rows of uncommitted cells
  - Routing channel

- **Committed Cell**
  - (4-input NOR)
Sea-of-gates primitive cells

Using oxide-isolation

Using gate-isolation
Base Cell of gate-isolated gate array
Flip-flop in gate-isolated array
Pre-wired Arrays

- Based on Programming Technique
  - Fuse-based (program-once)
  - Non-volatile EPROM based
  - RAM based

- Programmable Logic Style
  - Array-Based
  - Look-up Table

- Programmable Interconnect Style
  - Channel-routing
  - Mesh networks
Fuse based FPGA

Open by default, closed by applying current pulse
Array-Based Programmable Logic

- PLA
- PROM
- PAL

+ Indicates programmable connection
+ Indicates fixed connection
Programming a PROM

\[ f_0 = x_0 x_1 + \overline{x_2} \]

\[ f_1 = x_0 x_1 x_2 + \overline{x_2} + \overline{x_0 x_1} \]
More Complex PAL

programmable AND array \((2i, 3, jk)\):

\[ j \]

\[ i \text{ inputs} \]

\[ j \text{ inputs} \]

\[ j \text{-wide OR array} \]

\[ k \text{ macrocells} \]

i inputs, j minterms/macrocell, k macrocells
2-input mux as programmable logic block

<table>
<thead>
<tr>
<th>Configuration</th>
<th>A</th>
<th>B</th>
<th>S</th>
<th>F=</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0 X 1</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>0 Y 1</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>0 Y X</td>
<td></td>
<td></td>
<td></td>
<td>XY</td>
</tr>
<tr>
<td>X 0 Y</td>
<td></td>
<td></td>
<td></td>
<td>XY</td>
</tr>
<tr>
<td>Y 0 X</td>
<td></td>
<td></td>
<td></td>
<td>XY</td>
</tr>
<tr>
<td>Y 1 X</td>
<td></td>
<td></td>
<td></td>
<td>X1 Y</td>
</tr>
<tr>
<td>1 0 X</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>1 0 Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>1 1 1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Logic Cell of Actel Fuse-Based FPGA
Look-up Table Based Logic Cell

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>
Array-Based Programmable Wiring
Mesh-based Interconnect Network

Switch Box

Connect Box

Interconnect Point

Logic Tile
Transistor Implementation of Mesh
Hierarchical Mesh Network

Use overlayed mesh to support longer connections

Reduced fanout and reduced resistance
EPLD Block Diagram

Primary inputs

Macrocell
Architecture Reuse

- Silicon System Platform
  - Flexible architecture for hardware and software
  - Specific (programmable) components
  - Network architecture
  - Software modules
  - Rules and guidelines for design of HW and SW
- Has been successful in PC’s
  - Dominance of a few players who specify and control architecture
- Application-domain specific (difference in constraints)
  - Speed (compute power)
  - Dissipation
  - Costs
  - Real / non-real time data
Platform-Based Design

- A platform is a restriction on the space of possible implementation choices, providing a well-defined abstraction of the underlying technology for the application developer.
- New platforms will be defined at the architecture-micro-architecture boundary.
- They will be component-based, and will provide a range of choices from structured-custom to fully programmable implementations.
- Key to such approaches is the representation of communication in the platform model.