Unit-5

Implementation Strategies
ASIC vs Standard IC

- **Standard ICs** – ICs sold as Standard Parts
  - SSI/LSI/ MSI IC such as MUX, Encoder, Memory Chips, or Microprocessor IC

- **Application Specific Integrated Circuits (ASIC)**,
  - An IC Customized to a Particular System or Application – Custom ICs
    - Reduced Cost and Improved Reliability
    - A Chip for Toy Bear, Auto-Mobile Control Chip
Types of ASICs

- The classification of ASICs is shown below:

  - **ASICs**
    - **Full Custom**
      - Standard cell Based
    - **Semi-Custom**
      - Gate Array Based
    - **Programmable**
      - PLDs
      - FPGAs
        - Channeled
        - Channelless
        - Structured Gate
Types of ASICs

- Full-Custom ICs
  - **Wafer**: A circular piece of pure silicon
  - **Wafer Lot**: 5 ~ 30 wafers, each containing hundreds of chips (dies) depending upon size of the die
  - **Die**: A rectangular piece of silicon that contains one IC design
  - **Mask Layers**: Each IC is manufactured with successive mask layers (10 – 15 layers)
    - First half-dozen or so layers define transistors
    - Other half-dozen or so define Interconnect
• Full-Custom ASICs: Possibly all logic cells and all mask layers customized

• Semi-Custom ASICs: All logic cells are pre-designed and some (possibly all) mask layers customized
Full-Custom ASICs

Include some customized logic cells

❖ Have all their mask layers customized
❖ Full-custom ASIC design makes sense only
  ✔ When no suitable existing libraries exist or
  ✔ ASIC technology is new or/and so special that no cell library exits.

❖ Offer highest performance and lowest cost (smallest die size) but at the expense of increased design time, complexity, higher design cost and higher risk.

❖ Some Examples: High-Voltage Automobile Control Chips, Ana-Digi Communication Chips, Sensors and Actuators
Semi-Custom ASICs

- **Standard-Cell based ASICs** (CBIC- “sea-bick”)
  - Use logic blocks from standard cell libraries, other mega-cells, full-custom blocks, system-level macros (SLMs), functional standard blocks (FSBs), cores etc.
  - Get all mask layers customized- transistors and interconnect
  - Manufacturing lead time is around 8 weeks
  - Less efficient in size and performance but lower in design cost
Semi-Custom ASICs

- Standard-Cell based ASICs (CBIC- “sea-bick”)
Gate Array based ASICs

A gate array, masked gate array, MGA, or prediffused array uses macros (books) to reduce turnaround time and comprises a base array made from a base cell or primitive cell. There are three types:

- Channeled gate arrays
- Channelless gate arrays
- Structured gate arrays

- A type of ASIC chip that is partially finished with rows of the transistors and resistors built in but unconnected.
- The chip is completed by designing and adhering the top metal layers that provide the interconnecting pathways.
- These final masking stages are less costly than designing a full custom chip
Gate Array Based ASIC

A channeled gate array

- Only the interconnect is customized
- The interconnect uses predefined spaces between rows of base cells
- Manufacturing lead time is between two days and two weeks
Gate Array based ASICs

- A channelless gate array (channel-free gate array, sea-of-gates array, or SOG array)
  - Only some (the top few) mask layers are customized — the interconnect
  - Manufacturing lead time is between two days and two weeks.

- An embedded gate array or structured gate array (masterslice or masterimage)
  - Only the interconnect is customized
  - Custom blocks (the same for each design) can be embedded
  - Manufacturing lead time is between two days and two weeks.
The purpose of a PLD device is to permit elaborate digital logic designs to be implemented by the user in a single device.

Can be erased electrically and reprogrammed with a new design, making them very well suited for academic and prototyping.

<table>
<thead>
<tr>
<th>Device</th>
<th>AND-array</th>
<th>OR-array</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROM</td>
<td>Fixed</td>
<td>Programmable</td>
</tr>
<tr>
<td>PLA</td>
<td>Programmable</td>
<td>Programmable</td>
</tr>
<tr>
<td>PAL</td>
<td>Programmable</td>
<td>Fixed</td>
</tr>
<tr>
<td>GAL</td>
<td>Programmable</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
Programmable logic devices

AND gate before programming

AND gate after programming
Programmable logic devices

Example Schematic of a PAL

\[ f_1 = x_1 x_2 x_3 + x_1' x_2 x_3 \]

\[ f_2 = x_1' x_2' + x_1 x_2 x_3 \]
Programmable Logic Devices

Programmable Logic Array
Complex Programmable Logic Structure
FPGA Structure

- FPGA is an integrated circuit that contains many (64 to over 10,000) identical logic cells that can be viewed as standard components.
- Individual cells are interconnected by a matrix of wires and programmable switches.
- Array of logic cells and interconnect form a fabric of basic building blocks for logic circuits.

Difference between ASIC and FPGA:
- An ASIC is a unique type of integrated circuit meant for a specific application while an FPGA is a reprogrammable integrated circuit.
- An ASIC can no longer be altered once created while an FPGA can.
FPGA Structure
Different Categorizations of FPGAs

- Based on Functional Unit/Logic Cell Structure
  - Transistor Pairs
  - Basic Logic Gates: NAND/NOR
  - MUX
  - Look-up Tables (LUT)
  - Wide-Fan-In AND-OR Gates

- Programming Technology
  - Anti-Fuse Technology
  - SRAM Technology
  - EPROM Technology

- Gate Density
- Chip Architecture (Routing Style)
Different Types of Logic Cells

FPGA Architecture: Functional Units

- Functional units
  - RAM blocks (Xilinx): implement function truth table
  - Multiplexers (Actel): build Boolean functions using muxes
  - Logic gates, flip-flops: Such as carry chains. Used for high-performance computations
Different Types of Logic Cells – Cont’d

- Xilinx XC4000 CLB Structure

(a) Circuit for a two-input LUT

(b) $f_1 = \overline{x_1}x_2 + x_1x_2$

(c) Storage cell contents in the LUT

$\begin{array}{c|cc}
 x_1 & x_2 & f_1 \\
 0 & 0 & 1 \\
 0 & 1 & 0 \\
 1 & 0 & 0 \\
 1 & 1 & 1 \\
\end{array}$
Different Types of Logic Cells – Cont’d

- Actel Act Logic Module Structure
  - Use Antifuse Programming Tech.
  - Based on Channeled GA Architecture
  - Logic Cell is MUX which can be configured as multi-input logic gates

The Actel ACT 2 and ACT 3 Logic Modules. (a) The C-Module for combinational logic. (b) The ACT 2 S-Module. (c) The ACT 3 S-Module. (d) The equivalent circuit (without buffering) of the SE (sequential element). (e) The sequential element configured as a positive-edge–triggered D flip-flop. (Source: Actel.)
Different Types of Logic Cells – Cont’d

- Altera Flex / Max Logic Element Structure
  - Flex 8k/10k Devices – SRAM Based LUTs, Logic Elements (LEs) are similar to those used in XC5200 FPGA

The Altera MAX architecture. (a) Organization of logic and interconnect. (b) A MAX family LAB (Logic Array Block). (c) A MAX family macrocell. The macrocell details vary between the MAX families—the functions shown here are closest to those of the MAX 9000 family.
Different Types of Logic Cells – Cont’d

To SUMMARIZE, FPGAs from various vendors differ in their

- **Architecture** (Row Based or Matrix Based Routing Mechanism)
- **Gate Density** (Cap. In Equiv. 2-Input NAND Gates)
- **Basic Cell Structure**
- **Programming Technology**

<table>
<thead>
<tr>
<th>Vendor/Product</th>
<th>Architecture</th>
<th>Capacity</th>
<th>Basic Cell</th>
<th>Programming Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actel</td>
<td>Gate Array</td>
<td>2-8 k</td>
<td>MUX</td>
<td>Antifuse</td>
</tr>
<tr>
<td>QuickLogic</td>
<td>Matrix</td>
<td>1.2-1.8 k</td>
<td>MUX</td>
<td>Antifuse</td>
</tr>
<tr>
<td>Xilinx</td>
<td>Matrix</td>
<td>2-10 k</td>
<td>RAM Block</td>
<td>SRAM</td>
</tr>
<tr>
<td>Altera</td>
<td>Extended PLA</td>
<td>1-5 k</td>
<td>PLA</td>
<td>EPROM</td>
</tr>
<tr>
<td>Concurrent</td>
<td>Matrix</td>
<td>3-5 k</td>
<td>XOR, AND</td>
<td>SRAM</td>
</tr>
<tr>
<td>Plessy</td>
<td>SOG</td>
<td>2-40 k</td>
<td>NAND</td>
<td>SRAM</td>
</tr>
</tbody>
</table>
Programming Technologies

- Three Programming Technologies
  - The Antifuse Technology
  - Static RAM Technology
  - EPROM and EEPROM Technology
The Antifuse Technology

- Invented at Stanford and developed by Actel
- Opposite to regular fuse Technology
- Normally an open circuit until a programming current (about 5 mA) is forced through it

Two Types:

- Actel’s PLICE [Programmable Low-Impedance Circuit Element] - A High-Resistance Poly-Diffusion Antifuse
- QuickLogic’s Low-Resistance metal-metal antifuse [ViaLink] technology
  - Direct metal-2-metal connections
  - Higher programming currents reduce antifuse resistance

Disadvantages:

- Unwanted Long Delay
- OTP Technology
Static RAM Technology

- SRAM cells are used for
  - As Look-Up Tables (LUT) to implement logic (as Truth Tables)
  - As embedded RAM blocks (for buffer storage etc.)
  - As control to routing and configuration switches

- Advantages
  - Allows In-System Programming (ISP)
  - Suitable for Reconfigurable HW

- Disadvantages
  - Volatile – needs power all the time / use PROM to download configuration data
Programming Technologies – Cont’d

- EPROM and EEPROM Technology -

  - EPROM Cell is almost as small as Antifuse
  - Floating-Gate Avalanche MOS (FAMOS) Tech.

  - Under normal voltage, transistor is on
  - With Programming Voltage applied, we can turn it off (configuration) to implement our logic
  - Exposure to UV lamp (one hour) we can erase the programming
  - Use EEPROM for quick reconfiguration, also, ISP possible
### Programming Technologies – Cont’d

- **Summary Sheet**

<table>
<thead>
<tr>
<th>Programmable ASIC technologies</th>
<th>Xilinx LCA&lt;sup&gt;1&lt;/sup&gt;</th>
<th>Altera EPLD</th>
<th>Xilinx EPLD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming technology</strong></td>
<td>Poly–diffusion antifuse, PLICE</td>
<td>Erasable SRAM ISP</td>
<td>UV-erasable EPROM (MAX 5k)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EEPROM (MAX 7/9k)</td>
</tr>
<tr>
<td><strong>Size of programming element</strong></td>
<td>Small but requires contacts to metal</td>
<td>Two inverters plus pass and switch devices, Largest.</td>
<td>One n-channel EPROM device.</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Special: CMOS plus three extra masks.</td>
<td>Standard CMOS</td>
<td>Standard EPROM and EEPROM</td>
</tr>
<tr>
<td><strong>Programming method</strong></td>
<td>Special hardware</td>
<td>PC card, PIROM, or serial port</td>
<td>ISP (MAX 9k) or EPROM programmer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QuickLogic</th>
<th>Crosspoint</th>
<th>Atmel</th>
<th>Altera FLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming technology</strong></td>
<td>Metal–metal antifuse, ViaLink</td>
<td>Metal–polysilicon antifuse</td>
<td>Erasable SRAM. ISP.</td>
</tr>
<tr>
<td><strong>Size of programming element</strong></td>
<td>Smallest</td>
<td>Small</td>
<td>Two inverters plus pass and switch devices. Largest.</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Special, CMOS plus ViaLink</td>
<td>Special, CMOS plus antifuse</td>
<td>Standard CMOS</td>
</tr>
<tr>
<td><strong>Programming method</strong></td>
<td>Special hardware</td>
<td>Special hardware</td>
<td>PC card, PROM, or serial port</td>
</tr>
</tbody>
</table>

<sup>1</sup>Lucent (formerly AT&T) FPGAs have almost identical properties to the Xilinx LCA family.
Chip Architecture or Routing Style

Programmable Switch Elements

- Used in connecting:
  - The I/O of functional units to the wires
  - A horizontal wire to a vertical wire
  - Two wire segments to form a longer wire segment

Programmable Switch Elements: Implementation

- SRAM connected to the gate of a transistor (Xilinx)
- Fuse / Anti Fuse (Actel)
Chip Architecture or Routing Style – Cont’d

Routing Channels

- Note: fixed channel widths (tracks)
- Should “predict” all possible connectivity requirements when designing the FPGA chip
- Channel -> track -> segment

- Segment length?
  - Long: carry the signal longer, less “concatenation” switches, but might waste track
  - Short: local connections, slow for longer connections

Switch Boxes

- Ideally, provide switches for all possible connections

- Trade-off:
  - Too many switches:
    - Large area
    - Complex to program
  - Too few switches:
    - Cannot route signals

One possible solution

Xilinx 4000
Chip Architecture or Routing Style – Cont’d

- Trade-off between Longer and Shorter Tracks Explained Through Example

![Diagram showing chip architecture or routing style](image)
ASIC Design Process

S-1 Design Entry: Schematic entry or HDL description

S-2: Logic Synthesis: Using Verilog HDL or VHDL and Synthesis tool, produce a netlist - logic cells and their interconnect detail

S-3 System Partitioning: Divide a large system into ASIC sized pieces

S-4 Pre-Layout Simulation: Check design functionality

S-5 Floorplanning: Arrange netlist blocks on the chip

S-6 Placement: Fix cell locations in a block

S-7 Routing: Make the cell and block interconnections

S-8 Extraction: Measure the interconnect R/C cost

S-9 Post-Layout Simulation
ASIC Design Process – Cont’d

- Altera FPGA Design Flow – A Self-Contained System that does all from Design Entry, Simulation, Synthesis, and Programming of Altera Devices

Design Entry → Synthesis → MAX+PLUS II Place & Route → Pre- & Post-Layoff Simulation

- EDIF/VHDL
- EDIF
- SDF
- Timing Annotated Netlists in Verilog HDL, VHDL, EDIF, SDF, VITAL
ASIC Design Process – Cont’d

- Xilinx FPGA Design Flow – Allows Third-Party Design Entry SW, Accepts their generated netlist file as an input

  - Use Pin2xnf and wir2xnf SW to convert the netlist file to .XNF

  - xnfmap and xnfmerge programs convert .xnf files to create a unified netlist file (Nand/Nor Gates) .MAP file are generated

  - map2lca program does fitters job, produces un-routed .LCA file

  - apr or ppr SW does the routing job, post-layout netlist generated

  - makebits SW generates BIT files