Digital Integrated Circuits

Lecture 9

Jaeyong Chung
Robust Systems Laboratory
Incheon National University
FPGA vs. ASIC

- FPGA (A programmable Logic Device)
  - Faster time-to-market
  - No upfront non-recurring expenses (NRE)
  - Easier to design
  - Field reprogrammability

- ASIC
  - Higher performance/Lower power
  - Lower unit costs
  - Full custom capability
  - Smaller form factor
FPGA vs. ASIC

Device Only Cost (ASIC includes NRE)

<table>
<thead>
<tr>
<th>Unit's</th>
<th>FPGA Cost</th>
<th>ASIC Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$16,000</td>
<td>$350,150</td>
</tr>
<tr>
<td>10</td>
<td>$32,000</td>
<td>$350,300</td>
</tr>
<tr>
<td>50</td>
<td>$160,000</td>
<td>$351,500</td>
</tr>
<tr>
<td>100</td>
<td>$320,000</td>
<td>$353,000</td>
</tr>
<tr>
<td>150</td>
<td>$480,000</td>
<td>$354,500</td>
</tr>
</tbody>
</table>

FPGA $ 3,200 Each
FPGA NRE $ -
ASIC $ 30 Each
ASIC NRE $ 350,000

FPGA/ASIC Cost vs Units (250K Gates)

Total Unit Cost (US$)

# of Units
FPGA vs. CPU+Software

- ASIC is chips customized for specific tasks
  - CPU is general-purpose chips that run software
  - FPGA is generic chips that can be programmed

- FPGA
  - Good for parallelizable tasks
  - Provides true concurrency

- CPU + Software
  - Provide high performance for a serial task
  - Emulates concurrency
Chapter 3 – Programmable Logic Devices

Many Ways to Implement Combinational Circuit
- NAND, NOR
- Multiplexers
- Decoders
- ROMs (LUTs)

Programmable Logic Device
- Puts Different Building Blocks in Chip
- Provides Mechanism to Reconfigure (Program) It
<table>
<thead>
<tr>
<th></th>
<th>SPLD</th>
<th>CPLD</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>Low</td>
<td>Low to Medium</td>
<td>Medium to High</td>
</tr>
<tr>
<td></td>
<td>Few hundred gates</td>
<td>500 to 12,000 gates</td>
<td>3,000 to 5,000,000 gates</td>
</tr>
<tr>
<td>Timing</td>
<td>Predictable</td>
<td>Predictable</td>
<td>Unpredictable</td>
</tr>
<tr>
<td>Cost</td>
<td>Low</td>
<td>Low to Medium</td>
<td>Medium to High</td>
</tr>
<tr>
<td>Major Vendors</td>
<td>Lattice Semiconductor</td>
<td>Xilinx</td>
<td>Xilinx</td>
</tr>
<tr>
<td></td>
<td>Cypress</td>
<td>Altera</td>
<td>Altera</td>
</tr>
<tr>
<td></td>
<td>AMD</td>
<td></td>
<td>Actel</td>
</tr>
<tr>
<td>Example Devices</td>
<td>Lattice Semiconductor</td>
<td>Xilinx</td>
<td>Xilinx</td>
</tr>
<tr>
<td></td>
<td>GAL16LV8</td>
<td>CoolRunner</td>
<td>Virtex</td>
</tr>
<tr>
<td></td>
<td>GAL22V10</td>
<td>XC9500</td>
<td>Spartan</td>
</tr>
<tr>
<td></td>
<td>Cypress</td>
<td></td>
<td>Altera</td>
</tr>
<tr>
<td></td>
<td>PALCE16V8</td>
<td>Altera</td>
<td>Stratix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAX</td>
<td>Actel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accelerator</td>
</tr>
</tbody>
</table>
8-WORD x 4-BIT ROM

(a) block diagram

(b) truth table for ROM

\[
\begin{array}{cccc|cccc}
A & B & C & F_0 & F_1 & F_2 & F_3 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 \\
\end{array}
\]

typical data stored in ROM

(2^3 words of 4 bits each)
### ROM WITH N-INPUT AND M-OUTPUTS

![Diagram of ROM with input and output lines]

<table>
<thead>
<tr>
<th>$n$ Input Variables</th>
<th>$m$ Output Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>00⋯00</td>
<td>100⋯110</td>
</tr>
<tr>
<td>00⋯01</td>
<td>010⋯111</td>
</tr>
<tr>
<td>00⋯10</td>
<td>101⋯101</td>
</tr>
<tr>
<td>00⋯11</td>
<td>110⋯010</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>11⋯00</td>
<td>001⋯011</td>
</tr>
<tr>
<td>11⋯01</td>
<td>110⋯110</td>
</tr>
<tr>
<td>11⋯10</td>
<td>011⋯000</td>
</tr>
<tr>
<td>11⋯11</td>
<td>111⋯101</td>
</tr>
</tbody>
</table>

- $2^n$ words in ROM
- $m$ bits each word

- Typical data array stored in ROM
- $(2^n$ words of $m$ bits each)
BASIC ROM STRUCTURE

- Decoder
- Memory Array: $2^n$ words $\times m$ bits
- ROM
- $n$ Input Lines
- $m$ Output Lines
8-WORD BY 4-BIT ROM
EQUIVALENT OR-GATE FOR $F_0$

\[ F_0 = \sum m(0, 1, 4, 6) = A'B' + AC' \]
\[ F_1 = \sum m(2, 3, 4, 6, 7) = B + AC' \]
\[ F_2 = \sum m(0, 1, 2, 6) = A'B' + BC' \]
\[ F_3 = \sum m(2, 3, 5, 6, 7) = AC + B \]
TYPES OF ROMS

- ROM – Read Only Memory
- PROM – User Programmable ROM
- EPROM – Erasable PROM
- EEPROM – Electrically Erasable PROM
- Flash – Particular Type of EEPROM
  - Uses Special Type of Charge Storage
  - Usually Has Built-In Programming and Erasure Capability
    - Doesn’t Require Separate Programmer Hardware
PROGRAMMABLE LOGIC ARRAY

\[ F_0 = \sum m(0, 1, 4, 6) = A'B' + AC' \]
\[ F_1 = \sum m(2, 3, 4, 6, 7) = B + AC' \]
\[ F_2 = \sum m(0, 1, 2, 6) = A'B' + BC' \]
\[ F_3 = \sum m(2, 3, 5, 6, 7) = AC + B \]
PROGRAMMABLE ARRAY LOGIC (PAL)

- Programmable Array Logic (PAL)
  - OR Connections Fixed
  - Cannot Share Products
  - Less Flexible Than PLA

- Example
  - PAL for $I_1I_2' + I_1'I_2$
GALs

- Erasible/Reprogrammable PALs Called
  - GALs (Generic Array Logic)
    - EPROM AND EEPROM type
      - Require Hardware Programmer
    - Flash type
      - In-Circuit Programmable
    - Often PALs Have GAL Equivalents
      - E.g., PALCE22V10 Equivalent to GAL22V10
  - Most Also Have Macroblocks
    - Contain MUXes and Additional Programmability
COMPLEX PLDs

- Complex PLDs (CPLDs)
  - Equivalent to Several PLDs on Same Chip
  - Uses Programmable Interconnect Matrix
    - Built From Crossbar Switch

- $N \times M$ Crossbar Switch
  - Allows $N$ Input Lines to be Connected to Any $M$ Output Lines
  - Expensive to Build
  - Provides Predictable Interconnect Delay
Field-Programmable Gate Arrays (FPGAs)

- ICs Containing Array of Logic Blocks with Programmable Interconnections
- Larger than PALs, PLAs, CPLDs
- Invented around 1985
  - Xilinx XC2000
- Xilinx chips - SRAM programmable
- In-Circuit Reconfigurable
FPGAs vs. MPGAs

- Mask Programmable Gate Arrays (MPGAs)
  - Needs Custom Mask
  - Programmable Only in Factory
  - Larger Turn Around Time (Time to Market)
  - Low Flexibility (Error Necessitates New Mask)

- Field Programmable Gate Arrays (FPGAs)
  - Standard Off the Shelf Products
  - Needs No Custom Mask
  - User Can Reprogram Many Times
  - Negligible Turn Around Time (Fast Time to Market)
  - High Flexibility (Easy to Change Design and Fix Errors)
  - Cheaper Than MPGAs at Low Volumes
DISADVANTAGES OF FPGAs

- Less Dense than MPGAs
  - Extra Hardware Need to Support Field Programmability
- Slower than MPGAs
  - Programmable Interconnect Results in Higher Resistance and Capacitance (Longer RC Delay)
- Unpredictable Delays
- More Power
POPULAR XILINX FPGAs

- Spartan/XL - 5K to 40K
- Spartan 3 - 50K to 5M
- Spartan-3E - 100K to 1.6M
- Spartan-3L - 1M to 4M
- Spartan-II - 15K to 200K
- Spartan-IIIE - 50K to 600K
- Virtex - 57,906 to 1,124,022
- Virtex-4 FX - 12,312 to 142,128 LCBs
- Virtex-4 LX - 13,824 to 200,448 LCBs
- Virtex-4 SX - 23,040 to 55,296 LCBs
- Virtex-E - 71,693 to 4,074,387
- Virtex-E Extended Memory - 129,600 to 254,016
- Virtex-II - 40K to 8M
- Virtex-II Pro - 3,168 to 99,216 LCBs
- Virtex-II Pro/X - 22,032 to 74,448 LCBs
POPULAR ALTERA FPGAs

- ACEX 1K - 56K to 257K
- APEX 20K - 113K to 2.5M
- APEX II - 1.9M to 5.25M
- Cyclone - 2,910 to 20,060 LEs
- Cyclone II - 4,608 to 68,416 LEs
- FLEX 10K - 10K to 50K
- FLEX 6000 - 5K - 24K
- Mercury - 120K to 350K
- Stratix - 10,570 to 79,040 LEs
- Stratix GX - 10,570 to 41,250 LEs
- Stratix II - 15,600 to 179,400 LEs
- Stratix II GX - 33,880 to 132,540 LEs
POPULAR ACTEL FPGAs

- Axcelerator - 125K To 2M
- eX - 3K to 12K
- SX/SXA - 8K to 72K
- ProASIC3 - 30K to 3M
- ProASICplus - 75K to 1M
- ProASIC - 100 to 450K
- MX - 3K to 54K
Approximate Market Share of FPGAs (2005)

- Xilinx – 51.5%
- Altera – 32.3%
- Lattice Semi – 7.8%
- Actel – 5.4%
- QuickLogic – 1.4%
- Atmel – 0.5%
FOUR ARCHITECTURES FOR FPGAs

Matrix-based

Row-based

PLD-based

Sea-of-Gates
TYPICAL MATRIX BASED FPGA LAYOUT
SIMPLE CONFIGURABLE LOGIC BLOCK

X Function Generator
LUT4

Y Function Generator
LUT4

Chung
EPC6055
28
## MUX-BASED LOGIC BLOCK

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The table above shows the truth table for the multiplexer logic block. The output $F$ is determined by the inputs $A$, $B$, and $C$. The diagram on the right shows a 4-to-1 MUX with inputs $A$, $B$, $C$, and $C'$, and outputs $D00$, $D01$, $D10$, and $D11$. The output $F_1$ is determined by the select lines $S0$ and $S1$.
XILINX VIRTEX CLB

- **G Inputs**: 4 inputs
- **Additional Input F5IN**: 2 inputs
- **F Inputs**: 4 inputs
- **Cout**: Carry output
- **Cin**: Carry input
- **Mux**: Multiplexer
- **Control Logic**: Logic control circuit
- **D Q**: Data output
- **Y Q**: Output
- **Feedthrough Output**: 2 outputs
- **5-Variable Function Output**: 2 outputs

The diagram illustrates the connections and logic flow within a Xilinx Virtex CLB (Configurable Logic Block) module.
GENERAL ROUTING MATRIX
SWITCH MATRIX
DIRECT INTERCONNECTS
GLOBAL LINES

Logic Block

Logic Block

Logic Block

Logic Block

Tri-State Lines
EMBEDDED RAM

- Distributed RAM from LUTs
  - Create RAM from LUTs;
  - Done in Early FPGAs up to XC4000
    - Had No Dedicated Memory

- BlockRAM
  - Starting with VIRTEX
  - Blocks of Dedicated Memory Embedded in FPGAs
  - Variable Width RAM
## VARIABLE WIDTH RAM ASPECT RATIOS

<table>
<thead>
<tr>
<th>Width</th>
<th>Depth</th>
<th>Addr Bus</th>
<th>Data Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32 K</td>
<td>15 bits</td>
<td>1 bit</td>
</tr>
<tr>
<td>2</td>
<td>16 K</td>
<td>14 bits</td>
<td>2 bits</td>
</tr>
<tr>
<td>4</td>
<td>8 K</td>
<td>13 bits</td>
<td>4 bits</td>
</tr>
<tr>
<td>8</td>
<td>4 K</td>
<td>12 bits</td>
<td>8 bits</td>
</tr>
<tr>
<td>16</td>
<td>2K</td>
<td>11 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>
## COMPARE FPGAs

<table>
<thead>
<tr>
<th>Special Components</th>
<th>Xilinx Virtex-4 FX, Virtex-II Pro</th>
<th>Xilinx Spartan-3E, 3/3L</th>
<th>Altera Stratix II</th>
<th>Altera APEX, APEX II</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP Support</td>
<td>18 X 18 Multipliers</td>
<td>18 X 18 Multipliers</td>
<td>DSP Blocks Embedded Multipliers</td>
<td>Embedded Multipliers</td>
</tr>
<tr>
<td>Embedded Processor</td>
<td>IBM PowerPC 400MHz</td>
<td>MicroBlaze, PicoBlaze</td>
<td>Nios2</td>
<td>ARM, MIPS, Nios</td>
</tr>
<tr>
<td>Comm. Support</td>
<td>Multi-GigaBit Transceivers</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Block RAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Addressable Mem. (CAM)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Spartan-3 FPGA attributes

### Table 1: Summary of Spartan-3 FPGA Attributes

<table>
<thead>
<tr>
<th>Device</th>
<th>System Gates</th>
<th>Equivalent Logic Cells&lt;sup&gt;(1)&lt;/sup&gt;</th>
<th>CLB Array (One CLB = Four Slices)</th>
<th>Distributed RAM Bits (K=1024)</th>
<th>Block RAM Bits (K=1024)</th>
<th>Dedicated Multipliers</th>
<th>DCMs</th>
<th>Max. User I/O</th>
<th>Maximum Differential I/O Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S50&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>50K</td>
<td>1,728</td>
<td>16, 12</td>
<td>12K</td>
<td>72K</td>
<td>4</td>
<td>2</td>
<td>124</td>
<td>56</td>
</tr>
<tr>
<td>XC3S200&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>200K</td>
<td>4,320</td>
<td>24, 20</td>
<td>30K</td>
<td>216K</td>
<td>12</td>
<td>4</td>
<td>173</td>
<td>76</td>
</tr>
<tr>
<td>XC3S400&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>400K</td>
<td>8,064</td>
<td>32, 28</td>
<td>56K</td>
<td>288K</td>
<td>16</td>
<td>4</td>
<td>264</td>
<td>116</td>
</tr>
<tr>
<td>XC3S1000&lt;sup&gt;(2)&lt;/sup&gt;</td>
<td>1M</td>
<td>17,280</td>
<td>48, 40</td>
<td>120K</td>
<td>432K</td>
<td>24</td>
<td>4</td>
<td>391</td>
<td>175</td>
</tr>
<tr>
<td>XC3S1500</td>
<td>1.5M</td>
<td>29,952</td>
<td>64, 52</td>
<td>208K</td>
<td>576K</td>
<td>32</td>
<td>4</td>
<td>487</td>
<td>221</td>
</tr>
<tr>
<td>XC3S2000</td>
<td>2M</td>
<td>46,080</td>
<td>80, 64</td>
<td>320K</td>
<td>720K</td>
<td>40</td>
<td>4</td>
<td>565</td>
<td>270</td>
</tr>
<tr>
<td>XC3S4000</td>
<td>4M</td>
<td>62,208</td>
<td>96, 72</td>
<td>432K</td>
<td>1,728K</td>
<td>96</td>
<td>4</td>
<td>633</td>
<td>300</td>
</tr>
<tr>
<td>XC3S5000</td>
<td>5M</td>
<td>74,880</td>
<td>104, 80</td>
<td>520K</td>
<td>1,872K</td>
<td>104</td>
<td>4</td>
<td>633</td>
<td>300</td>
</tr>
</tbody>
</table>

<sup>(1)</sup> One CLB = Four Slices

---

Spartan-3 FPGA Family Data Sheet
Spartan-3 Family Architecture

Notes:
1. The two additional block RAM columns of the XC3S4000 and XC3S5000 devices are shown with dashed lines. The XC3S50 has only the block RAM column on the far left.

Figure 1: Spartan-3 Family Architecture

Spartan-3 FPGA Family Data Sheet
CLB of Spartan-3

- All four slices have following elements in common: two logic function generators, two storage elements, wide-function multiplexers, carry logic, and arithmetic and ROM functions.
- Besides these, the left-hand pair supports two additional functions: storing data using Distributed RAM and shifting data with 16-bit registers.
Slice of Spartan-3

Spartan-3 FPGA Family Data Sheet
Design Report of ICE

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice Flip Flops</td>
<td>412</td>
<td>40,960</td>
<td>1%</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>855</td>
<td>40,960</td>
<td>2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logic Distribution</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of occupied Slices</td>
<td>484</td>
<td>20,480</td>
<td>2%</td>
</tr>
<tr>
<td>Number of Slices containing only related logic</td>
<td>484</td>
<td>484</td>
<td>100%</td>
</tr>
<tr>
<td>Number of Slices containing unrelated logic</td>
<td>0</td>
<td>484</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Number of 4 input LUTs</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number used as logic</td>
<td>144</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as a route-thru</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number used as Shift registers</td>
<td>711</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>20</td>
<td>489</td>
<td>4%</td>
</tr>
<tr>
<td>Number of MULT18X18s</td>
<td>7</td>
<td>40</td>
<td>17%</td>
</tr>
<tr>
<td>Number of BUFGMUXs</td>
<td>1</td>
<td>8</td>
<td>12%</td>
</tr>
</tbody>
</table>
Placement of ICE
Routing of ICE
APPLICATIONS OF FPGAs

- **Rapid Prototyping**
  - 5 Million Gate FPGAs Allows Large Applications
  - Can Also Have Multiple FPGAs on Boards
- **Final Product in Medium Speed Systems**
  - Can Operate in 150-200 MHz Range
- **Reconfigurable Circuits and Systems**
  - Reconfigurable “Soft” Hardware Systems
- **Glue Logic**
  - Handle Interface Between Chips
  - Can Be Updated for Changes in Protocol
- **Hardware Accelerators/Coprocessors**
  - Implement Key Kernels to Speed Up Application
DESIGN FLOW FOR FPGAs

- Create Model of Design in HDL
- Simulate and Debug Design
- Synthesize Design Targeting Desired Device
- Run Mapping/Partitioning Program
  - Maps to CLBs
- Run Place and Route Program
  - Places CLBs and Routes Interconnect
- Run Program to Generate Configuration Bits
- Download Configuration Bits into FPGA
INITIALIZING SRAM-BASED FPGA

FPGA

Address

Data

EPROM
(contains configuration data)