**Outline**

- What are FPGAs?
- Why use FPGAs (a short history lesson).
- FPGA variations
- Internal logic blocks.
- Break/Announcements
- Designing with FPGAs.
- Specifics of Xilinx 4000 series.

---

**Why FPGAs?**

- By the early 1980’s most of the logic circuits in typical systems were absorbed by a handful of standard large scale integrated circuits (LSI).
  - Microprocessors, bus/IO controllers, system timers, ...
- Every system still had the need for random “glue logic” to help connect the large ICs:
  - generating global control signals (for resets etc.)
  - data formatting (serial to parallel, multiple etc.)
- Systems had a few LSI components and lots of small low density SSI (small scale IC) and MSI (medium scale IC) components.

---

**FPGA Overview**

- Basic idea: two-dimensional array of logic blocks and flip-flops with a means for the user to configure:
  1. the interconnection between the logic blocks,
  2. the function of each block.

---

**Why FPGAs?**

- Custom ICs where sometimes designed to replace the large amount of glue logic:
  - reduced system complexity and manufacturing cost, improved performance.
  - However, custom ICs are relatively very expensive to develop, and delay introduction of product to market (time to market) because of increased design time.
- Note: need to worry about two kinds of costs:
  1. cost of development, sometimes called non-recurring engineering (NRE)
  2. cost of manufacture
  - A tradeoff usually exists between NRE cost and manufacturing costs total.

---

**Why FPGAs?**

- Therefore the custom IC approach was only viable for products with very high volume (where NRE could be amortized), and which were not TTM sensitive.
- FPGAs were introduced as an alternative to custom ICs for implementing glue logic:
  - improved density relative to discrete SSI/MSI components (within around 10x of custom ICs)
  - with the aid of computer aided design (CAD) tools circuits could be implemented in a short amount of time (no physical layout processes, no mask making, no IC manufacturing)
  - lowers NREs
  - shortens TTM
- Because of Moore’s law the density (gates/area) of FPGAs continued to grow through the 80’s and 90’s to the point where major data processing functions can be implemented on a single FPGA.
Why FPGAs?

- FPGAs continue to compete with custom ICs for special processing functions (and glue logic) but now also compete with microprocessors in dedicated and embedded applications.
- Performance advantage over microprocessors because circuits can be customized for the task at hand. Microprocessors must provide special functions in software (many cycles).

Families of FPGA's differ in:

- physical means of implementing user programmability;
- arrangement of interconnection wires, and
- the basic functionality of the logic blocks.

Most significant difference is in the method for providing flexible blocks and connections:

- Anti-fuse based (ex: Actel)
- Latch-based (Xilinx, Altera, ...)
- Look up table (LUT)

Summary:

- Performance advantage over microprocessors because circuits can be reprogrammed.
- Result is a general purpose "logic gate".
- n-LUT can implement any function of n inputs.

ASIC = custom IC, MICRO = microprocessor

User Programmability

- Latches are used to:
  1. make or break cross-point connections in the interconnect
  2. define the function of the logic blocks
  3. set user options:
      - within the logic blocks
      - in the inputs/output blocks
      - global reset/clock
- Configuration bit stream can be loaded under user control:
  - All latches are strung together in a shift chain.

Idealized FPGA Logic Block

- 4-input look up table (LUT)
- Implants combinational logic functions
- Register
  - optionally stores output of LUT

4-LUT Implementation

- n-bit LUT is implemented as a 2^n x 1 memory:
  - inputs choose one of 2^n-memory locations.
  - memory locations (latches) are normally loaded with values from user's configuration bit stream.
  - Inputs to mux control are the CLB inputs.
- Result is a general purpose "logic gate".
- n-LUT can implement any function of n inputs.

LUT as general logic gate

- An n-lut as a direct implementation of a function truth table.
- Each latch location holds the value of the function corresponding to one input combination.

Example: 2-LUT

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
</tr>
<tr>
<td>01</td>
<td>0011</td>
</tr>
<tr>
<td>10</td>
<td>0101</td>
</tr>
<tr>
<td>11</td>
<td>0110</td>
</tr>
</tbody>
</table>

Implements any function of 2 inputs.

How many of these are there? 4
How many functions of n inputs?

ASIC = custom IC, MICRO = microprocessor

FPGA Variations

- Families of FPGA's differ in:
  - physical means of implementing user programmability;
  - arrangement of interconnection wires, and
  - the basic functionality of the logic blocks.
- Most significant difference is in the method for providing flexible blocks and connections:

  - Anti-fuse based (ex: Actel)
  - Latch-based (Xilinx, Altera, ...)
  - Look up table (LUT)

  - Non-volatile, relatively small
  - Fixed (non-reprogrammable)
Announcements

- Quiz results
- Administrative Q&A.
- New reading posted:
  - large section of Xilinx 4000 databook
  - All of chapter 2 in Mano

Example Partition, Placement, and Route

- Idealized FPGA structure:
  - collection of gates and flip-flops
- Check operation at full speed in real environment.
- Hardware description language (Verilog, VHDL)
- Schematic editor or

FPGA Generic Design Flow

- Design Entry:
  - Create your design files using:
    - schematic editor or
    - hardware description language (Verilog, VHDL)
- Design “implementation” on FPGA:
  - Partition, place, and route
- Load onto FPGA device (cable connects PC to development board)
- other software determines max clock frequency.
- Use Simulator to check function,

Xilinx FPGAs (4000 Series)

<table>
<thead>
<tr>
<th>Device</th>
<th>Logic Cells</th>
<th>Max D Flip-Flops</th>
<th>Max 4-input LUTs</th>
<th>Max 4-input LUTs</th>
<th>Max 4-input LUTs</th>
<th>Max 4-input LUTs</th>
<th>Total CLB</th>
<th>Flip-Flops</th>
<th>Max Use ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>xc4008e</td>
<td>770</td>
<td>7000</td>
<td>7000</td>
<td>7000</td>
<td>7000</td>
<td>7000</td>
<td>30</td>
<td>30</td>
<td>300</td>
</tr>
<tr>
<td>xc4008e</td>
<td>1519</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>50</td>
<td>50</td>
<td>500</td>
</tr>
<tr>
<td>xc4008e</td>
<td>1519</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>15000</td>
<td>50</td>
<td>50</td>
<td>500</td>
</tr>
<tr>
<td>xc4008e</td>
<td>2424</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>80</td>
<td>80</td>
<td>800</td>
</tr>
<tr>
<td>xc4008e</td>
<td>2424</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>24000</td>
<td>80</td>
<td>80</td>
<td>800</td>
</tr>
<tr>
<td>xc4008e</td>
<td>3070</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>100</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>xc4008e</td>
<td>3070</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>30000</td>
<td>100</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>xc4008e</td>
<td>4050</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>120</td>
<td>120</td>
<td>1200</td>
</tr>
<tr>
<td>xc4008e</td>
<td>4050</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>40000</td>
<td>120</td>
<td>120</td>
<td>1200</td>
</tr>
<tr>
<td>xc4008e</td>
<td>5075</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>150</td>
<td>150</td>
<td>1500</td>
</tr>
<tr>
<td>xc4008e</td>
<td>5075</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>50000</td>
<td>150</td>
<td>150</td>
<td>1500</td>
</tr>
</tbody>
</table>

Xilinx FPGAs (IOB detail)

Xilinx FPGAs (CLB detail)
Xilinx FPGAs (interconnect detail)

Xilinx 4000 series FPGAs

- How they differ from idealized array:
  - In addition to their use as general logic "gates", LUTs can alternatively be used as general purpose RAM.
  - Each 4-bit can become a 16x1-bit RAM array.
  - Special circuitry to speed up "ripple carry" in adders and counters.
  - Therefore adders in the "Xilinx Unified Library" operate much faster than adders built from gates and luts alone.
  - Many more wires, including tri-state capabilities.