FPGA Design
EECE 277

Choosing the Right Device

Dr. William H. Robinson
April 11, 2005

http://eecs.vanderbilt.edu/courses/eece277/

Topics

“There are always two choices. Two paths to take. One is easy. And its only reward is that it's easy.”

– Anonymous

• Administrative stuff
  – Grading of Exam 2
  – Laboratory Assignment #3 has been graded
  – Proposals for final project
  – In-Class Laboratory Day on Wednesday

• Design questions and specifications

• Evaluating alternatives

Final Project Selections

• Team 1 – Tim & Leighanna
  – Tic-Tac-Toe

• Team 2 – Dolores & Chris N.
  – Web-Based Server

• Team 3 – George & Daniel
  – Music Synthesizer

• Team 4 – Chris Heath & Adam
  – Hamming Code Transceiver

• Team 5 – Brett & Adrian
  – Tic-Tac-Toe

• Team 6 – Anu & Chris Holt
  – RHBD of a DC/DC Switching Converter

• Team 7 – Julian & Raj
  – Single-Error Detection in ALU

• Team 8 – Efosa & Drew
  – Chutes and Ladders
Final Project Selections

- Team 9 – Ryan & Brian
  – War (The Card Game)

- Team 10 – Ezgi, Jason, Ian
  – Simon Game

- Team 11 – Andre & Stephen
  – Simon Game

Features of FPGAs

- Besides primitive logic elements and programmable routing, some FPGA families add other features

- Embedded memory
  – Many hardware applications need memory for data storage.
  – Many FPGAs include blocks of RAM for this purpose

- Dedicated logic for carry generation, or other arithmetic functions

- Phase locked loops for clock synchronization, division, multiplication

- Microprocessor cores

Choosing an FPGA

- Chip size is not everything

- Evaluate your design needs
  – Number of I/O pins
  – Available logic resources
  – Special functional blocks

- History with a vendor
Questions to Ask

• Estimate of ASIC equivalent gates

• Input/Output
  – Pin requirements
  – Interface technologies

• Acceptable packaging options

• Requirement for specialized functions
  – Gigabit transceivers, embedded MAC, embedded RAM, embedded µP cores, etc.

• Requirement for Intellectual Property (IP)

I/O Technology

• Input/Output (I/O) has become very complex
  – Used to only have to worry about TTL vs CMOS
  – TTL had current drive requirements, CMOS just voltage level requirements
  – Both used full swing signals (0 to Vdd, used to be 5 V)

• New issues in I/O technology
  – Limit voltage swing to speed up signaling
  – Voltage swing about a reference voltage instead of between 0 and Vdd
  – Differential signaling to reject noise
  – Termination required to prevent signal reflections from corrupting signals
### FLEX 10K Device Family Overview

**Table 2. FLEX 10K Device Overview (Part 2)**

<table>
<thead>
<tr>
<th>Feature</th>
<th>EPF10K70</th>
<th>EPF10K100</th>
<th>EPF10K100A</th>
<th>EPF10K100E</th>
<th>EPF10K130V</th>
<th>EPF10K130E</th>
<th>EPF10K200E</th>
<th>EPF10K200S</th>
<th>EPF10K250A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical gates (logic and RAM)</td>
<td>70,000</td>
<td>100,000</td>
<td>130,000</td>
<td>200,000</td>
<td>250,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic elements (LEs)</td>
<td>3,744</td>
<td>4,992</td>
<td>6,656</td>
<td>9,984</td>
<td>12,160</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic array blocks (LABs)</td>
<td>468</td>
<td>624</td>
<td>832</td>
<td>1,248</td>
<td>1,520</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Embedded array blocks (EABs)</td>
<td>9</td>
<td>12</td>
<td>16</td>
<td>24</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total RAM bits</td>
<td>18,432</td>
<td>24,576</td>
<td>32,768</td>
<td>32,768</td>
<td>32,768</td>
<td>40,960</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes to tables:
1. For designs that require JTAG boundary-scan testing, the built-in JTAG circuitry contributes up to 31,250 additional gates.
2. The greater number in this cell describes FLEX 10KE devices.

### FLEX 10K Device Block Diagram

**FLEX 10K Logic Array Block (LAB)**

**FLEX 10K Embedded Array Block (EAB)**
Embedded Specialized Functions

- RAM blocks
- Multipliers
- Logic blocks

Hard Microprocessor Core

(a) One embedded core
(b) Four embedded cores

Altera’s Soft Microprocessor Core

- **NIOS II**
  - Features a general-purpose RISC CPU architecture
  - Three cores optimized for a specific price and performance range
    - Fast (Nios II/f) – less than 600 LEs
    - Economy (Nios II/e) – less than 1300 LEs
    - Standard (Nios II/s) – less than 1800 LEs
  - Supported by the Stratix and Cyclone FPGA families
    - Original NIOS processor supported by all FPGA families

(b) Four embedded cores

Intellectual Property (IP)

- Refers to any existing functional block (H/W)
- Can also take the form of software routines
- Sources of IP include
  - Internally created blocks from previous designs
  - FPGA vendors
  - Third-party IP providers

http://www.altera.com/products/ipprocessors/nios2/overview/n2-overview.html
Using IP in FPGA Design

- Generally encrypted by vendor
- Vendor has made effort to achieve optimal implementation

IP Core Generator

- Special parameterized tools that allow designers to specify functional elements, widths/depths, etc.

Speed Grades

- Databooks often list different speed grades for a part at the same temperature
- Simply test parts that come off the fabrication line and see how fast they are
  - Divide the parts into different speed bins
  - For three speed grades, a design goal might be to have 15% of your parts fall in the upper bin, 50% in the middle bin, and 25% in the lower bin
  - As the process matures, more and more fabricated parts will move into the upper speed bin, at which point you make a new upper speed bin.
  - Obviously, faster parts cost more (and are more profitable)

Selection of Speed Grade

- Rule of Thumb
  - Moving up a speed grade increases performance by 12 to 15 percent, but increases cost by 20 to 30 percent
  - If you can manipulate the architecture to improve performance by 12 to 15 percent, you can drop a speed grade and save 20 to 30 percent on cost
- For high volume, you want to use the lowest speed grade that is feasible
Summary

• Choosing the “right” FPGA is not easy!

• There are multiple dimensions to consider

• Most designs are I/O limited

• Be sure the device includes any specialized capabilities you need