Introduction to FPGA & Quartus II

Speaker: Y.S. Lin
Advisor: Prof. Li & Prof. Lu
May 27th, 2003

Outline

• PLD & FPGA introduction
  – Definitions & differences
  – Advantages & disadvantages

• Quartus II
  – Introduction
  – Design entry
  – Project compilation
  – Timing analysis
  – Function simulation
  – Device programming

• Conclusion
PLD & FPGA Introduction

PLD & FPGA

- PLD (Programmable Logic Device)
  - EEPROM Process & Product-term logic
  - SPLD (Simple PLD)
    - Hundreds of gates
    - Usually under 28 pins package
  - CPLD (Complex PLD)
    - Thousands to millions of gates
    - Usually hundred of pins
    - Altera names all its products as CPLDs
- FPGA (Field Programmable Gate Array)
  - SRAM Process & LUT (Look-Up-Table) logic
    - Need EPROMs to configure
  - Difference from PLDs
    - More registers, less gates
- Almost Merged
Advantages

• Improve time-to-market
• Allow try-and-error

Modern FPGA Features

• Increased capacity
  – Millions of gates
• Hierarchical structures
  – Logic, routing hierarchies
• Embedded memories
  – RAM (Single/Dual port), ROM, FIFO support
  – Implementation of general logic in embedded array blocks
• Enhanced embedded arithmetic resources
  – Carry chains, cascade chains
  – Embedded processor (e.g. Altera’s Excalibur)
• Support different I/O standards
  – PCI, AGP, LVDS
• Clock management circuits
  – Multiple PLLs suitable for multi-clock designs
Altera APEX 20K & KE Family

- Advanced Programmable Element MatriX (APEX)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Gates</td>
<td>35,000</td>
<td>60,000</td>
<td>100,000</td>
<td>150,000</td>
<td>200,000</td>
<td>300,000</td>
<td>400,000</td>
<td>600,000</td>
<td>800,000</td>
<td>1,000,000</td>
</tr>
<tr>
<td>Maximum System Gates</td>
<td>112,764</td>
<td>191,792</td>
<td>252,912</td>
<td>304,439</td>
<td>404,480</td>
<td>525,624</td>
<td>728,064</td>
<td>1,091,848</td>
<td>1,607,624</td>
<td>1,771,520</td>
</tr>
<tr>
<td>Logic Elements</td>
<td>1,200</td>
<td>2,560</td>
<td>4,150</td>
<td>6,400</td>
<td>9,520</td>
<td>11,020</td>
<td>16,640</td>
<td>24,020</td>
<td>39,000</td>
<td>51,640</td>
</tr>
<tr>
<td>Maximum RAM Bits</td>
<td>24,375</td>
<td>32,706</td>
<td>53,240</td>
<td>81,920</td>
<td>156,496</td>
<td>197,496</td>
<td>313,996</td>
<td>511,296</td>
<td>637,596</td>
<td>644,368</td>
</tr>
<tr>
<td>Maximum Microblocks</td>
<td>582</td>
<td>746</td>
<td>1,146</td>
<td>1,645</td>
<td>2,633</td>
<td>1,153</td>
<td>1,884</td>
<td>3,613</td>
<td>5,566</td>
<td>3,698</td>
</tr>
<tr>
<td>Maximum Interconn.</td>
<td>128</td>
<td>154</td>
<td>202</td>
<td>252</td>
<td>310</td>
<td>382</td>
<td>458</td>
<td>532</td>
<td>624</td>
<td>658</td>
</tr>
</tbody>
</table>

[1]

Introduction to Quartus II Software
Introduction to Quartus II

- Operate in a self-contained environment

Design Entry
- Graphic Design Entry
- Megafunctions
- Text Design Entry
- Hierarchical Design Entry

Design Compilation
- Logic Synthesis and Fitting

Verification & Programming
- Timing simulation
- Functional simulation
- Timing analysis
- Device Programming

Quartus II Design Methodology

Design Specification
- Design Entry Files
- Project Compilation
- Quartus Simulation
- Timing Analysis
- Verified?
- Device Programming
- PG & LA Verify
- Design Modification
Design Entry Files

- Multiple design entry methods
  - Quartus II
    - Block/Schematic Editor
    - Text editor
      - AHDL, VHDL, Verilog
    - Memory editor
      - .hex, .mif (Memory initialization file)
  - Third party EDA tools
    - EDIF, OrCAD schematics
  - Add flexibility and optimization to the design entry process by using LPM and Megafuctions
Quartus II Projects

Quartus II Project Management

- What is a Project?
  - A design file
  - A project is:
    - checked for design entry errors
    - compiled
    - simulated (functional or with timing)
    - analyzed for timing
    - used to generate programming file

- Projects can be archived
Quartus II Projects (1/4)

- New project wizard

1. Invoke New Project Wizard

2. Select Working directory

3. Name of Project. Recommendation: Use top-level design entity

Quartus II Projects (2/4)

4. Add design files
   - Graphic (.BDF, .GDF)
   - AHDL
   - VHDL
   - Verilog
   - EDIF

Notes:
- All files in the project directory do not need to be added
- Add top level file if file name and entity name are not the same

5. Add user library pathnames and files
Quartus II Projects (3/4)

5(cont.) Add user library pathnames and files
- User Libraries (ex. MegaWizard functions)
- MegaCoros/AMPP libraries
- Pre-compiled VHDL packages

Browse to file and click on Add.

Quartus II Projects (4/4)

6. Review results and click on Finish
Block Diagram/Schematic File Editor

- Block diagram entry is mainly for top-down design methodology
- Schematic file entry is the traditional schematic design entry
- User can enter blocks, primitives, and megafonctions from Quartus II-provided or user libraries

Block Editor – Create a New File

- Select File => New
- Select Block Diagram / Schematic File
- Save as *.BDF (Block Design File)
- Add file to current project

Click OK
Block Editor – Block Design File

Block and symbol editors

Block Editor – Enter Symbols

- Schematic file entry
- Click the symbol tool button

Symbol library Preview the symbol
Block Editor – Draw Block

- Text Design Entry
- Click the Block Tool button
- Click and drag on the block diagram and you can see the Block symbol

Block Editor – Specify I/Os

- Right click on the block symbol, choose Block Property
- Select I/Os, input your I/O name and type

Click OK
Block Editor – Make Connections

- Wire (single bit line)
- Bus (Multiple bits)
- Conduit
  - Connects blocks to any other objects

Block Editor – Mapper properties

- Map the block I/Os when the I/O names are different between the blocks.
- First, label the connector
  - Right-click on the connector and choose Properties
Block Editor – Mapper properties (cont.)

- Double-click on the mapper to open the Mapper Properties dialog box
- In the General tab, set the Mapper Type – Input, Output, Bidir
- In the Mappings tab, set the I/O on block and connector signal
- Click Add and hit OK

Block Editor – Make connections

- Enter mapper properties on both the blocks
- Now, the I/Os are connected
- Save the file as a .bdf file
Block Editor – Add I/O pins

- Click the symbol tool button

Block Diagram

- Arrange the Blocks, Primitives, and Megafuctions
Block Editor – Generate Design File

- Right-click on block symbol
- Select **Create Design file from Selected Block**

Block Editor – Create Design File

These lines are necessary for Quartus II to update the source code:

```vhdl
module hvalores;

// Do not remove this line!
end h;

// Port Declaration
input [1:0] sel;

// Do not remove this line!
endmodule;
```

Quartus II creates a design file that contains the port names that are specified in your block.
Quartus II MegaWizard Plug-In Manager

• The MegaWizard® Plug-In Manager (Tools menu) helps you create or modify design files that contain custom megafunction variations, which you can then instantiate in a design file.
• These custom megafunction variations are based on Altera-provided megafunctions, including library of parameterized modules (LPM) functions.
• The MegaWizard Plug-In Manager allows you to run a wizard that helps you easily specify options for the custom megafunction variations. The wizard asks questions about the values you want to set for parameters or about which optional ports you want to use.
• Altera recommends using the LPM functions to replace all other types of similar functions
MegaWizard Plug-In Manager

1. Select MegaWizard

2. Create New

3. Select a suitable Megafonctions

(1) Available Megafonctions:
- ABC Megafonction
- XYZ Megafonction
- P Megafonction

(2) Which megafonction would you like to create?
- ABC Megafonction
- XYZ Megafonction
- P Megafonction

(3) How to compile your project successfully? 
- Use the correct compiler version
- Ensure all dependencies are met

(4) You cannot use library functions here.
3. Configure the Megafunction

(1) Which operating mode do you want for the slave? (Select one):
- Address only
- Generate 'mil' only (use just to allow you to do both)
- None (Default)

(2) How wide should the 'data' and 'data2' inputs be?

(3) [Diagram showing configuration options]

(4) Is the slave on board input bus value a constant?
- Yes, define
- Yes, default

(5) [Diagram showing configuration options]

(6) Do you want any optional inputs or outputs?
- Yes, create an input
- Yes, create an output
- Create a condition output

(7) [Diagram showing configuration options]

(8) Do you want to prioritize the functions?
- Yes, I want an output priority list
- Create a non-blocking reset output
- Create a soft reset output

[Diagram showing configuration options]
MegaWizard Plug-In Manager

Add the add10B.v to the design entry.

Quartus II Compilation
Compiler Settings

- Compiler control information
  - Level of compilation
  - Device & Device options
  - Pin assignments
  - Synthesis & Fitting
  - Timing analysis
- Accessed via the Processing Menu
- Information stored in a Compiler Settings File (*.CSF)

Compiler Settings – General

- Select processing => Compile Mode
- Select processing => Compiler Settings...
Compiler Settings – Chips & Devices

Pin assignment – Add I/O pins

- Select a pin number
- Invoke **Node Finder** to find pin name or type
Pin assignment – Add I/Os

- Select **Filter: Pins: all** and click **Start**
- Add to assignment list

Compiling in Quartus II

- Select **Processing => Start Compilation**
The Compiler Report

- Contains all information on how a design was implemented in the targeted device
  - Device Summary Statistics
  - Compiler Settings
  - Floorplan Views
  - Device Resources Used
  - State Machines Implemented
  - Equations
  - Timing Analysis Results
  - CPU Resources
- This is a read-only window

Reporting Timing Results

- Timing information is part of the Compilation Report
  - Summary Timing Analyses
  - fmax (not include delays to/from pins)
  - fmax (include delays to/from pins)
  - Register-to-Register Table
  - tsu (Input Setup Time)
  - th (Input Hold Time)
  - tco (Clock to Out Delays)
  - tpd (Pin to Pin Delays)
- All timing results are reported here
Quartus II Simulator

Simulations

- **Run Functional Simulation**
  - Fast compilation
  - Logical model only, no logic synthesis
  - All nodes are retained and can be simulated
  - Outputs are updated without delay

- **Run Timing Simulation**
  - Slower compilation
  - Timing model: logical & delay model
  - Nodes may be synthesized away
  - Outputs are updated after delay
**Create Waveform file**

- Create a waveform for simulation
- Select File => New => Other Files
- Select Vector Waveform File
- Save as *.VWF (Vector Waveform File)

Click OK

**Setup End time**

- Specify maximum length of simulation end time
  - Select Time => End time
Add I/O pins – select

- Select View => Auxiliary Windows => Node Finder
- Select Filter: Pins: all and Click Start
- Select all pin and drop them into the Name in waveform file

Add I/O pins

- All pins are in the Name list
- With initial values or Hi-Z
Creating a Clock

- Right-click on the waveform name and choose Value => Clock

Creating Counting Pattern

- Right-click on the waveform name and choose Value => Count Value
Simulator Settings – General

- Select Processing => Simulate Mode
- Select Processing => Simulation Settings

Simulator Settings – Time/Vectors
Simulator Settings – Mode

Simulator Settings – Options
Simulator Result

- Select **Processing => Run Simulation**
FPGA Configuration

- FPGA are usually SRAM-Based
  - Must be reconfigured with every power-up

- Can be configured via:
  - Configuration device
    - Download the Programming Object File (.pof) to the Altera programming hardware by communication cable
    - Then configure the device
  - Download cable
    - Configure the device directly by download the SRAM Object file (.sof) through the communication cable

- Communication cable
  - ByteBlaster™: Uses Parallel port (LPT) of the PC
  - MasterBlaster™: Use USB or serial port of PC or workstation

ByteBlaster

Figure 1. ByteBlaster Parallel Port Download Cable
Configuration

- Select **Processing** => open programmer
- Save as *.CDF (Chain Description File)

![Configuration Image]

Configuration

- Select **Setup** => Add => **Hardware Setup** dialog box, select **ByteBlasterMV**

![Configuration Image]
Configuration

- **Select Add File**
  - *.SOF (SRAM Object File)
  - *.POF (Programmer Object File)
- **Check on Program/Configure.**

Click Start

Important !!

- The board can be damaged without proper anti-static handling!
- If the I/O pins on the board are **NOT** 5.0-V tolerant, it should **NOT** be directly connected to logic powered from a 5.0-V supply!
- Remember to add all needed files into your project!
- Be sure to connect pin1 of the JTAG cable to the pin1 of the header on the board!
- If you configure the FPGA device with *.pof files, notice that EPC devices configures the APEX device with the new data after power to the board is cycled!
- If you configure the FPGA device with *.sof files, keep the power on!
- Be careful to the settings of trigger & threshold of LA!
- The data sheets of Altera DSP development boards can be download from Altera’s website
Conclusion

- FPGA devices & Quartus II software are introduced
- Follow the standard design flow, IC design verification can be done efficiently
- If you have any problem with Quartus II, I will try my best to help you
- Rm333, 林晏生