Synthesis Challenges for Next-Generation High-Performance and High-Density PLDs

Jason Cong
Department of Computer Science
University of California,
Los Angeles, USA

Songjie Xu
Aplus Design Technologies, Inc.
Los Angeles, USA
Outline

◆ Introduction
  ◆ Synthesis Challenges for New Architectures
  ◆ Synthesis Challenges for High Density and High Performance
  ◆ Concluding Remarks
PLD Industry Growth

- Enjoyed the exponential growth as the rest of the semiconductor industry
- With an even faster rate
Definitions

- **PLD (Programmable Logic Device)**
  - **CPLD (Complex PLD)**
    - Extensions of early PAL
    - Consist of PLA-like blocks
    - Macrocell
  - **FPGA (Field Programmable Gate Array)**
    - Typically based on look-up tables (LUTs)
    - Multiple LUTs form a programmable logic block (PLB)
CPLD

- Example: Altera MAX 7000

![CPLD Diagram]
Macrocell

- Example: Altera MAX 7000
  - Each macrocell has a logic array, a product-term select matrix, and a programmable register
Definitions

◆ PLD (Programmable Logic Device)
  - CPLD (Complex PLD)
    - Extensions of early PAL
    - Consist of PLA-like blocks
    - Macrocell
  - FPGA (Field Programmable Gate Array)
    - Typically based on look-up tables (LUTs)
    - Multiple LUTs form a programmable logic block (PLB)
FPGA

◆ Example: Xilinx XC 4000
- Xilinx XC 4000
  - Each PLB has two 4-LUTs, one 3-LUT and 2 FFs
# Advance of PLD Architectures

<table>
<thead>
<tr>
<th></th>
<th>1980’s</th>
<th>1998/1999</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Altera</strong></td>
<td><strong>MAX 5000:</strong> 32-192 P-terms 600-3,750 usable gates</td>
<td><strong>APEX 20K:</strong> 51,840 Logic elements (LUTs) 442,368 RAM bits 3,456 P-term macrocells 60,000-1.5M usable gates</td>
</tr>
<tr>
<td><strong>Xilinx</strong></td>
<td><strong>XC 2000:</strong> 64-100 LUTs 1,200-1,800 logic gates</td>
<td><strong>Virtex:</strong> 58K-4M system gates 1Mb distributed RAM 832Kb embedded memory</td>
</tr>
</tbody>
</table>
PLD Synthesis Tends to Fall Behind ...

- Additional features and capabilities in the new architecture often place new requirements for synthesis tools
- Higher density and higher performance demand better scalability and more efficient optimization
- Devil is always in the software ...
  - Tool effort is often being underestimated
  - Quick customization from ASIC or existing PLD synthesis tool leads to considerably inferior results
  - Software is often the bottleneck of new PLD product release ...
Challenges to PLD Synthesis

- Support for new PLD architectures
  - Hierarchical architectures
  - Heterogeneous architectures

- Support for high-performance and high-density PLD designs
  - Layout-driven synthesis
  - Incremental synthesis
  - IP-based synthesis
Two important trends
- Hierarchical architectures
- Heterogeneous architectures

Synthesis needs
PLD Architecture Development Trend - Hierarchical Architectures

- **Basic Idea**
  - Group of basic logic blocks into clusters
  - Fast local programmable interconnects inside clusters
  - May have multiple levels of hierarchy

- **Benefits**
  - Exploit the inherent locality of interconnections in most applications
  - Lead to the improvement in both performance and density
Example Hierarchical Architectures

- **Altera FLEX 10K**
  - Each LAB has 8 LEs
  - Each LE has a 4-LUT and a programmable register
Two Types of Clusters

- **Hard-wired connection based cluster (HCC)**
  - Intra-cluster connection is formed by hard wires
  - e.g. CLB in XC4000

- **Programmable interconnection based cluster (PIC)**
  - Intra-cluster connection is formed by a local programmable interconnection array
  - e.g. LAB in FLEX 10K and APEX 20K

Synthesis Challenges for New Architectures
Existing Synthesis Results for HCC

- Traditional approach
  - Map into LUTs and then combine the LUTs to form HCCs in a heuristic post-processing step

- Recent advance [Cong & Hwang, FPGA’97]
  - Use Boolean matching techniques to completely characterize the set of functions that can be implemented in a HCC
  - Map a netlist directly into HCCs
Hard-Wired Connection Based Clusters (HCCs)

- **Example: Xilinx XC 4000 CLB**
  - Each CLB has two 4-LUTs connected to a 3-LUT
Characterization based on functional decomposition

- $f(X) = H(F(X1), G(X2))$,
- $f(X) = H(F(X1), G(X2), x)$,
- $f(X) = H(F(X1,x), G(X2), x)$,
- $f(X) = H(F(X1,x), G(X2,x), x)$.

Conditions

- $F$ and $G$ input sizes $\leq 4$

Result: matched all “difficult examples” (over 1,700) from Xilinx

- Best known tool produced only about 70% match
Example: Mapping to XC4K CLB

- Given a function \( f(0,1,2,3,4,5) \) where
  \[
  a = 1' + 3, \quad b = 1 + 3
  \]
  \[
  f = 0'245b' + 0'245'b + 0'145b + 012'5'a + 0'2'4'5a + 025b + 0'2'5'a' + 045a' + 05'b'
  \]

- How many XC4K CLBs are needed to implement \( f(0,1,2,3,4,5) \)?
Example: Mapping to XC4K CLB (Cont’d)

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Packing</th>
<th>#CLBs</th>
<th>#Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chortle-crf</td>
<td>simple</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>FlowMap</td>
<td>simple</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>FlowMap</td>
<td>functional</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Boolean</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The Boolean matching result

Synthesis Challenges for New Architectures
Programmable Interconnection Based Cluster (PIC)

- Example: Altera APEX 20K
  - Each LAB has 10 LEs (LUT + FF) connected through a fully programmable matrix

Synthesis Challenges for New Architectures
Existing Synthesis Results for PIC

- Common approaches
  - Map into basic logic blocks and then group them into clusters under size and pin constraints
  - Recent progress on circuit clustering
    - Performance driven clustering for combinational circuits [Lawler’69] [Yang & Wong, T-CAD’97]
Benefits of Considering Retiming during Clustering

- Proper clustering allows retiming to **hide** inter-cluster delays

(E.g., assume gate_delay = 1, inter_cluster_delay = 2)

Clustering A
- $\Phi = 8$
- Retiming reduces delay

Clustering B
- $\Phi = 8$
- Retiming cannot help

same cutsize
Major Challenge in Synthesis for Hierarchical Architectures

- Can we synthesize a design directly into a multi-level hierarchical architecture?
  - Most existing PLD synthesis algorithms transform a given design into a flat netlist of basic PLBs and then go through a separate clustering/partitioning step.
  - Very few consider synthesizing directly for hierarchical architectures.
Three types of heterogeneous architectures

- Type 1: Multiple sizes and/or configurations of the same type of logic blocks
  - e.g. ORCA 2C, VF1, XC4000

- Type 2: Multiple types of logic blocks
  - LUTs, macrocells, and MUXes
  - e.g. APEX 20K

- Type 3: Different kinds of resources on the same chip
  - Programmable logic blocks
  - Embedded memory blocks (EMBs)
  - Embedded processors
Type 1 Heterogeneous Architectures

- Example: Xilinx XC 4000
  - Each CLB can implement two 4-LUTs or one 5-LUT
Synthesis Results for Type 1 Heterogeneous Architectures

- **Area minimization**
  - [He & Rose, FPGA’94]
  - [Korupolu, et al, DAC’98]
  - [Cong, Ding & Wu, FPGA’99]

- **Delay minimization**
  - HeteroMap [Cong & Xu, DAC’98]
    - Delay optimal polynomial-time algorithm

- **Evaluation results show**
  - Heterogeneous architectures are superior to homogeneous ones for both area and delay
  - “One size fits all” doesn’t produce best results.
Architecture Evaluation—Homogeneous vs. Heterogeneous FPGAs

Delay(3-LUT) : Delay(4-LUT) : Delay(5-LUT) : Delay(6-LUT) = 1 : 1.3 : 1.7 : 2
Area(3-LUT) : Area(4-LUT) : Area(5-LUT) : Area(6-LUT) = 1 : 2 : 4 : 8

Synthesis Challenges for New Architectures
“AT²-Metric” for Homogeneous and Heterogeneous FPGAs

Delay(3-LUT) : Delay(4-LUT) : Delay(5-LUT) : Delay(6-LUT) = 1 : 1.3 : 1.7 : 2
Area(3-LUT) : Area(4-LUT) : Area(5-LUT) : Area(6-LUT) = 1 : r : r² : r³

Synthesis Challenges for New Architectures
Type 2 Heterogeneous Architectures

- An example: Altera APEX 20K
  - Embedded system blocks (ESB) can implement dual-port RAM, ROM, FIFO, CAM blocks, and P-term logic
  - In P-term mode, each ESB has 16 macrocells
    - Each macrocell has two P-terms
Synthesis for Type 2 Heterogeneous Architectures

- Very little work
- Preliminary study for a hybrid architecture of LUTs and Pterm blocks [Kaviani, Ph.D. thesis’99]
  - Use a greedy approach for hybrid mapping
    - Use LUTs for density optimization
    - Use Pterm blocks for performance optimization
Type 3 Heterogeneous Architectures

- An example: FLEX 10K (logic array + embedded memory blocks (EMBs))
  - 576 to 12,160 LEs
  - 3 to 20 embedded array blocks (EABs)
    - Each EAB has 2K bits (11x1, 10x2, 9x4, 8x8)
Field-Programmable System-on-a-Chip (FPSOC)

General-Purpose FPSOC  Application-specific FPSOC

- processor
- Programmable Logic
- memory

- processor
- ASIC
- memory
- Programmable Logic

Synthesis Challenges for New Architectures
Synthesis for Type 3 Heterogeneous Architectures

- Explore logic implementation using EMBs
  - Area minimization
    - EMB_Pack [Cong & Xu, FPGA’98]
    - With Delay constraint
    - SMAP [Wilton, FPGA’98]
  - Delay minimization
    - [Cong & Xu, ICCAD’98]

- The general synthesis problem for FPSOC is largely untouched
Synthesis Needs for FP-SOC

- Partition the design/application to heterogeneous resources. E.g.
  - Software/hardware partitioning
  - Memory/logic partitioning

- Efficient use of each type of resources. E.g.
  - Code generation for embedded CPUs
  - Automatic synthesis for FPGA

- Scheduling & synchronization of various components. E.g.
  - Real-time O/S

- Trade-off between heterogeneous resources

- Support for IP integration
Outline

- Introduction
- Synthesis Challenges for New Architectures
- Synthesis Challenges for High Density and High Performance
- Concluding Remarks
Important Synthesis Problems

- Layout-driven synthesis
- Incremental synthesis
- IP-based design

Synthesis Challenges for High Density and High Performance
Layout-Driven Synthesis

- Scaling of IC feature size \([\text{NTRS'97}]\)
  - Interconnect delay becomes more and more dominant in the overall circuit delay

- FPGA design
  - Interconnect delay has always been very significant (due to programmable switches)

- Layout design has a significant impact on performance

- Synthesis needs to consider impact on layout
Logic v.s. Local Interconnect v.s. Global Interconnect Delay

<table>
<thead>
<tr>
<th>Delay Resource</th>
<th>Delay Value (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Element (LE)</td>
<td>2.4</td>
</tr>
<tr>
<td>Local Interconnect</td>
<td>0.5</td>
</tr>
<tr>
<td>Row Interconnect</td>
<td>4.7</td>
</tr>
<tr>
<td>Column Interconnect</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Altera FLEX8K part

Synthesis Challenges for High Density and High Performance
Delay Distribution

- Logic: 30%
- Global Interconnect: 61%
- Local Interconnect: 9%

Synthesis Challenges for High Density and High Performance
Challenges and Opportunities for Layout-Driven Synthesis

- **Challenges:**
  - Interconnect design is not finalized until after placement and routing
  - Both synthesis and layout are highly complex. How to properly combine them without complexity explosion?

- **Opportunities:** substantial performance gain
  - Example: Mapping with consideration of fast interconnections (cascade chains)
Comparison between FlowMap and Fast Interconnection Mapping

- Delay Assumption: 4-LUT fast pin delay = 0.7 ns, fast interconnect delay = 0.2 ns, 4-LUT slow pin delay = 2.7 ns, general interconnect delay = 4.1 ns
- LUT fast interconnect is connected to the fast pin
Comparison between FlowMap and Fast Interconnection Mapping (Cont’d)

- Delay Assumption:
  - 4-LUT fast pin delay = 0.7ns
  - 4-LUT slow pin delay = 2.7 ns
  - Fast interconnect delay = 0.2ns
  - General interconnect delay = 4.1 ns
- LUT fast interconnect is connected to the fast pin
Incremental Synthesis

◆ Motivation
  ▪ The PLD designs are getting more complex
  ▪ All design process is iterative/incremental
  ▪ Resynthesizing the entire large design is not acceptable with consideration of multiple design iterations
  ▪ The highly incremental design process requires fast incremental synthesis capabilities
Requirements on Incremental Synthesis

- **Preservability**
  - Preserve as much information as possible from the existing synthesis solution

- **Efficiency**
  - A faster synthesis system will enable more design iterations and shorten the overall design time

- **Quality of the synthesis solution**
  - Delay, area, etc. should be as close as possible to that by complete re-synthesis
Status on Incremental Synthesis

- Very few works
  - ECO [Kukimoto & Fujita, ICCAD’92]
    - No structural change is allowed
    - Only functional change is allowed
  - Incremental mapping [Cong & Hui, DAC’2000]
    - Preserve optimal mapping depth
    - Achieve over 300X speed-up for circuits of about 100,000 gates compared to re-mapping by FlowMap

- Much more work is needed in this area
IP-Based Design

◆ Motivation
  - Design reuse to improve productivity
  - Better performance and density

◆ Example:
  - Altera IP MegaStore
<table>
<thead>
<tr>
<th>TP Megafusion Vendor</th>
<th>Free Test Drive</th>
<th>Buy Now</th>
<th>APEX™ (1)</th>
<th>FLEX® 10K (1)</th>
<th>Other Device Families Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit, 66-MHz PCI Master/Target (PLSM-PCI/G)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>1,200</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit, 33-Mhz PCI Master/Target with DMA (PLSM-PCI/A)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>1,050</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33 MHz</td>
</tr>
<tr>
<td>32-bit, 33-Mhz PCI Master/Target (PLSM-PCI/B)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>1,050</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33 MHz</td>
</tr>
<tr>
<td>32-bit, 33-Mhz PCI Target Only (PLSM-PCI/11)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>550</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FLEX 8000 FLEX 6000</td>
</tr>
<tr>
<td>32-bit Master/Target PCI Function (PLSM-PCI/MT32)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>65 MHz</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64-bit Master/Target PCI Function (PLSM-PCI/MT64)</td>
<td>TRY</td>
<td>BUY</td>
<td>(2)</td>
<td>(2)</td>
<td>55 MHz</td>
</tr>
<tr>
<td>Altera Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Requirements on IP-Based Design

- **IP representation** -- should allow migration between
  - Different FPGA vendors
  - Different FPGA generations

- **Characterization**
  - functionality
  - performance

- **Interface with synthesis tools**
  - automatic inference/instantiation
  - optimization and constraint propagation
  - simulation and verification

- **IP protection**
  - How to prevent un-authorized use?
  - E.g. Embed watermarks in FPGA mapping solutions
    [Kirovski, et al, ICCAD’98]
Outline

◆ Introduction
◆ Synthesis Challenges for New Architectures
◆ Synthesis Challenges for High Density and High Performance
◆ Concluding Remarks
Concluding Remarks

- PLD market is going through a rapid expansion
- PLD synthesis is facing many new challenges
  - Support for new PLD architectures
    - Hierarchical architectures
    - Heterogeneous architectures
  - Support for high-performance and high-density PLD designs
    - Layout-driven synthesis
    - Incremental synthesis
    - IP-based synthesis
- Many research and business opportunities
  - UCLA VLSI CAD Laboratory
  - Aplus Design Technologies, Inc.
PLD Synthesis Research at UCLA

- Advanced synthesis algorithms
  - Synthesis for heterogeneous architectures
  - Synthesis for sequential circuits with simultaneous mapping, retiming, and pipelining
  - Layout-driven synthesis
  - IP-based synthesis
  - Synthesis/compilation techniques for FPSOC ...
  - Software prototype: RASP system

- Architecture evaluation
  - Evaluation of PLB architecture
  - Evaluation of heterogeneous architectures
  - Evaluation of hierarchical architectures ...
  - Software prototype: fpgaEva tool

- URL: http://cadlab.cs.ucla.edu/~xfpga

Concluding Remarks
UCLA RASP Synthesis System for LUT-Based FPGAs

- EDIF netlist
- HDL design
- Internal netlist
- LUT Mapping Engine
- LUT netlist
- Vendor Specific netlist
  - Xilinx, Altera, ORCA
- Placement Routing
- PLB Mapping Engine
- Chip Programming Information

Concluding Remarks
FPGA Architecture Evaluation

Concluding Remarks

Welcome to evaluate heterogeneous FPGA architectures using fpgaEva.
A new start-up in PLD synthesis
- Based in Los Angeles (near UCLA)

Objective: provide **Advanced Programmable Logic Unified Solution (APLUS)**
- Unify architecture and synthesis
- Unify synthesis and layout

Products & Services
- Next generation synthesis tool for high-density, high-performance PLDs
- Architecture evaluation tool kits and services

Has already established strategic partnership with several major PLD vendors

URL: [http://www.aplus-dt.com](http://www.aplus-dt.com)
The Typical Design Flow Using LPMs

Synthesis Challenges for High Density and High Performance