DPGA-Coupled Microprocessors

Commodity IC’s for the Early 21st Century

by

Aman Sareen

School of Electrical Engineering and Computer Science
Ohio University
What’s going to be covered ??

Part 1

- Technology Trends
- Application Outlook
- Some Developed Reconfigurable Engines
- Applications of Reconfigurable Logic
- Common Objectives of Reconfigurable Devices
- Limitations of the Current Systems
What’s going to be covered ?? (cont.)

Part 2

- Uniform Computational Array Model
  - FPGA
  - SIMD Arrays
- Hybrid Arrays
  - DPGA
- DPGA Prototype
  - Highlights
  - Architecture
  - Implementation

Applications
Benefits
What’s going to be covered?? (cont...)

Part 3
- DPGA Coupled Processor Applications
- Costs and Benefits of Reconfiguration
- Challenges
- Conclusion
Technology Trends

What's going on in the industry??
- Operational performance of microprocessors is increasing by 60% each year.
- More and more transistors (25% increase per year) on a single chip.
- 12 million transistors on a single chip are estimated by the end of the century.

Disadvantages ??
- High performance is not always achievable.
- Cost ineffective.
- Risks overspecialization.
- Reduced volume utilization per design investment.

So what do we do ?? => Reconfigurable Design

What does it do ??
- Application acceleration.
- Implement system specific functions.
Application Outlook

There’s always a scope of additions/modifications

So what do we do ?? => Reconfigurable Design

What does it do ??

- It allows applications to specialize the hardware.
Some Developed Reconfigurable Engines

**PRISM** (Processor Reconfiguration through Instruction-Set Metamorphosis) built by Athanas and Silverman.
* couples a programmable element with a microprocessor.
* each application synthesizes new processor instructions for acceleration.

**CM-2** built at the Supercomputing Research Center by Cuccaro and Reese.
* the processor is augmented with reconfigurable logic to perform common operations.

**SPLASH** built at the Supercomputing Research Center.
* used in genome sequence matching.
Applications of Reconfigurable Logic

- Binary Operations.
- Arithmetic.
- Encryption/Decryption/Compression.
- Sequence and string matching.
- Sorting.
- Physical system simulation.
- Video and image processing.
Common Objectives in Reconfigurable Applications

- High performance.
- Clear potential for application acceleration.
- Exploring bit-level parallel computation.
- High performance through parallelism.
- Customize data paths.
Limitations of the Current Systems

- Low Bandwidth and High Latency Interface
  - Expected acceleration not achievable.
  - Prevents close cooperation between fixed and reconfigurable logic circuits.
  - Expensive.
  - Limits throughput.

- High Reconfiguration Overhead
  - Single configuration must be maintained throughout an application.
  - Multitasking/Time sharing not possible.
Unified Computational Array Model

Computational Block of AE

Instruction

Array Element Computational Unit

Inputs from local state or from other array elements

Outputs to local state or to other array elements
Unified Computational Array Model

Lookup Models for AE Computational Unit

Instruction

Inputs from local state or from other array elements

Lookup Table (Memory)

Address Inputs

Data Outputs

Outputs to local state or to other array elements

Instruction = Memory Programming

Inputs from local state or from other array elements

Outputs to local state or to other array elements

Address Inputs

Data Outputs
Unified Computational Array Model

Instruction Distribution

Ideally, different instruction for each AE on each computational cycle

Drawback:
- Instruction distribution resource requirement increases.
- Instruction bandwidth becomes unmanageable.

\[ I_{BW} = \frac{P \times \log_2(N_f)}{t_{cycle}} \]

\[ P = 100, \quad N_f = 64, \quad \text{Operational Freq.} = 10 \text{ MHz} \]

\[ I_{BW} \Rightarrow 6 \text{ Gbits/sec} \]
Unified Computational Array Model

Weakening Instruction Distribution

FPGA
- Instruction / AE
- Uniform in time
- Slow programming phase

SIMD Array
- Instruction / cycle
- Uniform in space

FPGA
Static Instruction
(distinct for each array element efficiently constant during operation)

SIMD Array
Global Instruction
(common to all elements in array)
FPGA v/s SIMD Computation

FPGA
- Fixed Function in Time
- Spatially Varying Computation
- Bit-Parallel Computation
- Build Computation Spatially
  * Low-latency

SIMD Array
- Operation Varies in Time
- Homogenous Computation in Space
- Bit-Serial Computation
- Build Computation in Time
  * High Throughput on Homogenous data
Dynamically Programmable Gate Arrays

Hybrid Model

- Multiple Context FPGA
- Broadcast a Context Identifier
- Indirect Instruction Lookup
- Features:
  - Rapid Context Switch
  - Exploits local, on-chip Bandwidth
  - Spatially and Temporally Varying Computation
  - High Logic Density
  - Reuse Gates and Wires in Time
Dynamically Programmable Gate Arrays

Configurable Instruction-Store View of DPGA AE

Computational Unit (Lookup Table)

- Configurational Unit function is configured by Instruction Store output

Address Inputs

- Data Outputs

Outputs to local state or to other array elements

Data Outputs

Global Context Identifier (common to all elements)

Address Inputs

Instruction Store (Lookup Table)

Programming may differ for each array element

Instruction

Inputs from local state or from other array elements
Dynamically Programmable Gate Arrays

Applications

- Rapid Context Switch FPGA
- Time-Slice Computation
- Temporal Pipelining
- Operation Cache
- Processor Assistance
- Multi-Stream SIMD
- Boundary Condition handling
- Virtual Cells
DPGA Prototype - Highlights

• 4 on-chip configuration contexts
• DRAM configuration cells
• Automatic refresh of dynamic memory elements
• Non-intrusive background loading
• Wide bus architecture for high-speed context loading
• Two-level routing architecture
DPGA Prototype - Overview

- Memory Block containing Multiple Configurations
- Hierarchy 1: Memory
- Array Element 4-Input Lookup Table
- SubArray 4x4 composition of Array Element
- Hierarchy 2: Context<1> configuration bits
- Context<2> configuration bits
- Context<3> configuration bits
- Context<4> configuration bits
- Hierarchy 3: Full DPGA 3x3 Composition of SubArrays with Programmable CrossBar

February 12, 1998
Aman Sareen
DPGA Prototype - Context Memory

3-Transistor DRAM Cell

32-Column by 4-Context Memory Block
DPGA Prototype - Array Element

1 set
6 local
8 global
15 interconnects

Level One Interconnect

4 sets of 16-Entry Lookup Table

4x16-bit Memory Block

16:1 Multiplexer

D Flip Flop

CLk
DPGA Prototype - Local Interconnect

Diagram showing a grid of array elements with connections to 8 global inputs from North, West, East, and South.
DPGA Prototype - Subarray Interconnect
### DPGA Prototype - Areas

<table>
<thead>
<tr>
<th>Unit</th>
<th>Size</th>
<th>Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die</td>
<td>6.8mm X 6.8mm</td>
<td>Core with pads</td>
</tr>
<tr>
<td>Core</td>
<td>5.6mm X 4.7mm</td>
<td>All internal logic except pads</td>
</tr>
<tr>
<td>Array Core</td>
<td>5.25mm X 4.4mm</td>
<td>3 X 3 subarrays including crossbars (no pads)</td>
</tr>
<tr>
<td>Subarray + crossbar tile</td>
<td>1460µ X 1750µ</td>
<td>Subarray + 4 adjacent 16 X 8 crossbars and memory</td>
</tr>
<tr>
<td>Crossbar</td>
<td>495µ X 270µ</td>
<td>Crossbar including memory</td>
</tr>
<tr>
<td>Subarray</td>
<td>1150µ X 1400µ</td>
<td>4 X 4 Array Elements + 4 Local Decodes</td>
</tr>
<tr>
<td>Local Decode</td>
<td>253µ X 167µ</td>
<td></td>
</tr>
<tr>
<td>Array Element</td>
<td>275µ X 240µ</td>
<td>Includes local routing channels</td>
</tr>
</tbody>
</table>

3 metal, 1µ drawn 0.85µ effective CMOS process
### DPGA Prototype - Area Percentages

<table>
<thead>
<tr>
<th>Function</th>
<th>Elements</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic</td>
<td>Total</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Memory array</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Memory decode</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Fixed Logic</td>
<td>3</td>
</tr>
<tr>
<td>Network</td>
<td>Total</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Memory array</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Memory decode</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Switching</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Wiring</td>
<td>27</td>
</tr>
<tr>
<td>Blank</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
# DPGA Prototype - Estimated Timings

<table>
<thead>
<tr>
<th>Path</th>
<th>Symbol</th>
<th>Slow Speed</th>
<th>Nominal Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK $\rightarrow$ configuration memory stable</td>
<td>$t_{\text{mem}}$</td>
<td>4 ns</td>
<td>2.5 ns</td>
</tr>
<tr>
<td>CLK $\rightarrow$ XBAR out</td>
<td>$t_{\text{xbar1}}$</td>
<td>8.5 ns</td>
<td>5 ns</td>
</tr>
<tr>
<td>XBAR in $\rightarrow$ XBAR out</td>
<td>$t_{\text{xbar}}$</td>
<td>4.5 ns</td>
<td>2.5 ns</td>
</tr>
<tr>
<td>LUT in $\rightarrow$ LUT out</td>
<td>$t_{\text{lut}}$</td>
<td>9 ns</td>
<td>3.5 ns</td>
</tr>
<tr>
<td>CLK $\rightarrow$ CLK maximum (DRAM leakage)</td>
<td>$t_{\text{clkmax}}$</td>
<td></td>
<td>200 ns</td>
</tr>
</tbody>
</table>

\[ t_{\text{cycle}} = t_{\text{mem}} + n_{l} \times t_{\text{lut}} + n_{x} \times t_{\text{xbar}} \]
DPGA-Coupled Processor Applications

- General-Purpose Workstations and Personal Computers.
- Special-Purpose Computing Machines.
- Embedded Systems.
- Multiprocessor Systems
Costs and Benefits of Reconfiguration

• Specialized design limits range of application.
• Moving exception handling into reconfigurable logic.
  * Feature Interaction.
  * Migrating critical control of fixed resources to reconfigurable logic.
Challenges

• Processor reconfigurable logic interfacing.
• Grain Size.
• Area and Pin allocation.
• Multitasking and state interaction.
Conclusion

- Prototype demonstrates that efficient DPGAs can be implemented
- DPGAs allow computation to vary both spatially and temporally
- DPGAs require no additional bandwidth
- Both bit-parallel and bit-serial computation in a single array structure
- Higher performance
- Higher flexibility
- Lower part count
- Microprocessors with tightly integrated, rapidly reconfigurable logic promise to be prime commodity building block.