Reconfigurable Supercomputing Systems

Tarek El-Ghazawi, The George Washington University
tarek@gwu.edu
Kris Gaj, George Mason University
kgaj@gmu.edu
David Pointer, National Center for Supercomputing Applications
pointer@ncsa.uiuc.edu
Agenda for the Day

- Reconfigurable Supercomputing Architectures
- High Performance Reconfigurable Computing Applications – Part I
- High Performance Reconfigurable Computing Applications – Part II
- Performance of Reconfigurable Supercomputers
- Languages and Tools for Reconfigurable Supercomputing
- Challenges and Promises of Reconfigurable Computing
Reconfigurable Supercomputing Architectures

Tarek El-Ghazawi & Kris Gaj

The George Washington University
George Mason University
Outline

- Reconfigurable Supercomputing Systems
- Reconfigurable Chips and Technology Trends
- Reconfigurable Computing Boards
- Networks of Reconfigurable Computers (NORCs)
- Reconfigurable Clusters
- Reconfigurable Massively Parallel Systems
- Conclusions
What is Reconfigurable Supercomputing (RSC)
Reconfigurable Computers

The microchip that rewrites itself

Scientific American – June 1997

- Computers that modify their hardware circuits as they operate are opening a new era in computer design.

- Reconfigurable computers architecture is based on FPGAs (Field Programmable Gate Arrays)

Source: [Sci97]
Reconfigurable Supercomputing (RSC)

- Efficient high performance computing using parallel and distributed systems of both reconfigurable hardware resources and conventional microprocessors

- This tutorial establishes the current status, the direction taken, and the potential for RSC
# Synergism between Reconfigurable and Conventional Processors

<table>
<thead>
<tr>
<th></th>
<th>µP</th>
<th>RP (FPGA-based)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallelism Type</strong></td>
<td>Coarse-Grain</td>
<td>Fine-Grain</td>
</tr>
<tr>
<td><strong>Processing Type</strong></td>
<td>Software (\rightarrow) Control Flow</td>
<td>Hardware (\rightarrow) Data Flow</td>
</tr>
<tr>
<td><strong>Partitioning Applications across Processors</strong></td>
<td>Relatively Easy</td>
<td>Harder</td>
</tr>
</tbody>
</table>
Microprocessor and FPGA Performance Increases

Conservative estimates for FPGAs
Performance = # of Gates x Clock Rate

Source: [SRC02]
What is an FPGA Chip?

- **Field Programmable Gate Array**
- A chip that can be configured by user to implement different digital hardware
- **Configurable Logic Blocks and Programmable Switch Matrices**
- Bit-Stream to configure: function of each block & The interconnection between logic blocks
CLB Structure

Left-Hand SLICEM
(Logic or Distributed RAM
or Shift Register)

Right-Hand SLICEL
(Logic Only)

COUT

COUT

SLICE X1Y1

SLICE X1Y0

SLICE X0Y1

SLICE X0Y0

CIN

CIN

Interconnect
to Neighbors

Switch Matrix
CLB Slice

[Diagram with CLB Slice components and labels: Carry & Control Logic, Look-Up Table, COUT, YB, Y, X, D, S, Q, CK, EC, R, SLICE, G4, G3, G2, G1, F5IN, BY, SR, F4, F3, F2, F1, CIN, CLK, CE]
LUT (Look-Up Table) Functionality

- Look-Up tables are primary elements for logic implementation
- Each LUT can implement any function of 4 inputs
Major FPGA Vendors

SRAM-based FPGAs

- Xilinx, Inc.
- Altera Corp.
- Atmel
- Lattice Semiconductor

{ Share over 60% of the market }

Flash & antifuse FPGAs

- Actel Corp.
- Quick Logic Corp.
Xilinx

- Primary products: FPGAs and the associated CAD software

- Main headquarters in San Jose, CA

- Fabless* Semiconductor and Software Company
  - UMC (Taiwan) {*Xilinx acquired an equity stake in UMC in 1996}
  - Seiko Epson (Japan)
  - TSMC (Taiwan)
Xilinx FPGA Families

◆ Old families
  – XC3000, XC4000, XC5200
  – Old 0.5µm, 0.35µm and 0.25µm technology. Not recommended for modern designs.

◆ Low Cost Family
  – Spartan/XL – derived from XC4000
  – Spartan-II – derived from Virtex
  – Spartan-IIE – derived from Virtex-E
  – Spartan-3

◆ High-performance families
  – Virtex (0.22µm)
  – Virtex-E, Virtex-EM (0.18µm)
  – Virtex-II, Virtex-II PRO (0.13µm)
  – Virtex-4 (0.09µm)
Virtex-II Architecture

- Block RAMs
- Multipliers 18 x 18
- I/O Block
- Configurable Logic Block
Virtex-II PRO™ Architecture

I/O Block
User Logic
Interface Core
Block RAM
Multipliers

Source: [Cray, MAPLD04]
Virtex 4

Source: [Xilinx]
## Parameters of most powerful Virtex devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Max. clock frequency</th>
<th>Slices</th>
<th>Maximum I/O pins</th>
<th>BlockRAM</th>
<th>Multiplier Blocks (18x18)</th>
<th>Power PC cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC2V6000</td>
<td>420 MHz</td>
<td>33.7 K</td>
<td>1,104</td>
<td>2,592 Kb</td>
<td>144</td>
<td>-</td>
</tr>
<tr>
<td>XC2V8000</td>
<td>420 MHz</td>
<td>46.5 K</td>
<td>1,108</td>
<td>3,024 Kb</td>
<td>168</td>
<td>-</td>
</tr>
<tr>
<td>Virtex II Pro 125</td>
<td>420 MHz</td>
<td>55.6 K</td>
<td>1,200</td>
<td>10,000 Kb</td>
<td>444</td>
<td>4</td>
</tr>
<tr>
<td>Virtex 4 LX 200</td>
<td>500 MHz</td>
<td>89.1 K</td>
<td>960</td>
<td>6,048 Kb</td>
<td>96</td>
<td>-</td>
</tr>
<tr>
<td>Virtex 4 FX 140</td>
<td>500 MHz</td>
<td>63.2 K</td>
<td>896</td>
<td>9,936 Kb</td>
<td>192</td>
<td>2</td>
</tr>
</tbody>
</table>
The Design Cycle
(That we want you to avoid !)

Design and implement a simple encryption unit with RC5 cipher with fixed key

Specification

HDL (Hardware Description Language) model

Functional simulation

Synthesis

Post-synthesis simulation

netlist
The Design Cycle
(That we want you to avoid!)

Implementation
(Mapping, Placing & Routing)

Timing simulation

Downloading and Testing

On board testing
Mapping
Placing

FPGA

CLB SLICES
Routing

Programmable Connections

FPGA
General Architecture of an FPGA-Based Board
Reconfigurable Computing Boards
(Accelerators)

- Boards may have one or several interconnected FPGA chips
- Support different bus standards, e.g. PCI, PCI-X, VME
- May have direct real-time data I/O through a daughter board
- Boards may have local onboard memory (OBM) to handle large data while avoiding the system bus (e.g. PCI) bottleneck
Reconfigurable Computing Boards (Accelerators)

- Many boards per node can be supported
- Host program (e.g. C) to interface user (and µP) with board via a board API
- Driver API functions may include functionalities such as Reset, Open, Close, Set Clocks, DMA, Read, Write, Download Configurations, Interrupt, Readback
Some Reconfigurable Boards Vendors

- ANNAPOLIS MICRO SYSTEMS, INC. (www.annapmicro.com)
- University of Southern California -USC/ISI (http://www.east isi.edu).
- AMONTEC (www.amontec.com/chameleon.shtml)
- XESS Corporation (www.xess.com)
- CELOXICA (www.celoxica.com)
- CESYS (www.cesys.com)
- TRAQUAIR (www.traquair.com)
- SILICON SOFTWARE: (www.silicon-software.com)
- COMPAQ: (www.research.compaq.com/SRC/pamette/)
- ALPHA DATA: (www.alpha-data.com)
- Associated Professional Systems: (www.associatedpro.com)
- NALLATECH: (www.nallatech.com)
Representative Example
Boards
From Annapolis Micro Systems (AMI)
http://www.annapmicro.com
&
Nallatech
http://www.nallatech.com
WILDSTAR™ II Pro
Nallatech's BenNUEY-PCI-4E
Networks of Reconfigurable Computers (NORCs)
Networks of Reconfigurable Computers (NORCs)

- Expensive reconfigurable workstations can be at times underutilized
- Large problems may need to be spread over a number of workstations
- Many problems may need a high throughput environment
- So, need S/W system to remotely schedule and monitor reconfigurable tasks, send data and bit-streams, and collect results
Example : GWU/GMU Extended JMS

http://www.gwu.edu/~hpc/lsf/
http://ece.gmu.edu/lucite/

◆ Team from GWU and GMU with DoD support

◆ Considered extending job management systems (JMS’s) to recognize reconfigurable computing resources and support the needed functionalities

◆ Evaluated many implementations of JMS’s and selected LSF (Load Sharing Facility) for implementation
Networked Reconfigurable Resources Management System
Architecture of a typical Job Management System

- **User Server**: Jobs & their requirements
- **Resource Requirements**
- **Job Scheduler**: Scheduling Policies
- **Available Resources**: Resource Allocation and Job Execution
- **Resource Monitor**: Job Dispatcher
- **Resource Manager**
LSF

- LSF (Load Sharing Facility) is the product of Platform Computing
- LSF is a layer of software services on top of UNIX and Windows NT operating systems
- The LSF Suite is a set of software modules that manage distributed computing resources and workloads
- LSF creates a single system view on a network of heterogeneous computers so that the whole network of computing resources can be utilized effectively and managed easily
GWU/GMU NORCs Testbed Used in Experiments

Submission & Master Host

Linux RH7.0 – PIII
450 MHz, 512 MB RAM

Workloads included crypto analysis and image processing (edge detection)

Execution Host 1

HPCL 2

FIREBIRD V1000

SLAAC-IV

Execution Host 2

HPCL 3

WILDFORCE

LINUX 2.2.5 – PII 450 MHz, 256 MB RAM

Execution Host 3

HPCL 5

Windows 2000 – PII 400 MHz, 128 MB RAM

Execution Host 4

HPCL 6

FIREBIRD V2000

Windows XP – PIV 1.3 GHz, 256 MB RAM
Parallel DES Breaker

Ciphertext=0X 8CA64DE9C1B123A7  Plaintext=0X 0000000000000000

Search Space from 0X 1010100C5663702 to 10101013C9BCB00

key found = 0X 0101010101010101

Estimated speed-up over Pentium 4 > 500
Reconfigurable Computing Clusters
Remember Beowulf Clusters?

- “Do-It-Yourself Supercomputers” - Science 1996

- Built around:
  - Pile of PCs (POP)
  - Dedicated High Speed LAN
  - Free Unix: Linux
  - Free and COTS Parallel Programming and performance Tools

- COTS Hardware permits rapid development and technology tracking
Reconfigurable Computing
Clusters

- Beowulf style clusters
- COTS reconfigurable boards as accelerators at each node
- Some parallel programming and execution model/tool
Example 1: HPTi Solution
Delivered and Benchmarked
http://www.hpti.com/

Source: [HPTi, MAPLD04]
Delivered and Benchmarked

- 48 nodes
- 2u, back-to-back (net 1u/node)
- 96 FPGA’s
- Annapolis Micro
- Xilinx Virtex II
- 34 Tera-Ops
- *In use today*
- *All Commodity Parts*
Tower of Power

http://ccm.ece.vt.edu/

◆ 16-node cluster of PCs
◆ WILDFORCE board on each PC
◆ Myrinet network connecting all PCs
◆ Runs ACS API (platform independent API for the configuration and control of multi-board systems)

Source: [ACS01]
SLAAC RRP (Research Reference Platform)

Source: [ISI 01]
Massively Parallel Reconfigurable Systems
Massively Parallel Reconfigurable Systems

- Large numbers of reconfigurable processors and microprocessors
- Everything can be configured
  - Functional units
  - Interconnects
  - Interfaces
- High-level of scalability
- Suitable for a wide range of applications
- Everything can be reconfigured over and over at run time (Run-Time Reconfiguration) to suite underlying applications
- Can be easily programmed by application scientists, at least in the same way of programming conventional parallel computers
Vision for Reconfigurable Supercomputers

\[ \text{μP} \quad \text{FPGA} \quad \text{μP} \quad \text{FPGA} \]

\[ \text{μP memory} \quad \text{FPGA memory} \quad \text{μP memory} \quad \text{FPGA memory} \]

Shared Memory and or NIC
Possible Classes of Reconfigurable Supercomputers

Independent Board Design

Joint Board Design

Tighter Integration
Possible Classes of Reconfigurable Supercomputers – cont.

Joint μP/RP Board

Tightener Integration

μP inside of RP Design

RP inside of μP Design
Current Reconfigurable Architecture

Microprocessor system

μP

μP memory

I/O

Reconfigurable system

FPGA

FPGA memory

Interface

Interface

I/O
Example 1: SRC 6E System
http://www.srccomp.com/

See the company website and the presentation by SRC Computers for more details

Source: [SRC, MAPLD04]

55
Example 2: Cray XD1
(OctigaBay 12K)
http://www.cray.com

See the company website and the presentation by Cray, Inc. for more details

Source: [Cray, MAPLD04]
Example 3: SGI Altix
http://www.sgi.com/servers/altix/

See the company website and the presentation by SGI for more details

Source: [SGI, MAPLD04]
Example 4: The Starbridge Hybrid Computer 62m
http://www.starbridgesystems.com/

See the company website and the presentation by Star Bridge, Inc. for more details

Source: [SBS, MAPLD04]
How to choose the system that best suits your needs?

Typical users’ criteria:

1. Number of FPGAs & number of microprocessors
2. Clock speed
3. Amount of memory
4. Price
How to choose the system that best suits your needs?

Typical users’ criteria:

1. Number of FPGAs & number of microprocessors
2. Clock speed
3. Amount of memory
4. Price
How to choose the system that best suits your needs?

Recommended users’ criteria:

1. Tools
   - right level of abstraction
   - ease of development & verification
   - progress & backward compatibility

2. Libraries
   - basic operations
   - examples of full applications

3. Technical support
How to choose the system that best suits your needs?

Recommended users’ criteria (cont.):

4. I/O Bandwidth

\[ \text{Reconfigurable Processor System} \]

\[ \mu P \text{ system} \quad \rightarrow \quad \text{external I/O devices} \]
How to choose the system that best suits your needs?

Recommended users’ criteria (cont.):

5. Scalability

- variable power and price
- efficient communication among the modules
Recommended users’ criteria (cont.):

6. Transfer of control overhead

Theoretical behavior

Actual behavior

Control transfer overhead
6. Reconfiguration overhead

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Task A
  - Reconf C
    - Task C
  - 65

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf B
    - Task B
  - Task B

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf C
    - Task C
  - Task C

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf C
    - Task C
  - Task C

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf B
    - Task B
  - Task B

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf C
    - Task C
  - Task C

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf B
    - Task B
  - Task B

- μP FPGA
  - Reconf A
    - Task A
  - Task A
  - Reconf C
    - Task C
  - Task C
6. Reconfiguration overhead (cont.)

µP

FPGA 1

Task A

Reconf A

Task B

FPGA 2

Task C

Reconf C

Reconf B
Recommended users’ criteria (cont.):

7. Number of FPGAs & number of microprocessors

8. Clock speed
   - maximum
   - variable vs. fixed

9. Amount of memory
Acknowledgements

- AMI
- Cray
- DoD/LUCITE
- HPTi
- NASA/GSFC
- SGI
- SRC
- Star Bridge
- Devrim Fidanci
- Esam El-Araby
- Esmail Chitalwala
- Mohamed Taher
- Proshanta Saha