Run time adaptive processing units using coarse and fine grain reconfigurable fabric

Ricardo dos Santos Ferreira
ricardo@ufv.br

Sabbatical TU Delft in collaboration with Stephan Wong
Outline

Where is Vicosa?

Research Lines

Big Picture: Run Time

1. Coarse-Grained Architecture and Modulo Scheduling
   - SAMOS paper
   - Paper to submit

2. Placement and Routing for FPGA
   - FPL paper
Where in the world is Viçosa?

Luigi Carro - UFRGS
Brazil:

- area: 8,514,877 km² (5th)
- population: 192 million (5th)
- economy:
  - GDP: U$ 2,5 trillion (6th)
  - per capita: U$ 13,000
1700km Porto Alegre - UFRGS
Sustainability
- Quality and Sustainability
- The Four Pillars
- Supporting the World's Coffee Growing Communities
- 22nd Ernesto Illy Prize
- Good for coffee. Good for the environment.
- Testimonials: In the Growers' Own Words
- Certifications

22nd ERNESTO ILLY QUALITY AWARD FOR ESPRESSO COFFEE

The 22nd Ernesto Illy Quality Award for Espresso Coffee has a winner: Edio Anacleto Miranda, from Araponga (Minas Gerais).
Number of students: 20,517
- undergraduate: 14,720
- graduate: 2,791
- graduate non degree: 1,293
- high school: 1,713
Student Life $

- No tuition fees for exchange students
- Master and PhD students can apply for a regular Brazilian scholarship:
  - € 500.00 / month for Master
  - € 750.00 / month for PhD
- University restaurant meals:
  - < € 1 for undergraduate and graduate students
- Room in a student house:
  - around € 160.00 / month
Undergraduate

Agribusiness
Agronomy
Animal Science
Agricultural Economics
Agricultural Engineering
Environmental Engineering
Dairy Science
Forestry
Biochemistry
Biological Sciences
Medicine
Nursing
Nutrition
Physical Education
Veterinary Medicine

Architecture
Chemistry
Chemical Engineering
Civil Engineering
Computer Science
Electrical Engineering
Food Engineering
Mathematics
Physics
Production Engineering
Administration
Business Administration
Economics
Geography
History
Law....
Graduate Agricultural Sciences

Agriculture Biochemistry
Agricultural Engineering
Agricultural Meteorology
Agrochemistry
Agroecology
Agricultural Microbiology
Cellulose and Paper
Entomology
Forestry
Genetics and Breeding

Plant Health Protection
Plant Pathology
Plant Production
Rural Extension
Soils and Plant Nutrition
Graduate

Biological and Health Sciences

Microbiology
Animal Biology
Animal Science
Botany
Ecology
Cellular Biology
Nutrition
Physical Education
Plant Science
Veterinary Medicine
Graduate

Exact Sciences

Applied Physics
Applied Statistics
Architecture
Civil Engineering
Computer Science
Msc
Food Science
Mathematics
Research

* GPU/FPGA and Bioinformatics:
  Gene Regulatory Networks
* Interconnections:
  run time Multistage Networks
* **run time scheduling, placement and routing for CGRA**
* **run time FPGA place&route**
Goals.....

* algorithms for just-in-time analysis of binary code

* runtime scheduling, placement and routing

* reconfigurable fabric architectures and hardware support for runtime accelerator implementation
Big Picture

Software Apps

Per program phase; Minimum energy; Max performance

ISA

Accelerators:
- CGRA
- FPGA

Previous Work
Outline

Where is Vicosa?

Big Picture: Run Time

1. Coarse-Grained Architectures and Modulo Scheduling
   - SAMOS paper
   - Paper to submit

2. Placement and Routing for FPGA
   - FPL paper
A Just-In-Time Modulo Scheduling for Virtual Coarse-Grained Reconfigurable Architectures

Ricardo Ferreira, Vinicius Duarte, Waldir Meireles
Universidade Federal de Viçosa, Brazil
Email: ricardo@ufv.br

Monica Pereira, Luigi Carro, and Stephan Wong
UFRN, Brazil  UFRGS, Brazil  TU Delft
Netherlands
Stream Computing

Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone.
Modulo Scheduling = software Pipeline

At same time, all operations are executed.....

One Clock Cycle THROUGHPUT!
Scheduling and Mapping in a CGRA
Scheduling and Mapping in a CGRA
Dynamic Modulo Scheduling $\neq$ dynamic scheduling

Initial Interval = 2 cycles

6 Units, t3
Dynamic Modulo Scheduling ≠ dynamic scheduling
CGRA Architectures

- Option 1 – MESH
  - DRESC, EMS, EPIMAP, REGIMAP, others.....
  - Scheduling, Place, Route is NP-complete
  - Mesh is scalable

- Option 2 – Global Net
  - MSPR (previous work 2011)
  - Scheduling is NP-complete
  - Place & Route O(1)
Option 1 – MESH
- DRESC, EMS, EPIMAP, REGIMAP, others....
- Scheduling, Place, Route is NP-complete
- Mesh is scalable

Option 2 – Global Net
- MSPR
- Scheduling is NP-complete
- Place & Route O(1)

- Runtime
- 16 Units
  - It is enough for more than 90% Loops
  - cost(Global) is similar to cost(Mesh)
Crossbar – Area $O(n^2)$

• Contribution

Cost of MESH ADRES is similar to CGRA Crossbar for 16 Functional Units!

Virtual CGRA on the top of Commercial FPGA XILINX XC6VLX75T

<table>
<thead>
<tr>
<th>Component</th>
<th>FlipFlop</th>
<th>LUTs</th>
<th>Mem Bank</th>
<th>Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xilinx XC6VLX75T</td>
<td>2.7 %</td>
<td>17.6 %</td>
<td>4.5 %</td>
<td>90 Mhz</td>
</tr>
<tr>
<td>Xilinx XC6VLX75T</td>
<td>2.5 %</td>
<td>14.7 %</td>
<td>16.0 %</td>
<td>110 Mhz</td>
</tr>
</tbody>
</table>
Scheduling Quality

- **Our Approach**

- **Minimal (optimal) Initial Interval**
Our Approach

Minimal (optimal) Initial Interval

Optimal 10 / 15
Instruction Level Parallelism
Instruction Level Parallelism

Dataflow 40 ops

40 / 16 $\rightarrow$ II=3

40/3 $\rightarrow$ 13.3

16 Units

9.3  8.0  8.5  10  7.3  11.5  12.3  10.2  8.8  11.4  11.8  13.2  10.1  13.1  10.6

582

11

10.1

10.6

35

126
Our Approach

At least 1000x faster!

→ Run time !!!!!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>clk</th>
<th>Graph</th>
<th>Node</th>
<th>Cycles</th>
<th>Reduction Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRESC</td>
<td>2.66</td>
<td>104</td>
<td>0.73</td>
<td>2.0 $10^9$</td>
<td>6.8 $10^6$</td>
</tr>
<tr>
<td>EPIMAP</td>
<td>2.66</td>
<td>30</td>
<td>0.17</td>
<td>4.5 $10^8$</td>
<td>1.6 $10^6$</td>
</tr>
<tr>
<td>RF</td>
<td>1.0</td>
<td>110</td>
<td>0.5</td>
<td>4.9 $10^8$</td>
<td>1.7 $10^6$</td>
</tr>
<tr>
<td>EMS</td>
<td>2.66</td>
<td>5.6</td>
<td>0.04</td>
<td>1.04 $10^8$</td>
<td>3.6 $10^5$</td>
</tr>
<tr>
<td>Gminor</td>
<td>2.66</td>
<td>3.4</td>
<td>0.04</td>
<td>1.04 $10^8$</td>
<td>3.6 $10^5$</td>
</tr>
<tr>
<td>RAM</td>
<td>2.66</td>
<td>3.9</td>
<td>0.01</td>
<td>2.7 $10^7$</td>
<td>9.3 $10^4$</td>
</tr>
<tr>
<td>MSPR</td>
<td>2.66</td>
<td>0.09</td>
<td>0.0002</td>
<td>4.6 $10^5$</td>
<td>1.6 $10^3$</td>
</tr>
</tbody>
</table>

Compile time
On Going Work

Previous work

Compiler → **Dataflow Graph** → Run time Modulo Scheduling → CGRA

Domain specific language
Stream language

Current work

Vex Binary Code → Detect and Translate The inner loops → CGRA Accelerator
Binary → Translate → Configuration Memory

# load/Store
# mults
# units
Only 1 ld/st per cycle?
Only 1 ld/st per cycle?
Outline

Where is Vicosa?

Big Picture: Run Time

1. Coarse-Grained Architectures and Modulo Scheduling
   - SAMOS paper
   - Paper to submit

2. Placement and Routing for FPGA
   - Winner for the M. Servit Award Fpl2013
FPL 2013

A RUN-TIME GRAPH-BASED POLYNOMIAL PLACEMENT AND ROUTING ALGORITHM FOR VIRTUAL FPGAS

Ricardo Ferreira* L. Rocha, A. Santos, J. Nacif – UFV Brazil
Stephan Wong - *TU Delft - Netherlands
Luigi Carro – UFRGS - Brazil
Motivation

- Run-time Place & Routing
  - Dynamic Behavior (Multitask Applications)
    - Share resources and increase the resource utilization
    - Previous Works
      - Configuration Bit stream were pre-computed at design time
  - Faults
  - Fragmented Regions
Context for Run-time

- Medium Complexity Modules
- Sequential x Parallel
  - Acceleration Parallel Part → Spatial and Temporal

- Bitstream Size (on-the-fly)
  - No need to store detail info (switches, wires)
Depth-First Traversal

- Previous Work – CGRA [Ferreira & Cardoso, 07]

Simple
Application is a graph
Target Architecture is a graph...

Depth-first in both GRAPHS
Depth-First Traversal

- Previous Work – CGRA [Ferreira&Cardoso,07]

Flexible
- Used area
Depth-First Traversal

Challenges for FPGA

- Quaternary Operation (4 in LUTs)
- Average Fanout = 3
- 35% of the edges could be local
- More complex target architecture:

LUT + local Wires + Switch Boxes
Graph model for the FPGA

Three Node types: LUT, Wires and switches
Nodes ID and Implicit Connections

![Diagram showing nodes ID and implicit connections with labels and connections.]
Cost = 1 and Cost = 2
Adjacent Nodes
Adjacent Nodes
Adjacent Nodes
Adjacent Nodes

![Diagram of adjacent nodes with numbers and labels]
Non Adjacent Nodes
Non Adjacent Nodes
Algorithm

For each edge $b \rightarrow a$ in DFS
    If $b$ is already placed, insert $a$ in Fanout list of $b$
    else If PLACE $b$ in Adj($a$)  // low cost & route
    else if PLACE $b$ in Non-Adj($a$)  // route
End FOR
For each Fanout list L
    Route L
end
Algorithm

For each edge \( b \rightarrow a \) in DFS
  If \( b \) is already placed, insert \( a \) in Fanout list of \( b \)
  else If PLACE \( b \) in Adj(\( a \))  // low cost & route
      else if PLACE \( b \) in Non-Adj(\( a \))  // route
Algorithm

For each edge $b \rightarrow a$ in DFS

If $b$ is already placed, insert $a$ in Fanout list of $b$
else if PLACE $b$ in Adj($a$) // low cost & route
else if PLACE $b$ in Non-Adj($a$) // route

End FOR

For each Fanout list L

Route L

end
Execution Time P&R

400 to 1500 times faster

VPR proposed approach
Placement Comparison

- Proposed Placement
  - VPR Fast Place
    - VPR Routing
  - VPR Critical Path
    - VPR Routing

- Circuit
  - Bounding Box
    - Timing Drive
Placement Execution Time

Proposed Placement

1024× faster

VPR Fast Place

13215× faster

VPR Critical Path

0.3 to 2.2 msec
Critical Path after VPR routing

Proposed Placement = VPR Fast Place

Proposed Placement

VPR Critical Path

26% worst

+ wires
More Wires...

- Proposed Placement: 46% worst
  - VPR Fast Place
- Proposed Placement: 46% worst
  - VPR Critical Path
Scenario 1:

- Apex2: 7% wires, 43% crit path
- Ex1010: 11% wires, 79% crit path
- Apex4: 24% wires, 42% crit path
- Too_large: 20% wires, 4% crit path
Scenario 2

- 23% wires, 48% crit path
- 54% wires, 35% crit path
- 42% wires, 69% crit path
- 12% wires, 19% crit path
About P&R for FPGA

• P&R is NP-complete but.....
  – Digital circuits are not random graphs....
  – Fanout=3, .....  
• It is possible to be three orders of magnitude faster than VPR
• Graph Model + Adjacent Node
• Greedy Approach
  – Just visit each node once based on the locality
• Optimize the Critical Path with greedy Place is possible
• Run-Time P&R is possible 1ms for 3000 LUTs!
• Flexible and manage fragmented regions
Future works for Run time P&R

- Explore others proprieties
- Domain specific applications
  - Accelerators for hardware models in Bioinformatics and Data mining
- Dynamic Datapaths
- Routing should be improved....
Conclusions
Adaptive platform

Programming Language

Source Code

Maintain SW abstraction
Adapt SW and HW
as a function of the QoS
as a function of required FT
Address
ricardo@ufv.br
 cacauvicosa@gmail.com