FPGA: Real needs and limits

Politecnico di Milano
Conference Room, Bld 20
19 November, 2015

Antonio R. Miele
Marco D. Santambrogio
Politecnico di Milano
Motivations

• Reconfigurable systems, while providing new interesting features in the field of hardware/software co-design, and more in general in the embedded system design, also introduce new problems in their implementation and management.

• This is particularly true for systems that implement self partial reconfiguration, such as Xilinx platforms.

• This talk will present the different scenarios (i.e. flexibility, resource lack...) where the reconfiguration can be effective showing also the drawbacks introduced by this new feature.

• We will show the presence of two different kinds of limits, theoretical and physical ones, trying to highlight possible solutions to both of these.
Outline

• Some real needs

• Limits and drawbacks
What’s next

• Some real needs
  – Behavioral and structural flexibility
  – Performance enhancement
  – Fault tolerance

• Limits and drawbacks
Behavioral and Structural flexibility

• Speedup the overall computation of the final system

• Increasing need for behavioral flexibility in embedded systems design
  – Support of new standards, e.g. in media processing
  – Addition of new features

• New applications too large to fit on the device all at once
HW vs SW

Feasible Solution Space

Problem: $A_d < A_{de}$
HW vs SW

Time

Shₜ

Shᵈₑ

Shᵥᵈ

Shₒ

Area

Aₜ

Aᵈ

Aᵈₑ

Aᵥᵈ

Aₒ

Feasible Solution Space
Digital Image Processing

• The canny edge detector is used to detect the edges in a given input image $i$ [Kb]

• 4 functionalities
  – Image smoothing $\rightarrow$ remove the noise
  – Gradient operator $\rightarrow$ highlight regions with high spatial derivative
  – Non-maximum suppression $\rightarrow$ reveal the edges
  – Hysteresis $\rightarrow$ remove false edges

• Each functionality has to be executed using an input of $j$ [Kb]
  – $j \leq i$ and $x = i/j$

• Time analysis to identify a first partition in HW core and SW core
  – Non-maximum suppression implemented as a SW core
  – Image smoothing, Gradient operator and Hysteresis implemented as HW cores
DIP: Partial Reconfiguration

- Static side and IP-Cores resources requirement analysis

<table>
<thead>
<tr>
<th>Application Functions</th>
<th>Occupied Slices</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Side, non-maximum suppression $f_c$</td>
<td>2662</td>
<td>54</td>
</tr>
<tr>
<td>image smoothing (FIR) $f_a$</td>
<td>245</td>
<td>4</td>
</tr>
<tr>
<td>gradient $f_b$</td>
<td>2168</td>
<td>44</td>
</tr>
<tr>
<td>hysteresis $f_d$</td>
<td>5343</td>
<td>108</td>
</tr>
</tbody>
</table>

- Time analysis, resources requirement and reconfigurability evaluation
  - Hysteresis implemented as a SW core
  - Image smoothing and Gradient operator implemented as reconfigurable cores

- Reconfiguration time ($f_a$ into $f_b$): 368ms
SRAM-based FPGAs are particularly sensible to radiation effects not only in critical environment, but also at terrestrial level

- alpha particles hitting devices cause temporary and permanent faults
- temporary faults can be modeled as
  - Modification in the data being processed
  - user-memory corruption
  - Modification of the functionality being performed
  - configuration-memory corruption

Embedded systems implemented on FPGAs need “robustness” to radiations, achieved by means of by-design fault tolerance
What’s next

• Some real needs

• Limits and drawbacks
  – Simulation and verification
  – Design flow support
  – Reconfiguration time overhead
Limits and Drawbacks

• Simulation and Verification

• Design flow: The need of a comprehensive framework which can guide designers through the whole implementation process is becoming stronger

• Reconfiguration times impact heavily on the final solution’s latency
Simulation and Verification

• A new way of intending simulation
  – Simulation used to explore the design space to find the best architectural solution

• Support to HW/SW codesing solutions but no standard ways to verify the overall (reconfigurable) design
  – Unfocused tools for the verification of all the reconfiguration related aspects
    • Xilinx Chipscope
    • Jbits (no longer supported)
Design flow

• Dynamic reconfigurable embedded systems are gathering, an increasing interest from both the scientific and the industrial world
  – The need of a comprehensive framework which can guide designers through the whole implementation process is becoming stronger

• There are several techniques to exploit partial reconfiguration, but...
  – Few approaches for frameworks and tools to design dynamically reconfigurable systems
    • They don’t take into consideration both the HW and the SW side of the final architecture
    • They are not able to support different devices
    • They cannot be used to design systems for different architectural solution
Reconfiguration challenges

• Reconfiguration times heavily impact on the final solution’s latency
  – Hiding reconfiguration time is not sufficient

• Possible solution:
  – Trivial
    • Bitstream dimension reduction
  – Complex
    • Maximize the reuse of configured modules
    • Reconfiguration hiding
    • Alternative implementation (SW execution)
    • Relocation
Tasks reuse

• Reconfiguration times impact heavily on the final solution’s latency, therefore:
  – Not only try to hide the reconfigurations
  – But try to maximize the reuse of reconfigurable modules

Schedule length is on average at least 18.6% better than the shortest one and 19.7% better than the average.
Reconfiguration hiding

Area/Time

2/1

A

2/2

C

2/2

F

1/2

B

1/1

D

1/1

E

2/2


Area

A

B

D

C

E

F

Reconf

Time
Reconfiguration hiding
Alternative implementation (SW execution)

- Object code implemented as hardware components do not always guarantee the best performance…
- Cryptography architecture
  - 1 GPP running Linux
  - 2 reconfigurable regions
  - 2 cryptography services (AES and DES)

<table>
<thead>
<tr>
<th>Input blocks</th>
<th>AES software throughput (kByte/s)</th>
<th>AES hardware throughput (kByte/s)</th>
<th>DES software throughput (kByte/s)</th>
<th>DES hardware throughput (kByte/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4847</td>
<td>0.0802</td>
<td>0.2574</td>
<td>0.0400</td>
</tr>
<tr>
<td>5</td>
<td>1.5671</td>
<td>0.4021</td>
<td>0.9736</td>
<td>0.2040</td>
</tr>
<tr>
<td>10</td>
<td>2.1613</td>
<td>0.8093</td>
<td>1.4985</td>
<td>0.4010</td>
</tr>
<tr>
<td>50</td>
<td>3.2479</td>
<td>3.9752</td>
<td>2.7554</td>
<td>2.0165</td>
</tr>
<tr>
<td>100</td>
<td>3.4954</td>
<td>8.0788</td>
<td>3.3482</td>
<td>4.0574</td>
</tr>
<tr>
<td>500</td>
<td>3.6717</td>
<td>39.2262</td>
<td>3.6307</td>
<td>20.2452</td>
</tr>
<tr>
<td>1000</td>
<td>3.6920</td>
<td>76.5226</td>
<td>3.7416</td>
<td>40.0855</td>
</tr>
</tbody>
</table>
Relocation: The Problem

People Demanding for Functionalities
Relocation: The Problem

People Demanding for Functionalities

Set of Available Functionalities

Legenda:
Area/Time $F_i$

FPGA

RR1  RR2  RR3
Relocation: The Problem

People Demanding for Functionalities

Set of Available Functionalities

F 2/2  B 1/2
2/1  A 2/2  C 1/1
D 1/1  E 1/1

Legenda:
Area/Time $F_i$

FPGA

RR1 RR2 RR3

RFU Implementations

A
RR1 RR2 RR3
RR1 RR2 RR3

B
RR1 RR2 RR3
RR1 RR2 RR3

C
RR1 RR2 RR3
RR1 RR2 RR3

D
RR1 RR2 RR3
RR1 RR2 RR3

E
RR1 RR2 RR3
RR1 RR2 RR3

F
RR1 RR2 RR3
RR1 RR2 RR3
Relocation: Scenario

A possible scenario

Time

RFU Implementations

Area

Legenda: Area/Time $F_i$
Relocation: Motivation

A possible scenario

Legenda:
Area/Time $F_i$
Relocation: Motivation

A possible scenario

Time

Legenda:

Area/Time

RFU Implementations

Time
Relocation: Motivation

A possible scenario

Time

Legenda:
Area/Time $F_i$

RFU Implementations

A
B
C
D
E
F

Area

Time
Relocation: Rationale

- Bitstreams relocation technique to:
  - speedup the overall system execution
  - reduce the amount of memory used to store partial bitstreams
  - achieve a core preemptive execution
  - assign at runtime the bitstreams placement

<table>
<thead>
<tr>
<th>Slots</th>
<th>Modules</th>
<th>Bitstreams</th>
<th>Bitstreams with reloc.</th>
<th>% Memory saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>6</td>
<td>50,0%</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>27</td>
<td>9</td>
<td>75,0%</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>55</td>
<td>11</td>
<td>80,0%</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>136</td>
<td>17</td>
<td>87,5%</td>
</tr>
</tbody>
</table>
Relocation: Virtual homogeneity
BiRF - Relocation management

- Create an integrated HW/SW system to manage relocation (1D and 2D) in reconfigurable architecture
  - Maintain information on FPGA status
  - Decide how to efficiently allocate tasks
  - Provide support for effective task allocation
  - Perform bitstream relocation

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Slices Used (#)</th>
<th>%</th>
<th>LUT Used (#)</th>
<th>%</th>
<th>BUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>xc4vlx40</td>
<td>Core 89 over 18432 0.5</td>
<td>161 over 36864 0.4</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 168 over 18432 0.9</td>
<td>213 over 36864 0.6</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xc4vlx60</td>
<td>Core 89 over 26624 0.3</td>
<td>161 over 53248 0.3</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 168 over 26624 0.6</td>
<td>213 over 53248 0.4</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xc4vlx100</td>
<td>Core 89 over 49152 0.2</td>
<td>161 over 98304 0.1</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 168 over 49152 0.3</td>
<td>213 over 98304 0.2</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xc4vlx40</td>
<td>Core 89 over 18432 0.5</td>
<td>161 over 36864 0.4</td>
<td>PLB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 157 over 18432 0.9</td>
<td>208 over 36864 0.6</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xc4vlx60</td>
<td>Core 89 over 26624 0.3</td>
<td>161 over 53248 0.3</td>
<td>PLB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 157 over 26624 0.6</td>
<td>208 over 53248 0.4</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xc4vlx100</td>
<td>Core 89 over 49152 0.2</td>
<td>161 over 98304 0.1</td>
<td>PLB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core 157 over 49152 0.3</td>
<td>208 over 98304 0.2</td>
<td>OPB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions...

FPGA: Real needs and limits

Politecnico di Milano
Conference Room, Bld 20
19 November, 2015

Antonio R. Miele
Marco D. Santambrogio
Politecnico di Milano