A System-Level Design Methodology for Reconfigurable Computing Applications

Esam El-Araby\textsuperscript{1}, Tarek El-Ghazawi\textsuperscript{1}, and Kris Gaj\textsuperscript{2}
\textsuperscript{1}The George Washington University, \textsuperscript{2}George Mason University, esam@gwu.edu, tarek@gwu.edu, kgaj@gmu.edu

Abstract

Reconfigurable Computers (RCs) can leverage the synergism between conventional processors and FPGAs by combining the flexibility of traditional microprocessors with the parallelism of hardware and reconfigurability of FPGAs. However, there exist multiple challenges that must be resolved to be able to develop efficient applications for reconfigurable computing systems. One of these challenges is the lack of formal modeling/design methodology. This work, addresses the need for a formal design methodology. In particular, a structured design life-cycle, which can be applied at the reconfigurable computing systems rather than at the chip-level as it has been traditionally the case, is conceptually proposed, formally modeled and experimentally verified.

1. Introduction

In order to optimize the design decisions for a reconfigurable computing environment, designers must be fully aware of the workload requirements and constraints of their applications as well as the physical constraints imposed by the underlying reconfigurable architecture. Modeling and quantifying the workload characteristics and needs of applications can result in optimal design decisions. In addition, designing applications by considering the physical machine-dependent parameters in a formal manner would result in optimal performance of these machines. Therefore, this paper aims to achieve the following objectives:

- Abstract and formulate a parameterized model for a generalized reconfigurable computer system
- Define the system parameters based on both the application and machine constraints
- Analyze the interactions among these parameters to set the stage for exploring the design space and its trade-offs
- Propose a system-level methodology for designing applications for reconfigurable platforms
- Demonstrate the applicability and the effectiveness of the proposed concepts

2. Design Methodology

The basic concept behind the proposed method is to leverage concepts such as superpipelining and superscaling \cite{1, 2} and use the application to create an optimal virtual architecture to which the application can be optimally mapped. The optimal virtual architecture is defined as one that does not impose any restrictions on the throughput of the application other than those imposed already by the critical path and degree of parallelism in the application. Once such optimal architecture is created, it is then mapped to the target platform. The optimal architecture is not necessarily unique and the mapping process is subject to the constraints of the target architecture, thus, produces a sub-optimal solution. Furthermore, the application resource requirements may lead to non-realistic solutions. Therefore, considering some of the constraints of the target architecture, namely the available resources, early during the step of creating the optimal architecture can lead to a more realistic architecture. Although this may sacrifice the optimality of solution, it would result in more realistic mapping of the virtual architecture to the target platform.

Based on this, the following steps for the design life-cycle of applications on reconfigurable computing architectures are proposed:

1) Provide basic problem description
2) Develop the application virtual architecture
   i. Construct a DFG for the underlying application
   ii. Provide time alignment to construct a basic linear pipeline
   iii. Recursively breakdown DFG to develop a virtual superpipeline
   iv. Determine the virtual architecture superscalar degree based on:
      - Real-Time throughput constraint
      - Resources usage constraint
3) Map the virtual architecture onto the target architecture
   i. Exploit maximum concurrency through I/O and computational overlapping
   ii. Determine the system minimum partitionability

3. Case Study

To demonstrate the applicability and the effectiveness of the proposed concepts we apply them to representative real-life workloads from cryptography and NASA hyperspectral imagery, i.e. RC5 \cite{3}, IDEA \cite{4}, and wavelet hyperspectral dimension reduction \cite{5, 6} respectively. In addition, we consider two versions of a contemporary reconfigurable computer, i.e. both Pentium 3 and Pentium 4 flavors of RC-6E \cite{7}. 

0-7803-9407-0/05/$20.00 © 2005 IEEE
Table 1. Experimental Results on SRC-6E (P4 version)

<table>
<thead>
<tr>
<th>Application</th>
<th>SRC-6E (P4 Version)</th>
<th>databases</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet Hyperspectral Dimension Reduction</td>
<td>Before: 6,394</td>
<td>6,349</td>
<td>3,185</td>
</tr>
<tr>
<td></td>
<td>After: 6,394</td>
<td>6,349</td>
<td>3,185</td>
</tr>
<tr>
<td>RC5 Encryption</td>
<td>Before: 6,308</td>
<td>6,371</td>
<td>6,373</td>
</tr>
<tr>
<td></td>
<td>After: 6,308</td>
<td>6,371</td>
<td>6,340</td>
</tr>
<tr>
<td>IDEA Encryption</td>
<td>Before: 12,700</td>
<td>10,027</td>
<td>10,583</td>
</tr>
<tr>
<td></td>
<td>After: 10,837</td>
<td>9,792</td>
<td>10,564</td>
</tr>
</tbody>
</table>

4. Experimental Work and Observations

Table 1 shows the experimental results of applying our methodology/technique to the implementation of the three applications, wavelet- based dimension reduction, RC5, and IDEA, on SRC-6E (P4). The results shown are for the two cases before and after the applications have been optimized using overlapping (streaming) techniques. The speedup figures are with reference to implementations on a conventional PC configured as: Pentium 4, 2.4 GHz, 512 KB of cache, and 512 MB of RAM. Table 2. shows the accuracy of our model expectations as compared to the actual experimental improvements.

Observations

The applications performance is bounded by the limitations of the underlying platform I/O. By improving the I/O on SRC-6E systems by a factor of 2.61 for the P4 version over the P3 version, the performance, using the proposed technique, has improved by almost an identical factor. Our experiments showed that this factor is practically 2.42 for a single superscalar implementation, e.g. wavelet dimension reduction, while a practical value of 4.75 has been measured for two superscalar implementations for IDEA. The experimental measurements for all the applications considered on both machines of SRC-6E have shown approximately an order of magnitude improvement over implementations on conventional platforms. These results confirm the effectiveness and the applicability of the proposed methodology.

5. Conclusions

The work in this paper, in contrast to the conventional ad hoc process, has introduced a formal/quantitative system-level methodology suitable for design of applications for reconfigurable platforms. The basic concept behind the proposed methodology is to use the application to create an optimal virtual architecture to which the application can be optimally mapped. Once such an optimal architecture is created, it is then mapped to the target platform. This mapping depends on overlapping the computations on the User FPGAs with the I/O transfer. This overlapping requires dividing data transfers into multiple transfer parcels that can be overlapped with partial computations.

The applicability and the effectiveness of the proposed concepts were demonstrated by applying them to NASA hyperspectral imagery and cryptography algorithms. The presented methodology has been experimentally verified using the P3/P4 SRC-6E architectures. Both theoretical analysis and experimental results proved that this technique is efficient in speeding up the execution time when compared to corresponding implementations on traditional microprocessors.

References