Algorithm Analysis and Mapping Environment for Adaptive Computing Systems

Eric Pauer, Cory Myers, Ken Smith, and Paul Fiore
(pauer,cory,jmsmith,pfiore)@sand.com
Sand, a Lockheed Martin Company
Nashua, NH 03061

Statement of the Problem

Reconfigurable computing technology offers leap ahead performance, e.g. 10X ops per watt and/or ops per cubic inch, over general purpose programmable solutions without the need to develop custom hardware. However, today generation of a working implementation requires hardware design expertise and generation of a good implementation requires many slow iterations between an algorithm designer and a hardware developer.
Adaptive Computing Performance Gain

<table>
<thead>
<tr>
<th></th>
<th>CHAMP</th>
<th>TMS 320C28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>256 x 256</td>
<td>256 x 256</td>
</tr>
<tr>
<td>Implementation Time</td>
<td>44 Days</td>
<td>28 Days</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>305 frames/sec</td>
<td>12 frames/sec</td>
</tr>
<tr>
<td>Latency</td>
<td>68 usec</td>
<td>82,000 usec</td>
</tr>
<tr>
<td>Processing Load</td>
<td>4.7 Bops</td>
<td>0.2 Bops</td>
</tr>
<tr>
<td>Utilization</td>
<td>73%</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
| Gates                | 510k          | N/A            

(Greater than 10X performance
Design time measured in weeks)

Reconfigurable Architectures

Analysis:
- Bit Widths
- Latency (L)
- Cells Used (C)

Analysis and Mapping in ACS Environment

- SNR analysis
- Alternative implementations
- Functional approximations
- Timing and sizing estimation
- Scheduling – FSM and contexts
- Partitioning within a resource node
- Device program
- Interface program

Adaptive Computing System Design and Implementation using the ACS Domain
Automated Float to Fixed Point Translation

- Optimize wordlengths
- Fixed point realization
- Yields:
  - Bit widths for each flow
  - Cost estimates (area/complexity)
  - Quantization noise (SNR)

\[
\begin{align*}
\text{Maximize } & \text{SNR}(b), \\
\text{subject to } & \text{Cost}(b) \leq C_0 \text{ and } b \geq b_{\min}, \\
\text{Minimize } & \text{Cost}(b), \\
\text{subject to } & \text{SNR}(b) \geq \text{SNR}_0 \text{ and } b \geq b_{\min}.
\end{align*}
\]

Ptolemy and the ACS Domain

- Ptolemy - simulation/design environment from the University of California, Berkeley (http://ptolemy.eecs.berkeley.edu)
- New ACS domain developed to facilitate movement among simulation and code/design generation (released in 0.7.1, 6/98)
- ACS Stars (basic building block) are composed of a Corona (interface) and multiple cores (implementations)
- Core (implementation) selection is via targeting mechanism

Available cores/targets

- Floating point simulation core
- Fixed point simulation core
- C code generation core
- FPGA design generation core
- FPGA Java generation core

Available cores/targets

Retargetable implementations

UCB BRASS Project
Top Level Example

- FIR filter to be implemented in both floating point and fixed point simulation

Selecting Among Alternative Implementations

- Alternative implementations are represented as “targets”
- Targets can have parameters
- Floating point simulation, fixed point simulation, and C code generation are integrated today. FPGA generation almost ready.
Comparing Implementations

- Comparison of floating point and fixed point implementations

Winograd-based FSK Receiver
ACS Domain - CGFPGA Target

Winograd dataflow (ACS domain)

CGFPGA target yields:
VHDL design and schedule

Dataflow/Hardware schedule

The results are sent to synthesis
and place/route, yielding complete FPGA implementation!

Winograd schedule
FPGA-based Winograd DFT

- Address generator
- Word counter
- Data multiplexer
- Sequencer/State Machine

Winograd DFT computations

Xilinx 4062XL

Hardware-in-the-loop

SDF Galaxy

SDF Wildforce star executes complete FPGA design in hardware on Annapolis Wildforce FPGA board
Related ACS Work at Sanders

- Context Switching
- Reconfigurable Computing
- Reconfigurable Algorithms for Adaptive Computing
- Algorithm Analysis and Mapping Environment for Adaptive Computing Systems
- Adaptive Computing Smart Modules
- Efficient Mathematical Algorithms for Image Processing Applications
- Advanced Sensors Federated Laboratory

Adaptive Computing System Design and Implementation using the ACS Domain