FPGA-based High-Level Design Strategies to Accelerate a 3D Path Planning Algorithm

João Diogo Teixeira (ee06128@fe.up.pt), Ali Azarian (azarian@fe.up.pt), João M. P. Cardoso (jmpc@fe.up.pt), José Carlos Alves (jca@fe.up.pt)

1 Faculty of Engineering, University of Porto, Portugal
2 INESC TEC (formerly INESC Porto) and Faculty of Engineering, University of Porto, Portugal

Abstract
This work presents the results of applying diverse high-level optimization strategies for FPGA hardware acceleration of a computational intensive application, using a commercial C to RTL high-level synthesis tool (CatapultC). The application was provided by an industry partner and is the critical part of a real-time path planning algorithm for unmanned aerial vehicles operating on three dimensional environments with dynamic obstacles.

In this work we studied various acceleration strategies for guiding the high-level synthesis process, applied to plain C code. These include loop transformations, data reorganization, SW/HW control and data synchronization, HW/SW shared memory organization, HW private memory organization, and HW specialization for different calling parameters of the same function. The effectiveness of dynamic reconfiguration was also analysed for using instance specific specializations of the HW core. Special attention was devoted to strategies allowing an efficient utilization of the available embedded memory blocks.

This study was done for a high-performance embedded computing architecture based on a Virtex5 XILINX FPGA device with a PowerPC 440 core attached to a custom computing unit built through the high-level synthesis flow. The EDK/ISE implementation tools were used. Results show effective speedups for the whole application of more than 6x, comparing the HW/SW solution with the original SW solution compiled with maximum optimization effort.
A fixed-point implementation of this function was selected for hardware acceleration.

Input: two 3D matrices, output: one 3D matrix

In each iteration this function is called 12x with three different fixed-size matrices:

- Large obstacle map: 32x64x16 (32K elements)
- Medium obstacle map: 16x32x8 (4K elements)
- Small obstacle map: 8x16x4 (512 elements)

**Objectives**

- To evaluate the effectiveness of hardware acceleration of a plain C code application using a commercial high-level synthesis tool
- To study the impact of various high-level optimization strategies in the hardware performance of the custom accelerator
- To evaluate the overhead of data transfer between the host processor and the custom core

**3D Path Planning Application**

Computes in real time a 3D trajectory for a UAV in an environment with known obstacles.

- The function `griditerate` is responsible for 90% of the original software application execution time

**Design Methodology and Tools**

- Embedded system design built with ISE/EDK 12.3 for a Virtex 5 FX70 FPGA (ML507 board, embedded PowerPC @400MHz)
- Original application profiled with gprof embedded into EDK tools to identify the critical function for hardware acceleration
- High-level synthesis with CatapultC
- RTL synthesis with Precision Synthesis
- Hardware template based on the EDK generated general purpose peripheral, using shared memory blocks.

**Optimization Strategies**

High-Level Synthesis of the `griditerate` function considering:

- original C code, hardware core only used for the large map
- external 12x loop moved into the `griditerate` function
- input matrix `pot[]` replicated into 3 memories
- three specialized cores, 3 memories, inner loop unrolled 2x

**Experimental Results**

Global speedups (software compiled with -O2)

<table>
<thead>
<tr>
<th></th>
<th>1.00</th>
<th>1.94</th>
<th>5.94</th>
<th>6.68</th>
<th>6.72</th>
<th>0.00</th>
<th>1.00</th>
<th>2.00</th>
<th>3.00</th>
<th>4.00</th>
<th>5.00</th>
<th>6.00</th>
<th>7.00</th>
<th>8.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW a)</td>
<td>0%</td>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
<td>50%</td>
<td>60%</td>
<td>70%</td>
<td>80%</td>
<td>90%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HW execution time (including data transfer)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data transfer overhead

<table>
<thead>
<tr>
<th></th>
<th>00.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large map</td>
<td>a)</td>
<td>b)</td>
<td>c)</td>
<td>d)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium map</td>
<td>a)</td>
<td>b)</td>
<td>c)</td>
<td>d)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small map</td>
<td>a)</td>
<td>b)</td>
<td>c)</td>
<td>d)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FPGA resource utilization (Virtex5FX70)

<table>
<thead>
<tr>
<th></th>
<th>DSP48</th>
<th>34k BRAM</th>
<th>LUTs</th>
<th>Flip-flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td>2470 (5.5%)</td>
<td>956 (2.1%)</td>
<td>939 (2.1%)</td>
<td>901 (2.0%)</td>
</tr>
<tr>
<td>HW</td>
<td>1182 (2.6%)</td>
<td>36k BRAM</td>
<td>34 (23%)</td>
<td>34 (23%)</td>
</tr>
</tbody>
</table>

**Conclusions**

- Effective use of high-level synthesis for creating a custom processing core synthesized from the original C code.
- Achieved important speedups for the whole application of more than 6x, for a FPGA device with an embedded PowerPC 440@400MHz, with the custom core running at 100 MHz.
- Interface between the processor and core implemented with shared dual-port memories.
- Important performance bottleneck due to the data transfer between main processor and the custom core.
- Future developments
  - evaluate the performance with floating-point operators synthesized from C functions
  - move additional processing stages to the custom processing core to reduce the amount of data transferred between processor and core

The authors acknowledge the support provided by the European Community’s Framework Programme 7 (FP7) under contract No. 248976. The authors also acknowledge the support of Zlatko Petrov and Kamil Kratky from Honeywell regarding the 3D Path Planning application.