Evaluation of a Locomotion Algorithm for Worm-like Robots on FPGA-embedded processors

By Juan Gonzalez-Gomez, Ivan Gonzalez, Francisco Gomez-Airribas and Eduardo Boemo
Computer Engineering School, Universidad Autonoma de Madrid (Spain)
E-mail: Juan.Gonzalez@uam.es

Abstract
In this paper, a locomotion algorithm designed for an eight modules worm-like robot has been successfully tested on three different FPGA-embedded processors: MicroBlaze, PowerPC and LEON2. The locomotion of worm-like robots, composed of a chain of equal linked modules, is achieved by means of wave propagation that traverse the body of the worm. The time the robot needs to generate a new motion wave, also known as the gait recalculation time, is the key to achieve an autonomous robot with real-time reactions. Algorithm execution time for four different architectures, as a function of the total number of articulations of the robot, are presented. The results show that a huge improvement of the gait recalculation time can be achieved by using a float point unit. The performance achieved using the LEON2 with FPU is 40 times better than LEON2 without FPU, using only 6% of additional resources.

“Cube Revolutions”: the worm-like robot
- It is composed of 8 similar linked Y1 modules
- The modules are connected in the same orientation
- The robot only can move forward or backward
- The electronic and power supply are located off-board

Y1 MODULES
- One degree of freedom
- Made of plastic
- Actuated by a servo
- Can be connected in two configurations:

Locomotion Algorithm
- The problem to solve is the generation of a sequence that allows the robot to move forward or backward.
- The shape of the robot at instant i is described by the angular position vector.
- The sequence of movement is characterized by a Matrix, rows are the angular position vectors at different instants.
- The locomotion algorithm calculates this matrix, based on the propagation of waves along the body of the robot.

Step 1: Fitting the worm to the wave
Step 2: Shifting of the wave
Step 3: Fitting the wave to the robot again

The algorithm operates as follows:
- Step 1: The articulation’s angles are calculated so that the worm fits the wave.
- Step 2: The wave is shifted.
- Step 3: The worm is fitted to the wave again.
- Steps 2 and 3 are repeated until the wave has moved a distance equal to the wavelength.

Implementation on FPGA

<table>
<thead>
<tr>
<th>Target architectures</th>
<th>Processor</th>
<th>Frequency</th>
<th>FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LEON1</td>
<td>25MHz</td>
<td>Virtex XC2000E</td>
</tr>
<tr>
<td>2</td>
<td>LEON1 + FPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MicroBlaze</td>
<td>50MHz</td>
<td>Virtex II Pro</td>
</tr>
<tr>
<td>4a</td>
<td>PowerPC</td>
<td>100MHz</td>
<td></td>
</tr>
<tr>
<td>4b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Synthesis results:

- Processor: LEON1
- Clock: 25MHz
- BRAM: 128K (50%)
- LUTs: 483 (29%)
- LEON1 + MicroBlaze: 50MHz
- LEON1 + PowerPC: 100MHz
- LEON2 + MicroBlaze FPU: 50MHz (40%)
- LEON2 + PowerPC FPU: 100MHz (50%)

Algorithm execution time

Conclusions and future work
- The locomotion algorithm for worm-like robots locomotion has been tested on four different architectures.
- The gait reconfiguration time (GRT) can be drastically improved by means of the use of an FPU unit.
- A 25 MHz LEON2 with an MicroBlaze FPU is almost one order of magnitude faster than a PowerPC working at 100MHz.
- When low GRT is required, the use of the MicroBlaze is a good solution.
- It saves about the 75% of area, leaving this percentage free for the implementation of new hardware cores.
- For further work, the architecture chosen is the LEON2 + FPU.
- The locomotion on 2D problem (2D) has to be solved.
- The same algorithm will be used but using two waves: one for the joints in the plane parallel to the ground and the other for the joints in the perpendicular plane.
- The final locomotion will be generated as a composition of the two waves.