A Hardware/Software Codesign Case Study: 
Design of a Robot Arm Controller

Mohamed ABID¹, Adel CHANGUEL, Ahmed JERRAYA

Laboratoire TIMA, System-level Synthesis Group,
46 Av. Félix Viallet, 38031 Grenoble Cédex, FRANCE

Abstract
This paper deals with exploration of hardware-software design space. The hardware/software partitioning is illustrated using a real-time robot arm controller. Several architectural solutions are discussed with regard to their performance and cost. The goal is to select the best solution that satisfies the real-time constraints and minimizes the cost.

1 Introduction
The goal of hardware/software codesign is to produce an efficient implementation that satisfies the performance and minimizes the cost, starting from the initial specification. There exists a diversity of technological solutions based on available hardware-software components. Different solutions are analyzed to choose the optimal solution.

The objective of this work is to discuss the codesign of a real-time robot arm controller and to illustrate the advantages of mixed hardware-software implementations.

2 The Robot Arm Controller
The robot arm controller under discussion acts between a "Host Machine" and a robot arm that makes use of 18 stepper motors (figure 1). The host fixes the trajectory of the arm. It then sends the commands to the robot arm. The controller makes use of an adaptive speed control to smooth on motion of the arm.

Figure 1: Robot Arm Controller Sub-System

So, the system performs two principal tasks: adaptive distribution of pulse packets and adaptive speed control in real-time. To ensure the control in real-time, the speed variation must be ensured during a lapse of time smaller than the response time of the motor (6 ms).

3 Hardware/Software Partitioning of the Robot Arm Controller
For the sake of simplicity we restrict the type of components used. A software component will be made of a 80286-based architecture; while a hardware component will be made of a Xilinx 4005 FPGA.

In order to be able to measure the performance of the solution, an execution time model is used. However, an additional assumption is that a cost is associated with each component.

We explore five alternatives of implementation:
3.1 Pure-software solution using one software processor.
3.2 Pure-software solution using two processors.
3.3 Pure-software solution using 19 software processors.
3.4 A pure-hardware solution.
3.5 Hardware/Software solution.

After the analysis, only two of the solutions met both cost and real-time constraints (Table 1). In this case we select the best one. In fact, the last solution is cheaper than the fourth solution. It has a better cost compared to solution iv.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Real-time violation</th>
<th>Cost violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Results of analysis

4 Conclusion
The advantages of mixed hardware-software codesign are illustrated using a design of a real-time robot arm controller. Future work aims to the generalisation of hardware/software method in order to allow the automatisation of the partitioning process. Research efforts will be currently deployed to formulate estimation models and objectives functions.

1 Assistant professor on leave from ENIM (Ecole Nationale d'Ingénieurs de Monastir) - 5000 Monastir - TUNISIA

ED&TC '96
0-89791-821/96 $5.00 © 1996 IEEE