Embedded hardware/software co-design for machine control

Free Claessens, Steven Kauffmann, Stefan de la Haye, Peter Slaets
Katholieke Universiteit Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300B box 2420, B-3001 Leuven, Belgium.
Katholieke Hogeschool Limburg,
Campus Diepenbeek, Agoralaan block B box 1, B-3590 Diepenbeek, Belgium.
Katholieke Hogeschool Kempen,
Campus Geel, Kleinheofsstraat 4, B-2440 Geel, Belgium
E-mail: peter@slaets.org

INTRODUCTION

This article deals with the development of a real-time machine control application on an embedded platform. Nowadays advanced industrial control applications are evolving from distributed architectures to embedded standalone architectures. These standalone architectures require significant computational power to perform complex operations. Traditional embedded systems provide a low cost, reconfigurable and compact solution, but lack computational power to guarantee fast real-time performance. To overcome this limitation our embedded system is equipped with an FPGA that enables fast computations. This combined hardware/software platform separates task executions in software running on the processor and dedicated hardware running on the FPGA. The current open machine control system, consisting of an industrial control PC combined with various data acquisition cards, is replaced by a relative cheap hardware/software platform where the data acquisition is implemented on the embedded FPGA and the high-level control on the embedded processor. Therefore the embedded system provides a low-cost and open alternative for industrial machine control.

HARDWARE/SOFTWARE CO-DESIGN FOR PERFORMER MK2 ROBOT CONTROLLER

The chosen machine control platform is a Performer MK2 robot with five joints actuated by PWM driven DC-motors. The position is measured using incremental optical encoders. The software platform consists of an embedded PowerPC processor controlled by an RTAI patched Linux kernel. The root filesystem is currently (during the experimental phase) provided by a remote NFS-server. Communication with the user is provided using Secure Shell Handler. The hardware platform consists of an FPGA with a reprogrammable hardware architecture, which is programmed at boot-up from a Compact Flash Card. Figure 1 shows the platform setup.

The machine controller needs to perform three basic steps: sensor decoding, motion controlling and PWM-generation. In our application all three steps are executed on the FPGA architecture. Our sensor decoder uses the incoming encoder signals (quadrature signals) to determine the position, speed and acceleration of the machine based only on the original
encoder signals. Figure 2 shows the difference between the approximated (derived) velocity and acceleration compared to the "direct" measurements using the FPGA. This sensor data is then used in a motion control loop to perform an accurate machine positioning. Additionally, the sensor data can be monitored by the PowerPC for user information. The FPGA implementation of the sensor decoder receives various parameters from the software, making the decoder implementation portable, flexible and open. Therefore the FPGA can replace expensive data acquisition cards without losing flexibility.

A proportional motion controller is implemented on the FPGA at the position level and the gain factor is provided at software level. An extension to more complex controllers (PI, PID) or robot control at velocity level is straightforward.

The PWM-generator is based on a PWM-block cycle. Therefore the numeric output of the motion controller is converted to a duty cycle of the PWM-block cycle. The various setpoints of the PWM-signal are set in software, making this implementation platform independent.

![Fig. 2. This figure compares the velocity and acceleration measurements of a robot axis derived from the position measurement (Approximated) and derived from the incremental encoder signals (Measured).](image)

**Conclusion**

This hardware/software co-design platform ensures a fast machine control loop on the FPGA combined with a hard real-time setpoint computing on the processor. Thanks to the flexible data exchange between the FPGA and the PowerPC, various machine parameters can be set on the hardware, making the platform portable between similar machines. Because of the flexible FPGA-architecture, the extension with additional sensors like distance, force or vision sensors is straightforward and a subject of future work. Additionally Table I shows a significant price reduction obtained when the hardware/software platform is used instead of the industrial open software machine controller.

<table>
<thead>
<tr>
<th></th>
<th>Software</th>
<th>Hardware/software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal computer</td>
<td>€800</td>
<td>€0</td>
</tr>
<tr>
<td>Embedded system</td>
<td>€0</td>
<td>€1500</td>
</tr>
<tr>
<td>Data-acquisition</td>
<td>€1830</td>
<td>€0</td>
</tr>
<tr>
<td>Power drives</td>
<td>€1250</td>
<td>€125</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>€3880</strong></td>
<td><strong>€1625</strong></td>
</tr>
</tbody>
</table>

**TABLE I**

Comparison of the price between an industrial controller and the hardware/software co-design platform.