The application of reconfigurable logic to high speed
CNC milling machines controllers

Roque Alfredo Osornio-Rios\textsuperscript{a,*}, Rene de Jesus Romero-Troncoso\textsuperscript{b},
Gilberto Herrera-Ruiz\textsuperscript{a}, Rodrigo Casta\~{n}eda-Miranda\textsuperscript{a}

\textsuperscript{a}Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas s/n, 76010 Querétaro, Qro., México
\textsuperscript{b}Electronics Department, FIMEE-Universidad de Guanajuato, Tampico 912, Col. Bellavista, 36730 Salamanca, Gto., Mexico

Received 4 April 2006; accepted 10 August 2007
Available online 29 September 2007

Abstract

Reconfigurable logic has gained relevance in high-speed computer numerical control (CNC) digital controllers where high processing speed at the servo loop update time is critical. Reconfigurable devices like field programmable gate array (FPGA) have advantages over microprocessors and DSP with similar logic complexity because their open architecture makes them suitable for system on-a-chip applications. The contribution of this work is the development of an FPGA-based proportional–integral–derivative (PID) controller applied to a high-speed CNC milling machines in order to achieve the servo loop update time requirements. Results show PID system functionality, which has been synthesized into a low-cost FPGA platform, successfully applied to a high-speed CNC milling machine.

Keywords: Reconfigurable logic; High-speed control; FPGA; Servo loop update time

1. Introduction

High-speed computer numerical control (CNC) machine motion control involves feed-rates superior to 40 m/min as well as acceleration dynamics above 1 G and in some cases 2 G (Wang, Liu, & Ai, 2003). Due to these facts, a motion controller for high-speed CNC machines has higher demands on the electronics design compared with a conventional speed CNC, this requires higher sampling rates (servo loop update time) which reduce the available time for on-line processing. PMAC (Delta Tau, 2005), and DMC-18×0 (Galil Motion, 2004) are some of the most popular commercially available CNC controllers based on off-the-shelf digital signal processor (DSP) and microcontrollers for the implementation of their proprietary control algorithms, where additional resources are also for processor support on a fixed and closed architecture, which is well suited for standard speed processing. Due to the sequential processing data flow on commercially available DSP and microprocessors, the increase in sampling rate for high-speed control can impose severe restrictions on the processor performance; therefore, other alternatives for signal processing must be considered in order to achieve on-line operation.

High-speed servo system controllers are generally harbored in a personal computer (PC) as an add-on board that frees the PC from calculations that demand a great deal of resources. Field programmable gate array (FPGA) devices have been winning market in system on-a-chip (SOC) applications, because they can integrate processing units defined by the user and related peripheral logic in hardware, combining open architectures that do not depend on the manufacturer or specific platforms. To integrate processing units along with their related peripheral logic support as a SOC into an FPGA, the device gives the designer enough freedom to achieve the specific task developing an architectural level that also generates a low-cost single chip unit. An FPGA is an array of basic logic blocks where the user can define its interconnectivity, making them programmable in a fully open architecture. Therefore, an FPGA provides the advantages of a general
purpose processor and a specialized circuit that can be reconfigured as many times as it is necessary until the required functionality is achieved. The speed and size of the FPGA are comparable with the application-specific integrated circuit (ASIC), but FPGA is more flexible and its design cycle is shorter because of its reconfigurability. FPGA applications go beyond the simple implementation of digital logic; they can be used for implementations of specific architectures for speeding up some algorithm. A specific architecture for an algorithm, implemented into an FPGA could have 10–100 times higher performance than the implementation on a DSP or microprocessor. On the other hand, DSP and microprocessors have a fixed sequential architecture for computation, which can be easily overloaded when the processing time between samples is significantly reduced, as in high-speed control, while FPGA have a natural parallel architecture for high-speed computation. Along with the advantages already cited, FPGA development is performed under hardware description languages (HDL), which make the design portable and platform independent, which is not the case for commercially available DSP or microprocessors.

The main objective of this paper is to show the profitability and advantages in the design of a proportional–integral–derivative (PID) controller for high-speed CNC using FPGA compared with other devices, highlighting the versatility, utility and practicality that results in industrial applications. The experimental results show an economic and simple option that satisfies the accuracy and on-line processing time requirements on the control of CNC machine tools.

2. Background

2.1. FPGA applications in mechatronics

During the last years, FPGA applications in mechatronics have been growing in relevance as an economic and reliable option made to fulfill the requirements of critical process. For example, Wegryn, Adamski, and Monteiro (1998) implemented the FPGA-based control systems using Petri Network, highlighting its application in mechatronics. Lin, Wang, and Huang (2004) combined a dead-time lag with a neural fuzzy network to control the speed of an induction motor, showing FPGA and reconfigurability as a desirable property in control. Romero, Herrera, Terol, and Correa (2004) showed the development of an on-line system for detecting the tool breakage in CNC machine tools, based on an FPGA, where they exploited the multiprocessing and open architecture capabilities of FPGA to give an on-line solution to a highly complex computation algorithm. The authors also proved that DSP or PC are not suited for the task. Wook and Kim (2002), developed an FPGA acceleration and deceleration profile generator for robots and machine tools, where hardware description languages and FPGA are shown to be the leading technology in control because its portability and vendor independence, in addition to reconfigurability, compared with other standard technologies such as microprocessors, microcontrollers and DSP. Jones, Goodall, and Gooch (1998) showed the design of a processor architecture for the implementation of high performance controllers with FPGA, demonstrating that low-cost solutions are easily developed under FPGA and hardware description languages. Sancho and Goodall (2007) developed a processor for real-time embedded mechatronic systems, based on an FPGA, they believe that this will apply particularly for demanding applications where the computational capability of general purpose devices is stretched, or when more complex control algorithms would offer superior performance but the algorithms currently require excessive computation.

The foregoing researches are some good examples of how the FPGA technology is the optimal solution for specific mechatronics problems such as low-cost, high processing speed, design simplicity and reconfigurability.

2.2. High-speed machining

High-speed machining refers to chip removal in the CNC and offers substantial economic benefits due to increased metal cutting productivity (Knospe, 2007). In praxis, it basically consists on substituting few slow passings of larger cut depth for many fast passings of smaller cut depth, obtaining an important increase in the quantity of evicted material. In high-speed machining, the goal is to increase the capacity in a remarkable way to speed up the chip removal, which clearly reduces machining time. This factor is especially important in applications, which involve operations of complex contours, as in molds.

Table 1
Relationship between the sampling period with speed and machining precision

<table>
<thead>
<tr>
<th>Servo cycle time tcs (ms)</th>
<th>Cycles/s (1/tcs)</th>
<th>Cycle distance with ( f = 3 \text{ m/min} ) (mm)</th>
<th>Cycle distance with ( f = 10 \text{ m/min} ) (mm)</th>
<th>Cycle distance with ( f = 30 \text{ m/min} ) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>50</td>
<td>1</td>
<td>3.33</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0.5</td>
<td>1.66</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>333</td>
<td>0.15</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>0.05</td>
<td>0.16</td>
<td>0.5</td>
</tr>
<tr>
<td>0.4</td>
<td>2500</td>
<td>0.02</td>
<td>0.06</td>
<td>0.2</td>
</tr>
<tr>
<td>0.1</td>
<td>10,000</td>
<td>0.005</td>
<td>0.016</td>
<td>0.05</td>
</tr>
</tbody>
</table>
The employment of a higher cutting speed allows reducing the chip removal width without production process time loss which reverts in the increasing of tool life and in machining precision, eliminating final stages of superficial finish. The increase of the feedrate must be accompanied by a raise in the axes advance.

As mentioned above, a control for a high-speed CNC machine has higher designing demands in comparison with a conventional CNC. The main designing requirement is the sampling rate (servo loop update time) which in conventional machining is 1 KSPS (kilo samples per second), while in high-speed machining the sampling rate is 10 KSPS or even higher. Table 1 shows the relevance of having fast servo cycle times to work quickly and accurately, demanding more cycles at the servo at higher speed (Gordon & Hillery, 2005).

3. FPGA controller applied to high speed CNC milling machines

There are few works about high-speed CNC machine controllers developed in hardware and most of them use a DSP, microprocessor and commercial boards for the implementation. Gordon and Hillery (2005) developed a high-speed CNC cutting machine using a stand-alone commercial controller. Erkorkmaz and Altintas (2001) presented the implementation of a high-speed tracking and contouring control of feed drives based on a DSP board. Erkorkmaz, Yeung, and Altintas (2006) used a Fadal 2216 machining center, controlled by an in-house-developed open architecture CNC, which is based on a DSP board. Tang, Landers, and Balakrishnan (2006) developed only in simulation, an important hierarchical optimal control methodology for the simultaneous regulation of servomechanism position errors, contour error and machining forces. After an exhaustive search over specialized literature, no references for FPGA implementation of high-speed controllers have been found to date.

In this paper, the case of study is a PID controller for high-speed CNC milling machines. Frequency response is the method used for controller design and tuning, and the hardware implementation is done into a low-cost FPGA. The novelty of this work is that the FPGA has the capacity in servo loop update time requirements for high-speed controllers and is inexpensive compared with DSP and microprocessor because it can integrate the whole solution (i.e. control law, timing, interface and support logic) in a single chip. Another highlight in the use of FPGA is its reconfigurability, which allows an open architecture for the controllers with the advantage of an easy-to-perform architectural redefinition without hardware modifications.

The developed design is divided into several components: identification, PID tuning, trajectory generation and controller implementation. In Fig. 1, a general block diagram of the complete controller system for high-speed machining is shown. The block diagram for the high-speed CNC controller is shown in Fig. 2. The main filter is a PID, but any filter can be implemented with the structure designed in FPGA without changing the system hardware via reconfiguration.

4. FPGA implementation

The control law implemented in this work is a standard PID. The algorithm is similar to the one used by commercial boards (Galil Motion) which will be developed in this section. The main contribution of this work is the FPGA implementation through an area-efficient high-speed digital structure. The development in FPGA devices allows obtaining higher sampling rates (servo loop update time) in comparison with other processors like DSP or microcontroller. On the other hand, the FPGA integration and portability is very important for the complementary structures designed in hardware, such as profile generation, feedback counter and system interfacing which is embedded on the same integrated circuit for a SOC approach.
while DSP or microcontrollers require separate and dedicated logic for these peripherals. It is also important to note that FPGA is reconfigurable, which means that whenever a modification on the inner digital structure is required, major changes on the PID algorithms, or even on additional control structures when PID control is not enough, can be done without changing the hardware by a software-like reconfiguration on the FPGA via hardware description languages. Though DSP can implement those changes in the control algorithm via software, the sequential nature of the processor cannot guarantee high sampling rates when the algorithm requires more cycles for computation, while FPGA do not have that problem because of their inherent parallel structure.

4.1. Controller hardware design

4.1.1. PID hardware

The design developed in this work has the purpose of carrying out a general controller in reconfigurable hardware for several control laws such as: proportional (P), proportional–integral (PI), proportional–derivative (PD), proportional–integral–derivative (PID), lead filters, lag filters and lead–lag filters. All the previously cited laws, with higher sampling rates allow higher resolutions. The digital design of the hardware for these control laws was developed based on the equation in differences of a second–order infinite impulse response (IIR) filter shown in (1), where \( b \) and \( a \) are constant coefficients that can be changed according to the particular control law, \( x(k) \) is the excitation and \( y(k) \) is the system output:

\[
y(k) = a_0 x(k) + a_1 x(k - 1) + a_2 x(k - 2) - b_1 y(k - 1) - b_2 y(k - 2).
\]

The design consists of a digital structure development for a second-order difference equation. The design can be flexible since it can generate a reconfigurable structure in hardware to implement any equation in differences and thus, any filter or control law. Fig. 3 shows the general controller structure described under very high-speed integrated circuit hardware description language (VHDL), which is the language for digital design in a portable and manufacturer insensitive platform. Thanks to VHDL description, the design remains portable to any present and future implementation technology and allows an easy to perform reconfiguration for every digital structure without hardware modifications. The controller structures are: controller coefficients, sample coefficients, sample block, multiplier accumulator (MAC) and barrel shifter.

The main controller structure is a 36-bit MAC. This structure was designed through a multiplier, accumulator, adder and one finite state machine that has the multiplication and accumulation control of the difference equation; the structure was designed under VHDL.

The sample coefficient structure is in charge of sample shifting, so that they multiply with its respective coefficient in the difference equation. Its digital structure was made using a multiplexer pointed by an index and the registers. The digital structure that integrates the controller coefficient module is a read only memory (ROM) described under VHDL, containing the digital coefficient equivalent to the equation previously found through the tuning methods. The resolution of these coefficients is 36 bits. Both structures were designed under VHDL.

Fixed-point arithmetic was used for the controller computations and a barrel shifter has the task of decimal point adjustment at the result. The digital structure of this programmed module was based on a register with shifting.

4.1.2. Position feedback counter

The servomotor position is obtained from an opto-electronic position codifier (encoder). There, two quadrature square pulse waveforms are generated to determine the motion direction; the quantity of pulses delivers the absolute position. But those waveforms cannot be received directly by the digital controller; they should be processed by a digital structure that gives the absolute position count at the encoder and use it to take control actions. A digital counter is the main block for the position encoder that gives the absolute position with a 32-bit resolution. Other digital structure is the synchronization register, used to avoid errors at the encoder inputs signals. Fig. 4 shows the block diagram of this structure.

4.1.3. Complementary digital structures

In the block diagram shown in Fig. 5, other complementary blocks to the overall controller system were developed under VHDL. Next, a summary of these complementary blocks follows. The profile generator is an independent computational unit based on accumulation, which can be reconfigured, using the FPGA flexibility, to provide the desired motion profile. This module has a multiplexer and data register that receives the speed profile parameters, which feed the accumulator to generate the motion profile, controlled by a finite-state machine.

Fig. 2. Control diagram to high-speed CNC.
Position profile is generated by integrating the speed profile with another accumulator.

A 32-bit digital adder is used for the adding point at the control loop. Another necessary element for the controller is a digital to analog converter (DAC) that gives the control signal to the servo amplifier. For this work a 16-bit Burr Brown (1998) DAC8531 is used. This converter has a serial interface and a state machine is used to provide the driving signals from the controller to the physical DAC device.

4.2. Servo system model identification

The controller design requires knowledge of the system dynamics or in other words, it is required to have a model. The model is obtained in two parts: the first is the identification of the elements: amplifier, servomotor and encoder, by computational methods; and the second is the gain computation of the DAC and the zero-order hold (ZOH) sampler.

4.2.1. Recursive least-square method

The computational identification methods have been evolving throughout the last few years as the best choice to obtain a system model. There are different identification methods that have been proven, obtaining the best results with recursive methods, in this specific case, recursive least square. The recursive least-square method was programmed in C++, according to Aguado and Martinez (2003), using an algorithmic form based on (2), where $C(t)$ is the covariance estimation matrix, $\phi$ is the exponential omission coefficient, on the other hand, the matrix $g(t)$ (3) is the product between the measurement matrix $z(t)$ and the estimate covariance $C(t)$, where $x^2$ is a constant determined by (4). The identified model is described by (5). The identification of the model includes the servomotor, amplifier and encoder:

$$C(t + 1) = \frac{1}{\phi^2} \left[ C(t) - \frac{1}{x^2(t + 1)} g(t + 1) g^T(t + 1) \right],$$

(2)
\[ g(t + 1) = C(t)z(t + 1), \]  
\[ z^2(t + 1) = z^2 + z^0[(t + 1)g(t + 1)], \]  
\[ B(s) = \frac{(s + 761.68)(s - 679.8)}{(s + 4.75)(s + 1.95)}. \]

### 4.2.2. DAC and ZOH models

A DAC is characterized by its resolution \( n \), it varies typically between 8 and 16 bits and the common output interval is \( \pm 10 \) V, which is the input range for most servo amplifiers. In this work, a 16-bit converter is used with output voltage ranges of \( \pm 10 \) V. The DAC gain is represented by the following equation:

\[
K_a = \frac{V_{\text{max}} - V_{\text{min}}}{2^n} = \frac{10 - (-10)}{2^{16}} = 0.0003051.
\]

The ZOH represents the effect of the sampling process where the motor command is refreshed every sampling period \( T \). Since high-speed machining requires faster sampling rates than the conventional machining (1 ms), the sampling period that is used in this design is 100 \( \mu \)s. Therefore, the transfer function of the ZOH in this research is described through the following equation:

\[
H(s) = \frac{1}{1 + (Ts/2)} = \frac{1}{1 + ((0.0001)s)/2} = \frac{20000}{s + 20000}.
\]

### 4.3. PID controller analysis

The model employed was the one proposed by Tal (1989) in the motion controllers for the commercial Galil Motion control boards. Tal claims that the model has a transfer function \( D(z) \) as shown in (8) with a sampling period \( T \). Filter parameters \( K, A \) and \( C \) are selected by \( KP, KD \) and \( KI \) instructions, respectively. The relationships between the filter coefficients and their related instructions are presented in Table 2. After that, Tal shows the equivalence between a conventional PID filter with transfer function \( G(s) \) as presented in (9) along with Table 3 where proportional (P), integral (I) and derivative (D) gains are related to \( KP, KD \) and \( KI \) instructions. Finally, (10) is obtained by the substitution of instructions in Table 2 into (8) and this is the digital PID expression related to \( KP, KD \) and \( KI \). The continuous model is shown in (9) and its digital equivalent was implemented in hardware, expressed in the transfer functions (8) and (10). as

\[
D(z) = \frac{Kz - A}{z} + \frac{Cz}{z - 1},
\]

---

**Fig. 4.** Block diagram of the position feedback counter.

**Fig. 5.** Block diagram of complementary digital structures: (a) profile generator and adder point and (b) DAC driver.
Table 2 Relationship between digital filter parameters and J. Tal method instructions

<table>
<thead>
<tr>
<th>Digital filter parameters</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>(KP + KD)²4</td>
</tr>
<tr>
<td>A</td>
<td>KD/(KP + KD)</td>
</tr>
<tr>
<td>C</td>
<td>K1/2</td>
</tr>
</tbody>
</table>

Table 3 Relationship between continuous filter parameters and J. Tal method instructions

<table>
<thead>
<tr>
<th>PID continuous coefficient</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>4KP</td>
</tr>
<tr>
<td>I</td>
<td>K1/2T</td>
</tr>
<tr>
<td>D</td>
<td>4T²KD</td>
</tr>
</tbody>
</table>

\[
G(s) = \left(\frac{P + DS + I}{s}\right), \quad (9)
\]

\[
D(z) = \left[4KP + 4KD(1 - z^{-1}) + \frac{KI}{2(1 - z^{-1})}\right]. \quad (10)
\]

4.3.1. PID tuning by frequency response method

This method is based on two important aspects of the system response: the stability grade and the response speed. The stability of a system indicates the response nature, while the response speed indicates the system capacity to carry out small corrections without causing oscillations. The grade of stability is expressed by means of the phase margin and the response speed of the system is directly related to the crossing frequency. It is assumed that the design objective can be expressed in function of the crossing frequency (\(\omega_c\)) and the phase margin (\(\theta_m\)). Thus Eqs. (11) and (12) are satisfied:

\[
|L(j\omega_c)| = 1, \quad (11)
\]

\[
\theta_m = 180^\circ + \arg[L(j\omega_c)], \quad (12)
\]

where \(L(j\omega)\) it the open loop transfer function of the servo system that, as shown in Fig. 6, represents the identification product \(B(s)\), the DAC gain \(K_d\), the ZOH function \(H(s)\) and the PID digital filter.

The system design process includes the specification of the parameters \(\omega_c\) and \(\theta_m\) such to satisfy (11) and (12). The stability rule for the frequency response says that the system is stable if the phase margin is positive, and unstable when it is negative. Values of phase margin between 30° and 45° indicate a response with good reduction, higher values on phase margin can generate overshoot and overdamping in the system response. The crossing frequency indicates the response speed. High values mean quick responses; in general, the constant response time (\(\tau\)) can be expressed in function of (13):

\[
\tau = \frac{1}{w_c}. \quad (13)
\]

The advantage of the described tuning method is that if a new tuning is required for another system, the tuning algorithm is adjusted by defining the phase margin (stability) and the cross-frequency (speed response). This method assumes an off-line identification for different conditions as commercially available controllers do (i.e. Galil Motion controllers).

The block diagram of Fig. 6 shows the blocks that integrate the function \(L(s)\) which is used for the tuning calculation. The data used in the controller design applied to high-speed machining and the coefficients of the PID digital transfer function are obtained from the preceding procedure and are shown in Table 4.

4.4. Hardware implementation of the PID controller for high-speed CNC

Digital PID Eq. (8) can be rewritten as (14) where the difference Eq. (15) is obtained. \(U(z)\) is the PID output and \(E(z)\) is the error signal at the PID input:

\[
U(z) = \frac{KAz^{-2} - (KA + K)z^{-1} + (K + C)}{1 - z^{-1}}, \quad (14)
\]

\[
u(k) = C_1e(k-2) + C_2e(k-1) + C_3e(k) + u(k-1), \quad (15)
\]

where the coefficients \(C_1\), \(C_2\) and \(C_3\) equivalence with the tuning values are stated in the following equations:

\[
C_3 = KA, \quad (16)
\]

\[
C_2 = KA + K, \quad (17)
\]
\[ C_1 = K + C. \] (18)

With the tuning process, the PID coefficients in (15) are obtained. From the generic module, designed in hardware based on (1) for a second-order differences equation, it is required to give the PID coefficients to the general hardware structure controller.

Fig. 7 shows a general block diagram of the implemented controller in FPGA, including the control law and support hardware like the feedback of the encoder counter, DAC driver, frequency divisor and interface. The PC, via the interface, sends the profile parameters to the profile generator. The profile generator feeds the reference position to the error adder, which uses the encoder feedback position to feed the error signal to the PID digital controller. The PID digital controller was previously tuned as shown in Section 4.3. The PID output is then sent to the DAC state machine to provide the required output to the servo amplifier, which supplies the servomotor current.

4.5. Motion profiles

Some of the most employed trajectories used as reference in the CNC machine systems are the trapezoidal and triangular velocity profiles that give soft positions when integrated. The lineal and circular trajectories are used as references in the high-speed machining. In the present work, a trapezoidal profile sequence of velocity that generates a position profile has been used. The test profiles applied to the controller have been the step, trapezoidal and triangular velocity profiles. This block is described under VHDL.

5. Results

5.1. PID controller tuning

The identified model has been validated using a position sequence and an accurate behavior was obtained because the reference and output of the system showed a similar behavior. These results allowed to continue with the design.

The model tuning gives the difference equation in (17) that has been implemented in the digital controller:

\[
\begin{align*}
 u(k) &= 3596.4e(k - 2) - 7271.753e(k - 1) \\
 &\quad + 3596.4e(k) + u(k - 1).
\end{align*}
\] (19)

The designed controller applied to the servo system for high-speed machining was subjected to several references as the step and other triangular and trapezoidal position profiles to obtain the system results.

The step excitation and system response of the tuned controller, as implemented, are shown in Fig. 8. From this figure it can be seen that the generated overshoot is excessive but it is used to show the worst case tuning of the system, which with further fine tuning on the parameters (phase margin and crossing frequency) this overshoot is significantly reduced. Fig. 9 shows a trapezoidal speed profile which gives the position profile at the output system shown in Fig. 10 along with the position reference command for an end position of 80,281 counts in 0.3 s, which implies 1 G on acceleration and deceleration parts of the trapezoidal speed profile. Figs. 9 and 10 show the advantages in the use of motion profiles over the step position command. It can also be seen in Fig. 10 the fine tuning efficiency of the PID controller where the actual
output follows the reference command satisfying the real-time running requirements.

5.2. FPGA controller implementation

The designed control system using VHDL, shown by the block diagrams of the previous section, was implemented in a SPARTAN-3 FPGA (Xilinx, 2005). This device has 200,000 gates, 216 kb random access memory (RAM), 12 18×18 bit multipliers and 173 inputs and outputs defined by the user. The reference clock runs at 50 MHz.

Synthesis results show an average 30% FPGA usage, equivalent to 60,000 system gates for one axis controller. Further axis controllers can be implemented into the FPGA without hardware modifications by using parallel, combined with sequential architectures. Due to its natural parallel architecture, the developed design has an overall delay of 12 clock cycles for data processing at each sample computation. It must be considered that all the constitutive blocks work in parallel, therefore the delay is minimum. As comparison, considering a standard commercially available fixed-point DSP from Texas Instruments (2005) TMS320C5409, which is equivalent in cost and integration complexity to the FPGA in this case of study, gives an overall processing time of 2050 clock cycles which is significantly higher than the FPGA implementation. Now, considering an equal clock frequency of 50 MHz for both options, the delay at the FPGA is 240 ns while the delay at the DSP is 41 μs at each sample computation.

In order to estimate the FPGA implementation performance, Table 5 shows the minimum servo loop update theoretical time for FPGA, DSP and commercially available controller Galil 18×0 for different number of axes. The required minimum servo loop update time

<table>
<thead>
<tr>
<th>Controller</th>
<th>Minimum servo loop update time (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-axis</td>
</tr>
<tr>
<td>FPGA</td>
<td>0.24</td>
</tr>
<tr>
<td>DSP</td>
<td>41</td>
</tr>
<tr>
<td>Galil</td>
<td>125</td>
</tr>
</tbody>
</table>

Table 6
Minimum servo loop update time (μs) for given feed rates and controller resolutions for one axis

<table>
<thead>
<tr>
<th>Feed rate (mm/min)</th>
<th>Controller resolution (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>1500</td>
<td>4000</td>
</tr>
<tr>
<td>15,000</td>
<td>400</td>
</tr>
<tr>
<td>30,000</td>
<td>200</td>
</tr>
<tr>
<td>60,000</td>
<td>100</td>
</tr>
</tbody>
</table>
related to controller resolution and feed rate is given by (18) as

\[ \text{Update time} = \frac{\text{Controller resolution}}{\text{Feed rate}}. \]  

(20)

Table 6 shows several minimum servo loop update time requirements for four different feed rates: 1500, 15,000, 45,000 and 60,000 mm/min; and three controller resolutions: 0.1, 0.05 and 0.01 mm. From Tables 5 and 6, it can be seen that Galil 18 \( \times \) 0 controller is suitable for few cases, while DSP-based controller is appropriate for some others, and the FPGA implementation is well matched for all cases.

5.3. Experimental setup

The designed prototype has been implemented on the Spartan-3 board. An analog output interface board was also designed for the high-speed CNC servosystem controller. In order to verify the FPGA controller functionality, several experiments were performed in order to test the FPGA controller performance with the following machining parameters: CIATEQ high-speed CNC milling machine, Baldor vector SERF. 1.0 mod. 76H037W03SZ1 motor with belt array set to 40,000 rpm spindle, milling on 6063 aluminum and a servo loop update time of 0.1 ms. Experiment 1 uses a Maxon 2266.85-73216-2000 servomotor with 5.6:1 planetary gear head for axis drive and 1200 mm/min feed rate. Experiment 2 uses a Baldor BSM90N-175 brushless servomotor with 4000 rpm for axis drive and 40,000 mm/min. Fig. 11 shows the implemented prototype with the servomotors and the high-speed CNC for the described machining process. For both servomotors the developed methodology for PID design was used.

6. Conclusions

An FPGA development of a digital controller for high-speed CNC applications was shown. Speed performance was achieved by FPGA because of the parallel architecture, which can be very limited for general purpose processor. As shown by the analysis stated in Table 6, the minimum servo loop update time for high-speed feed rates cannot be achieved by the commercially available Galil 18 \( \times \) 0 controller; DSP-based controllers can achieve the stated requirements only for few cases, while FPGA-based controller does not have restrictions in processing time to achieve the required performance, even for multi-axes implementations as can be seen in Table 5.

Besides the speed and processing performance of FPGA over DSP and microprocessor, a single chip solution can be implemented into the FPGA without the requirement of support logic due to its intrinsic resources, with an evident cost reduction. On the other hand, VHDL design to be implemented into FPGA gives a portable and platform insensitive design for present and future developments, which is virtually impossible for general purpose processors. Finally, the natural open architecture of FPGA provides better suitable solutions for CNC controllers due to its reconfigurability, where structural changes in logic architecture can be easily implemented without hardware modifications. For the present prototype, a 20 US dollar FPGA was used for a single-chip solution.
Acknowledgements

This research was partially supported by CONACyT scholarship No. 177102.

References


