DEVELOPMENT OF A LOW COST INTEGRATED CONTROL SYSTEM BASED ON THE FPGA TECHNOLOGY

Kassio Maciel Kienitz , kassio.kienitz@gmail.com
Frederico Alevato Ramalho Filho, fcnc05@gmail.com
Jordano Fernandes Cordeiro, jordano.f@gmail.com
Rodrigo Ribeiro Nunes, rodrigo_rn1@hotmail.com
Eduardo José Lima II, ejlima2@yahoo.com.br
Alexandre Queiroz Bracarense, bracarense@ufmg.br
Universidade Federal de Minas Gerais, Departamento de Engenharia Mecânica, Av. Antônio Carlos, 6627 – 31270-901, Belo Horizonte, MG, Brazil

Abstract. *The use of CNC systems are becoming more common every day, and such systems are not only for industrial use anymore, with the presence of miniature CNC systems in laboratories, mechanic shops and even small design studios and for hobbyist use. This means that CNC systems have become mainstream on the past few years and processes like milling, turning, plasma cutting and rapid prototyping techniques like plastic fused deposit modeling are the most common methods. But all those systems have in common the electronics necessary to interpret the commands sent from a computer or read from a memory card and to translate that in electric current for the motors. Besides the quality of the machine, its rigidity and the motor power, the electronics have to be capable to provide the necessary movement control in order to provide the best movement with minimal follow up error in the trajectory. This can be done individually for each motor based on the necessary movement read on the G code, but this individual planning does not provide the best control algorithm or control strategy, and also does not provide the best cost benefit solution because of all the wiring and individual boards necessary. When the machine have only Cartesians axis, the control of each axis is straightforward, but machines with serial and even parallel configurations are also becoming more common, and does not have a bigger development mainly due to the need for a kinematics algorithm, and there isn't an open source hardware system capable of processing the movement and kinematics simultaneously. This paper proposes an integrated solution based on FPGA technology in order to provide all the necessary hardware to control up to 6 axes in a single board, versatile enough so it can be used in different CNC system or even retrofitted industrial robots.*

Keywords: *FPGA, Low Cost, CNC, controller, industrial robot, retrofitting*

1. INTRODUCTION

The LRSS (Laboratory of Robotics, Simulation and Welding of the Federal University of Minas Gerais UFMG) has been developing robotic's technology some time ago. Starting with the study and training on industrial commercial robots passing through the reform of these robots up to the development of new technologies in robotics and building/mounting robots with various types and characteristics. Thus boosting research and development of control systems more versatile in order to meet the demands of the different characteristics of each project or Retrofitting. Always keeping the focus on low cost, ease of construction and quality of the system.

Making a brief history of control systems of robots, starting with the OEM dedicated systems based in a huge and complicated digital logic circuits commanding a linear amplifiers with low energy efficiency, to compact micro-processor dedicated systems commanding high-efficiency switched drivers, but with prohibitive cost for small applications, especially because characteristic rigidity of the system which makes difficult the customization and development of innovative solutions. With the evolution of power electronics, we had a sensitive cheapening of the switched drivers, especially for servo-motors, DC motors and precision step-motors, this added to interfaces capable of connecting these drivers to PCs, thus enabling the development of low cost control as demonstrated by Cordeiro *et. al.* (2009). These PC based systems enabled the construction of small machines for use in teaching and production of low and medium scale. Working very well for CNCs and the simple kinematics robots, but are limited by transmission capacity of low-speed outdated ports or modern faster ports than are not deterministic and hinder the continuous control of speed and position on several axes simultaneously Then came the need to develop a system that does not depended on old ports and bypass the problems of lack of determinism on the modern ports and operating systems, i.e. it was independent of a PC, but had not the high cost of a closed OEM system, being able to control complex robot kinematics, such as parallel, serial, orbital, among others, or that require speed and flexibility far beyond as possible on a PC based system. This all, compact enough to be attached to the robot itself, eliminating external cabinets.

This paper proposes a control system based on configurable hardware (FPGAs) fully integrated with the robot and independently from a PC to perform their routines. So expect a system compact and capable of executing control algorithms for robots with different types of kinematics, as well as generate and correct the trajectory of several axes really at the same time, not interpolating between the movements of the axles. That being flexible enough for teaching purposes, and simple and inexpensive enough to be used in applications of low and medium businesses.

2. Development of a Low Costs FPGA based control system

2.1. CNC and Robot low cost control systems

We start from the concept of low-cost robots control system, that have been developed on our laboratory and described by Cordeiro *et. al.* (2009), where we had a system based on a trajectory generator followed by a pulse generator that will command the motor drivers using step and direction signals In the now proposed system, the difference is in the trajectory generator, before that was a function of software on a PC will be implemented in hardware as well as the pulse generator. We expect a large reduction on the number of components and cables, as will be only necessary the connections between the control system, power drivers and sensors, i.e. a frequency signal (Step) controlling the speed, acceleration and motor position , the motor direction signal and the encoders feedbacks . Another function that will be easily implemented is the closed loop pulse generator, providing greatly reduction of pulses loss, noise errors and reporting the real time joint position to the trajectory generator.

In the previous system we used two basic configurations:

2.2.1 Low cost control systems using PC based pulse generators:

At the first, we had the PC as the trajectory generator and pulse generator (Figure 1), using the parallel port as a data port and a breakout board to interface electronics between the PC and the motor drivers, thus making the conditioning voltage levels and isolation between the logic and the high power system. We managed this way, a very simple system and with the lowest cost ever since.

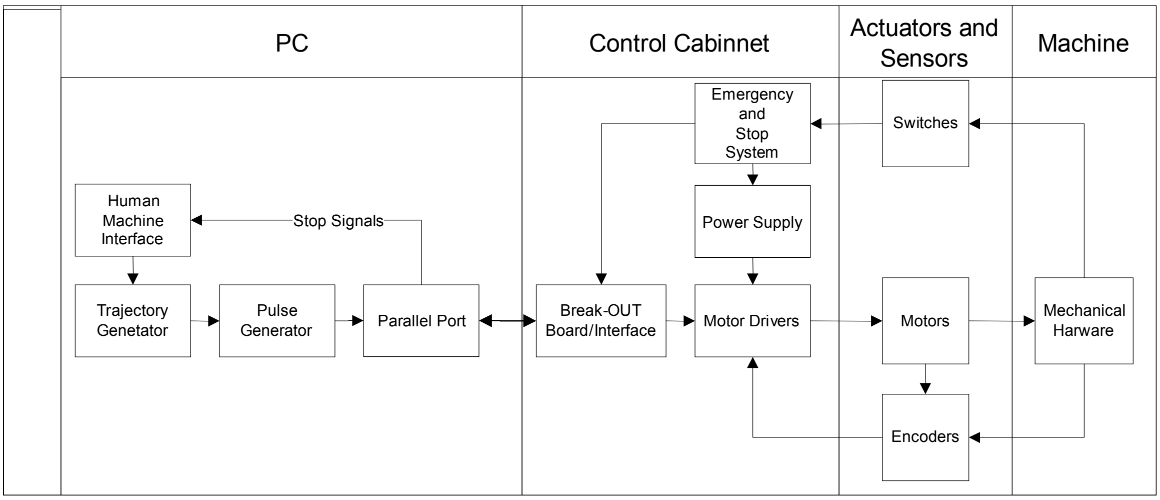

Figure 1. Robotic and CNC system PC based pulse generator.

This model worked well for machines up to 5 axes controlled by open loop over the trajectory. Is suitable for small CNC machines without the need correction speed and acceleration on the fly. I.e., applications where you can give high accuracy, particularly where we have multiple axes interpolation, as outlines of circles and spheres, these machines usually have oversized motor power for its inertia and the effort required for the work to develop, In order to minimize the loss of pulses and acceleration faults. Another way to improve slightly the accuracy of these machines is the use of precision large reductions in motors, but greatly increasing the cost of the mechanical system and putting more joints backslash errors.

This solution proved to be effective but there is a tendency to extinction of the parallel port on most modern PC leaving the system exposed to a rapid process of obsolescence. Another problem is the existence of a great noises sensitivity and frequency limiting on the parallel ports, which directly affects the resolution and speed of motor command, causing the system to be subject to a trade-off speed / accuracy.

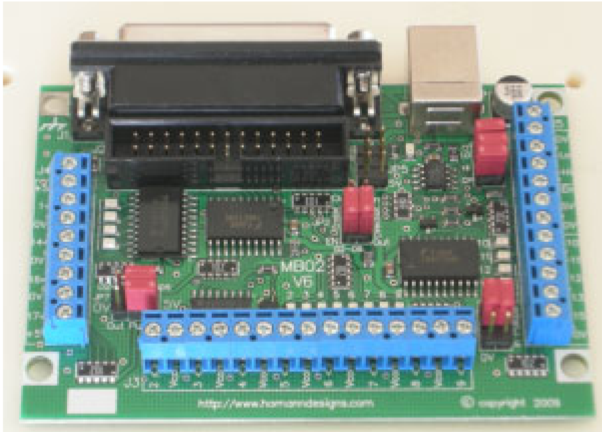
Figure 2. Parallel Port Break-out board example.

2.1.2 Low cost control systems using a dedicated pulse generator electronic board:

The second option was to use the PC only as trajectory generator and use a specific external hardware as the pulse generator using the USB port (for being the most versatile and easy development today), as shown in Figure 3. At first this option showed itself capable of supporting closed loop systems in the pulse and trajectory generators with several axes simultaneously. Mainly by the number of available I / Os, and can thus bring the motion signal of the motors and even additional joints position sensors back to trajectory generator, this all at frequencies well above the parallel port.

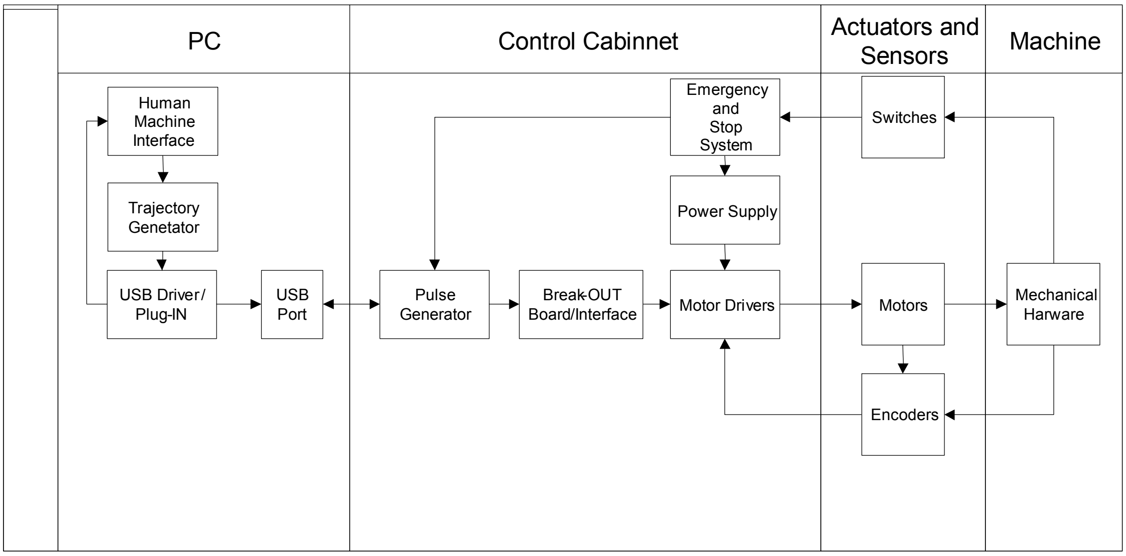

Figure 3. Robotic and CNC system with external pulse generator.

But as we began developing the system we found a series of problems, especially regarding the drivers of dedicated boards that are not open, so we depending on the APIs and SDKs provided by the manufacturer, which are not always easy to work with or are not widely distributed. Despite the USB port have data transmission speeds much higher than the parallel port, the increasing of software layers between the trajectory generator and pulse generator greatly reduces its efficiency.

Another undesirable feature of the USB port is the lack of determinism in its data transmission, i.e., the information reaches its destination at variable times, depending on the efficiency of the software driver and the PC operating system, this complicates the synchronization between several axes. We can utilize a large buffer to minimize the time differences between the commands of multiple axes, but we will lose the ability to correct the velocity and acceleration of the motors during the movement, i.e. no trajectory correction on the fly. So we get a faster and modern system than using the parallel port but with more complex and expensive development, depending on variables difficult to control, like the characteristics of the operating systems and hardware drivers.

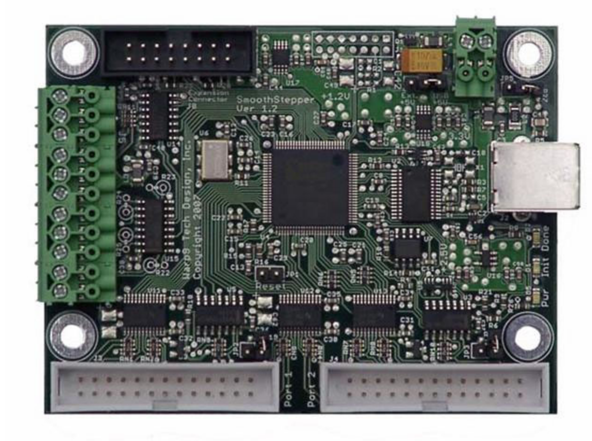

Figure 4. Example of External Pulse Generator using USB port with GpI/O

2.1.3 Defining benchmarks and parameters for a specific and dedicated control system.

So we assume the task of developing a specific system for robots and CNC control, where we could better control the system variables. Without relying on a PC, an operating system and all software layers that are today needed to generate and correct a trajectory during the movement, doing this by the motion feedback or some external variable such as a vision system for example. This system should be flexible enough to: introduce advanced control algorithms for several geometries, vision automatic positioning and trajectory systems, proximity control by voltage measurement on coated electrode welding, automatic torch height control of a plasma cutting, among others. For this we started a research to choose what the best development platform, and we found two viable options: the first would be a microprocessor system, based on micro-controllers in a similar way to OEM commercial systems but in order to be low cost; the second would be an innovative system based on FPGAs (high-capacity programmable hardware). We ended up choosing the second by flexibility, its affinity for frequency control and the large pool of useful applications already developed for the robotic systems.

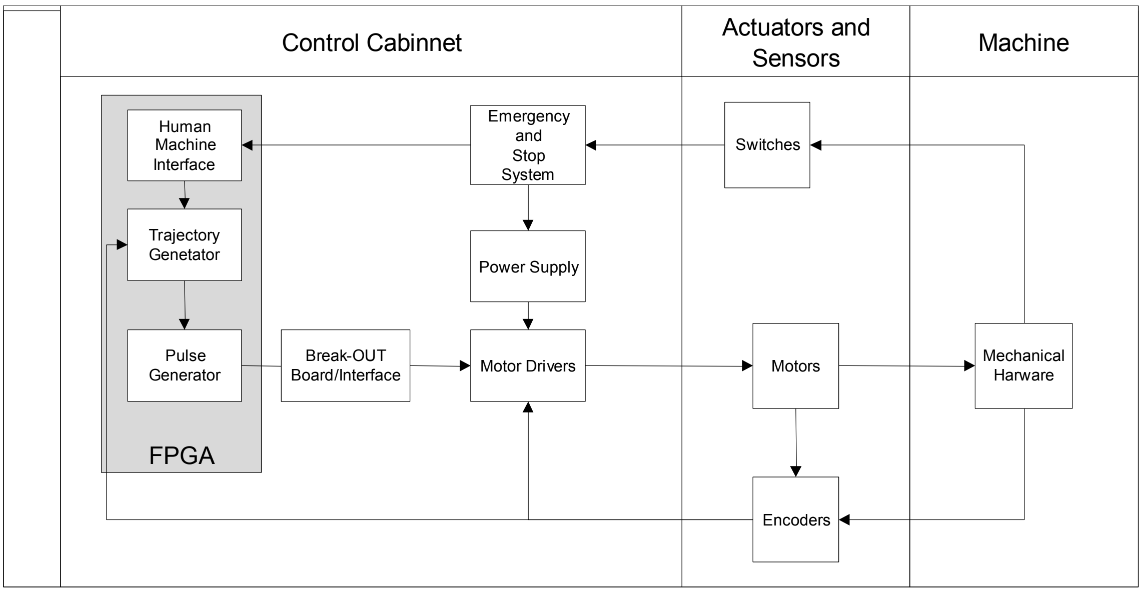

Figure 5. FPGA based CNC and robotics control system

2.3. FPGAs, CPLDs and LPDs

Below we will better define the characteristics that led us to choose the FPGA as development platform as well as their origin and history. The LPDs (Programmable Logic Devices) exist for some time and consist of a group of electronic devices capable of being programmed to assume the function of a logic circuit described in your code With its development were becoming more powerful to the point that being able to put a complete control system in its interior, in the concept called "system on a chip", thus giving us the flexibility to completely replacing a system without necessarily change the hardware.

Today there are two main types of LPDs, the CPLDs (Complex Programmable Logic Devices) and the FPGAs (Field Programmable Gate Arrays). The first group is much cheaper, has its structure based on a non-volatile EEPROM system, that is, its programming only changes when requested. That limits their working frequency and amount of logic gates in a single device; already the second group has its programming volatile and must be reprogrammed every time it is turned off, so it is necessary to an external memory or a specific area in your building containing the program memory on stand-alone systems. But he is able to contain a largest number of logic ports in the same device; it is able to work with a larger number of modules and arithmetic and works in a much higher frequency than the first.

For too long the LPDs were only used to perform discrete logic, or replace discontinued logic devices Today however, already exist FPGAs highly skilled in mathematical problems solving ,and had even able to replacing the DSP (Digital Signal Processors) in motor control applications as demonstrated by Le et al. (2007). Thus enabling the calculations for trajectory generation in real time for multiple axes, quickly and efficiently.

One of major LPD´s features is the ability to actually perform tasks in parallel, unlike a microprocessor that performs its tasks sequentially. Thus enabling a precise control of the time that their task will be executed, leaving the system highly deterministic. This makes the application of FPGAs as pulse generators for motor-drivers based on frequency signals (Step) much simpler than in a microprocessor system. Where control would be based on interruptions that would limit the number of axes to be controlled and the system would be little deterministic, because the math would be interrupted for the execution of the frequency signal routine. Another great advantage of the parallel system, when works with clock frequencies lower or equal to those of micro-controllers its have a much faster response.

Today we can also use a wide range of so-called virtual devices to include on the FPGA's codes. With even the possibility that we will not dwell on a specific manufacturer, simply the host chip has the needed features by the logic device so that it can be used. Thus we can abstract the hardware and basically thinking only in the logic and tasks that we want to run on the FPGA. This flexible and greatly facilitates the development stage, because a already tested solution can be acquired and incorporated into the system so simple and cheap, since it is logical (intellectual property) and non-physical components what we can buy a device that is to thousands of miles away and just receive it via e-mail. This abstraction of the hardware also allows us to use standard hardware and change only the FPGA's programming. So we can skip the step of prototyping of the physical system, using an already tested board that has the necessary parameters, greatly reducing development time and costs. Thus we can use the same hardware to implement very different control systems.

2.4. Low cost FPGA based control system

To develop our dedicated CNC and robots control system, we started from a low cost development board containing a very simple Xilinx CPLD CoolRunnerII 256K With a 1MHz working clock and programming interface to the CPLD. On this board we developed a controlled pulse generator for six axes with maximum output frequency of 50 KHz and a resolution of 28bit for the speed control and 16 bits for acceleration control. To control the motors we used the servo motor driver Geckodrive G320X which already has its frequency inputs (Step) and direction duly opto-isolated and compatible with the voltage levels of the CPLD. To simulate de mechanical hardware we use “Maxon” DC servo-motors coupled on 2000 pulse/revolution optical encoders. We set up a system where so inserted manually, via buttons on the board, the velocity, acceleration and number of pulses to be generated. So we tested the algorithm to generate pulses with success and already at the maximum speed equal to the parallel port but sixteen times smaller than the USB port (this to her working at 480Mbps with a maximum error rate at less than 4.76%. After the first tests we decided to increase the work clock frequency to 250MHz to and work with maximum output frequency in the range of 100KHz, as that are the maximum operating frequency supported by the input of the motor driver used, and so we can get the maximum error in velocity was lower than 0.04% in controlling six axes simultaneously. We also performed experiments with the output frequency of 4 MHz that is equal to the higher frequency than the USB external pulse generator owned by the laboratory is able to control and so we get a maximum error less than 1.57% in speed control. These results were achieved without using any advanced algorithms to reduce discretization errors or the linearity due to characteristic of simplicity of mathematical functions which the used CPLD was able to run natively.

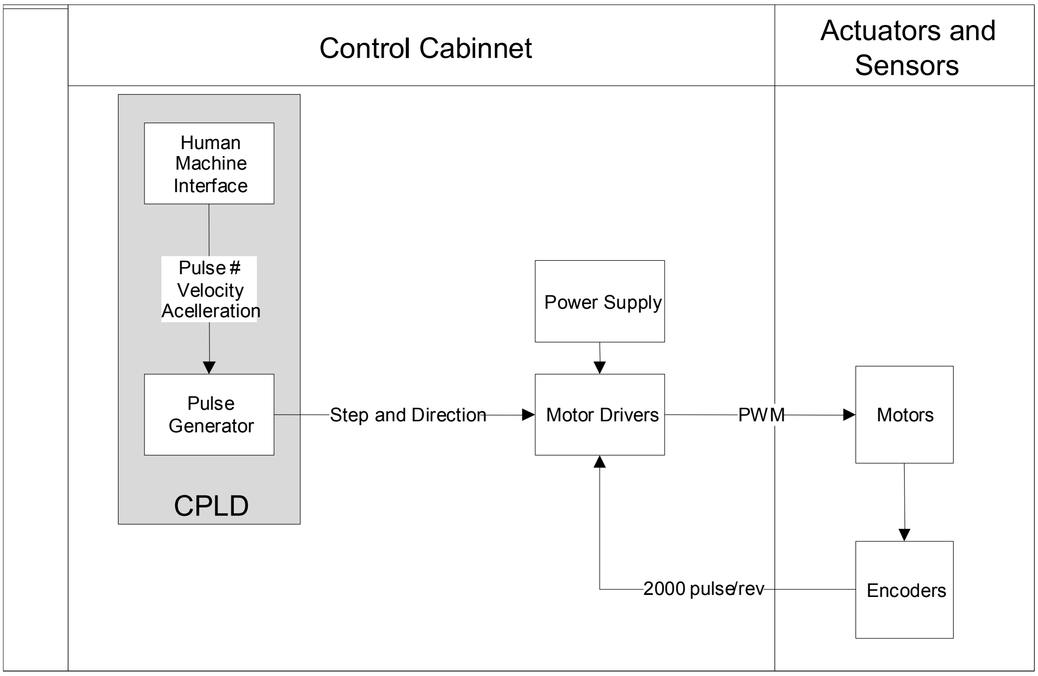

Figure 6. LPD based pulse generator development platform

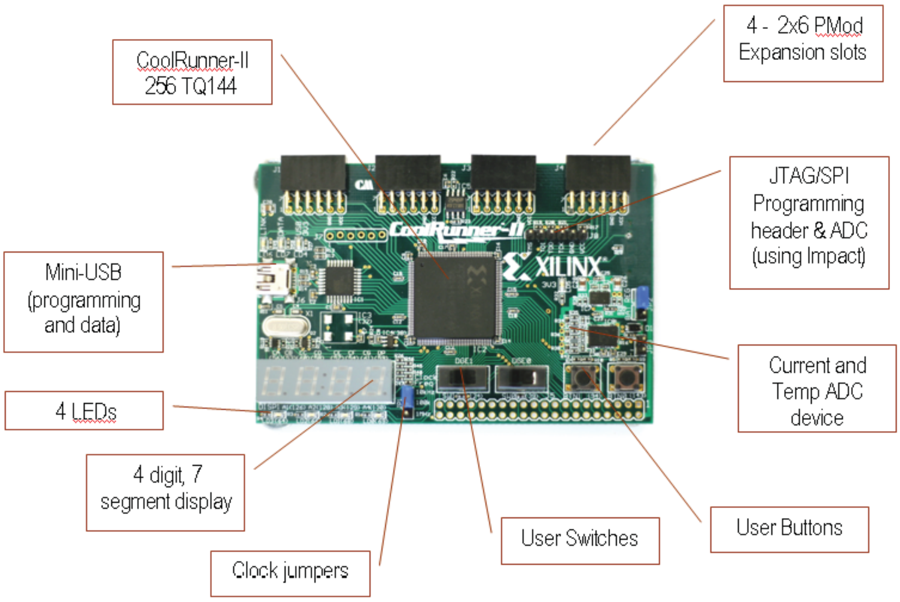

Figure 7. CPLD Board used on pulse generator development.

Upon finishing the pulse generator test, we migrated the project to the development board containing a Xilinx FPGA Spartan3, because the CPLD despite running well did not contain the arithmetic units required for the development of the trajectory generator. For this we transform the pulse generator in a virtual device and it fit into the FPGA's programming, along with an algorithm able to control the pulses generation in closed loop control. Thus enabling the trajectory generator know the exact position of the joint and make corrections to speed and acceleration on the fly, i.e. during the movement, and virtually extinguish the speed error ,now that the error will depend on the tuning of the system.

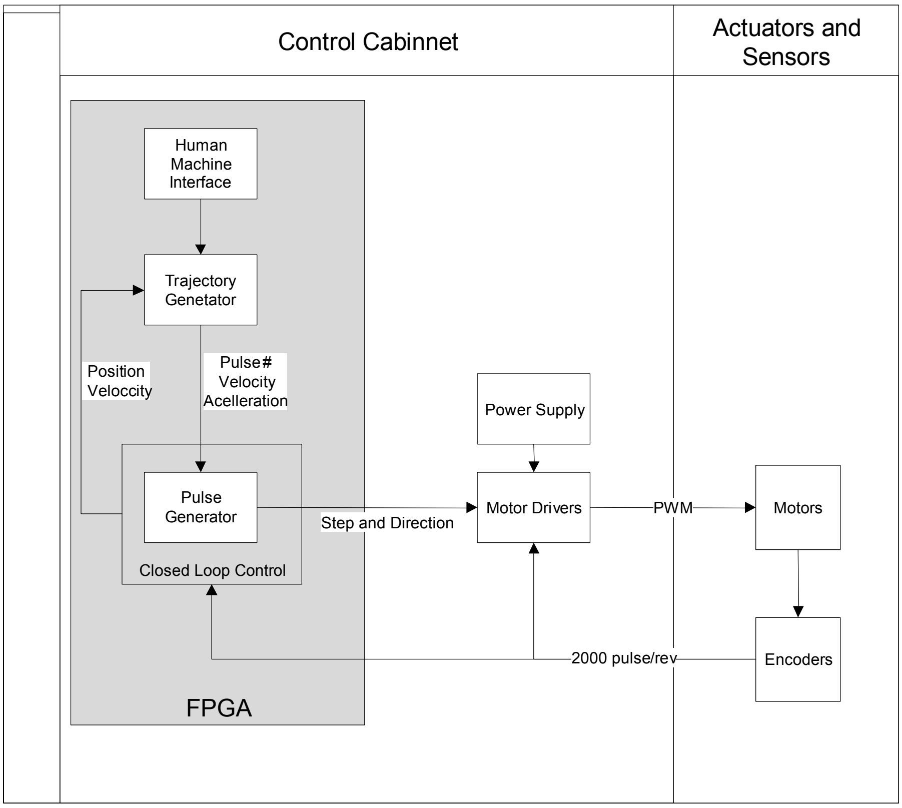

Figure 8. LPD based development platform.

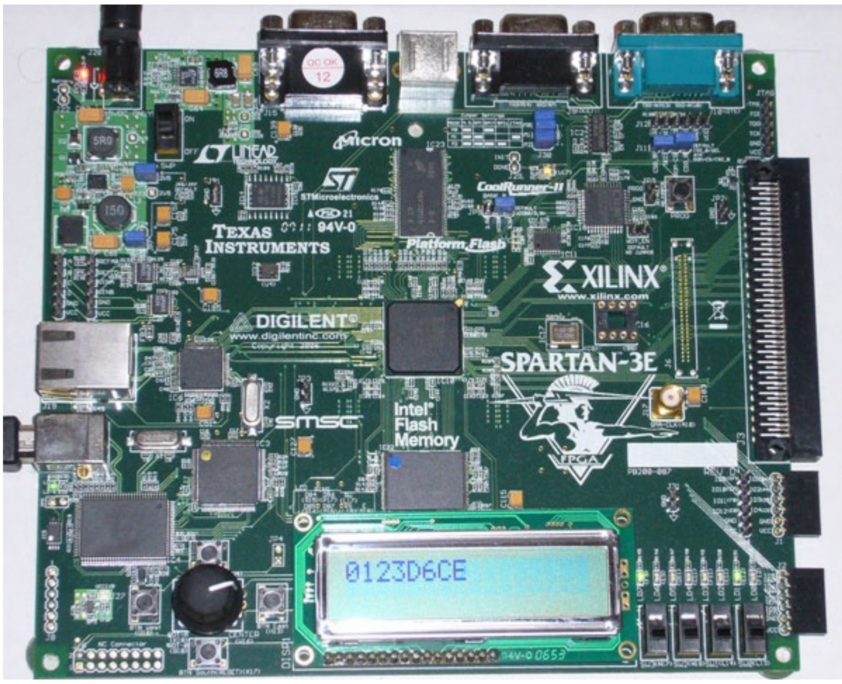

Figure 9. Spartan3 FPGA Board used on the Trajectory Generator development.

3. CONCLUSION

We conclude that the use of FPGAs for the development of a control system for CNCs and robots flexible is efficient and feasible. And that despite the apparent greater complexity of programming, set-up and system design; we could develop an extremely powerful and compact system, capable of controlling multiple axes in real parallelism in a deterministic way. We also conclude that the system cost is significantly reduced since the cost of a board containing the basic components for implementing the system is not much higher than that of an external pulse generator, thus eliminating the cost of a PC and including leaving the system ready for a stand-alone mounting.

4. ACKNOWLEDGEMENTS

We thank the LRSS and the Mechanical Engineering Department (DEMEC) of the Engineering School of UFMG.

5. REFERENCES

Cordeiro, J. F. and Kienitz M. K. ,2009, "DEVELOPMENT OF CONTROL CABINET FOR INDUSTRIAL ROBOTS ABCM Symposium Series in Mechatronics - Vol. 4 <http://www.abcm.org.br/symposiumSeries/index.shtml>

Le, N. Q. and Jeon, J. W., 2007. "An Open-loop Stepper Motor Driver Based on FPGA". International Conference on Control, Automation and Systems, COEX, Seoul, Korea.

Hackney, C., 2005, "FPGA Motor Control Reference Design". September 16, 2005, Application Note: Spartan and Virtex FPGA Families, <http://www.xilinx.com>

Xilinx, "CoolRunner II Evaluation Board Reference Manual" UG501 May 15,2008 http://www.xilinx.com

Meshram, U. D. and Harkare, R. R. 2010 "FPGA Based Five Axis Robot Arm Controller" International Journal of Electronics Engineering, 2(1), 2010, pp. 209-211

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.