A COMPARISON OF FPGA AND FPAA TECHNOLOGIES FOR A SIGNAL PROCESSING APPLICATION

Roberto Selow

Departamento de Engenharia Elétrica,
Universidade Positivo
Rua Prof. Pedro Viriato P. de Souza, 5300 -
81280-330 - Curitiba (PR) - Brazil
rselow@up.edu.br

Heitor S. Lopes, Carlos R. Erig Lima

Departamento de Eletrônica,
Universidade Tecnológica Federal do Paraná,
Avenida 7 de Setembro, 3165-
80230-901 - Curitiba (PR) - Brazil
hslopes@pesquisador.cnpq.br ,
erig@utfpr.edu.br

ABSTRACT

This paper presents a comparison between two technologies for reconfigurable circuits that are FPGA'se the FPAA's. The comparison is based on a case study of the area of industrial control. The results of simulations with both types of reconfigurable devices is presented. Several design issues are discussed, including the ease of implementation, accuracy, capacity, consumption, size, among others. Based on the case study, we present qualitative directions to choose the most suitable reconfigurable device for similar applications.

1. INTRODUCTION

For a long time, signal processing was accomplished by microprocessors and digital signal processors (DSPs), with almost no exception. When reconfigurable computing raised powerful devices, such as Complex Programmable Logic Devices – CPLD´s and Field Programmable Gate Arrays – FPGA´s, they become an alternative for signal processing applications [1].

More recently, Field Programmable Analog Arrays – FPAA´s [2] devices appeared in this scenery as an interesting alternative, but still limited due to small number of suppliers. On the other hand, sophisticated development systems make easy projects with FPAA´s for specific applications. Therefore, it is expected that there will be a growing interest in using FPAA´s for engineering projects [15].

For some signal processing applications, FPGA´s are indicated but, for others, FPAA´s are clearly the most appropriated alternative. However, there are applications where both technologies can be used. In these cases a more detailed study of the advantages and drawbacks of each reconfigurable technology should be done. This paper intends to shade a light in this issue, by comparing FPGA and FPAA, using a case study the simulation of an industrial control system.

Considering that this sort of comparison is not found in current literature, we believe that this work can provide interesting subsidies for engineers, designers and researchers in reconfigurable computing. It is important to mention that this study is done for a specific problem, and for other signal processing problems other design parameters, not analyzed here, may be relevant.

2. STATE OF THE ART IN RECONFIGURABLE COMPUTING

The current status of development of FPAAs that are in the market is based on devices built from operational amplifiers with a few elements per chip. Recent studies exploring the use of floating-gate devices as the core element in the programmable signal processing FPAA [3]. Studies with this technology show a large number of elements per chip, high precision and the high efficiency gain that can be obtained when compared to DSP processors (Figure 1).

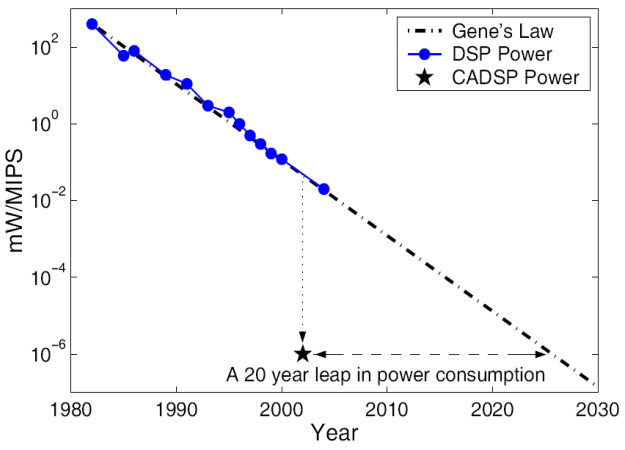

Fig. 1. Data from [4] showing the power consumption trends in DSP microprocessors along with data taken from a recent analog, floating–gate integrated chip developed by the CADSP team [5, 6, 7].

The other important group of programmable devices is that of FPGAs. The high performance of these devices is known and according to Berkeley Design Technology Inc. (BDTI) which regularly publishes analysis comparing the performance between FPGAs and DSPs, FPGAs have reduced the cost per channel for more than 95% compared to the DSP processors [8]. In addition, the power consumption of FPGAs is known as high [9]. New releases of the industry, however, show the concern of manufacturers in the reverse situation is considerable and results can already be observed [10].

3. METHODOLOGY

This work is based on a case study of an industrial control system. Two solutions are proposed, under the same directions, using FPGA and FPAA. The implementations are then compared according some design parameters.

The control system focused in this work consists of a classical configuration composed by a process and a controller, as shown in Figure 2. The choice of a second-order process and the PID (Proportional-Integral-Derivative) controller was due to the large number of industrial applications where this type of system can be found [11], [12].

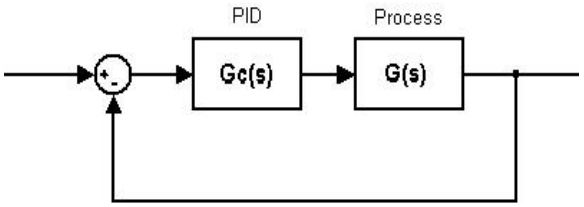

Fig. 2. Blocks diagram of a simple industrial control system

The process *G(s)* corresponds to the transfer function of a second-order process given by equation (1). The transfer function of the controller $G_c(s)$ is given by equation (2), in which the gains of the proportional, integral and derivative elements are, respectively, K_p, K_i, K_d and Eo and Ei as the input and output signals.

$$G(s) = \frac{Eo(s)}{Ei(s)} = \frac{1}{as^2 + bs + c} \tag{1}$$

$$Gc(s) = K_p + K_i \frac{1}{s} + K_d s \tag{2}$$

3.1. Implementation in FPGA

The implementation of equation (1) in a FPGA device requires the application of the Z transform to the equation[1]. The result of such transformation is equation (3) and the corresponding differences equation is shown in (4) – that will be useful in the development of the digital hardware.

$$G(z) = \frac{Eo(z)}{Ei(z)} = \frac{0.006912z + 0.006641}{z^2 - 1.873z + 0.8869} \tag{3}$$

$$\begin{aligned} Eo(k) = {} & 0.006912.Ei[k-1] + \\ & 0.006641.Ei[k-2] + 1.873.Eo[k-1] - \\ & 0.8869.Eo[k-2] \end{aligned} \tag{4}$$

Since the input and output signals of the FPGA are strings of bits, it is necessary the conversion of the values of the difference equation (4) from real to integer. This conversion is shown in Table 1 and it is accomplished simply by multiplying the real value by constant 1000 and truncating off the decimals. In equation (4), the terms [k-1] are subject to one time delay while the terms [k-2] are subject to two. After being delayed, the signals are multiplied by their respective gains. The final blocks diagram with integer values is shown in Figure 3.

Table 1. Conversion from real to integer in the difference equation

Real	Integer
Ar = 0.006912	*Ai = 69*
Br = 0.006641	*Bi = 66*
Cr =1.873	*Ci =18730*
Dr = 0.8869	*Di = 8869*

The PID controller is constructed in such a way to operate in parallel with the process. This is done not only to take advantage of the parallelism offered by the reconfigurable device, but also, to achieve a faster response in the simulation. The transfer function of the controller is given by equation (5). The proportional $P[k]$, integral $I[k]$ and derivative $D[k]$ actions are given, respectively, by equations (6), (7) and (8)[2]. The blocks diagram of the PID controller is shown in Figure 4.

[1] For obtaining a Z transform we used the indexes a=1, b=1 and c=1 for equation (1) and a sampling period of 0.12s.

[2] The PID gains used are Kp=1, Ki=0.5 and Kd=0.00025.

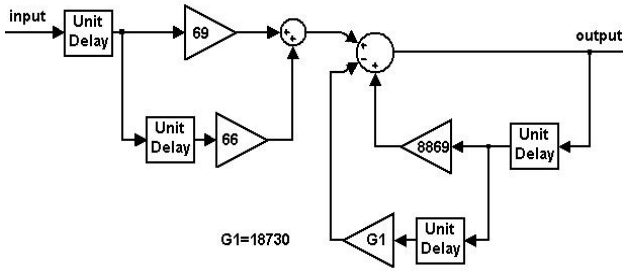

Fig. 3. Blocks diagram of the difference equation of the transfer function $G(s)$

$$Gc(k) = P[k] + I[k] + D[k] \tag{5}$$

$$P[k] = K_p.e[n] \tag{6}$$

$$I[k] = I[k-1] + \frac{K_p.T_S}{2.T_i}(e[k] - e[k-1]) \tag{7}$$

$$D[k] = \frac{(pT_S - 2)}{(pT_S + 2)}.D[k-1] + \frac{2.K_p.T_d}{T_S.(pT_S + 2)}(e[k] - e[k-1]) \tag{8}$$

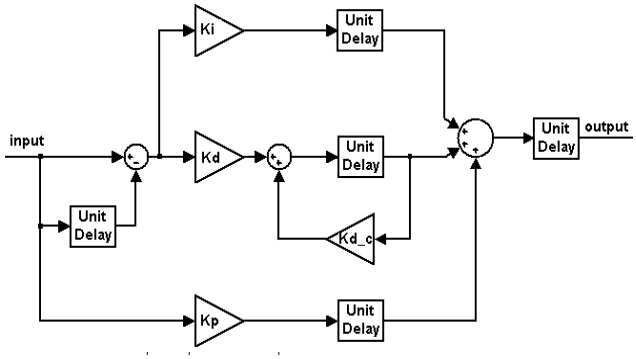

Fig. 4. Digital version of the PID controller referred as $Gc(s)$

The FPGA device used in this implementation was the Altera's EP2C35F672C6. This device has a total of 33,216 logic elements (LE) and only 13% (4,472 LE's) of the device capacity was used in the whole implementation. The development was done purely in VHDL using the Quartus II development system [13].

The final blocks diagram of the implementation in FPGA is shown in Figure 5. Although each of the four blocks of the system have a specific function, it was possible to group them into a single FPGA device. Notice that the block referred to as **sim-enabler** is responsible for generating the PID controller gains as well as the input step signal entering the system.

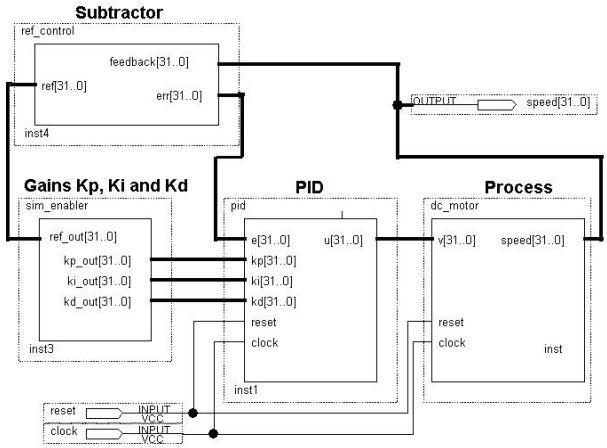

Fig. 5. Complete blocks diagram of the FPGA implementation

3.2. Implementation in FPAA

The implementation of process *G(s)* in the FPAA is based on equation (1), without the need of any further transformation. The building blocks of the FPAA used in this implementation are: two integrators, two adders and three programmable-gain amplifiers, with gains *1/a*, *b/a* and *c/a* as shown in [11]. Figure 6 shows the blocks diagram of the analog version of the process.

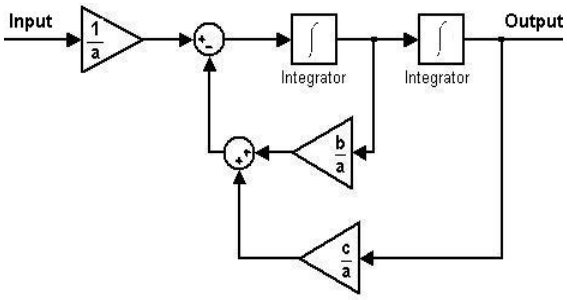

Fig. 6. Blocks diagram of the transfer function of the process

The PID controller of equation (2) is implemented as a sum of the proportional, integral and derivative actions over the error signal entering the controller. The gain of each of

these actions is given by the specific block in the implementation. The corresponding blocks diagram of the controller is shown in Figure 7.

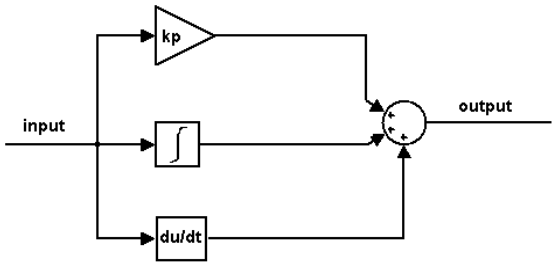

Fig. 7. Analog version of the blocks diagram of the PID controller

The FPAAs supplied by Anadigm [14] has the advantage of having multiple functional blocks ready to use and, therefore, the analog project is straighforward. Besides, the implementation of the PID in the FPAA was done using the AnadigmPID development system for PID controllers 14]. This software is freely available in the AnadigmDesigner2 development system. The designer needs only to set values for the gain of each block and everything else is automatically done.

The FPAA device used in this implementation was the Anadigm's AN231E04. Two FPAA devices were used for the whole implementation, each of them having eight operational amplifiers, 32 capacitors, 4 Successive Approximation Register (SAR's) that performs 8-bit analog-to-digital signals conversion and 4 comparators. The PID block was implemented in the first FPGA using five operational amplifiers and 15 capacitors. The process block was implemented in the second FPGA with eight operational amplifiers and 26 capacitors. Therefore, most of the FPAA's resources were used in the implementations.

Due to the gain limitations of each analog block, more than a single block was necessary to achieve the desired behavior in the simulations, as shown in Figure 8.

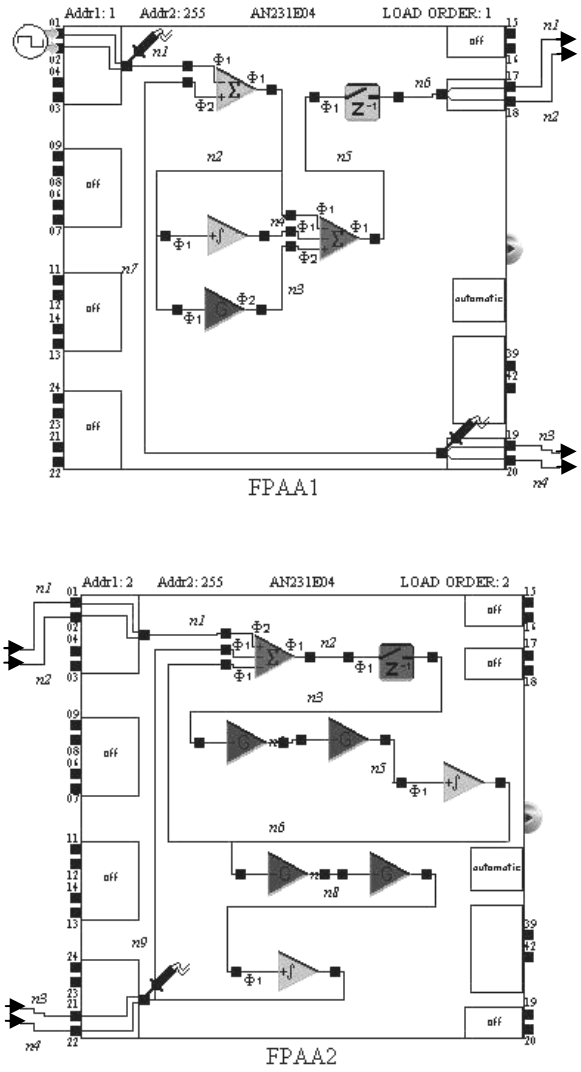

Fig. 8. Simulated circuit implemented in two FPAAs

4. SIMULATIONS AND RESULTS

The simulation of the FPGA implementation was done by applying a unity voltage step to the input of the system. The resulting output of the FPGA is shown in Figure 9.

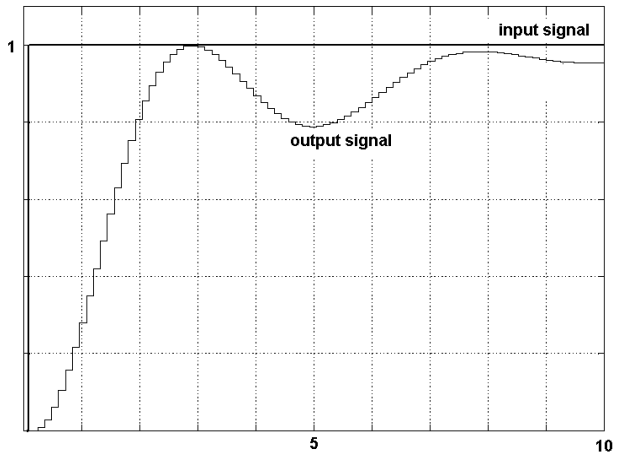

Fig. 9. Simulation of the FPGA implementation using Quartus II and plotted with a graphics software.

The second part of the simulations was done with the FPAA implementation. A 0.1 V step was applied to the input of the FPAA. This small amplitude was necessary to avoid saturation of the system and thus allow adequate functioning of the FPAA. The result of this simulation is shown in Figure 10.

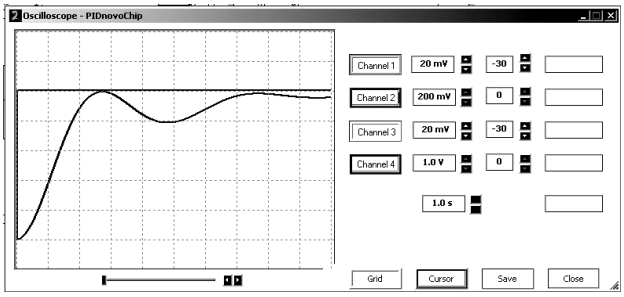

Fig. 10. Simulation of the FPAA implementation using AnadigmDesigner2

5. DISCUSSION

Simulation results from both implementations were quite similar, suggesting their electronic equivalence. However, there are several design parameters that are worth to analyze, as follows.

Regarding the expressiveness of the models implemented in the reconfigurable devices, the FPAA approach is closer to the mathematical model than the FPGA approach. In the FPGA, the coefficients a, b and c of the transfer function as well as the gains K_p, K_d, and K_i of the PID controller are not explicit in Equations (5) and (6). This is due to the application of the Z transform and further conversion to integer numbers. Overall, the procedure for implementation in the FPGA is much more complex than in the FPAA. Furthermore, the former requires the use of signal converters (analog-to-digital and digital-to-analog), whereas the later can cope with analog signals directly, without the need of auxiliary devices.

The capacity of each device can be an important limiting factor in electronic designs. The FPGA approach used only 13% of the capacity of the device. On the other hand, the FPAA approach required two devices for the same project. It is a matter of fact that nowadays there are many options of FPGA devices, but the same does not holds for FPAA devices, since this technology is relatively recent. Although a direct comparison is not possible between technologies, generally speaking, current FPGA devices have large capacity, but FPAAs available have very limited capacity. Of course, FPGAs have an obvious competence to deal with digital signals, while FPAAs are specially suited for dealing with analog signals.

Any design that employs reconfigurable devices should use a development platform. For both, FPGA and FPAA projects there are software tools capable of offering many resources to the designer, from circuit edition to realistic simulation. The tools available for the design of FPGAs are robust and have a great facility for the development of large projects. This feature is due the hierarchical design that the projects are develops. With respect to the configuration tool available for FPAAs, which is a tool provided by the manufacturer Anadigm, it offers features that include the use of existing analog blocks. This feature simplifies the programming and makes the work of designer simple.

The power consumption of the implementations was estimated using the development tools. For the FPGA approach it is estimated around 113 mW. For the two FPAAs the power consumption is estimated, respectively, in 105±32mW and 158±47mW. At first sight the FPGA presents advantage over the FPAA, but it should be taken into account that the FPGA needs external devices that will increase the overall power consumption.

The maximum operation frequency of the devices may limit their use in some applications. The FPAA device used in this study runs at 16MHz and, for most cases, these devices can process signals up to 2MHz. FPGAs, in general, can operate at much higher frequencies, in the order of 500MHz. However, the actual running frequency is strongly dependent of the implementation and the optimization done by the development software.

Another aspect of great importance that should be considered is the possible values of gain between the reconfigurable devices. For FPAAs, the characteristics of the construction of devices allow gains limited to a range of values that can vary, for example, from 0.01 to 100 with increments of 0.01. With respect to the FPGAs, the gains are achieved through mathematics operations with integers and can reach values higher than 32 bits, equivalent to very high gains in a FPAA.

6. CONCLUSIONS

Reconfigurable computing has provided solutions to countless engineering problems. When using reconfigurable devices, namely FPGA and FPAA, the electronic designer has to meet the project constraints, taking into account the constructive features of the devices. The several issues raised in the electronic design, such as difficulty of implementation, capacity or power consumption, will suggest the use of a given device.

Due to the limited availability of FPAA devices, it is difficult to establish a detailed study including all possible application areas of both approaches. As mentioned before, FPGAs have a clear applicability in digital processing, while FPAA, in analog processing. Notwithstanding, we believe that the main design parameters that determine the choice of one of the reconfigurable technologies can be summarized in Table 2. In this table, the design parameters are ranked qualitatively into four classes for both implementations. Finally, it is important to stress that the comparison was done keeping in mind the case study described in this work and, for other applications the results can be somewhat different.

Table 2. Qualitative comparison of reconfigurable technologies.

Design parameter	FPGA	FPAA
Facility of implementation for signal processing application	+	+++
Capacity	+++	-
Development tools	+++	+++
Power consumption	+	-
Running frequency	+++	+
Gain	+++	+

+++ very good ++ good + average - weak

As future work aims to make an extension of this research through the broader study of the different devices FPGAs available in the market applied to the case proposed. It is true the development of comparative studies between the reconfigurable devices in other areas related to signal processing, of which we mention applications involving filters and adaptive filters.

7. REFERENCES

[1] S. Hauck and A. DeHon, *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing*, Morgan Kaufman, 2008.

[2] S. Bains., "Analog's answer to FPGA opens field to masses," EE Times, vol. 1510, 2008, pp.1.

[3] T.S. Hall, C.M. Twigg, P. Hasler, D.V. Anderson, *Application performance of elements in a floating-gate FPAA*, Proceedings of the 2004 International Symposium on Volume 2, Issue , 23-26 May 2004 Page(s): II - 589-92 Vol.2

[4] G. Franz, "Digital signal processor trends," IEEE Micro, vol. 20, no. 6, pp. 52–59, Nov–Dec 2000.

[5] R. Ellis, H. Yoo, D. Graham, P. Hasler, and D. Anderson, "A continuous-time speech enhancement front-end for microphone inputs,"in Proceedings of the IEEE International Symposium on Circuits and Systems, Phoenix, AZ, 2002, vol. 2, pp. II–728 – II–731.

[6] P. Hasler, P. Smith, R. Ellis, D. Graham, and D. V. Anderson, "Biologically inspired auditory sensing system interfaces on a chip," in 2002 IEEE Sensors Conference, Orlando, FL, June 2002, Invited Paper.

[7] P. D. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, "Mel–frequency cepstrum encoding in analog floating-gate circuitry," in Proceedings of the International Symposium on Circuits and Systems, Phoenix, AZ, May 2002, pp. IV–671 – IV–674.

[8] Altera Sponsored White Paper, "FPGAs for High-Performance DSP Applications", FPGA and Programmable Logic Journal, May 2005

[9] Koji Gardiner, "Comparing Power Consumption of FPGAs with Customizable Microcontrollers" FPGA and Structured ASIC Journal, March 18, 2008.

[10] STRATIX IV FPGAs: the lowest power high-end 40-nm PFGA, URL: http://www.altera.com/products/devices/stratix-fpgas/stratix-iv/overview/power/stxiv-power.html, 2009.

[11] OGATA, K. Modem control engineering. 4. ed. New Jersey: Prentice-Hall, 2001.

[12] K. Åström, T. Hägglund, "PID Controllers: Theory, Design, and Tuning," in *Instrument Society of America*, Research Triangle Park, NC, USA, 2nd, 1995.

[13] Altera Corporation (2006), URL: http://www.altera.com, 2006.

[14] Anadigm (2008), URL: http://www.anadigm.com, 2008.

[15] Jean-Marc Galliere, A control-systems FPAA based tutorial, 2nd WSEAS/IASME International Conference on EDUCATIONAL TECHNOLOGIES (EDUTE'06) Bucharest, Romania, October 16-17, 2006.