A DIGITAL PID CONTROLLER FOR REAL TIME AND MULTI LOOP CONTROL: a comparative study

L. SAMET, N. MASMOUDI, M.W. KHARRAT, L. KAMOUN

ELECTRONIC and COMPUTER SCIENCE LABORATORY
SFAX ENGENEERS SCHOOL, BP W 3038 SFAX TUNISIE
Tel. 216-4-274088 Fax. 216-4-275595

ABSTRACT:

We introduce in this paper a comparative study for three digital PID (Proportional Derivative Integral) controller architectures. So designed this PID can be used for real time applications where the command speed is a crucial parameter. For multi-loops systems this controller allows tuning parameters calculations in the minimal time delays. The implementation of the digitized PID algorithm in one FPGA (field Programmable Gate Array) is designed as a specialized computation processor. This FPGA PID circuit can be embedded into a process calculator (PLC, Microcomputer, and microprocessor board...). This hardware implementation can be done in three different architectures: Serial, Parallel or Mixed.

In all cases this kind of electronic implementation is very fast response than a software. The digital PID so designed is based essentially in architectures including state machines, multipliers, adders and some others logic circuits.

The three electronic architectures are described. These FPGA-PID are integrated in Xilinx FPGA:

- **For parallel architecture: the XC4013, which has 576 CLBs (Configurable Logic Bloc), is used.**
- **For serial architecture: the XC4005, which has 196 CLBs, is used.**
- **For mixed architecture: the XC4006, which has 256 CLBs, is used.**

The occupation rate is about 90%. The calculation time depends on the architecture and varies from 0.2 µs to 3.5 µs.

The obtained results with FPGA-PID are confronted to those obtained by the MATLAB-SIMULINK simulator.

Keywords: Control, PID, FPGA, digital, architecture, implementation.

1. INTRODUCTION

The modern digital control systems require more and more strong and fastest calculation components. This type of elements becomes yet indispensable with the utilization of some new control algorithms control like the fuzzy control, the adaptive control, and the sliding mode control...

Although the PID controllers are the oldest they represent the most used controllers in the industrial control systems. The simplicity tuning and the efficiency of this algorithm in 90% of cases justify this use. To decrease the influence of the disruptions, industrial process is generally tuned. In this case, regulators try to maintain the output process to a value near the operator provided one. However, the modern automation, with the data processing developments and especially in the micro-computer sciences, make more and more call to the digital methods for the industrial process tuning. In fact, the numeric regulators enable the digital elaboration of tuning algorithms, which replace in this way the analog classics regulators [3].

2. EQUATIONS OF THE DISCREET PID

According to the study done in [9] the digitized PID equation is brought back to:

$$L_K = l_{K-1}+e_K(K_P+K_D+K_I/2)+e_{K-1}(K_I/2-K_P-2K_D)+e_{K-2}K_D \quad (1)$$

The figure 1 gives the loop of one variable discreet digital loop.

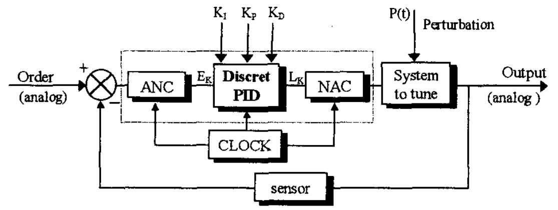

Figure 1. Position of the PID in single variable tuning loop.

3. THE DIGITAL PID ARCHITECTURES

3.1. The parallel architecture

For the design of a fast PID the fastest architecture is known as ASAP (As Soon As Possible) architecture [4]. This architecture requires the availability of all calculation operators in each phase. For this architecture very important material resources are requested.

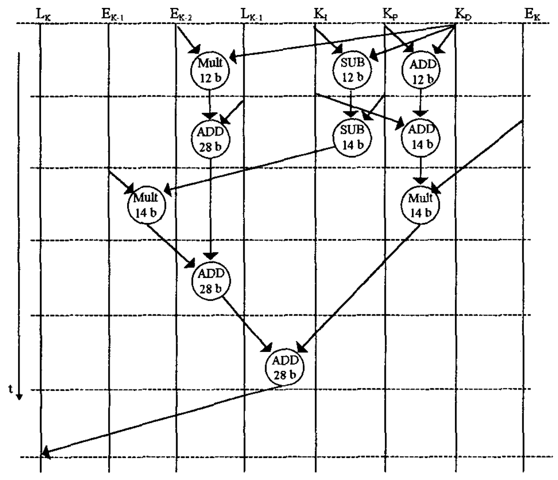

Figure 2. Parallel PID Architecture.

The required operator's are available for the algorithm calculation. The architecture operator's is completely combinational logic in this case [4]. The figure 2 gives this architecture.

The occupation rate of such PID is important due to the big number of used operators (3 multipliers and 7 adders). Indeed, the fastest combinatorial multiplier is the one achieved by the WALLACE tree technique. According to the study done in [1][4][9] this multiplier is the fastest combinatorial multiplier. The principle is based on cascades of one-bit adders' stages, who allows to total the partial products of the multiplication. The figure 3 gives the 6 bits (for schema simplification) multiplier architecture.

In the fact that it is completely combinatorial, this multiplier occupies the widest space on the FPGA chip. It occupies, in an FPGA, about 160 CLBs (Configurable Logic Bloc) for 12 bits and 200 CLBs for 14 bits.

The four 12 bits adders and three 28 bits adders requires about (24+42=66) CLBs [7][9]. So, the PID will be implemented in one FPGA having more than 500 CLBs.

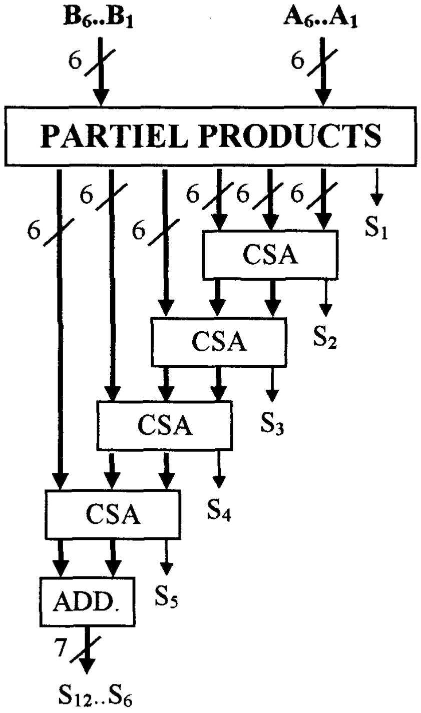

Figure 3. WALLACE tree multiplier.

A fast evaluation for the combinational logic PID performances allows us to deduce its execution time. Indeed, the calculation time of such circuit would be equal to its propagation time. According to the equation (1), this is the time of two 12-bit additions (T_{A12}); one 14-bit multiplication (T_{M14}) and one 28-bit addition (T_{A28}).

$$T = T_{A12} + T_{A12} + T_{M14} + T_{A28}$$

The result of [7] and [9], for the fastest operators, allowed us to deduce respectively:

$$T \approx 25 + 25 + 110 + 40 = 200 \text{ ns}$$

3.2. The serial architecture

The serial architecture is achieved by a design based on the ALAP (As Last As Possible) architecture [9]. This design is obtained by using only two sequential logic operators: one adder and one multiplier. A finite state machine is used to control the execution of PID algorithm operations. The figure 4 gives this architecture.

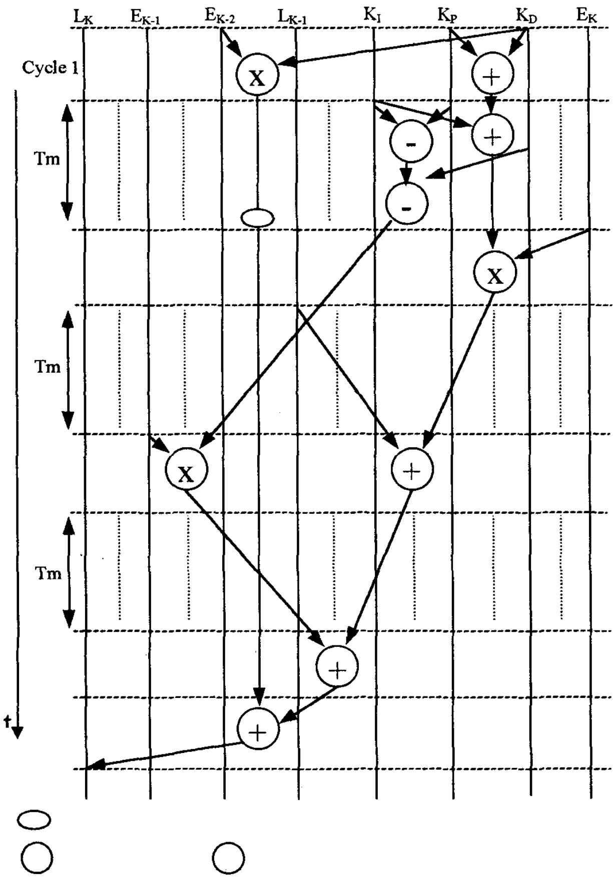

Figure 4. Serial PID Architecture.

The used adder is the RCA (Ripple Carry Adder).

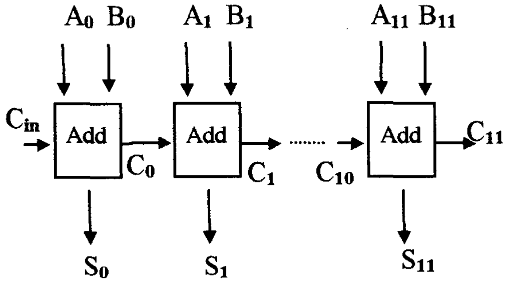

Figure 5. The RCA Architecture.

For the implementation in an FPGA circuit, this adder architecture is the most optimized. The best results for adder are achieved by using the fast carry logic technique available in every CLB and by an adequate logic distribution (placement) in the FPGA [5][6][7]. The 28-bit addition is done in 36ns. The implementation requires 14 CLBs.

The used multiplier has an improved serial parallel architecture. By the use of CSA (Carry Save Adder) cells [11] this multiplier calculation time and occupation rate can be modulated as needed [10]. For the fastest configuration we have the WALLACE multiplier and for the slowest architecture we have the serial one. The multiplier architecture is given by the figure 6.

So designed this multiplier require 72 CLBs and 4 clock cycles for the calculation of one 14x12 bits operand multiplication.

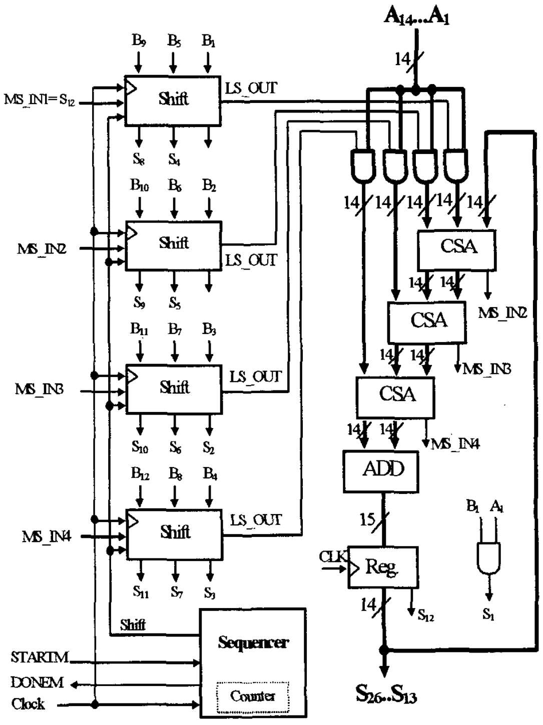

Figure 6. Serial parallel multiplier Architecture.

3.3. The mixed architecture

To improve the speed and minimize the cost while offering clearly good performances, one mixed architecture is used that includes essentially: one combinational logic multiplier, two adders and one accumulator. One finite state machine is in this case necessary to manage the whole exchanges and data transfer operations. The figure 7 gives the adopted architecture.

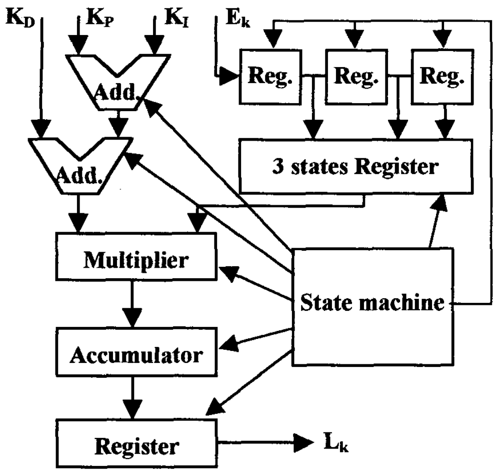

Figure 7. Mixed PID Architecture.

The adders and the accumulator are designed by the use of the ripple carry adder (RCA) architecture. For one 12 bits operation 25 ns are necessary. The rate of occupation is 6 CLBs for a 12-bits adder.

A 28-bits adder followed-up by storage register composes the accumulator essentially. The implemented architecture stills the RCA one. The occupation rate is 42 CLBs; the propagation time is 40 ns.

We used one 14*12 bits multiplier. The multiplier architecture is the WALLACE tree one. 170 CLBs are so necessary for implementation. The propagation time obtained after temporal simulation is near 110 ns.

For the realization of finite state module, XABEL language is used that enables, starting from different state descriptions, to provide one sequential circuit. The state machine was designed in order to generate the different necessary states for the PID running. This circuit allows executing the different steps described by the running PID state diagram, to provide finally L_K.

4. THE PID IMPLEMENTATION

The PID design is achieved by the means of the specialized software WORKVIEW® of Viewlogic Company. The routing and the placement of PID is done for the circuits:

- XC4013PQ240-4, which has 576 CLBs and 192 inputs /outputs for the parallel architecture.
- XC4005PC84-4, which has 196 CLBs and 192 inputs /outputs for the serial architecture.
- XC4006PC84-4, which has 256 CLBs and 61 inputs /outputs for the mixed architecture.

The choice of those part types is the best to have an optimal implementation. In fact, for a largest part type the occupation rate is very weak and the number of occupied (not packed) CLBs is very high. In this case a big CLBs number is partially used.

The relative informations about the PID implementations are extracted from the routing report. The implementation required:

- For parallel architecture: 512 CLBs for about 88% occupation rate.
- For serial architecture: 176 CLBs for about 89.8%.
- For mixed architecture: 225 CLBs for about 87% occupation rate.

Those rate sill high results of the sustained attention agreed to logic FPGA blocs' connections in the placement. Indeed, the disposition of these CLBs facilitates the distribution of registers content, which follows the privileged data circulation direction proper to the XC4000 family. Some constraints were introduced on the location of inputs, outputs and control lines to optimize the implementation.

5. THE PID SIMULATION

After the functional simulation, which allows verifying the right PID and logic working, a temporal simulation is processed. Indeed, after the PID routing some delays are introduced in the circuit.

The performances expected for the PID module has normally a direct relationship with the imposed time constraints and they are expressed in the technological specification [8]. The realization of PID circuit for serial and mixed architecture being synchronous to a basic clock, the execution time to directly a multiple of this clock cycle.
This clock cycle is dependent of the:

- execution time or operators propagation;
- the finite state machine cycles;
- Some delays added in the routed circuit.

For the parallel architecture the calculation speed is fixed by the propagation time through circuit logic layers.

To evaluate the PID execution time the temporal simulation is used for the worst case and for the critical values inputs (FFF and 000).

a) Parallel architecture
The calculation time supplied by this PID architecture is 210 ns. This time conform to a clock period given by temporal simulation. For this architecture the calculation time is:

$$\mathbf{T_{cPID} = 220\ ns}$$

b) Serial architecture
For this architecture the state machine includes 28 states executed in 28 clock cycles. The minimal simulation clock period is 120 ns. So the calculation time is :

$$\mathbf{T_{cPID} = 28 * 120 = 3360\ ns}$$

c) Mixed architecture
For this architecture the state machine includes 6 states executed in 6 clock cycles.
The maximal clock frequency that could pilot our PID is determined by temporal simulation. It is around of 8.7 MHz, which is equivalent to a period of 115 ns. By where:

$$\mathbf{T_{cPID} = 6 * 115 = 660\ ns}$$

6. THE PID VALIDATION

After the simulation for the three architectures a validation is processed.

The validation of the PID circuit consists in the checkup of PID behavior in one tuning loop. To proceed to this verification, first and second order systems are simulated. Then the outputs of FPGA-PID are compared with the one provided by MATLAB.

The architecture of the experimental loop is given by the figure 8.

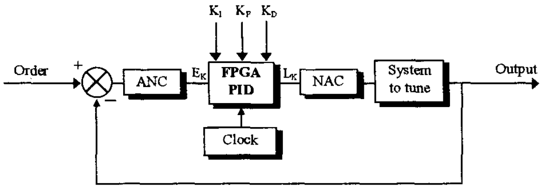

Figure 8. Loop tuning architecture.

The simulated first order system equation:

$$\frac{S(p)}{I(p)} = \frac{1}{1+0.6p} \qquad (2)$$

The second order system equation simulated:

$$\frac{S(p)}{I(p)} = \frac{1}{1+2Zwp+p^2/w^2} = \frac{1}{1+0.6p+p^2} \qquad (3)$$

For the two previous systems, the error curves of the PID regulator are plotted, the reference being the theoretical curves obtained by MATLAB.

For the three architectures the obtained validation results are similar. The above conclusion can be considered true for all of them.

The analysis of different experimental curves and of error ones show that the FPGA-PID thus designed presents conform answers to those provided by the theoretical simulation.

The error on the commands is not very important, less than 0.5% in steady state.
We notice that the error on the command of second order system is higher than to the one of first order.

Indeed, the system of second order amplifies the error, which is du to the quantification that is done in our case on 12 bits.

These errors are in fact, more important in the transient state than in the steady state.

7. CONCLUSION

Playing the role of a specialized processor, a digital PID circuit will integrate efficiently process calculators and make real time control easier.

Generally the PID applications in the process control don't require fast response times.

However the speed becomes interesting if the PID is used to do the control on several loops in real time process.

For the three previous architectures this comparative study show that the choice of implementation elements is very important to fix the performances of the target circuit. We note that the best architecture adapted for FPGA implementation is the mixed one [4]. In fact, this architecture presents good compromise between cost and speed.

References

[1] V.G. OKLOBDZIJA, D. VILLEGER
Improving Multiplier Desing by Using Improved Column Compression Tree and Optimized Final Adder in CMOS Technology.
IEEE TRANSACTIONS OF VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, Vol. 3, N° 2, Juin 1995.

[2] A. VODA, I.D. LANDU
A Method of the Auto-calibration of PID Controllers.
Automatica, Vol. 31, N° 1, pp. 41-53 1995.

[3] FARGEON C.
Commande numérique des systèmes.
444 p, 1986, Masson.

[4] N.H.E.WESTE, K. ESHRAGHIAN
Principles of VLSI CMOS Design, a System Perspective.
713 p, 1994, Edition ADDISON WESLEY.

[5] J.ROSE, EL A. GAMAL and A. SANGIOVANNI-VICENTILLI, "Architecture of Field-Programmable Gate Array" in *Proceeding of the IEEE Vol. 81 N° 7 JUL. 93.*

[6] S.TRIMBERGER, "A Reprogrammable Gate Array and Appli." in *Proceeding of the IEEE Vol. 81 N° 7 JUL. 93.*

[7] L. SAMET, M. BEN AYED, N. MASMOUDI, KAMOUN L.
Intégration des additionneurs numériques
Architectures, simulations et comparaison. CMGE'95, Tunisie.

[8] J.P.CLAVEZ, Spécification et Conception des ASICs.
Paris 1993, Edition MASSON.

[9] L. SAMET, "Etude de l'intégration électronique en technologie FPGA d'un algorithme de contrôle de processus: le PID" Thèse Docteur Ingénieur, ENIS-TUNISIE, décembre 1996.

[10] HAYES J.P. , Computer Architecture and organisation, MC Graw-Hill, 2$^{\text{ème}}$ édition 1988.

[11] MASMOUDI N., SAMET L., KAMOUN L. "Intégration des multiplieurs numériques" JTEA' 8-9 Novembre 1996. Nabeul Tunisie.