A New Synchronous-Asynchronous Mixed Pipeline Architecture with Clock-Gating

Duarte L. Oliveira[1], Nicolly N. M. Cardoso[1], Gracieth C. Batista[1], Diego A. Silva[1], Leonardo Romano[2]
[1]Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – ITA – IEEA – SJC – São Paulo – Brazil
[2]Departamento de Engenharia Elétrica – Centro Universitário da FEI – São Bernardo do Campo – São Paulo – Brazil

***Abstract*— Digital systems design are usually synthesized in the synchronous paradigm using the global clock signal and they can be implemented in Field Programmable Gate Array (FPGA) and Very Large Scale Integration (VLSI) using Deep-Sub-Micron CMOS (DSM_CMOS) technology. These digital designs implemented in DSM_CMOS technology have the global clock signal as an obstacle which makes it difficult to comply with requirements, such as: performance, power consumption, reusability, etc.; because the wires have significant delays (latency). One solution is to synthesize modules that are insensitive to wires delays, that is, the module adapts to the latency of communication between the modules. In this paper, we propose a new pipeline architecture that is insensitive to latency because it has the property of elasticity, obtained by the asynchronous control responsible for the pipeline communication. Our pipeline can receive data at a frequency unrelated to the global clock signal. Through a case study, a Finite Impulse Response (FIR) filter of order five was used in reason to prove the efficiency of our proposal. Compared to a conventional synchronous pipeline, we achieved a throughput increase of up to 14.7% on Altera's FPGA platform.**

***Keyword*— Elastic Circuits, STG specification, Adaptive to Latency.**

I. INTRODUCTION

Contemporary digital systems may require high integration capability, high performance, low power consumption [1,2]. They can be implemented in hardware (dedicated processors) plus software (programmable processors). General or dedicated processors, such as Digital Signal Processor (DSP), are mainly implemented in the pipeline architecture [3]. These systems are implemented in Very Large Scale Integration (VLSI) and Field Programmable Array (FPGA), both in the Deep Sub-Micron CMOS (DSM_CMOS) technology. This technology needs to operate with low noise and the difference between the maximum and minimum delay (in the wires and gates) is higher when compared to other CMOS technologies, and the delay in a wire can be greater than the delay in a gate [4].

Digital systems are traditionally designed in the synchronous paradigm, i.e., they use a global clock signal to synchronize their operations. In DSM-CMOS technology, a global clock signal requires attention due to the wires delays and power consumption. Besides these factors, the clock signal distribution is a task with increasing complexity due to the clock skew problem, which leads to a decrease in performance (latency and cycle times).

Fig. 1 shows six modules in an Integrated Circuit (IC) that operate with a global clock. In the past, data communication between these modules could be done with a single clock cycle because the delays of data in wires could be neglected. In DSM-CMOS technology, the data in wires have significant delays, so transmitting data with a clock cycle leads to a strong reduction of the clock rate and increased clock skew [5].

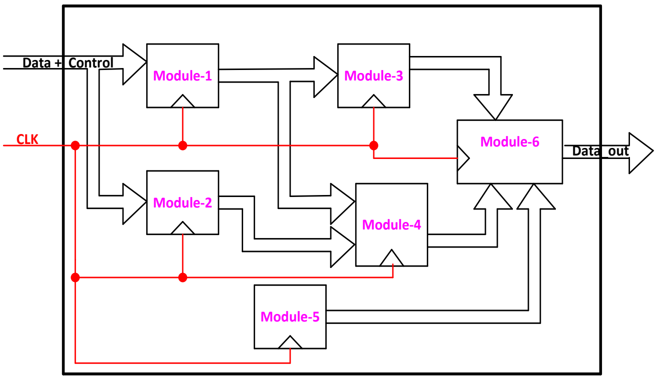


Fig. 1. IC with six modules synchronized with global clock.

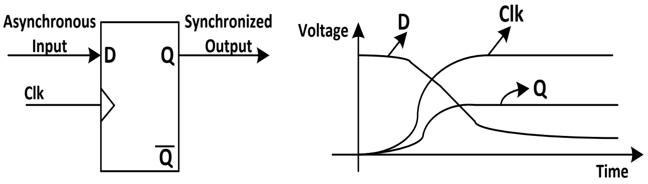
Carloni et al. [6,7] proposed a theory for the design of Latency-Insensitive (LI) synchronous circuits, also known as elastic synchronous circuits [8,9]. It explores some of the advantages of asynchronous projects, which is applied in synchronous systems to turn them into systems that are insensitive to latency. LI systems are systems that are insensitive to wire delays. For this purpose, LI protocol also known as an elastic protocol has been developed to synchronize handshake-like signals with the system global clock signal. LI design was introduced to help synchronous circuits to tolerate excessive latency (delay) from wires [4]. Therefore, LI theory has introduced a new synchronous design paradigm that allows transmitting data between the transmitter and receiver type modules using a number of clock cycles, where this number can be an integer or a rational number [10].

A. The synchronization problem

Designing complex systems with global clocking in DSM_MOS technology has become impractical as it becomes increasingly difficult to distribute global clock to all parts of the chip. The solution of using some asynchronous modules or using modules that operate at different clock rates leads to the synchronization problem of asynchronous input signals. This can be done using a single D-type flip-flop, as shown in Fig. 2a. However, if the clock edge gets very close to the data arriving in asynchronous form, the circuit can enter in a metastable state in which its output is not at logical level 0 neither 1, but rather somewhere intermediate [11]. This behavior is shown in Fig. 2b. Assume that Q is initially low and that D has recently risen, if D goes down again at approximately the same time that CLK goes up, output Q can

Duarte L. Oliveira, duarte@ita.br, Tel +55-12-3947-6813; Nicolly N. M. Cardoso, nicollynmcardoso@gmail.com; Gracieth C. Batista, gracieth@ita.br; Diego A. Silva, dasilva@ita.br; Leonardo Romano, leoroma@uol.com.br

start to rise and then get stuck between logical levels as D falls. Should Q rise or fall? In fact, any answer would be fine, but the flip-flop becomes indecisive. At some point, Q can continue at a logical level 1 or it may fall to logical level 0. When this happens, however, it is theoretically unlimited. If during this period of indecision, the circuit following from this flip-flop looks at the synchronized input, it sees an indeterminate value. The value can be interpreted by different subsequent logical stages, as logic 0 or 1. This can lead the system in an illegal or incorrect state causing the system to fail, such failure is traditionally called synchronization failure [11]. If care is not taken, any asynchronous communication between the synchronous modules can lead to a probable unacceptable failure. Fig. 3 shows the conventional solution for the synchronization problem which is to insert a further D flip-flop.



(a) (b)

Fig. 2. Interaction with Flip-Flops: a) Simple, dangerous synchronizer; b) Oscilloscope view of metastable behavior.

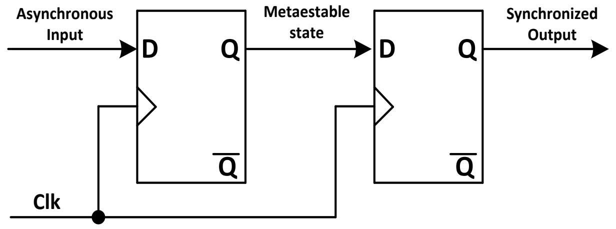


Fig. 3. Double flip-flop solution to reduce synchronization failure.

B. Pipeline design

A style of synchronous digital design that achieves high performance is the pipeline style [12]. A basic synchronous pipeline system, shown in Fig. 4, has three limitations intrinsic to the architecture and that are aggravated by DSM_CMOS technology: a) the difficulty of knowing when the data is valid at the pipeline output; b) clock activating the registers of each stage, even without new data to store; c) difficulty in diagnosing the valid data arriving at the pipeline input, because data transfer may need several clock cycles, and as commented, the number of cycles can be an integer or a rational number.

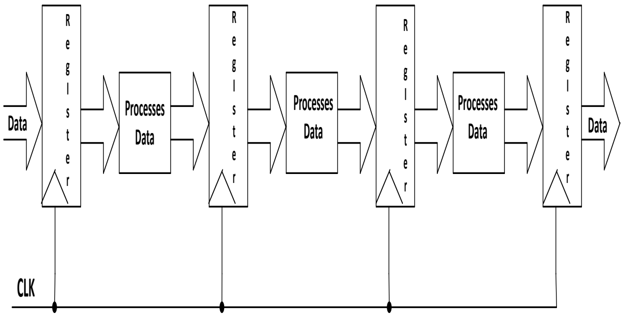


Fig. 4. Conventional Scheme of a Linear Synchronous Pipeline.

Different proposals for LI pipeline architectures have been made, but they use a sophisticated control to synchronize the data with the clock signal [7]. In this paper we propose a new gated-clock architecture, shown in Fig. 5, to implement linear pipeline systems that is insensitive to latency and are principally intended for applications that involving high throughput as digital signal processing and image processing, which are widely used areas in the military and aerospace sector.

The proposed pipeline architecture was generated using the synchronous and asynchronous paradigms. The synchronous concept allows the no need for insertion of delay elements at each stage of the pipeline (use of the clock signal). The value of the inserted delay elements, for example, in an FPGA is obtained in the exhaustive form and it can be a sub-optimal value. The asynchronous concept allows the introduction of elasticity in the pipeline, obtained through protocol handshaking. The elasticity of the pipeline becomes insensitive to latency of the wires communication.

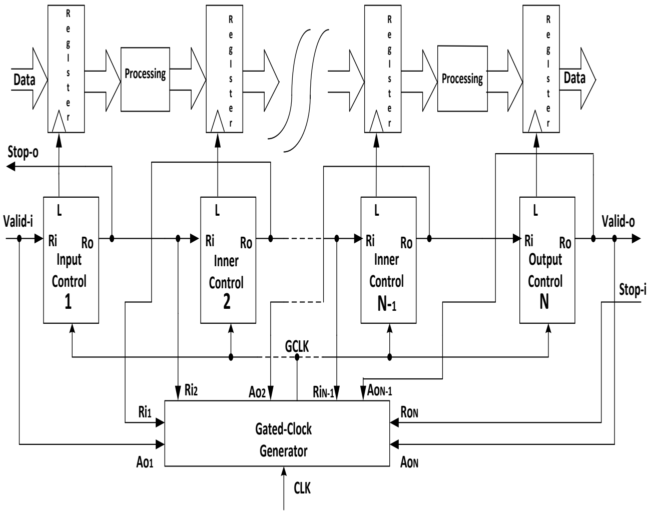


Fig. 5. Proposal of the synchronous-asynchronous mixed pipeline with clock-gating.

II. LI SYNCHRONOUS PIPELINE DESIGN: REVIEW

The pipeline technique applied in the design of digital systems has been instrumental in increasing parallelism. It has been applied with great success in high performance processors, multimedia, signal processors, etc. Architecture of a pipeline system can be classified into different classes: linear or non-linear [3]. A second classification is if the pipeline architecture is synchronous [12], asynchronous [13-16], synchronous insensitive to (elastic) latency [17-19], or mixed mode (synchronous and asynchronous) [20]. In a synchronous pipeline system, a complex module is partitioned into smaller sub-modules and registers are inserted between the sub modules, forming stages where the registers are activated by a global clock signal.

In a LI pipeline system, the events are synchronized with a global clock signal, but the data arrives at different times in a large temporal variability, the synchronization with the clock signal is made by the LI protocol that has a pair of "Valid" and "Stop" signals showing when data is valid or not (see Fig.6). There are different variants in the design of LI synchronous systems, for example, an interesting variant is the interlocked pipeline proposed in [17] which strongly uses principles of the handshake protocol. This pipeline is limited by the lack of the stop signal, being that the valid signal and input data must be synchronized. Fig. 6 shows a LI synchronous pipeline that operates on the protocol that contains only the *Valid* signal [18]. Fig. 7 shows a LI synchronous pipeline that operates with the protocol based on the *Valid* and *Stop* signals that was proposed in [19].

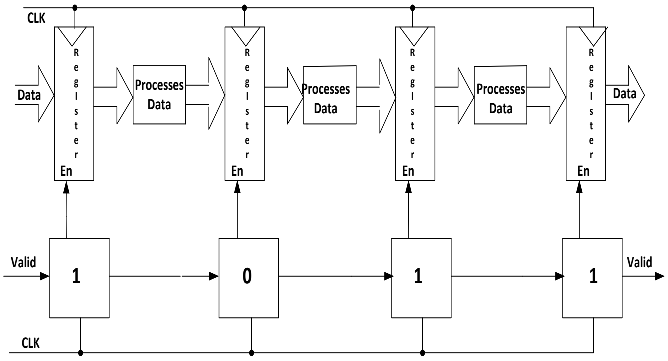


Fig. 6. LI Synchronous Pipeline without *Stop* signal of [18].

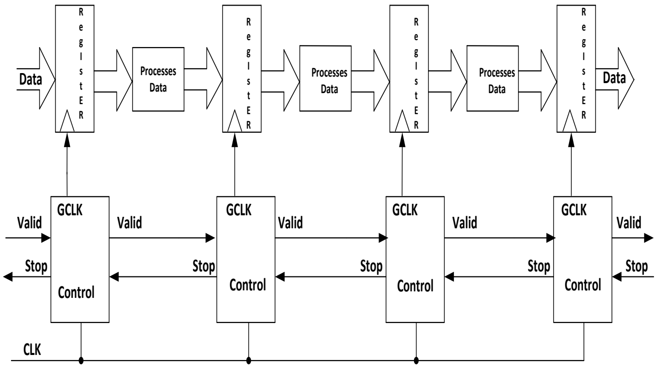


Fig. 7. LI Linear synchronous pipeline: only stores valid data that is independent of the clock cycles with the help of Stop signal [19].

III. CONTROL FOR SYNC-ASYNC MIXED PIPELINE WITH CLOCK-GATING

The proposed LI pipeline architecture is composed of processing modules, D flip-flops based on registers, clock-gating generator and a control in each register. The pipeline uses three different controls where there are the input (shown in Fig. 8a) and output controls (shown in Fig. 8b) completing the internal control (shown in Fig. 8c). The control consists of an Asynchronous Finite State Machine (AFSM) and a synchronizer which synchronizes with the global clock, respectively, of the input and output Valid signals (Valid-i,Valid-o), Stop (Stop-i,Stop-o), and that operate on 2-phase protocol. For the internal control there is the Ai signal (acknowledge input signal), meaning that the initial register has accepted the request to store. The AFSM decide the write operation in the register and it was described in the STG (Signal Transitions Graph) specification of [21], shown in Fig. 9. It is composed by the following signals: Ri (valid data at the input stage - request-store in the initial register), Ro (request output, stage final register is being able to accept new data) and L (stores the data).

The proposed AFSM was synthesized by the Petrify tool [22]. Fig. 10 shows the logic circuit generated by the Petrify tool, where it did not need to introduce internal signal to satisfy the CSC (Complete State Coding) [5]. The proposed pipeline uses the clock-gating technique to optimize the power consumption related to the clock signal. Fig. 11 shows the proposed clock-gating for the pipeline circuit, being for each stage register, an XOR gate, which represents when the register can store.

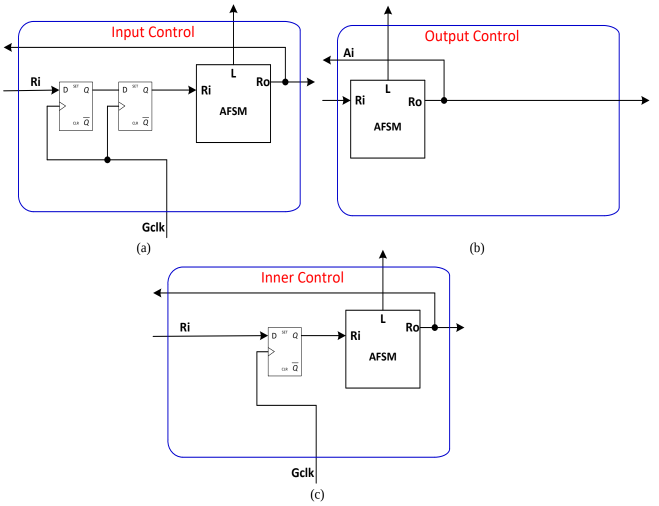


Fig. 8. Proposed control: a) input schema; b) output schema; c) inner control.

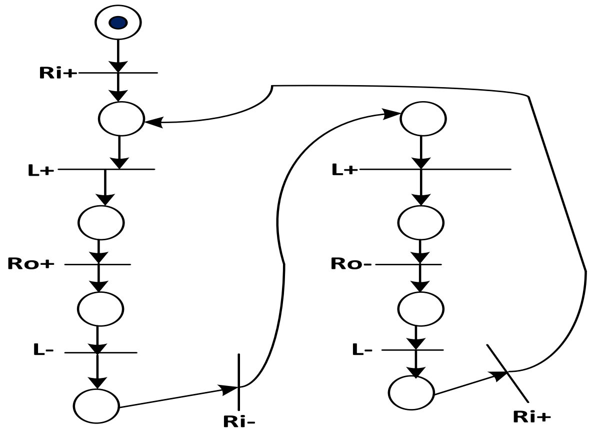


Fig. 9. Signal transition graph (STG) specification of AFSM.

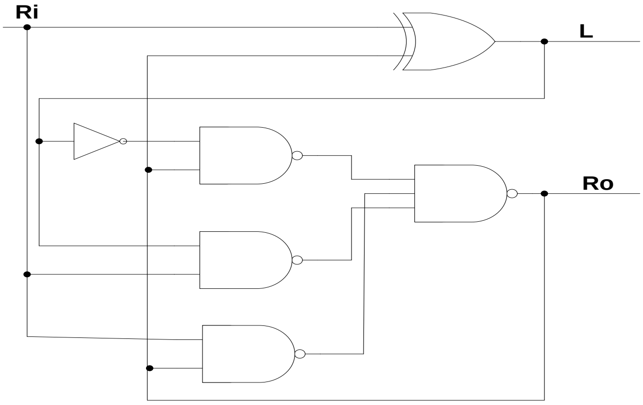


Fig. 10. Logic circuit of AFSM.

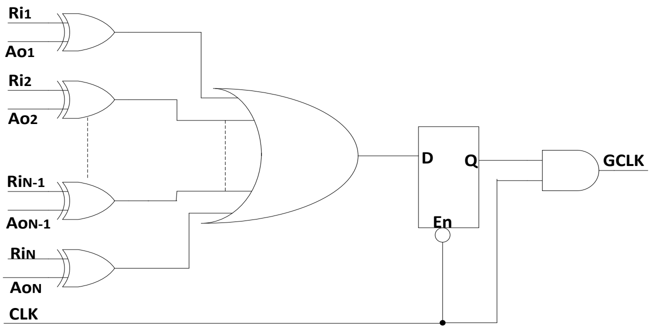


Fig. 11. Proposed clock-gating generator.

IV. CASE STUDY

To illustrate our LI synchronous pipeline architecture behavior, we used the FIR filter of order five (see equation 1), which are important in the area of digital signal processing. Using the behavioral synthesis procedure of [23], the FIR filter design was generated from the stepped data flow graph, which was obtained by the list scheduling algorithm, with the resource constraints of two multipliers and an adder. Fig. 12 shows the 6-stage datapath pipeline obtained in [24].

$$Y[t]= \sum_{i=0}^{N-1} H[i]*X[t-1] \qquad (1)$$

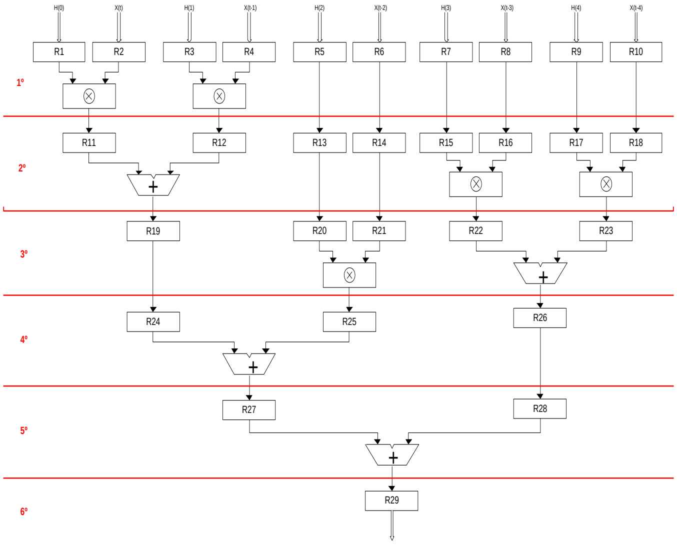


Fig. 12. FIR filter: pipeline data-path of [24].

V. SIMULATION & RESULTS

In order to demonstrate the architecture performance, it was realized the simulation and synthesis of the FIR filter in the proposed synchronous-asynchronous mixed pipeline architecture, was described in structural VHDL and implemented in the FPGA platform. Murray et al. [25] shows that the LI paradigm improves performance on the FPGA platform; Simulation and synthesis were performed in QUARTUS II software version 9.1 of ALTERA [26], family CYCLONE II in device EP2C35F672C6 and in family STRATIX II in device EP2S15F484C3.

Fig. 13 and 14 show, respectively, the simulations of the six stages FIR filter in proposed architecture. The simulation in Fig. 13 shows the operation of the pipeline, presenting in time the waveforms of the input signals {x} and output {y}, where constants *h*, *x* and *y* are integer values.

The simulation in Fig. 14 shows the FIR filter of order five generating the expected processing, with six control circuits, and showing the *Gclk* signal, the *Ri (ri)* signal (*Valid* - in the transition of 0→1 and 1→0), in pipeline input and signal *Ai (ai)* (transition 0→1 and 1→0) confirming initial storage. At the output of the pipeline there is the signal *Ro (ro)* that at the transition from 0→1 and 1→0, signals the new values of *y*.

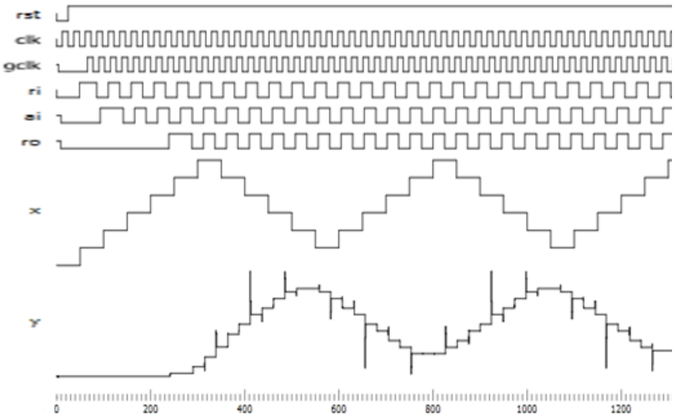


Fig. 13. Simulation: Filter FIR and input/output signals.

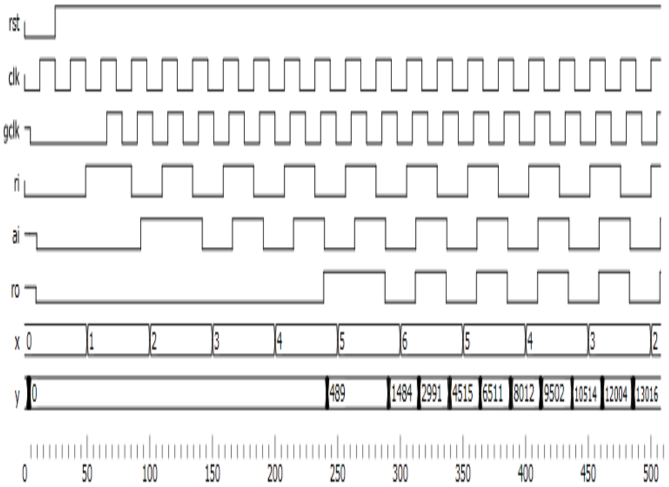


Fig. 14. Simulation: Filter FIR in the synchronous-asynchronous mixed pipeline of six-state with clock-gating.

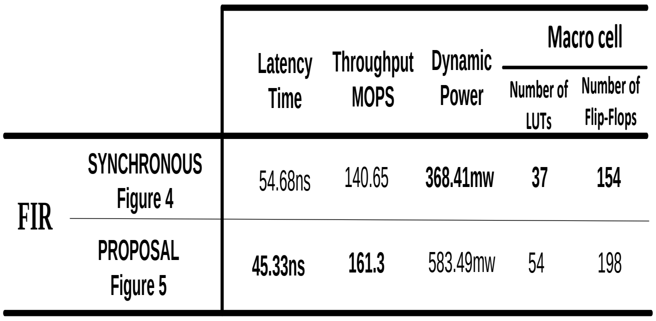
The tables below presents results of the FIR filter in the proposed synchronous-asynchronous mixed architecture, LI synchronous architectures of [24] and a basic synchronous version. All the architectures were implemented in FPGA and the results involves latency time, area (LUTs + FFs), dynamic power and throughput.

Table I presents results of the proposed architecture and the basic synchronous version shown in figure 4. The two implementations were performed in the STRATIX II family on the EP2S15F484C3 device. We can highlight the reduction and increase respectively of 17.1% and 14.7% in latency time and throughput in the proposed architecture, and also we had area penalty (LUTs + FFs) and dynamic power.

Table II involves the same implementations of Table I, but using the CYCLONE II family and EP2C35F672C6 device. We can highlight the 14.17% increase in throughput in the proposed architecture. There was a penalty of 14.7%, 17.6% and 74.9% respectively in the latency time, area (LUTs + FFs) and dynamic power.

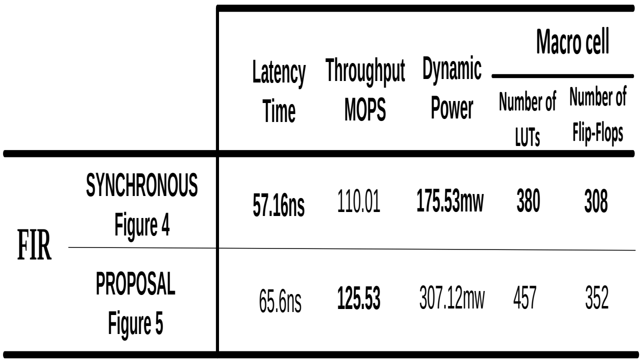
Table III presents results of the proposed architecture and of three LI synchronous pipeline architectures of [24]. The four implementations were made in the STRATIX II family on the EP2S15F484C3 device. We can highlight the average reductions of 57.7%, 38.8% respectively latency time and area in the proposed architecture. There was an average penalty at the dynamic power of 45.6%.

TABLE I: RESULTS OF FIR FILTER OF FIFTH ORDER: FAMILY STRATIX II IN DEVICE EP2S15F484C3



		Latency Time	Throughput MOPS	Dynamic Power	Macro cell	
					Number of LUTs	Number of Flip-Flops
FIR	SYNCHRONOUS Figure 4	54.68ns	140.65	**368.41mw**	**37**	**154**
	PROPOSAL Figure 5	**45.33ns**	**161.3**	583.49mw	54	198

TABLE II: RESULTS OF FIR FILTER OF FIFTH ORDER: FAMILY CYCLONE II IN DEVICE EP2C35F672C6



		Latency Time	Throughput MOPS	Dynamic Power	Macro cell	
					Number of LUTs	Number of Flip-Flops
FIR	SYNCHRONOUS Figure 4	**57.16ns**	110.01	**175.53mw**	**380**	**308**
	PROPOSAL Figure 5	65.6ns	**125.53**	307.12mw	457	352

TABLE III: RESULTS OF FIR FILTER OF FIFTH ORDER IN THE LI PIPELINES AND SYNC-ASYNC MIXED PIPELINE

ARCHITECTURES	**Latency Time**	**Number of Flip-Flops**	**Number of LUTs**	**Dynamic Power**
LI Pipeline Figure 7 of [24]	82,78ns	240	72	404,66mw
LI Pipeline Figure 8 of [24]	123,40ns	240	66	**368,85mw**
LI Pipeline Figure 9 of [24]	115,23ns	240	78	428,49mw
PROPOSAL Figure 5	**45.33ns**	**154**	**37**	583.49mw

VI. CONCLUSION

Digital pipeline systems are traditionally designed in the synchronous paradigm, so they use a global clock signal to synchronize their operations, besides they are implemented in DSM_CMOS technology, then the wires can have significant latency and the communication between synchronous modules may require several clock cycles, so not degrading performance. In this paper, we present a new pipeline architecture that mixes the synchronous and asynchronous paradigms, generating an architecture that has the LI property and does not use delay elements that represent the critical path of data-path of each stage of the pipeline. Because, asynchronous pipeline in the Bundled-data class i.e. uses conventional functional units, synchronization is done on protocol handshaking and delay elements [15]. Through a case study, in the case of a FIR filter application, we show the performance of the proposed synchronous-asynchronous mixed pipeline and the operation correctness in the latency independent environment, where it only stores valid data regardless of the time intervals in which they occur.

REFERENCES

[1] C. Constantinescu, "Trends and Challenges in VLSI Circuits Reliability," *IEEE Micro*, 23 (4), 2003.

[2] K. D. Muller-Glaser, et al. "Multiparadigm Modeling in Embedded Systems Design", *IEEE Trans. on Control Systems Technology*, vol. 12, no. 2, March 2004.

[3] S. M. Nowick and M. Singh, "High-Performance Asynchronous Pipelines: An Overview," *IEEE Design & Test of Computers*, September/October, pp.8-22, 2011.

[4] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou, "Coping with the variability of combinational logic delays," *ICCD*, pages 505–508, 2004.

[5] C. J., Myers, "*Asynchronous Circuit Design*", Wiley & Sons, Inc., 2004, 2ª edition.

[6] L. P. Carloni, K.L. McMillan, and A.L. Sangiovanni-Vincentelli, "Theory of latency-insensitive design," *IEEE Transactions on Computer-Aided Design*, 20(9):1059–1076, September 2001.

[7] L.P. Carloni and A.L. Sangiovanni-Vincentelli, "Coping with latency in SoCdesign," *IEEE Micro, Special Issue on Systems on Chip*, vol. 22, no. 5, pp.12, Octo. 2002.

[8] J. Cortadella et al., "Synthesis of Synchronous Elastic Architectures," Proc. DAC, pp. 657–662, 2006.

[9] J. Cortadella, et al., "SELF: Specification and design of synchronous elastic circuits," Proc. ACM/IEEE Int. Workshop on Timing Issues, TAU'06, pp.1-6, 2006.

[10] L.P. Carloni et al., "A Methodology for Correct-by-Construction Latency Insensitive Design," Proc. ICCAD, pp. 309–315, 1999.

[11] R. Ginosar, "Metastability and Synchronizers: A Tutorial," *IEEE Design & Test of Computers*, vol.:28, Issue:5, pp.23-35, 2011.

[12] T. C. Chen, "Parallelism, Pipelining and Computer Efficiency," *Computer Design*, pp. 69-74, January 1971.

[13] M. Singh and S. M. Nowick, "The Design of High-Performance Dynamic Asynchronous Pipelines: Lookahead Style," *IEEE Trans. on VLSI Systems*, vol.15, no. 11, pp.1256-1269, November 2007.

[14] M. Singh and S. M. Nowick, "The Design of High-Performance Dynamic Asynchronous Pipelines High-Capacity Style", *IEEE Trans. on VLSI Systems*, vol.15, no.11, ´pp.1270-1283,November, 2007.

[15] M. Singh and S. M. Nowick, "MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines", *IEEE Trans. on VLSI Systems*, vol.15, no. 6, pp.684-698, June 2007.

[16] D. L. Oliveira, et al., "Using FPGAs to Implement Asynchronous Pipeline," 5th IEEE Latin American Symposium on Circuits and Systems, Santiago, Chile, 2014.

[17] H.M. Jacobson et al., "Synchronous Interlocked Pipelines," Proc. ASYNC, pp. 3–12, 2002.

[18] H. M. Jacobson, et al., "Stretching the Limits of Clock-Gating Efficiency in Server-Class Processors," Proc. 11th Int. Symposium on High-Performance Computer Architecture (HPCA-11 2005), pp.1-5, 2005.

[19] A. Islam, et al., "A New Synchronous circuit for Elastic Pipeline Architecture," International Conference on Materials, Electronics & Information Engineering, ICMEIE-2015, pp. 1-4, 2015.

[20] M. Singh et al., ''An Adaptively Pipelined Mixed Synchronous-Asynchronous Digital FIR Filter Chip Operating at 1.3 Gigahertz,'' *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 7, pp. 1043-1056, 2010.

[21] T. –A. Chu, "*Synthesis of Self-Timed VLSI Circuits from Graph-Theory Specifications*," PhD. Thesis, June 1987, Dep. Of EECS, MIT

[22] J. Cortadella, et al. "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Trans. on Information and Systems*, E80-D(3), March, pp.315-325, 1997.

[23] D. D. Gajski, "*Principles of Digital Design*," Prentice Hall, 1997.

[24] D. L. Oliveira, et al. "Uma Nova Arquitetura para Sistemas Pipeline Síncrono Insensíveis à Latência," XXII Iberchip Workshop, Florianópolis, Brazil, pp.17-20, 2016.

[25] K. E. Murray and V. Betz, "Quantifying the Cost and Benefit of Latency Insensitive Communication on FPGAs," Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays, FPGA'14, pp.223-232, 2014.

[26] Altera Corporation, 2019, www.altera.com.