ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Novel Implementation of FIR filters using Sequential Circuits

Rishitej Chaparala
Department of ECE,
Sasi Institute of Technology & Engineering, Tadepalligudem.

Naga Raju Ravada
Assistant Professor,
Department of ECE,
Sasi Institute of Technology & Engineering, Tadepalligudem.

Sudheer Kanuri
Assistant Professor,
Department of ECE,
Sasi Institute of Technology & Engineering, Tadepalligudem.

Abstract:

The asynchronous paradigm has interesting features due to the lack of the clock signal and it is another option for the project of digital systems. This paradigm has several design styles, where the micro pipeline style is the most suitable one for FPGA platforms, due to the simplicity of its control. In this paper, we propose a five stage data path pipeline architecture that implements both array and booth multipliers, considering FPGAs as target devices. Through a case study, we show that the booth multiplier presents 14.99ns reduction of timing to that of array multiplier.

Keywords:

Micropipeline, multipliers;

I. INTRODUCTION:

FPGAs devices (Field Programmable Gate Array) have become a very popular way to develop and implement digital circuits due to their cost and design time. High performance FPGAs are implemented in Deep-Sub-Micron MOS technology (DSM-MOS). In this technology, the delay on the lines should be considered and may be higher than the delay of the logic gates [1], [2]. Digital systems are traditionally designed in synchronous paradigm, i.e. they use a global clock signal to synchronize their operations. They are quite popular due to their simplicity of design and availability of commercial CAD tools for automatic synthesis. In DSM-MOS technology, a clock signal requires attention due to its noise generation, electromagnetic interference and power consumption. Besides these factors, the distribution of the clock signal along the chip is a task with increasing complexity because of the clock skew problem, which drops the system performance.The overhead caused by the clock signal can reach 130% in a VLSI (Very Large Scale Integration) implementation [3] and worsens when FPGAs are employed.

Considering the design style of asynchronous systems, it is easier to implement the micro pipeline style in commercial FP-GAs due to the simplicity of its control [4], [5]. This is important because of the difficulty to achieve hazard-free Considering the design style of asynchronous systems, it is easier to implement the micro pipeline style in commercial FPGAs due to the simplicity of its control [4], [5].This is important because of the difficulty to achieve hazard-free [7], [8], [9], [10], [11], [12], [13]. However, most of these pipeline architectures are focused on VLSI implementations, employing full-custom control [8], [9], [10]. FPGA oriented pipelines have been proposed [12], [13], but one uses a complex delay mechanisms [12] and the other does not obey the fundamental-mode [13].

The pipeline architecture known as MOUSETRAP, proposed in [11], has a good performance, it is based on logic gates and can be implemented on FPGAs (see Fig. 1).In this paper, we propose a FPGAs oriented architecture for micro pipeline design style (see Fig. 2). The proposed linear pipeline architecture employs flip-flops as registers, due to the large availability of these elements in FPGAs. The control is based on logic gates and it is mapped into two LUTs (Look-Up Table).The new pipeline architecture has a better performance, when compared to the MOUSETRAP architecture implemented on FPGA, because it has a smaller latency and a greater throughput. A digital FIR filter of fourth order shows the efficiency of our architecture on a FPGA plat-form.

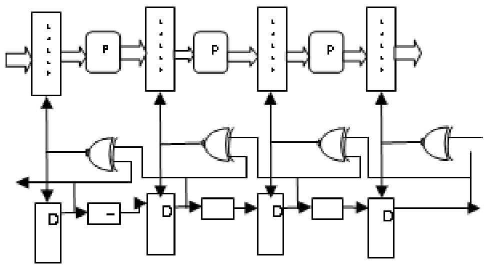


Fig. 1. MOUSETRAP linear micro pipeline

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

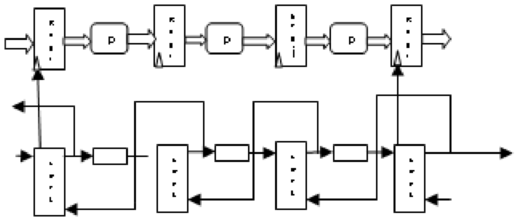


Fig. 2. Proposed linear micro pipeline architecture.

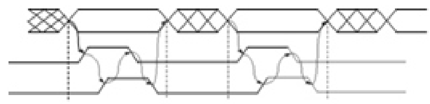
II. MICROPIPELINE PROJECT EMPLOYING FPGAS:

Programmable devices, such as FPGAs, are designed for synchronous designs [2]. Efforts for prototyping asynchronous design in commercial FPGAs [14], [15] and academic FPGAs [16], [17] been reported recently. The problems in implementing asynchronous systems in commercial FPGAs are related to the control. They area) Process of mapping risk-free Boolean functions in logic blocks (macro cells).

The commercial tools used for de-composition and mapping Boolean functions in LUTs are not prepared to meet the requirements of logical hazard. This may cause a circuit malfunction, if manual intervention to fix the problem is not performed. The mapping function must satisfy the decomposition requirements proposed in Sigel et al. [18]. b) Internal routing process among macro cells can introduce significant delays. These delays can result in essential hazard and lead to the circuit malfunction [4], [5].

The circuit delay model defines how to solve the problem of essential hazard: the insertion of delay elements in the feedback lines or the employment of macro cells that satisfy the isochoric fork condition [4], [5]. The micro pipeline architecture can be linear or nonlinear. [19]. In this paper we focus only on the linear architecture. The micro pipeline style has as main feature, the simplification of the pipeline control, an important characteristic for asynchronous systems implemented in FPGAs.

The control is either distributed between stages or centralized, and it is responsible for the communication between the pipeline stages. In a FPGA platform, the control must be distributed in order to avoid hazard problems, as previously mentioned. The pipeline communication employs the handshake protocol with Request and Acknowledge signals[4],[5]. The communication between the stages can be performed in two different protocols: 4-phase or 2-phase. Fig. 3 shows the behavior of these protocols.



(a) 4 phases handshaking control

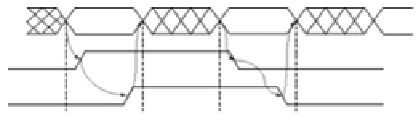


Fig.3. Handshake protocol:a) 4-phase b)2 –phases

The linear micro pipeline design has two variants. In the first one, it is used only components of the synchronous paradigm (single-rail) and delay elements between stages. The delay is defined considering the critical path of each stage. The bundled-data implementation is one of the data encoding schemes used in asynchronous circuits. It represents N bits of data with N+2 lines, called “bundled”, where the two additional lines are the handshake signals, request and acknowledge. This architecture is called “micropipelinebundled-data”.Thesecond architecture employs dual rail components [4], [5]. The components are synthesized using Delay Insensitive(DI) codes or dual-rail. In the case of dual-railcode,eachFig.6. Controller’s logic circuit. signal is encoded with two bits. Four cases would be possible for a signal: a0a1=00 (null); a0a1=01 (1); a0a1=10 (0); and a0a1=11 (never occurs).

The design employing this code generates a signal of “end of operation” without the need of delay element, and it can be implemented with a simple circuit. The dual-rail linear micro pipeline architecture can be implemented in FPGA platforms, but there is a significant increase in the number of used macro cells, that compromises their performance and power consumption. On the other hand, some advantages need to be considered:this architecture does not need delay elements and timing analysis, moreover, it is more robust.

The proposed micro pipeline architecture operates in 2-phase handshake protocol and, therefore, the processing request occurs on both edges of this signal. Fig. 4 shows the general lay-out of a stage, which consists of a flip-flop based register and control. The control was specified in STG (signal transition graph) [20] (see Fig. 5). It is composed by the signals: Ri (request input), Ao (acknowledge output), Ai (acknowledge input), L (load) and Ro (output request). Fig. 6 shows the circuit of the logic control.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

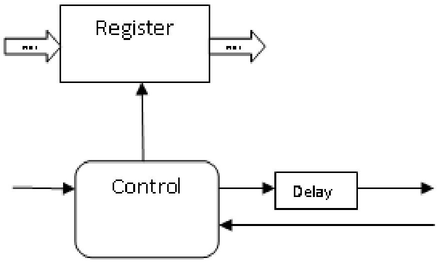


Fig. 4. Linear micro pipeline target for FPGA.

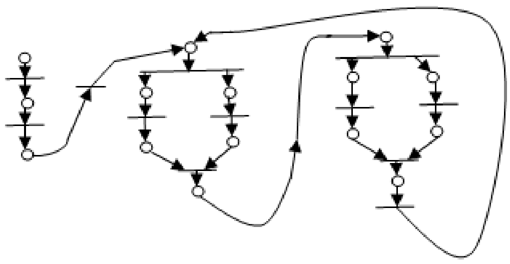


Fig. 5. Controller's signal transition graph

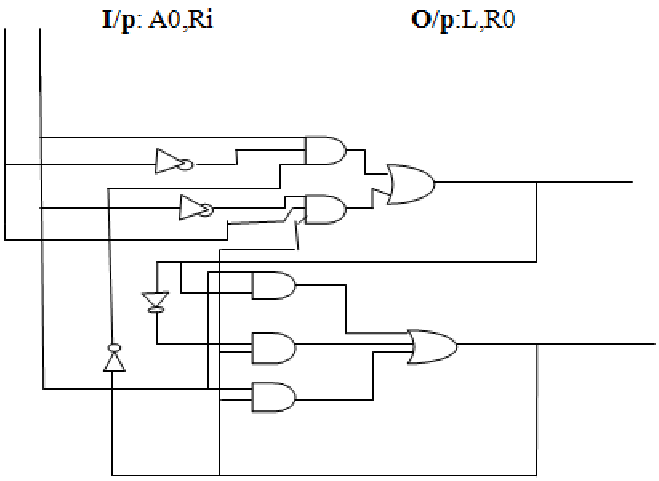


Fig.6. Controller's logic circuit

The procedure used to synthesize bundled-data micro pipeline systems attempts to use components and synthesis tools of the synchronous paradigm. The behavioral synthesis starts from the Control Data Flow Graph (CDFG), which represents the operations and their data dependency. The method can be divided into five steps: 1) Generation of the scheduled DFG employing the list scheduling algorithm [21]. 2) Based on the scheduled DFG, generation of the pipeline data-path employing single-rail components and the pipeline synthesis according to [21]. This step generates the synchronous pipeline. (see Fig. 7). 3) Execution of the de synchronization process; detection of the critical path of each pipeline step and calculation of the respective delay element.

4) Change the clock signal by the proposed asynchronous pipeline controller (see Fig. 8).5) Synthesize the asynchronous pipeline generated in step 4.

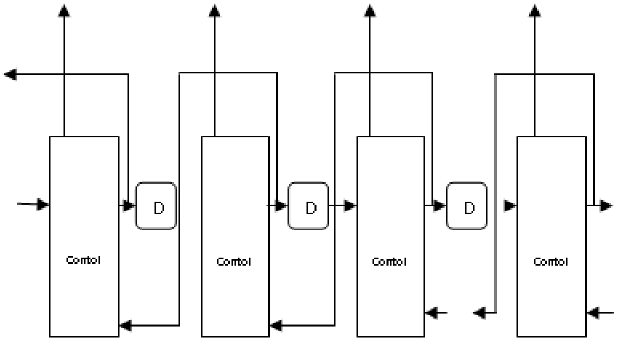


Fig.7.Asynchronous pipeline architecture

In order to illustrate the use of proposed micro pipeline architecture, an asynchronous version of 4th order FIR filter was designed, according to the equation:

y[n] = ¦hi ´ x[n −i]   (1) i=0

where, y[n] and x[n] are the output and input signal samples, and h(i) are the FIR filter coefficients.Fig. 9 shows the first step, the scheduled GFD of the filter, obtained from the List Scheduling algorithm with three resource constraints: two multipliers and one adder. The second step of the proposed method generates a data-path pipeline with five stages. It was obtained by the behavioral synthesis [21], which defines the synchronous pipe-line. Fig. 10 shows the pipeline data-path of the filter. The steps 3 and 4 make the desynchronization of the pipeline, generating the proposed pipeline, requiring six control sand sixdelayelements

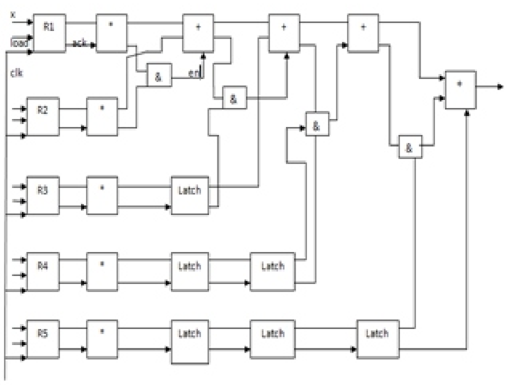


Fig. 8. Scheduled GFD of the filter.

III. EXPERIMENTAL RESULTS:

A comparison was made with the MOUSETRAP pipeline architecture in order to evaluate the performance of the proposed architecture.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The simulations were performed employing the Xilinx software, version 14.2, considering an Spartan 6 as target de-vice. Fig. 11 and 12 show the simulations of the mousetrap control and the stages of the FIR filter. Fig. 13 and 14 show the simulations of the proposed pipeline control and FIR stages.

Table.1.ARRAY MULTIPLIER

Area report:

DEVICE UTILIZATION SUMMARY			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slices	464	768	60%
No. of Slice flip-flops	251	1536	16%
No. of 4-input LUT's	808	1536	52%
No. of bonded IOB's	103	124	83%
No. of GCLK's	3	8	37%

Table.2

Delay report:

Minimum Period	17.486ns
i/p arrival time(before clock)	30.149 ns
O/p required time(after clock)	6.141 ns

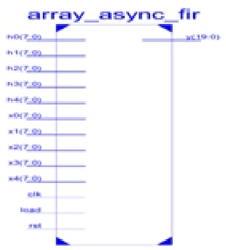


Fig.9.Block diagram for Array asynchronous pipeline

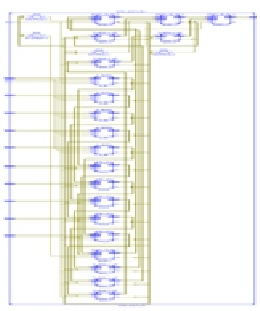


Fig.10. RTL schematic (array)

The waveform depicted in the fig.11. Shows the result of 5 stage scheduled GFD filter section with the placement of array multiplier block. Files are inserting filter method in multiplication process. Because of inserting these methods we can get both filtered and multiplier outputs. Therefore, the output consists of both filtered & multiplier output values. To calculate this output initially multiply x0 value with impulse response values h0,h1,h2,h3,h4 Now, x1 value with h0,h1,h2,h3,h4 and x2 value with h0,h1,h2,h3,h4 and repeat same for x3,x4 values. So, by doing this process we will get the exact output. With this process we will get combined outputs nothing but, Intermediate values(filter) & multiplier values(original). By placing booth multiplier block the same output result obtains to the 5 stage scheduled GFD filter. The output waveform for booth multiplier block resembles similar causalities to that of array multiplier output waveform result. In this project the scheduled GFD using array multiplier output result is said to be existed system where as, scheduled GFD using booth multiplier block result is the extension system.

Table.3 BOOTH MULTIPLIER

Area report:

DEVICE UTILIZATION SUMMARY			
Logic Utilization	**Used**	**Available**	**Utilization**
No. of Slices	416	768	54%
No. of Slice flip-flops	241	1536	15%
No. of 4-input LUT's	749	1536	48%
No. of bonded IOB's	99	124	79%
No. of GCLK's	3	8	37%

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Table 4 Delay report:

Minimum Period	14.994ns
i/p arrival time(before clock)	18.741 ns
O/p required time(after clock)	6.141 ns

The tables 1&2 depicted above shows the hardware utilization summary of 5 stage scheduled GFD of the filter for both array & booth multipliers. The hardware utilization summary for booth multiplier reduces to that of using array multiplier.

The complexity also reduces for booth multiplier schematic compared to that of array multiplier.The tables 1.1&2.1 depicted above shows the delay report of 5 stage scheduled GFD of the filter for both array & booth multipliers. Compared to the operation of array multiplier the booth multiplier operation display less period than that of array multiplier. Therefore efficiency increases.

Fig.12.Block diagram for Booth asynchronous pipeline

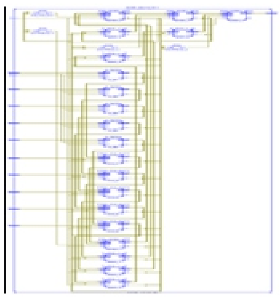


Fig.13.RTL schematic (booth)

Fig.14.WAVEFORM (booth multiplier)

IV. CONCLUSION:

The linear micro pipeline style of the asynchronous paradigm is advantageous because the performance is increased and the control is simplified. In this paper, we proposed new pipeline architecture oriented for FPGAs implementations. The control employs only two LUTs, and the registers used are based on flip-flops, allowing a better distribution of macro cells in a FPGA. The implementation of the proposed FPGA control is free of essential hazard. Through a case study, we show that our architecture has a better performance than the MOUSETRAP architecture, considering the FPGA implementation.

REFERENCES:

[1]D. Goldhaber-Gordon, et al., “Overview of Nanoelectronic Devices,” Proc. of the IEEE, vol. 85, No. 4, pp.521-540, April 1997.

[2]J. J. Rodriguez, et. Al., “Features, Design Tools, and Applications Domains of FPGAs”, IEEE Trans. on Industrial Electronics, vol. 54, No. 4, pp.1810-1823, August 2007.

[3]J. Cortadella, et al., “Coping with the variability of combinational logic delays,” ICCD, pages 505–508, 2004.

[4] C. J., Myers, “Asynchronous Circuit Design”, Wiley & Sons, Inc., 2004, 2a edition.

[5] J. Sparsø e S. Furber, “Principles of Asynchronous Circuits Design”, Kluwer Academic Publishers, 2001.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[6]I. E. Sutherland, "Micro pipelines", Communication of the ACM, vol. 32, No.6, pp.720-738, June, 1989.

[7]S. B. Furber et al., "Four-Phase Micro pipeline Latch Control Circuits," IEEE Trans. on VLSI Systems, vol.4, no. 2, pp.247-253, June, 1996.

[8]G. S. Taylor and G. M. Blair, "Reduced Complexity Two-Phase Micro pipeline Latch Controller," IEEE Journal of Solid-State Circuits, vol. 33, Nro. 10, pp.1590-1593, October, 1998.

[9]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.

[10] 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.

[11] 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.

[12] T.N. Prabakar G. Lakshminarayanan, K. K. Anilkumar, "FPGA Based Asynchronous Pipelined Multiplier with Intelligent Delay Controller," International SOC Design Conference, 304-309, 2008.

[13] D. L. Oliveira, et al., "An Architecture for High Performance Micro pipeline Systems directed to FPGAs," XV SIGE, (in Portuguese).

[14] E. Brunvand, "Using FPGAs to Implement Self-Timed Systems", Journal of VLSI Signal Processing, Special issue on field programmable logic, vol.6 (2), pp.173-190, August, 1993.

[15] M. Tranchero and L. M. Ryneri, "Implementation of Self-Timed Circuits onto FPGAs Using Commercial Tools", 11th Euromicro Conf. on Digital System Design Architectures, Methods and Tools, pp.373-380, 2008.

[16] R. Payne, "Asynchronous FPGA architectures," IEE Proc. Comp. Digit. Tech., vol.143, no.5, September, pp.282-286, 1996.

[17] N. Huot, et. al., "FPGA architecture for multi-style asynchronous logic," Proc. Of the Design, Automation and Test in Europe Conference and Exhibition, 2005.

[18] P. Sigel, G. De Michele and D. Dill, "Decomposition methods for library binding of speed independent," Proc. Int. Conf. Computer-Aided Design, pp.558-565, 1994.

[19] R. O. Ozdag, et al., "High-Speed Non-Linear Asynchronous Pipelines," Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE.02), 2002.

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

[21] D. D. Gajski, "Principles of Digital Design," Prentice Hall, 1997. [22] Altera Corporation, 20013, www.altera.com.

Author's Profile:

Rishitej Chaparala pursuing his M.Tech in Very Large Scale Integration from Sasi Institute of Technology and Engineering, West Godavari District, Tadepalligudem, Andhra Pradesh 534101.

Naga Raju Ravada Working as a Assistant Professor, Electronics and communication Engineering in Sasi Institute of Technology and Engineering, West Godavari District, Tadepalligudem, Andhra Pradesh 534101.

Sudheer Kanuri Working as a Assistant Professor, from Electronics and communication Engineering in Sasi Institute of Technology and Engineering, West Godavari District, Tadepalligudem, Andhra Pradesh 534101. Sudheerkanuri@sasi.ac.in