

Globally Asynchronous Locally Synchronous (GALS) Microprogrammed Parallel FIR Filter

Gouri Wazurkar¹, Dr. S. L. Badjate²

¹Department of Electronics Engineering, Shri Ramdeobaba College of Engineering & Management, Nagpur.

²Principal, S. B. Jain Institute of Technology, Management & Research, Nagpur.

Abstract: In this paper, we propose the design of globally asynchronous locally synchronous (GALS) microprogrammed parallel finite impulse response (FIR) filter using pipelined GALS Baugh Wooley Multiplier. The primary objective is to demonstrate low power implementation of microprogrammed parallel GALS FIR filter for digital signal processing applications. Fully synchronous microprogrammed parallel FIR filter and GALS microprogrammed FIR filter are implemented using same FPGA and almost same logic cells for fair benchmarking. The four taps synchronous and GALS microprogrammed parallel FIR filter is coded in VHDL and implemented in vertex 5 FPGA device. GALS microprogrammed parallel FIR filter is more power efficient as compared to synchronous filter.

Keywords: Low power, GALS, Microprogrammed, Parallel, FIR Filter.

I. Introduction

Low power operation is desirable in all the digital signal processing systems. Most of the digital signal processing systems is fully synchronous but a trend, as detailed in the International Technology Roadmap for Semiconductors (ITRS) [1], for an increasing use of asynchronous logic from the present 15 % to 49 % in 2024. In some of these SoC's, asynchronous signaling scheme [2], [3] were used for synchronization between the different fully synchronous modules that is opposed to fully asynchronous systems where asynchronous signaling scheme are used for both between modules (inter-modules) and within modules (intra-module). This hybrid inter-module asynchronous cum intra-module synchronous, termed Globally-Asynchronous-Locally-Synchronous (GALS) may be advantageously exploited to simplify some challenging design issues [4]. For the asynchronous-to-synchronous data transfer [2], [5] or vice-versa, the GALS approaches may be generally categorized by its clocking schemes, pausable clocking and the data-driven clocking.

FIR filter is the fundamental digital signal processing (DSP) operation for many DSP systems. It finds applications in audio, image and video processing, wireless communication, noise removal etc. In most of the applications digital filters are used to implement frequency-selective operations. Therefore, specifications are required in the frequency-domain in terms of the desired magnitude and phase response of the filter. FIR with constant coefficients is a linear time invariant digital filter. The output of an FIR of order or length (N), to an input time-series $x[n]$, is given by a finite version of the convolution sum given in (1) and (2),

$$y[n] = x[n] * h[n] \tag{1}$$

$$y[n] = \sum_{k=0}^{N-1} x[k]h[n-k] \tag{2}$$

where $h[n]$ is called as filter coefficients or impulse response and $y[n]$ is the output signal.

For linear time invariant system, it can be expressed in Z domain as given in (3)

$$y(z) = x(z)h(z) \tag{3}$$

where $h(z)$ is the FIR filter transfer function defined in Z domain by (4)

$$h(z) = \sum_{k=0}^{N-1} h[k]z^{-k} \tag{4}$$

Direct form implementation of linear time invariant FIR filter using delay element, adder and multiplier is shown in fig 1.

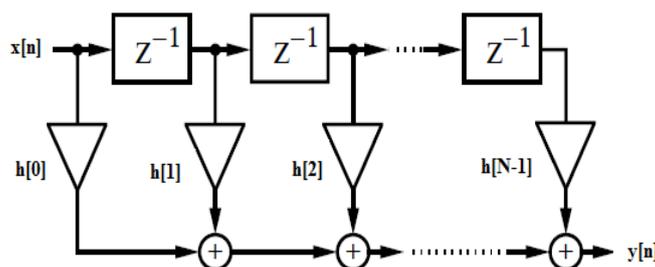


Fig. 1 Direct form FIR filter

The difference equation for 4-tap FIR filter (N = 4) is given in (5)

$$y[n] = \sum_{k=0}^3 x[k]h[n-k]$$

$$y[n] = x[0]h[n] + x[1]h[n-1] + x[2]h[n-2] + x[3]h[n-3] \quad (5)$$

Direct form FIR filters are also known as tapped delay line or transversal filters. The size of the FIR filter is determined by the number of coefficients $h[n]$. N-tap FIR filter consist of N delay elements, N multipliers and N-1 adders or accumulators. Generally a linear phase response in the pass band is desirable for many applications especially in communication. It is shown in [6] that linear phase is achieved if the impulse response is symmetric or anti-symmetric and hence it is preferable to use anti-causal framework [7] given in (6) obtained from (4)

$$h(z) = \sum_{k=-(N-1)/2}^{(N-1)/2} h[k]z^{-k} \quad (6)$$

Due to advances in technology, many researchers are trying to design FIR filter architecture which can offer one or more of the following design advantages such as high speed, low power consumption and less area. DSP functions are generally implemented in general purpose DSP processors where built in multiply accumulate (MAC) engines are used to perform mathematical operations. Application specific integrated circuits (ASICs) can also be used where high performance is needed or design volume is high enough to justify the non-recurring engineering (NRE) cost [8]. However, field programmable logic (FPGA) offers the better of the two technologies in addition to the re-configurability feature of the hardware platform. An important factor in a DSP processor is the limitation on hardware resources such as MAC engines. This is not an issue with FPGAs since these devices offer sufficient capacity to fit plenty of MAC processors into a single device. The performance of the parallel FIR is determined by multiplier. Modified Booth (MB) encoding reduces to half the number of partial products resulting in reduced area, critical delay and power consumption. However, a dedicated encoding circuit is required and the partial products generation is more complex [9]. Baugh Wooley 2's complement multiplier offers better sign bit management, uniform VLSI structure and no complex encoding circuits that result in compact circuit. The biggest advantage of compact and uniform structure is implementation of pipelining that easily divides the partial product generation stages and increases speed of operation [9].

In this paper, we proposed FPGA implementation of GALS microprogrammed parallel 4-tap FIR filter and its comparison with fully synchronous parallel microprogrammed 4-tap FIR filter using GALS & synchronous Baugh Wooley multiplier respectively given in [10]. The primary objective of the design is to demonstrate low power implementation of GALS FIR Filter. The paper is organized as follows section I introduces GALS and FIR Filter, section II describes Baugh Wooley multiplier and section III describes microprogrammed FIR filter architecture. Section IV provides in detail design of synchronous and GALS microprogrammed parallel FIR filter. Results are discussed in section V and finally concluded in section VI.

II. Baugh Wooley Multiplier

The Baugh Wooley multiplication algorithm [11] is developed to designed regular 2's complement multipliers. It effectively handles sign bit during the computation of partial products. Let a and b be the two n-bit signed numbers can be represented as,

$$a = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i \quad (7)$$

$$b = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \quad (8)$$

The result of multiplication of a and b is represented as

$$p = axb$$

$$= \left(-a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i \right) x \left(-b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} 2^j b_j \right)$$

$$= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} 2^i a_i \sum_{j=0}^{n-2} 2^j b_j - a_{n-1}2^{n-1} \sum_{j=0}^{n-2} 2^j b_j - b_{n-1}2^{n-1} \sum_{i=0}^{n-2} 2^i a_i \quad (9)$$

The last two terms in equation (9) are n-1 bits each that are extended from position 2^{n-1} to 2^{2n-3} . We pad zeros to remaining bits to obtain 2n bit number in order to extent binary weight from 2^0 to 2^{2n-1} . Rather than subtracting the last two terms, we can obtain 2's complement of the last two terms and add all terms to obtain final product. Let z be one of the last two terms, it can be represented in equation (10) with zero padding.

$$z = -0 x 2^{2n-1} + 0 x 2^{2n-2} + 2^{n-1} \sum_{j=0}^{n-2} 2^j z_j + 0 x \sum_{r=0}^{n-2} 2^r \tag{10}$$

Table I Bit values for -Z

Bit position	Bit Values
2n-1	1
2n-2	1
2n-3	Z _{n-2}
2n-4	Z _{n-3}
2n-5	Z _{n-4}
...	...
n	Z ₁
n-1	Z ₀₊₁
...	...
1	0
0	0

Table II Bit patterns

Bit position	2n-1	2n-2	n	n-1
+	1	1		1
	1	1		1
Carry in	1	0 / 1	1	0 / 1
Sum		0 / 1		0 / 1

After obtaining 2's complement of z, the new bit value for -z is shown in table I. Let z₁ and z₂ be last two terms in equation (3) then addition of -z₁ + (-z₂) results in following bit patterns at most significant bits and bit position n shown in table II. Hence the product p in equation (9) can be given as

$$p = a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} 2^{i+j} a_i b_j + 2^{n-1} \sum_{j=0}^{n-2} 2^j \overline{b_j a_{n-1}} + 2^{n-1} \sum_{i=0}^{n-2} 2^i \overline{a_i b_{n-1}} - 2^{2n-1} + 2^n \tag{11}$$

Let us assume if a and b are 8-bit numbers then product p is given as

$$p = a_7 b_7 2^{14} + \sum_{i=0}^6 \sum_{j=0}^6 2^{i+j} a_i b_j + 2^7 \sum_{j=0}^6 2^j \overline{b_j a_7} + 2^7 \sum_{i=0}^6 2^i \overline{a_i b_7} - 2^{15} + 2^8 \tag{12}$$

Fig. 2 shows the implementation structure of 4-bit Baugh Wooley multiplier and fig. 3 shows the corresponding internal structure of cells.

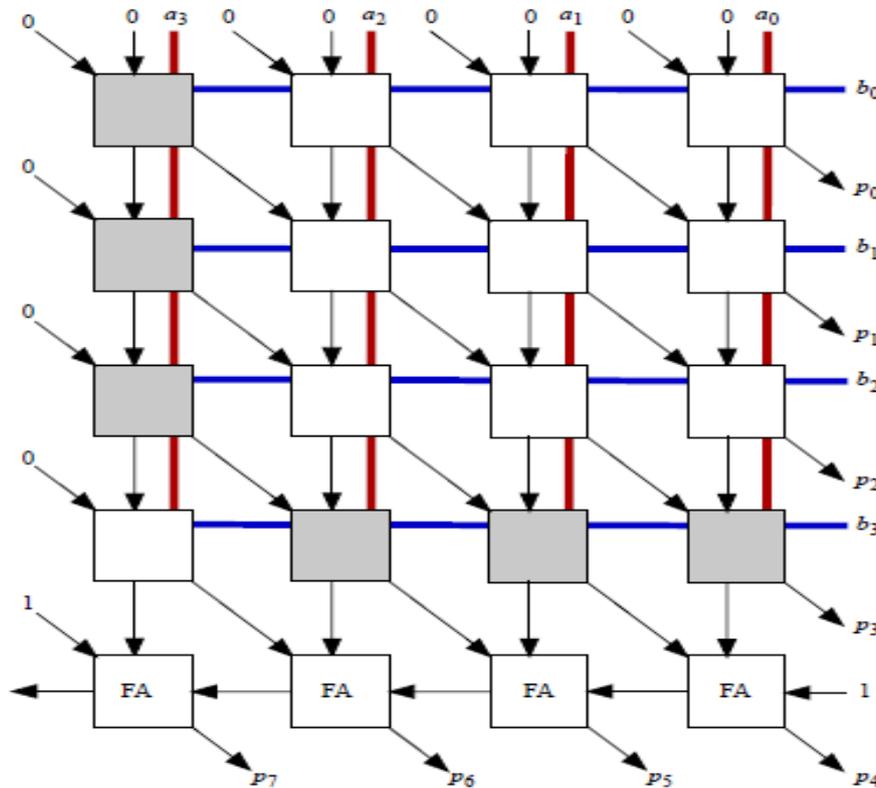


Fig. 2 Structure 4-bit Baugh Wooley multiplier

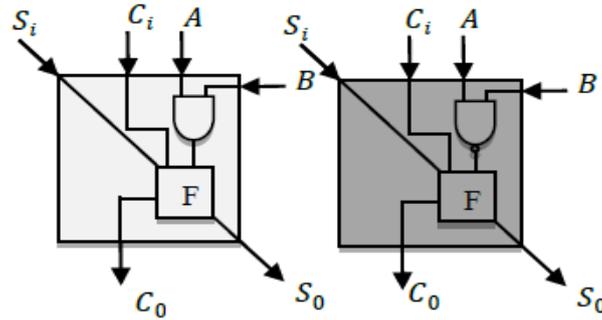


Fig. 3 Internal structure of cells

III. Microprogrammed FIR Filter Architecture

The microprogrammed FIR filter architecture consist of datapath unit and control unit [12]. The function of data path unit is to perform multiplication and addition operation on the applied input signal and impulse response. Control unit generates various control signals for data path. Fig. 4 shows the block diagram of microprogrammed FIR filter.

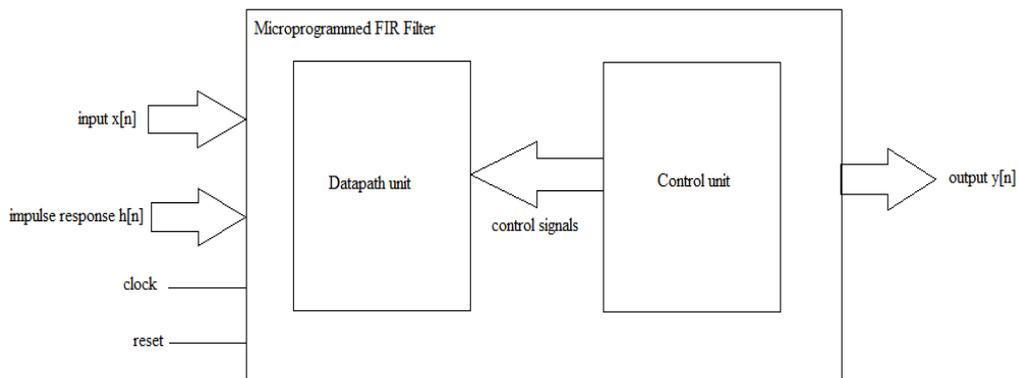


Fig. 4 Microprogrammed FIR filter architecture

The architecture of the data path unit can be classified as sequential and parallel depending upon the method adopted for computing output signal. The architecture of datapath completely depends on the nature of application. Typically it consists of multipliers, adders, data registers and multiplexers. Data registers acts as a memory to hold input signal and filter coefficients for computing. Multiplexer are used to route the appropriate data to multipliers in accordance with (2). Two approaches can be adopted for designing control unit, hardwired and microprogrammed. Microprogrammed control unit stores the microinstructions inside the memory that can be fetched using address decoding logic. These microinstructions generate the control signals for data path unit. The mainadvantage of the microprogrammed control unit is its flexibility to modify themicroprogram in the memory [12]. Microprogrammed control unit consist of address decoding logic and memory. Fig. 5 shows the simplified block diagram of microprogrammed control unit.

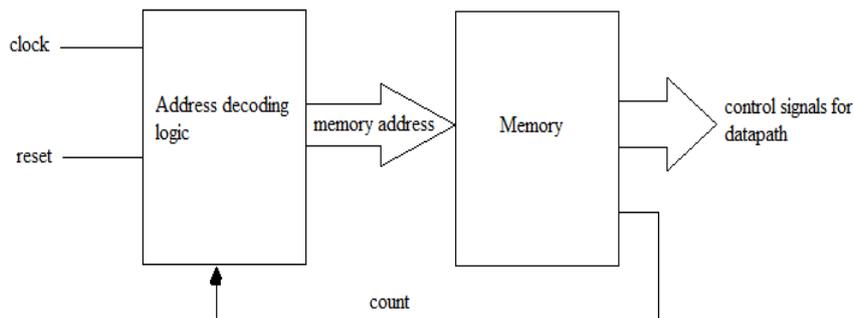


Fig. 5 Block diagram of microprogrammed control unit

The control signals from microprogrammed control unit are fed to data path unit that performs necessary operations such as load data registers with input signal and filter coefficients, perform multiplication on appropriate data, addition and latch output signal. The microinstruction also has a bit to indicate address decoding logic to stop or continue generating memory address signal.

IV. Implementation of Microprogrammed Parallel FIR Filter

The 16 x 16 bit Baugh Wooley multiplier with 18 pipelined stages implemented using fully synchronous logic and globally asynchronous locally synchronous using clock divider and decoder module given in [10] is used in the implementation of microprogrammed parallel FIR filter. GALS parallel 4-tap FIR filter that consist GALS 16-bit pipelined Baugh Wooley multipliers, carry look ahead adder and GALS microprogrammed control unit is implemented. For fair benchmarking synchronous parallel 4-tap FIR filter that consist synchronous 16-bit pipelined Baugh Wooley multipliers, carry look ahead adder and synchronous microprogrammed control unit is also implemented using same FPGA and almost same logic cells.

A. Synchronous Microprogrammed FIR Filter

Fig. 6 illustrates the block diagram of synchronous microprogrammed 4-tap FIR filter. It consists of synchronous pipelined Baugh Wooley multiplier, carry look ahead adder, synchronous microprogrammed control unit and data registers to hold input signal and filter coefficients. All the registers, multipliers and control unit are clocked simultaneously by global clock signal. Pipelined Baugh Wooley 16-bit multiplier requires 18 pipelined stages therefore it takes 18 clock cycles to generate output. Four (4) clock cycles are required to load data into both registers simultaneously. Finally two (2) clock cycles at the adder stages are required to achieve pipeline at each stage of FIR filter. Thus 24 clock cycles are required to generate final output of the filter. Since all the pipelined registers are clocked simultaneously at higher clock rate, considerable amount of power is dissipated in the circuit.

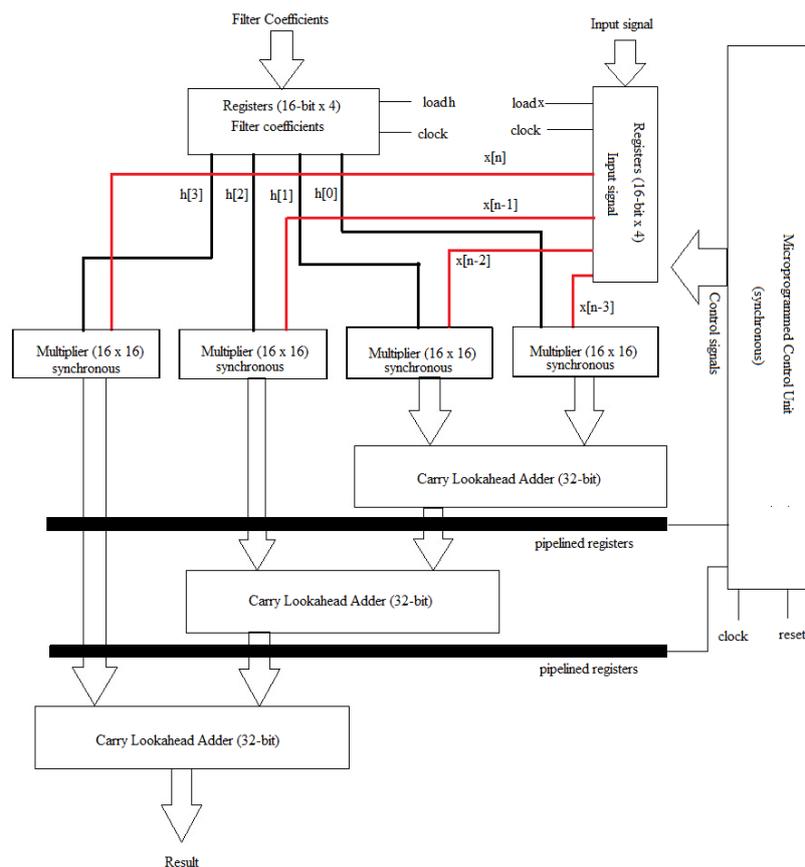


Fig. 6 Synchronous microprogrammed 4-tap FIR filter

B. GALS Microprogrammed FIR Filter

Fig. 7 illustrates the block diagram of GALS microprogrammed 4-tap FIR filter. It consists of GALS pipelined Baugh Wooley multiplier, carry look ahead adder, GALS microprogrammed control unit and data registers to hold input signal and filter coefficients. All the registers, multipliers and control unit are not

clocked simultaneously by global clock signal. GALSmicroprogrammed control unit receives a global clock signal that generates enable signals for all the pipelined stages and memory. On reception of the enable signal, memory generates various control signals to load data into the registers. Enable signals to the multiplier and pipelined stages at adder facilitate to perform operation in (2) to generate output. Since the global clock signal is applied only to the control unit termed as locally synchronous, while each subblocks of the FIR filter are not synchronized termed as globally asynchronous. The enable signals generated by the control unit are at much lower rate as compared to global clock rate, therefore the switching power dissipation reduces without affecting the speed of operation in GALS FIR filter.

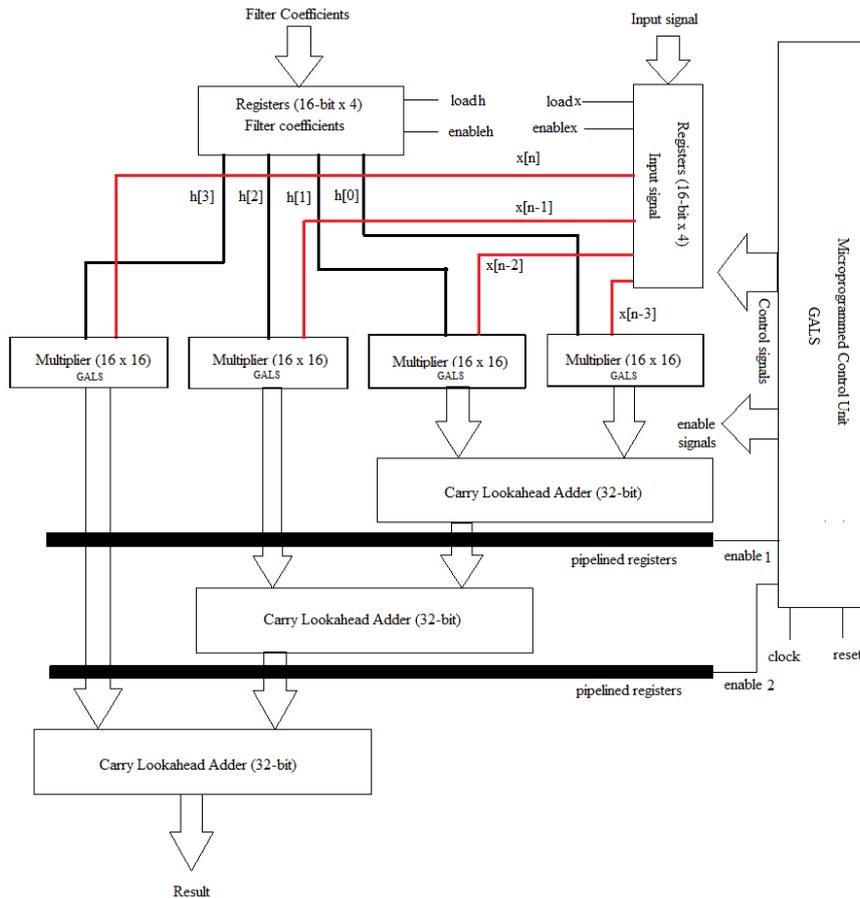


Fig. 7 GALS microprogrammed 4-tap FIR filter

Table III Results

FPGA Resources / Parameters	Fully Synchronous	GALS
Number of Slices	2347	2154
Number of LUTs	2340	2448
Number of FFs	2675	2501
Delay	8.011ns	8.011ns
Maximum Frequency	124.82 MHz	124.82 MHz
Clock Frequency	1 GHz	1 GHz
Total Power	2.516 W	0.478 W
Dynamic Power	2.17 W	0.156 W
Leakage Power	0.346 W	0.322 W

V. Results & Discussion

Virtex-5 FPGAs offer the best solution for addressing the needs of high-performance logic designers, high-performance DSP designers, and high-performance embedded systems designers with unprecedented logic, DSP, hard/soft microprocessor, and connectivity capabilities [13]. Built on a 65-nm state-of-the-art copper process technology, Virtex-5 FPGAs are a programmable alternative to custom ASIC technology [13]. The 16 x 16 bit fully synchronous and GALS pipelined MAC unit is coded in VHDL and implemented in virtex 5 FPGA (xc5vlx20t-2ff323) device. The obtained results are also confirmed on other FPGA devices such as Spartan 5,

vertex 6, and Spartan 6. The output of the each block of FIR filter verified using Xilinx ISE web pack 13.1 simulation and synthesis tool. Table III summarizes the result obtained after simulation and implementation of synchronous and GALS FIR filter. Results clearly indicate that fully synchronous FIR Filter dissipates 5.26 times more power as compared to GALS FIR filter. But at the cost of increased area GALS FIR Filter requires 1.046 times more number of slices LUT as compared to fully synchronous FIR filter.

VI. Conclusion

The fully synchronous and GALS pipelined microprogrammed FIR filter coded in VHDL and implemented in vertex 5 FPGA (xc5v1x20t-2ff323) device. The primary objective is to demonstrate low power implementation of microprogrammed parallel GALS FIR filter for digital signal processing applications. Fully synchronous microprogrammed parallel FIR filter and GALS microprogrammed FIR filter are implemented using same FPGA and almost same logic cells for fair benchmarking. Results clearly indicate that fully synchronous FIR filter dissipates 5.26 times more power as compared to GALS FIR filter. But at the cost of increased area GALS FIR filter requires 1.046 times more number of slices LUT as compared to fully synchronous FIR filter. GALS microprogrammed FIR filter can be used as basic building block in GALS implementation of digital signal processor.

References

- [1]. Semiconductor Industry Association, "International Technology Roadmap for Semiconductors," <http://www.itrs.net>.
- [2]. L. A. Plana et al., "A GALS infrastructure for a massively parallel multiprocessor," IEEE Design and Test of Computers, vol. 24, no. 5, pp. 454–463, Sep.–Oct. 2007.
- [3]. S. Dasgupta and A. Yakovlev, "Comparative analysis of GALS clocking schemes," IET Computer & Digital Technology, vol. 1, no. 2, pp. 59–69, Mar. 2007.
- [4]. Kwen-Siong Chong, et al, "Synchronous-Logic and Globally-Asynchronous-Locally-Synchronous (GALS) Acoustic Digital Signal Processors", IEEE Journal Of Solid-State Circuits, vol. 47, no. 3, pp 769 – 780, March 2012.
- [5]. R. Dobkin, R. Ginosar, and C. P. Sotiriou, "Data synchronization issues in GALS SoCs," in Proc. Int. Symp. Async. Circuits Syst. (ASYNC), pp. 170–179, 2004.
- [6]. V. K. Ingle, J. G. Proakis, "Digital Signal Processing using Matlab", in 2nd Edition, Cengage Learning, 2007.
- [7]. Uwe Beyer Baese, "Digital Signal Processing using Field Programmable Gate Array", in Springer Series on Signal & Communication Technology, 2007.
- [8]. Clive Max Maxfield, "The Design Warrior's Guide to FPGAs," Elsevier Publication, 2006.
- [9]. Gouri Wazurkar, S. L. Badjate, "Power Efficient GALS Pipelined MAC Unit for FFT with Complex Numbers", in IEEE International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016.
- [10]. Gouri Wazurkar, S. L. Badjate, "Globally Asynchronous Locally Synchronous (GALS) Pipelined Signed Multiplier", in IEEE International Conference on Computing, Analytics and Security Trends (CAST), 2016.
- [11]. R. C. Baugh and A. B. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Computers, Vol. C-22, No. 12, pp. 1045-1047, Dec. 1973.
- [12]. Syed Manzoor Qasim and Mohammed S. BenSaleh, "Hardware Implementation of Microprogrammed Controller Based Digital FIR Filter", in Springer IAENG Transactions on Engineering Technologies, 2012.
- [13]. Xilinx, "Virtex-5 Family Overview," <http://www.xilinx.com>, v5.1, 2015.