RESEARCH ARTICLE OPEN ACCESS

High Performance FPGA Based Optimization Techniques for DSP Blocks

B.Krishna[1], M.Srinu[2]
[1]*Assistant Professor ECE AAR Mahaveer Engineering College*
[2]*Assistant Professor ECE AAR Mahaveer Institute of Science and Tecnology*

ABSTRACT
A digital circuit Optimization is needed to attain higher performance include terms like minimizing the area occupancy and increasing the speed of operation. In any digital circuit the critical path (longest path delay) decides the operating frequency of the system. The operating frequency of digital circuits can be increased by several techniques such as pipelining and wave-pipelining. The proposed technique is evaluated by implementing 4x4 array multiplier, 4-tap FIR filter using array multiplier and 4-tap DA based FIR filter by using three different schemes: non-pipelining, pipelining and wave-pipelining on Spartan 3E FPGA. The WP array multiplier and FIR filters are operating at higher frequency than by using conventional pipelining and non-pipelining techniques.
Keywords: Clock skew, DA, MAC, WP, OSPAM, NRE.

Date of Submission: 08-08-2017 Date of acceptance: 22-11-2017

I. INTRODUCTION

Wave pipelining is high performance circuit designs which an implements pipelining in logic without the use of intermediate registers has the ability to improve speed, efficiency, economy in every aspects. The operating speed of the wave-pipelined (WP) circuit can be increased by the following three tasks: adjustment of the clock period, clock skew and equalization of path delays. The path-delay equalization task can be done theoretically but the real challenge is to accomplish in the presence of various different delays. So to solve the path delay equalization problem insert the control circuit in WP based circuit which will act as critical path for the data moves from input to output.

The conventional Finite Impulse Response (FIR) filters use multipliers, adders and delay elements to produce the required output. The multipliers which multiply the input with the fixed content significantly occupy more area to store their temporary values and also increase the power consumption. The multipliers in FIR filter are replaced with multiplier less Distributed Arithmetic (DA) based technique.

Most of the Digital Signal Processing (DSP) algorithms require multiplication and addition in real time the unit carrying out this function is called Multiply Accumulate (MAC). The three types of technology exist for the implementation of DSP algorithms are: Programmable DSP (PDSP) chips, Application-Specific Integrated Circuits (ASICs), Field-Programmable Gate-Arrays (FPGAs). Typically PDSP chips have only one MAC unit that can perform one MAC is less than a clock cycle. DSP processors or PDSP chips are flexible but they might not be fast enough. The reason is that the DSP processor is general purpose and that architecture requires constant instructions to be fetched, decoded and executed.

ASICs have multiple dedicated MACs that perform DSP functions in parallel but they have high cost, low volume production and the inability to make design modifications after production makes them less attractive. FPGAs offer amazing capabilities for many embedded systems from network infrastructure to military and medical. Many front-end DSP algorithms are Fast Fourier Transforms (FFTs), Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters built with ASICs or PDSPs are now replaced by FPGAs. FPGAs have advantages over ASICs such as rapid prototyping, circuit programmability, lower Non-Recurring Engineering (NRE) costs and more economical designs.

II. MOTIVATION

Multipliers are extensively used in common engineering tasks are correlations, convolutions, filtering and frequency analysis. The multiplication specifically in case of higher data path is an expensive, slow process and forms the lowest denominator for the performance of any design with the trend in modern digital systems increasingly focused on design solutions are high reliability and low cost along with power restrictions, design

verification and reliability are becoming two critical design cost components. Wave-Pipelined (WP) multipliers using the filters are found to be faster and require low power than those using pipelined multipliers.

But in case of Wave-pipelining system multiple waves of data are propagated between storage elements. The wave-pipeling was first introduced by cotton [3] and he observed that the maximum rate at which logic can propagate through a circuit depend not only on the longest path delay and also the difference between the longest and shortest path delay. A several computational "waves" (logic signals) are related to different clock cycles can propagate through the simultaneous logic. The operating speed of the WP circuit can be increased by the following three tasks: adjustment of the clock period, clock skew and equalization of path delays. A new high speed control circuit is used in [4] which will act as critical path.

III. CONCEPT OF PIPELINING

A. Logic Pipelining

A pipeline is a set of data processing elements connected in series so the output of one element is the input of the next one. Let's consider the following circuit performing a calculation on a stored value and storing the result. Flip-flop ---> combinatorial logic (10 gates) ---> flip-flop

Assume that the combinational logic is a sequence of 10 logic gates each introducing a delay of 1 ns means that it takes 10 ns for a change in the first flip-flop to reach the second and therefore this circuit can be clocked at 100 M Hz only. It takes 1 clock cycle or 10 ns for the data to reach the output and the throughput of this circuit is 1 value per cycle or 1 value per 10 ns.

Now, split the combinational logic in half as shown below: flip-flop ---> logic (5 gates) ---> flip-flop ---> logic (5 gates) ---> flip-flop.

Now it takes 5 ns for data from the first flip-flop to reach the middle and 5 ns from the middle to the end the key is that these two paths are now independent so this circuit can be clocked at 200 M Hz. The latency is 2 clock cycles for 10 ns. In addition we get 1 value out per cycle so the throughput has doubled to 1 value per 5 ns means we can give an input and will get the correct result at output 10ns later. But, we give inputs at each 5ns for this Pipelining circuit has doubled the clock speed and throughput without affecting the latency.The clock frequency is determined by the biggest delay of a combinational logic part. When this delay is reduced then the usable clock frequency is increased so the way to decrease the delay by chopping up the combinational logic part in smaller chunks with synchronous registers in between.

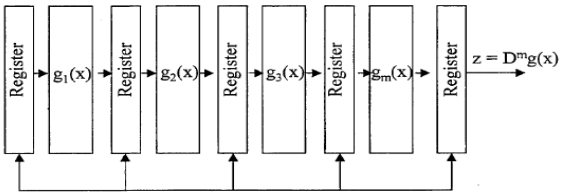

Fig.3.1 optimization schemes for FPGA based Wave pipelining multipliers and FIR filters

B. 4x4 *PIPELINED* ARRAY *MULTIPLIER*

A 4 x 4 array multiplier functions M0, M1, M2 and M4 (M's are either HAs or FAs) are shown in Fig 3.2. X3X2X1X0 is the 4 bit multiplicand and Y3Y2Y1Y0 is the 4 bit multiplier full adder is the important component in each cell. Each cell consist of AND gate which determines whether a multiplicand bit Xj is added to the incoming partial product bit based on the value of the multiplier bit Yi. PPi is unchanged and passed vertically downward if Yi=0 else each row adds the multiplicand (appropriately shifted) to the incoming partial product PPi to generate the outgoing partial product PP (i+1).

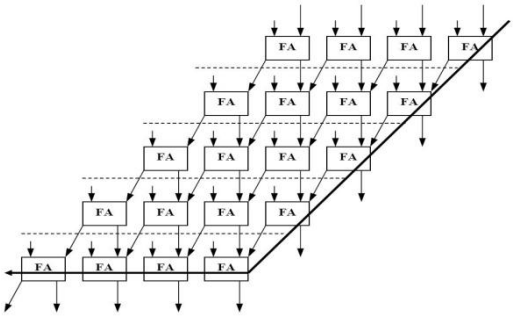

Fig 3.2 a 4 x 4 pipelined array multiplier

C**. 4x4 OPTIMALLY SYNTHESIZED PIPELINED ARRAY MULTIPLIER**

The objective of the synthesis technique proposed in [5] is to ensure that all of the four inputs of the LUTs in FPGAs are effectively engaged. Consider the optimization of a 4x4 array multiplier given in Fig 3.2, the stages involving HAs utilize the LUTs inefficiently these stages may be modified to engage all of the four inputs of the LUT as follows: Stage1 may be modified to compute the partial products due to the two least significant multiplier bits. The last 'N' stages may be reduced to 'N/2' stages by replacing the HAs with suitable functional blocks and feeding the sum and carry outputs from one stage to another stage properly. The resulting pipelined array multiplier is referred to as Optimally Synthesized Pipelined Array Multiplier (OSPAM) is shown in Fig 3.3. It consists of five stages of combinational logic blocks these stages uses the

functional blocks M0, M2, M5, M6, M7, M8, M9, and M10.

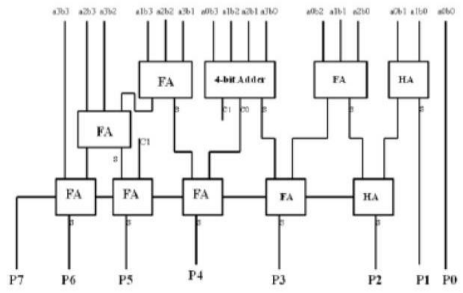

Fig 3.3 Optimally Synthesized Pipelined Array Multiplier

D. CONCEPT OF CONTROL CIRCUIT USED FOR CRITICAL PATH SCHEME

Control circuit is consists of flip-flops and XOR gates as shown in Fig 3.5 is basically a high speed circuit will be placed in the WP architecture as shown in Fig 3.4 to improve the operating speed. In WP circuit data will move from input to output in different waves in the previous work designers calculated maximum (D_{max}) and minimum (D_{min}) distances which varies from input to output and taking the difference between these two distances ($D_{max} - D_{min}$) they improve the speed. First calculate all the distances after that maximum and minimum distance to be sort out is a very time consuming process and requires more hardware. The modified WP circuit is shown in Fig 3.4. In Fig 3.5 whatever the data will enter at the input X, the same data will be at the output at very high speed. The advantage of this circuit is automatic high speed and it act as a critical path for the data moving from input to output. This control circuit is used for implementing the 4x4 WPAM and 4-tap WP FIR filter.

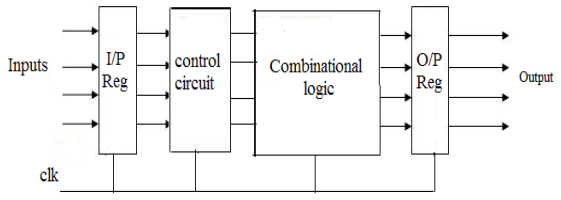

Fig 3.4 Modified WP Circuit with Control Circuit.

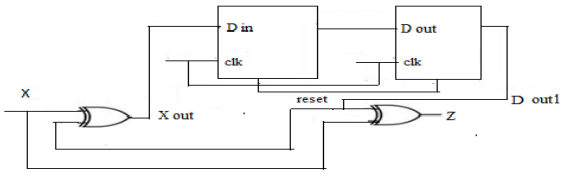

Fig 3.5 Control Circuit having D flip-flops and XOR gates.

E. FIR FILTER USING AS ARRAY MULTIPLIER

The FIR filter is shown in Fig 3.6. The input samples and the impulse response coefficients of each 4-bit are stored in the RAMs. The products of the inputs with the filter coefficients are accumulated in the adders. The outputs of these adders are finally accumulated to get the filter output. To improve the throughput of the FIR filters using the pipeline multipliers and adders. In this case pipelined array multiplier is used and the registers are placed between multipliers and adders are separated by registers. A 4-tap WP FIR filter is implemented by using 4x4 WPAM, adders and the control circuit.

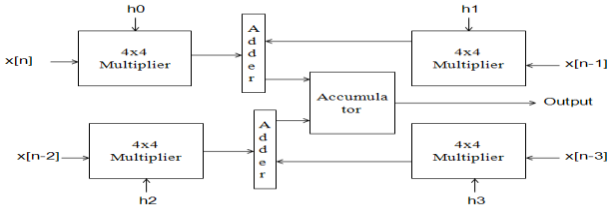

Fig.3.6 FIR Filter Using as array multiplier

F. DA BASED FIR FILTER

Memory-based structures are well suited for many DSP applications. Memory elements like Random Access Memory (RAM) or ROM are used as a part or whole of an arithmetic unit. Memory-based structures are more regular compared with the MAC structures and many other advantages are very greater potential for high throughput and reduced-latency in implementation (since the memory-access-time is much lesser than the usual multiplication-time) and expected less dynamic power consumption due to less switching activities for memory-read operations when compared to the conventional multipliers [1]. The multipliers are replaced with memory based structures to reduce area and latency of the system.

IV. IMPLEMENTATION RESULTS

A. SIMULATION RESULTS

The simulation results of 4x4 array multiplier, 4-tap FIR filter using array multiplier and 4-tap DA based FIR filters implemented in the three techniques non-pipelining, pipelining and wave-pipelining are shown in Fig 4.1 to Fig 4.9.

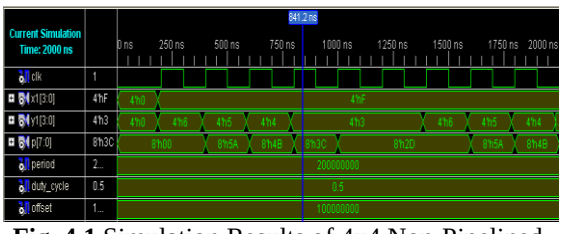

Fig .4.1 Simulation Results of 4x4 Non-Pipelined Array Multiplier

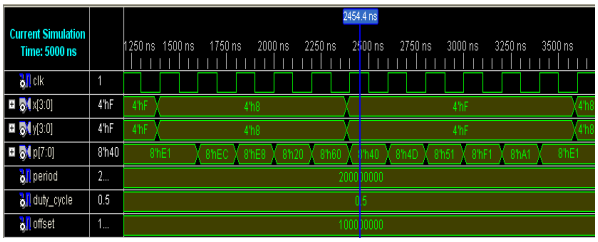

Fig .4.2 Simulation Results of 4x4 Pipelined Array Multiplier

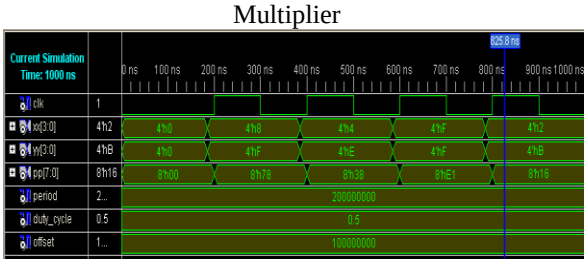

Fig .4.3 Simulation Results of 4x4 WPAM

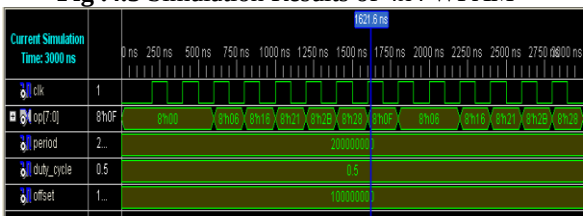

Fig .4.4 Simulation Results of 4-tap FIR Filter using Non-Pipelined Array Multiplier

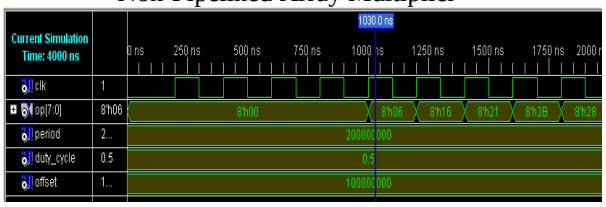

Fig .4.5 Simulation Results of 4-tap FIR Filter using Pipelined Array Multiplier

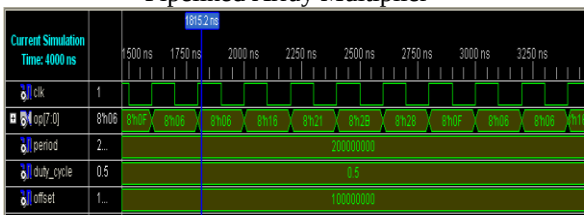

Fig .4.6 Simulation Results of 4-tap FIR Filter using WPAM

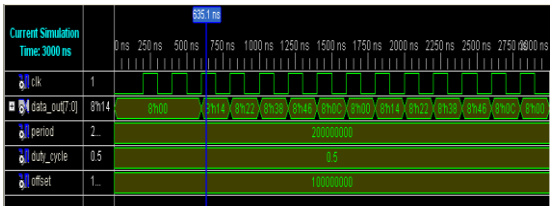

Fig .4.7 Simulation Results of 4-tap Non-Pipelined DA FIR Filter

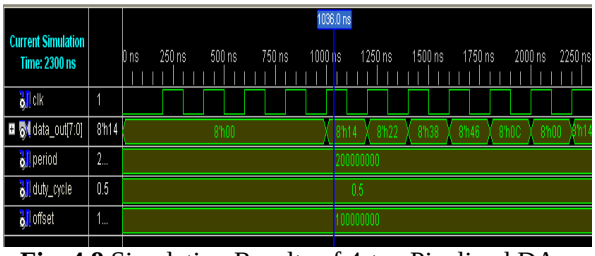

Fig .4.8 Simulation Results of 4-tap Pipelined DA FIR Filter

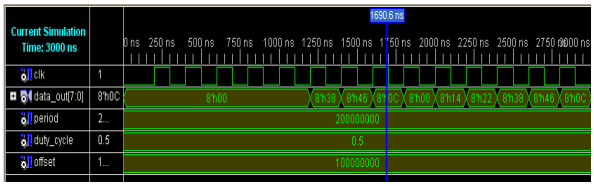

Fig .4.9 Simulation Results of 4-tap WP DA FIR Filter

B. SYNTHESIS RESULTS

The implementation results of 4x4 non-pipelined, pipelined, WP array multipliers and 4-tap FIR filter are shown in Table 4.1 and 4.2 respectively. And the implementation results of 4-tap DA based FIR filter using non-pipelining, pipelining and wave-pipelining are shown in Table 4.3.

Table 4.1 Implementation Results of 4x4 Non-Pipelined, Pipelined and WP Array Multipliers

Array Multiplier	Number of Slices	Number of Slice F- F	No. of LUTs	Min period (ns)	Max Frequency (MHz)
Non-pipelining	17	16	28	5.137	194.678
Pipelining	22	38	31	2.658	376.230
Wave-pipelining	18	12	30	1.997	500.663

Table 4.2 Implementation Results of 4-tap FIR filter using Non-Pipelined, Pipelined and WP Array Multipliers

FIR using Array Multiplier	No. of Slices	No. of Slice F-F	No. of LUTs	Min period (ns)	Max Frequency (MHz)
Non-pipelining	27	29	42	6.273	159.406
Pipelining	33	54	55	4.113	243.114
Wave-pipelining	30	31	48	3.062	326.541

Table 4.3 Implementation Results of 4-tap DA based FIR Filter using Non-Pipelining, Pipelining and Wave-Pipelining

FIR using DA	No. of Slices	No. of Slice Flip-Flops	No. of LUTs	Min period (ns)	Max Frequency (MHz)
Non-pipelining	15	20	25	4.273	236.019

Pipelining	22	38	26	3.492	286.369
Wave-Pipelining	16	17	23	3.023	330.764

V.FPGA IMPLEMENTATION

This project is implemented on FPGA Spartan 3 XC3S100E. Crystal oscillator generates the clock signal of 24MHz frequency, which is the clock source for FPGA. The block diagram of hardware implementation on FPGA is shown in Fig. 5.1.

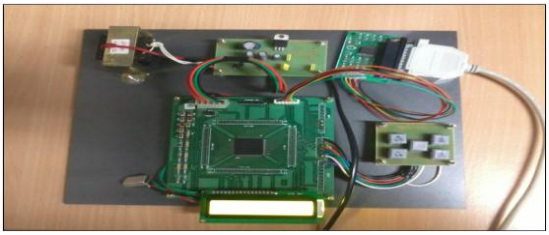

Fig .5.1 Spartan-3e FPGA Board

This is the normal Spartan-3e FPGA board. The WP DSP blocks are implemented on this board.

A. LCD display

Fig 5.2 LCD Displayed Output of 4x4 WPAM

The input samples and the filter coefficients are {3, 5, 2, 1} and {2, 4, 3, 6} respectively used for the 4-tap FIR filter using WPAM. The result of the FIR filter is {6, 22, 33, 43, 40, and 15}.

The input samples and the filter coefficients are {10, 2, 0, 0} and {2, 3, 5, 6} respectively used for the 4-tap WP FIR filter using DA. The result of the FIR filter is {20, 34, 56, 70 and 12}.

Fig 5.3 LCD Displayed Output of 4-tap FIR Filter using WPAM

Fig 5.4 LCD Displayed Output of 4-tap WP FIR Filter using DA

VI. CONCLUSION

The optimized multiplier is obtained for multiplication of two four bit Numbers. By replacing the gates that does not utilize all the inputs available efficiently which will be an important factor while using an FPGA. Further the delay is reduced from pipelined multiplier to optimized multiplier which shows the improved performance. By applying Wave-Pipelining technique the performance of the array multiplier and FIR filter increased. Traditionally, direct implementation of a K-tap FIR filter requires K MAC blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. An alternative to computing the multiplication is to decompose the MAC operations into a series of LUT accesses and summations. Advantage of this method is the LUTs readily available in the FPGAs can be utilized efficiently. This work presents the proposed DA architecture for FIR filter, i.e., multiplier-less architecture. Then, the complexity is reduced. Hence there is low power consumption. Then performance increases. Then the speed increases.The proposed architecture provides an efficient area-time implementation which involves significantly less latency and less area-delay complexity when compared with existing structures for FIR Filter.The implementation of highly efficient DA algorithm was presented in this work. The results were analyzed for 4-tap FIR filter using DA LUT on Xilinx 10.1i as a target of SPARTAN-3E FPGA device.

REFERENCES

[1]. Jiafeng Xie,Jianjun He, Guanzheng Tan, "FPGA Realization of FIR filters for high-speed and medium-speed by using modified distributed arithmetic architectures", Microelectronics journal 41(2010) 365-370.

[2]. Antonion, "Digital Filters: Analysis, Design, and Applications", McGraw-Hill, New York, 1993.

[3]. L.Cotton, "Maximum rate pipelined systems," in Proc.AFIPS Spring Joint Comput.Conf., 1969. [4] Charanjit Singh, Balwinder Singh, "Design of High Performance Modified Wave pipelined DAA Filter with Critical Path Approach", International Journal of Electrical and Electronics Engineering (IJEEE), Vol.1, No.2, 2011.

[4]. G. Lakshminarayanan, B. Venkataramani, "Optimization Techniques for FPGA-Based Wave-Pipelined DSP Blocks" , IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 7, July 2005, pp. 783–793.

[5]. Nyathi, J. And Delgado-Frias, G. J. A hybrid wave pipelined network router. *IEEE Trans.*

Circ. Syst. I: Fundam. Theory Appl. 49, 12, 1764–1772. 1999.

[6]. G.Seetharaman, B.Venkataramani, "Automation Schemes for FPGA Implementation of Wave-Pipelined Circuits", ACM Transactions on Reconfigurable Technology and Systems, Vol. 2, No. 2, Article 11, Pub.date: June 2009.

[7]. Hirak Kumar Maity, Mitra Barun Sarkar and A.Chakrobarty, "Wave Pipelining: An Analysis for High Performance Digital Circuits", International Journal of Electronic Engineering Research, Vol. 1 Number 3 (2009) pp. 269–278.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

B.Krishna High Performance FPGA Based Optimization Techniques for DSP Blocks." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 11, 2017, pp. 50-55.