Design and Implementation of FIR Filter using Low Power and High-Speed Multiplier and Adders

O. Venkata Krishna
Assoc. Professor, CVR College of Engineering/EIE Department, Hyderabad, India
Email: venkatakrishna.odugu@gmail.com

Abstract: **The Finite Impulse Response (FIR) filter is robust and high stable architecture rather than Infinite Impulse Response (IIR) Filter for the speech and image processing applications. In this paper, a high speed and low power FIR filter is designed and implemented using Radix-4 modified Booth Multiplier and Carry Look Ahead (CLA) adder. The Booth multiplier reduces the accumulation computation time in the multiplication of filter inputs and coefficients. CLA is used to reduce the critical path delay of the normal Ripple carry adder, which is used for the addition for the FIR filter. The 8-tap direct form FIR filter is implemented using Booth multiplier and CLA, and it is simulated and synthesized. The delay and power corresponding to these blocks are computed and presented. The utilization summary with respect to target FPGA of the each block is presented.**

Index Terms: **FIR, CLA, Booth multiplier, FPGA, HDL, Verilog and Low Power.**

I. INTRODUCTION

Digital filters are most frequently used for the speech processing, image processing and video processing applications. These digital filters are of two types Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. The FIR filter is preferred for the above applications due to the simplicity of the design and of high stability. The following expression (1) represents the N-tap FIR filter.

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \cdots \qquad (1)$$

Where a_0, $a_1 \ldots a_n$ are the coefficients of the filter and x[n] and y[n] are the input and outputs of the FIR filter respectively. This FIR filter can be designed in many ways, like such as direct form, transpose form and hybrid forms. The block diagram of direct form FIR filter is shown in figure.1

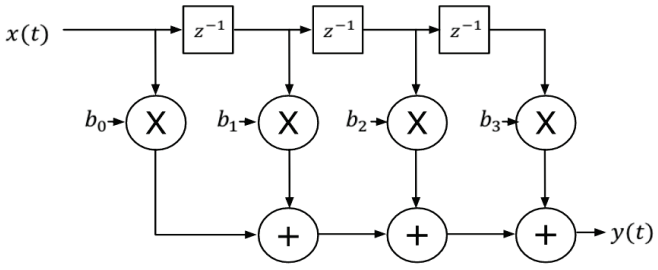


Figure 1. Block diagram of FIR filter

One of the important hardware blocks in the FIR filter is multiplier, this is used to generate the product of filter coefficients and inputs. There are different types of multipliers such as, Array multiplier, Booth's multiplier and Wallace tree multiplier, which are mostly used for the VLSI design. The other blocks are adders and delay elements. The D Flip-flops are used as delay elements in the filter, the adders are basically, Carry Look Ahead Adder (CLA) and Carry Save adders (CSA) are used in the implementation of the FIR architecture.

For the low power and high performance, the optimization can be done in block level. The optimized multiplier and adders can reduce the delay and power as well as area also. For the low power, the CLA adder and Booth multipliers are used in this work.

II. BOOTH MULTIPLIER IMPLEMENTATION

The performance of the FIR filter depends mainly on multiplier. Hence, the optimized multiplier is implemented in this section. For the low power and less area, the booth multiplier is selected for the implementation. This approach reduces half of the accumulations and hence overall area, delay reduced.

Generally, the multiplication of two binary numbers requires partial product generation, reduction of partial products and final addition of partial products. For these operations, multiplier takes long time and more hardware. The Booth multiplier which is based on the Booth's Algorithm introduced by "Andrew Donald Booth" can be used for the reduction of delay [1]. There are radix-2 and radix-4 Booth multipliers available. The radix-2 Booth multiplier requires more additions compared to radix-4 multiplier. In this work radix-4 Booth multiplier (Modified Booth multiplier) is implemented and used for the design of FIR filter.

The Booth's concept is applicable for both signed and unsigned numbers. The figures 2 and figure 3 shows the flowchart of the radix-4 Booth's algorithm for unsigned and signed numbers respectively.

The booth multiplier multiplies two binary words with length of N x N or any length. Both the numbers signed or unsigned and combination of two types of numbers can be applied as the inputs for the multiplier. Multiplication means partial product generation, shifting and final addition. The Radix-4 modified Booth Multiplier reduces the number of partial products into half [2] [3].

doi:10.32377/cvrjst1514

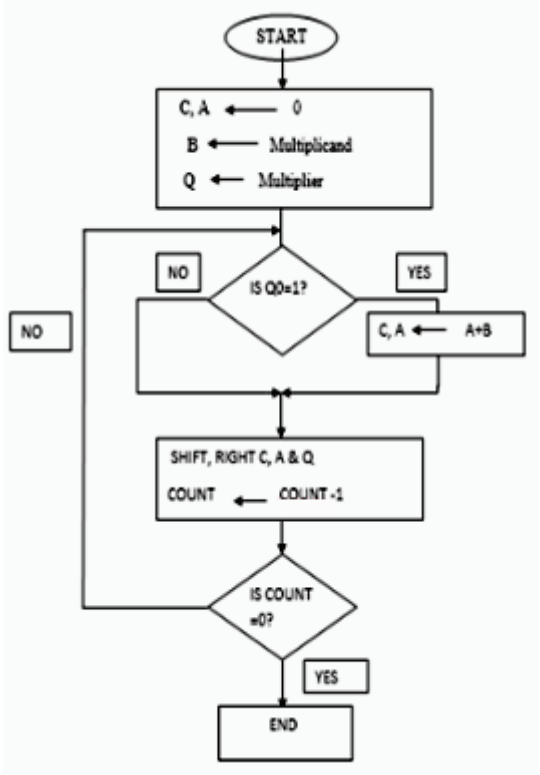


Figure 2. Booth's algorithm flowchart for the unsigned numbers.

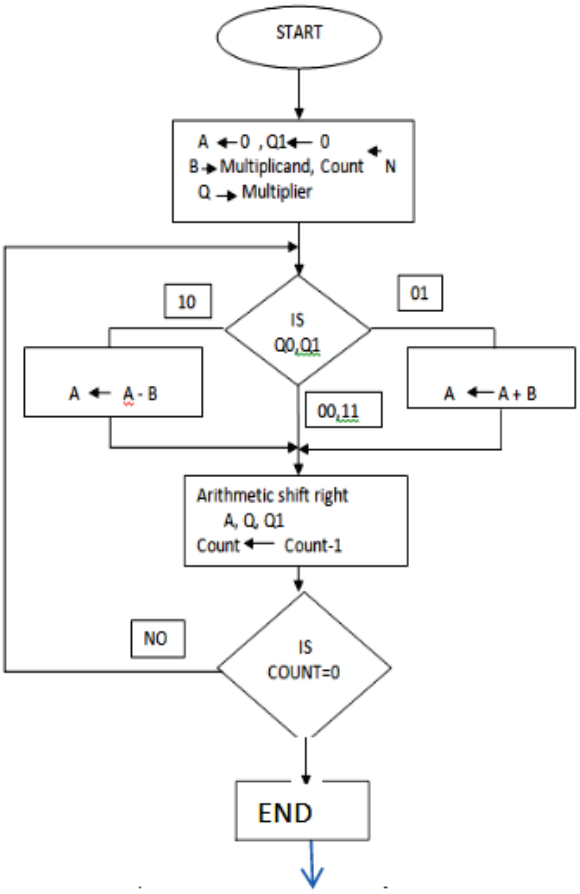


Figure 3. Booth's Algorithm in flow chart for the signed numbers.

The flow charts are converted into the logic structure, which consists of control logic, 2's compliment logic circuit, partial product register, adder and shifting circuits. The hardware structure of the radix-4 modified Booth multiplier [4] [5], is shown in figure 4. The same structure is also implemented and synthesized using Xilinx ISE tool. The RTL schematic view is shown in the figure 5.

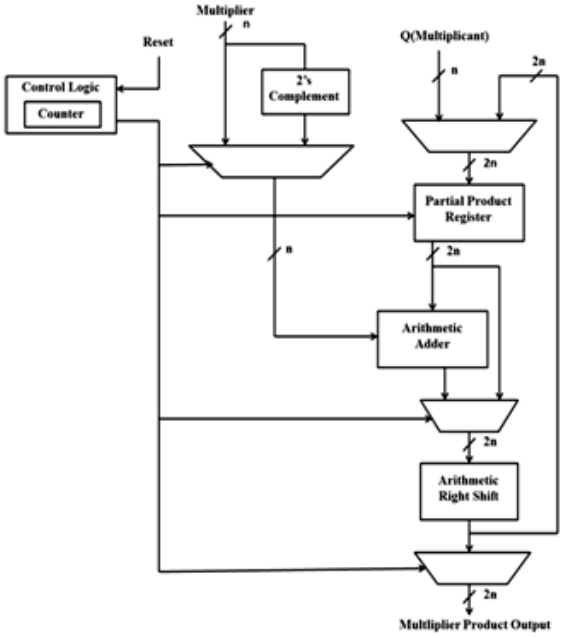


Figure 4. General architecture of Radix-4 Booth multiplier

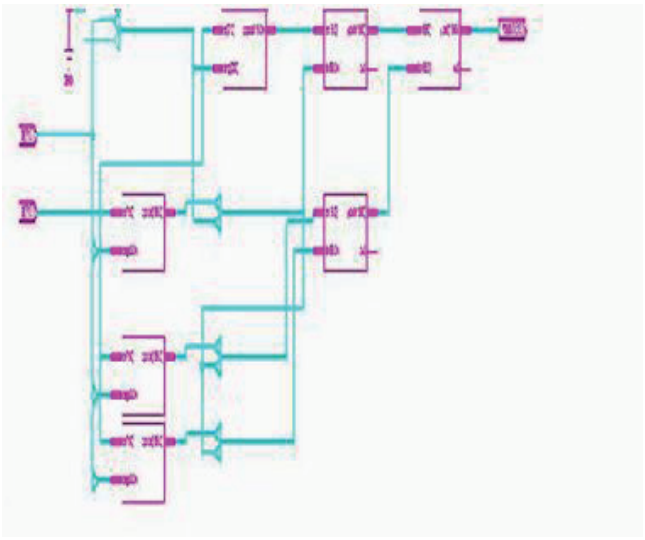


Figure 5. RTL view of Radix-4 Booth multiplier

The synthesis report of the radix-4 Booth multiplier is represented in the table I this tables reports the hardware utilization for the Booth multiplier with respect to target device as Vertex Pro FPGA. The delay and power values for the Booth Multiplier are shown in the results section.

TABLE I.
DEVICE UTILIZATION SUMMARY OF BOOTH MULTIPLIER FOR THE TARGET DEVICE VERTEX PRO FPGA

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	75	64000	0%
Number of Slice LUTs	278	64000	0%
Number of fully used LUT-FF pairs	74	193	38%
Number of bonded IOBs	132	640	20%
Number of BUFG/BUFGCTRLs	3	32	9%
Number of DSP48Es	80	256	31%

III. IMPLEMENATION OF CLA

Many High-speed adder architectures are available for the implementation of Addition, such as Carry Select, Carry Save, Carry Skip and Carry Look Ahead adders (CLA). The CLA is a robust structure that reduces the hardware and improves the speed.

The CLA overcomes the carry rippling delay, and also overcomes the latency problem in normal Ripple carry adder. The CLA adder depends on the carry generating term and carry propagate terms of the Full adder. In this, the carry is calculated for the possible carry bits and next it calculates the sum using appropriate hardware. Hence the delay is reduced with respect to the addition computation [6]. The architecture of the 4-Bit CLA is shown in the figure6.

The CLA operation depends on the propagate terms and generate terms of the full adder. The propagate and generate terms of the full adder is considered as given in the equations (2) and (3).

$$Pi = Ai \oplus Bi \quad (2)$$

$$Gi = Ai.Bi \quad (3)$$

The modified full adder sum and carry out are calculated with the help of the following equations (4) and (5) in CLA adder [7] [8].

$$Ci + 1 = Gi + (Pi.Ci) \quad (4)$$

$$Si = Pi \oplus Ci \quad (5)$$

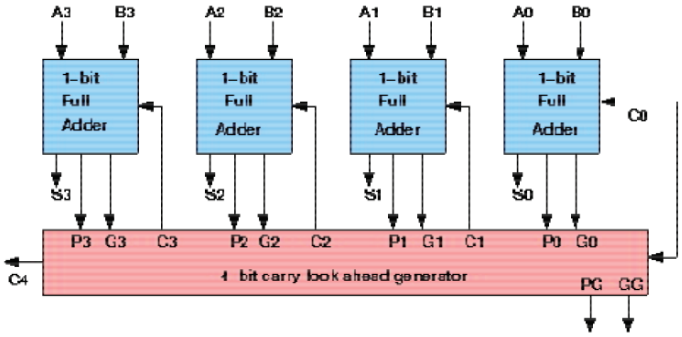
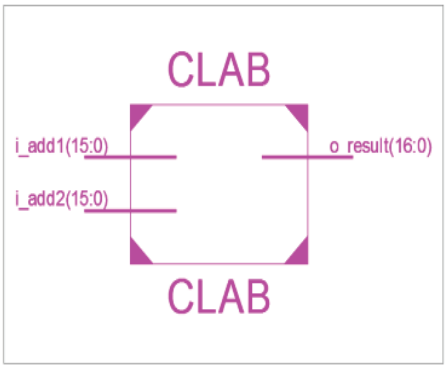
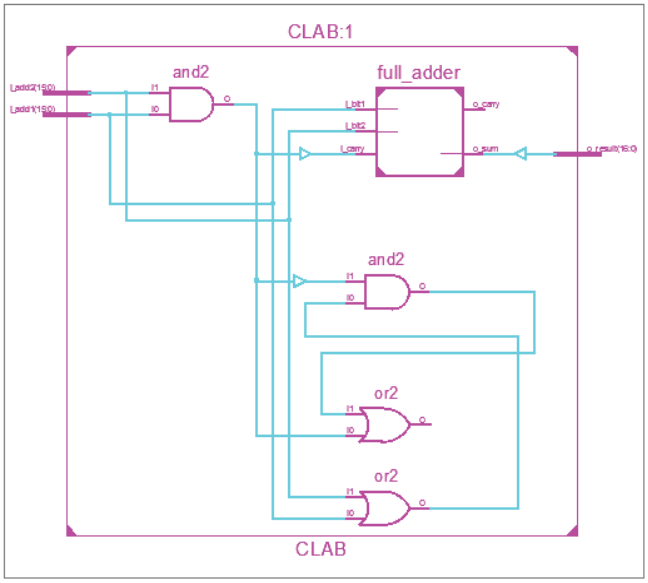


Figure 6. General Architecture of the CLA

As per the above architecture, the HDL code is written to design the 16- bit CLA and simulated. The same HDL code is also synthesized using Xilinx ISE Tool. The RTL view of the 16-bit CLA as shown in the figure 7. The synthesis report of the CLA is shown in the table II.



(a)



(b)

Figure 7. RTL view of (a) the 16-bit CLA symbol (b) internal RTL view of 16-bit CLA

TABLE II.
DEVICE UTILIZATION SUMMARY OF CLA FOR THE TARGET DEVICE VERTEX PRO FPGA

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	29	64000	0%
Number of fully used LUT-FF pairs	0	193	0%
Number of bonded IOBs	132	640	20%

IV. IMPLEMENTATION OF FIR FILTER

The 8-Tap FIR filter is designed and implemented using Booth multiplier and CLA. The block based direct form FIR architecture [9] [10] selected and coded using HDL in Xilinx ISE tool and synthesized. The block-based concept is introduced for the parallel proceeding for the FIR filter to reduce the delay [11]. In this work 4 inputs are grouped as block and applied to filter input [12] [13]. The parallel processing also reduces the power consumption of the FIR filter [14]. The RTL view and the simulated output of the FIR filter is shown in the figure 8 and figure 9 respectively.

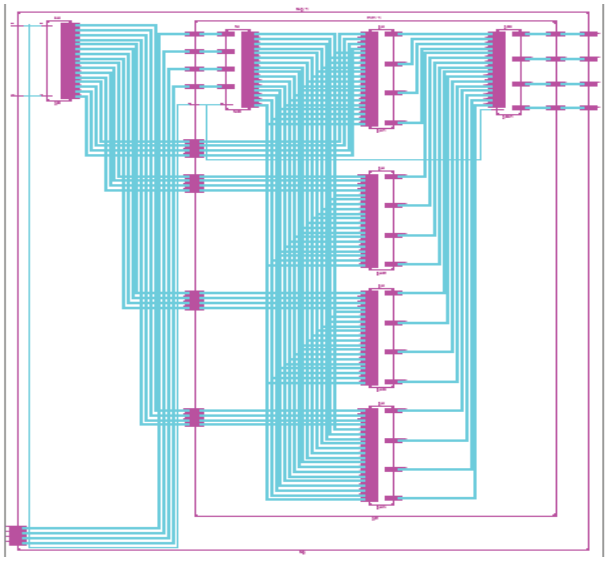


Figure 8. RTL Schenatic view of 8-tap FIR Filter

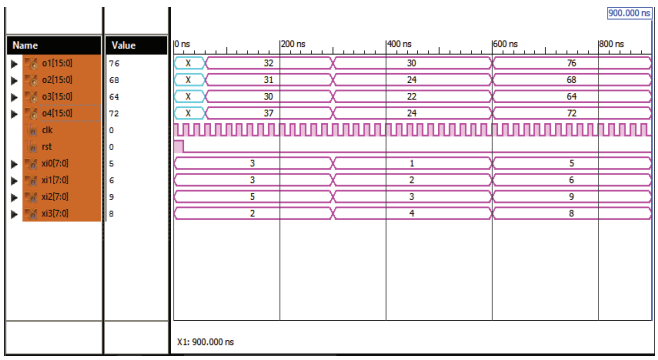


Figure 9. Simulated output results of FIR Filter for the random inputs.

The utilization summary report of the 8-tap block-based FIR filter using Xilinx software is shown in the table III. Total 80 DSP blocks are required for the entire FIR filter. The total LUTs and LUT-FF pairs are 192 and 128 utilized for the 8-tap FIR filter. The input output blocks are 98 in total.

TABLE III.
DEVICE UTILIZATION SUMMARY OF FIR FILTER FOR TARGET DEVICE VERTEX PRO FPGA

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	129	64000	0%
Number of Slice LUTs	192	64000	0%
Number of fully used LUT-FF pairs	128	193	66%
Number of bonded IOBs	98	640	15%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48Es	80	256	31%

V. RESULTS

In this section, the Radix-4 Booth multiplier and 16-bit CLA power consumption, area and delay are tabulated. The proposed 8-tap FIR Filter architecture is based on Booth multiplier and CLA adder power consumption, area and delay are computed using RTL compiler from CADENCE Tools. The TSMC 180nm CMOS technology is also used for synthesis of the FIR filter. Here, the same Xilinx HDL code is also synthesized using RTL compiler and it can be optimized. The RTL complier also generates the area, power and delay reports of the proposed 8 tap FIR filter. The table IV shows the area, delay and power of the FIR filter and important blocks of the FIR filter. In the previous section, Xilinx results corresponding to 8-tap FIR filter are discussed.

TABLE IV.
VLSI PARAMETERS OF FIR FILTER AND OTHER INTERNAL BLOCKS.

Name of the structure	Area (μm^2)	Delay (ns)	Power (μW)
CLA Adder	24544	8.371	6076
Booth Multiplier	38659	27.198	8906
8-tap FIR Filter	91925	35.391	16967

VI. Conclusions

The 8-tap FIR filter is designed and implemented using high speed adder and high-speed multiplier. The Radix-4 modified Booth multiplier and 16-bit CLA adder is considered for the implementation of high speed block based direct form 8-tap FIR filter, which is used for the digital signal processing applications. The VLSI parameters, such as area, delay and power for the optimized 8-tap FIR filter are calculated and presented. The summary report of the hardware utilization with respect to target FPGA also generated and presented in this paper for Booth multiplier, CLA adder and FIR filter. The HDL code is written and simulated, synthesized using Xilinx and RTL Compiler from CADENCE also.

References

[1] Oscal T. -C. Chen, Sandy Wang, and Yi-Wen Wu, "Minimization of Switching Activities of Partial Products for Designing Low-Power Multipliers", IEEE Transactions on VLSI Systems, vol. 11, no. 3, June 2003.

[2] Shiann-Rong Kuang and Jiun-Ping Wang "Design of power efficient configurable booth multiplier" IEEE Trans. Circuits Syst. I Regular Papers vol. 57, no.3, pp. 568-580, March2010.

[3] TusharV.More, Dr.R.V.Kshirsagar, "Design of Low Power Column Bypass Multiplier using FPGA" IEEE journal of solid-state, circuits, vol 31,pp 1535-1546,July 2011.

[4] Akanksha Sharma, Akriti Srivastava, Anchal Agarwal, Divya Rana, Sonali Bansa "Design and Implementation of Booth Multiplier and Its Application Using VHDL. International Journal of Scientific Engineering and Technology, Volume No.3 Issue No.5, pp: 561 –563, May2014.

[5] A.RamaVasantha, M.Sai Satya Sri" Design and Implementation of FPGA Radix-4 Booth Multiplication Algorithm" in International Journal of Research in Computer and Communication Technology, Vol 3, Issue 9, September – 2014.

[6] Jagannath Samanta, Mousam Halder, Bishnu Prasad De "Performance Analysis of High Speed Low Power Carry Look-Ahead Adder" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, Jan-2013.

[7] Fu -Chiung Cheng Stephen H. Unger "Delay - Insensitive Carry - Look ahead Adders" Manoj Kumar, Sandeep K. Aryaand Sujata Pandey "Single bit full adder design using 8 transistors" International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.4, December 2011.

[8] M.C.B. Osorio, C.A. Sampaio, A. I. Reis, R.P. Ribas., etc.al. "Enhanced 32 - bit Carry Look Ahead Adder using Multiple Output Enable-Disable CMOS Differential Logic". SBCCI, pp. 181-185, 2004.

[9] A. Umasankar and N. Vasudevan," Design a and Analysis of Various Slice Reduction Algorithm for Low Power and Area Efficient FIR Filter", ICCTET13, IEEE Conf. July 2013.

[10] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Computer-Aided Design Integr. Circuits Syst., vol. 29, no. 2, pp. 275–288, Feb. 2010.

[11] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 7, pp. 511–515, Jul. 2014.

[12] B. K. Mohanty and P. K. Meher, "A high- performance FIR Filter Architecture for Fixed and Reconfigurable Applications," IEEE Trans. on VLSI systems, vol. 24, issue 2, pp.444 –452, 2016.

[13] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," IEEE Trans. Wireless Commun., vol. 7, no. 5, pp. 1669–1675, Jul. 2006.

[14] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," IEEE J. Solid State Circuits, vol. 39, no. 2, pp. 348–357, Feb. 2004.