Design of low power fir filter technique with booth multiplier and delay buffer

[1,]Gummadi.Lakshmi Pratap , [2,]P.suresh
M.Tech student,Professor
Dept of ECE,Sri Sunflower college og Engg.&Technology
Challapalli, Krishna (D), A.P - 521131

ABSTRACT

This paper presents the methods to reduce dynamic power consumption of a digital Finite Imppulse Respanse (FIR)filter these methods include low power serial adder, combinational booth multiplier And low-powerdelay buffer. The proposed delay buffer uses several new techniquesto reduce its power consumption. Since delay buffers are accessed sequentially, it adopts a ring-counter addressing scheme. In the ring counter, double-edge-triggered (DET) flip-flops are utilized to reduce the operating frequency by half and the C-element gated-clock strategy is proposed. A novel gated-clock-drivertree is then applied to further reduce the activity along the clock distribution network. Moreover, the gated-driver-tree idea is also employed in the input and output ports of the memory block to decrease their loading, thus saving even more power. folding transformation in linear pheasarchitecture and applied to firfilters to power consumption reduced thus reduce power consumption due to glitching Is also reduced. The proposed FIR filters were synthesized implemented using Xilinx ISE Virtex IVFPGA and power is analized using Xilinx XPower analyzer.

I. INTRODUCTION

Multipliers are one of the most important arithmetic units in microprocessors and DSPs and also a major source of power dissipation. Reducing the power dissipation of multipliers is key to satisfying the overall power budget of various digital circuits and systems. If multiplication of two 8-bit numbers is performed, then the result is 16-bit. To produce that result array multiplication generates 8 partial products. NOR gates can also be used instead of AND in accordance with the De Morgan's Law:

$$A.B = (A' + B')'$$

The proposed work is MAC FIR filter based BOOTH multiplier which reduces the number of partial products by half compared to a traditional implementation. Using the pass logic based implementations; the number of transistors was reduced, resulting in hardware-reduced and consequently power-aware designs. This multiplier has two stages. In the first stage, the partial products are generated by the Booth encoder and the partial product generator (PPG), and are summed by compressors. In the second stage, the two final products are added to form the final product through a final adder [3]. The booth encoder reduces the generation of number of partial products there by reducing power. The objective of the project is to design and implement the FIR filter which consumes less power. To achieve the requirements a low power booth multiplier, adder and delay buffer are used. Along with the low power consumption the proposed technique also increases the speed.

II. BOOTH MULTIPLIER

One of the solutions of realizing high speed multipliers is to enhance parallelism which helps to decrease the number of subsequent calculation stages there by reducing the power consumption.

2.1.Delay Buffer

A serial access memory is needed in temporary storage of signals that are being processed (i.e. delay buffer). Currently, most circuits adopt static random access memory (SRAM) plus some control/addressing logic to implement delay buffers and it is shown in below Figure.

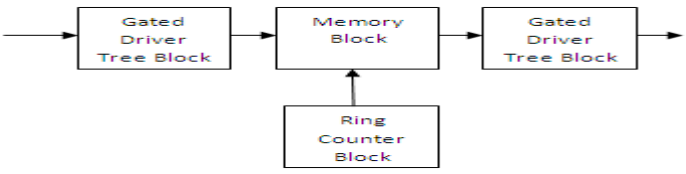


Fig System development

2.2.Implementation

The realization of FIR filters can be accomplished by using the following design procedure

1. Choose filter structure
2. Choose number representation, e.g. signed magnitude, two's compliment
3. Implement software code, or hardware circuit, which will perform actual filtering.
4. Verify the simulation that the resulting design meets given performance specifications.

A simple low pass filter is designed. Any filter specifications can be specified but for the sake of simplicity the following specifications are used in this thesis. Hanning windowing technique is used to generate the finite impulse response of the filter. Filter coefficients are generated by using the Matlab.

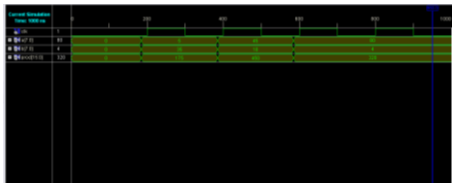
2.3.Simulation results

FIR filter using Array Multiplier

In implementing the FIR filter using Array Multiplier there are various blocks. They are the array multiplier, full adder, 16-bit adder. In this project all the blocks are simulated using Xilinx simulator.

2.4. Multiplier

The below Figure shows the output waveform of an array multiplier with different inputs (i.e., a, b) and outputs (i.e., prod). Here the multiplier (5, 45, 80) and multiplicand (35, 10, 4) both are 8-bit and the final product (175, 450, 320) is 16-bit.



2.5.Full Adder

The full adder consists of three inputs (i.e., a, b, c) and it produces two outputs (i.e., sum, carry) and its corresponding output waveforms are shown in below Figure.

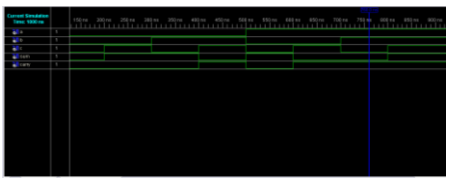


Fig: Output waveform of Full Adder

2.6.16-bit adder

In this project the 16-bit ripple adder is used and it consists of 16 full adders which adds two 16-bit numbers (a, b) and produces two outputs one is sum (here it is y_{out}) of 16-bit and carry (here it is c_{out}) of 1-bit. The corresponding simulated waveforms are shown in below Figure. For example the two 16-bit numbers taken are 33, 43 and its sum is 76 with a carry 0. The ripple adder is shown in below Figure. It consists of the result for the various inputs also.

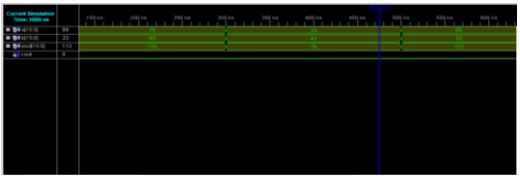


Fig: 16-bit Adder

2.7.FIR filter using Array Multiplier

In the design of FIR filter using array multiplier five inputs are taken (X1, X2, X3, X4,X5) and the single output(Y), the output responds to the inputs in synchronous with the clock signal and it is shown in below Figure.

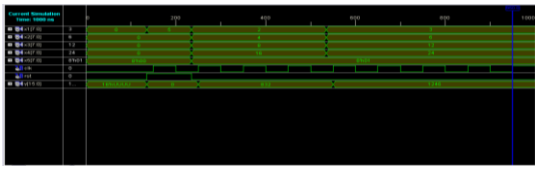


Fig: FIR filter using Array Multiplier

2.8.FIR filter using BOOTH Multiplier

In implementing the FIR filter using Booth Multiplier there are various blocks. The BOOTH multiplier, BOOTH encoder, full adder, 16-bit adder are the various blocks. The outputs of full adder and 16-bit adder are same as the implementation in array multiplier.

2.9.BOOTH Encoder

Booth encoder encodes the given multiplicand depending on the multiplier bits. In this depending on the arg, the input 'a' is encoded and its result 'pprod' is shown in Figure.

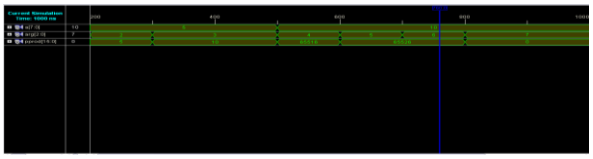


Fig: Output waveform of BOOTH Encoder

2.10.BOOTH Multiplier

It performs multiplication operation depending on the encoded value. Here the inputs are 'a','b' both are 8-bit and it produces result of 16-bit Y_{out}. It also indicates overflow (ovf) if the result exceeds the 16-bit, it is shown in Figure.

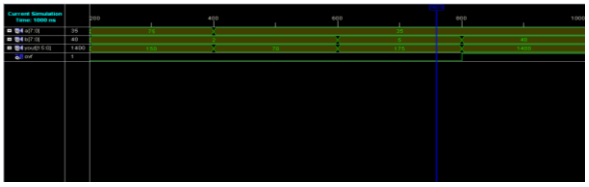


Fig: BOOTH Multiplier

2.11.FIR filter using Booth Multiplier

The filter design contains 5 inputs (X1, X2, X3, X4, X5) according to the specifications. If asynchronous input 'rst' is equal to '1', then the output 'Y' equal to '0'. Otherwise the output responds to the inputs in accordance with the synchronous input 'clk' and is shown in Figure

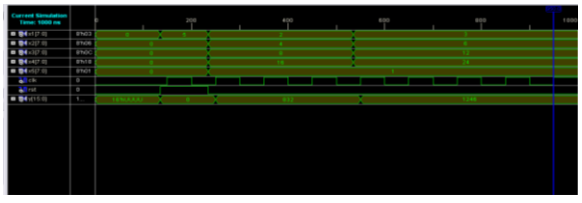


Fig: Output waveform of FIR filter using Booth Multiplier

2.12.Delay Buffer

Delay Buffer consists of the ring counter, gated driver tree, memory block. The ring counter consists of Dual Edge Trigered flip flop, c-element, gated clock.

2.13.C-Element

The output (c_out)of the c-element does not changes when both the inputs (a, b) are not same and it is equal to the input when they are same and it is shown in Figure.

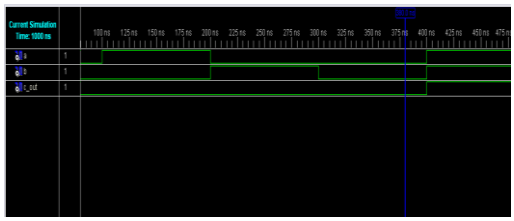


Fig: Output waveform of c-element

2.14.Power report

Power report of FIR filter using Booth Multiplier

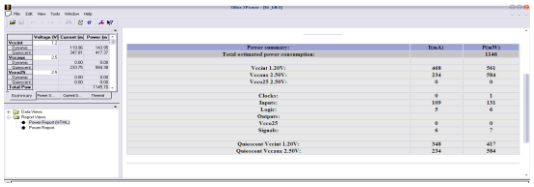


Fig: Power report of FIR filter using Booth Multiplier

Power report of FIR filter using Array Multiplier

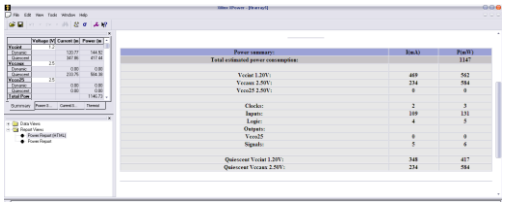


Fig: Power report of FIR filter using Array Multiplier

First Figure shows the power report of FIR filter using array multiplier and it shows that the power consumed by that filter is more when compared with the FIR filter using booth multiplier is as shown in Second Figure. since it requires N/2 partial products for multiplication operation. Power report is generated by considering 50MHz frequency.

Device utilization summary of FIR filter using Booth multiplier

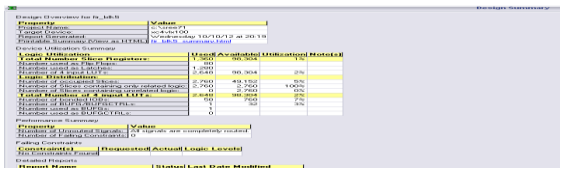


Fig: Device utilization summary of FIR filter using Booth Multiplier

Device utilization summary of FIR filter using Array multiplier

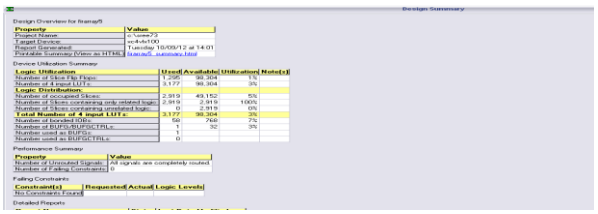


Fig: Device utilization summary of FIR filter using Array Multiplier

The above figures shows the device utilization summary of both the techniques. FIR filter using booth multiplier utilizes less number of devices when compared with the FIR filter using array multiplier, eg.,the number of 4 input LUT's required is 2648 for the present technique where as later one requires 3177 out off 98304 LUT's. Similarly the different devices occupied are also less for FIR filter design using booth multiplier The maximum frequency of the FIR filter design using booth multiplier is 122.763MHz and it is 120.179MHz for the previous technique. Therefore speed also increases with the new technique. There are so many parameters for comparison but here only 5 are considered. As shown in the figure the no. of LUT's used, the no. of multiplexers used and the no. of occupied slices are less for FIR filter design using booth multiplier so that this technique consumes less power when compared with the FIR filter using array multiplier technique and the maximum frequency is also more the new technique when compared with the other one. With this the minimum delay for each operation is reduced so that speed also increases.

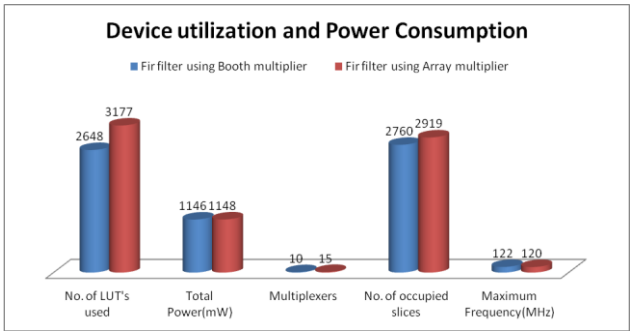


Fig: Performance chart of FIR filter using Booth and Array multiplier for LPF

III. CONCLUSION

In this project, results shows that FIR filter using booth multiplier employs the booth encoder and it generates less number of partial products than the implementation of the FIR filter using the array multiplier. For storing the filter coefficients delay buffer is used to reduce the power. Therefore speed also increases with the new technique. From the results, one can conclude that FIR filter using booth multiplier is the best method and consumes less power.

[1] Bahram Rashidi, Majid Pourormazd,"Design And Implementation Of Low Power
[2] Digital FIR Filter Based On Low Power
[3] Multipliers And Adders On Xilinx FPGA",
[4] IEEE conference on electronic computer technology (ICECT),Volume 2, 2011.
[5] Po-Chun Hsieh, Jing-Siang Jhuang, Pei-Yun

REFERENCES

[1] Tsai,Tzi-Dar Chiueh, "A Low-Power DelayBuffer Using Gated Driver Tree", IEEE
[2] Transactions On Very Large Scale Integration
[3] (VLSI) Systems, Vol. 17, No. 9, September 2009
[4] Thankachan, Shibi, "64 x 64 Bit Multiplier Using Pass Logic" Computer Science Theses,2006.
[5] Nam-Phuong D. Nguyen, Hiroyuki Kuwahara, Chris J. Myers, James P. Keener, "The Design of a Genetic Muller C-Element", Async07, March 2006.
[6] Mirzaei S., Hosangadi A. and Kastner R. , "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", International Conference on Computer Design (ICCD),pp 308-313, 2006
[7] Jones, D. L., "FIR Filter Structures", Version 1.2: Oct 10, 2004.
[8] Ronak Bajaj,SaranshChabra,SreehariVeeramachaneni, M B Srinivas, "A Novel, Low-Power Array Multiplier Architecture", 2002.
[9] Yun-Nan Chang, J han H. Satyanarayana, and Keshab K. Parhi, "Design And Implementation Of Low Power Digital Serial Multipliers", circuits and systems,1998.
[10] Massoud Pedram, Qing Wu, Xunwei Wu, "A New Design for Double Edge Triggered Flip-flops", IEEE, 1998.
[11] Prokis J. G., Manolakis D. G., "Digital Signal Processing", 4th Edition, PHI publication.
[12] Vallavraj & Salivhanan, "Digital Signal Processing", TMH publications.
[13] M.Morris Mano, "Digital design", PTR publications.
[14] J.Bhasker, " VHDL Primer", 3rd edition, PB publications.