



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

DESIGN OF DIGIT SERIAL FIR FILTER

SHAILESH S. NICHAT

Dept. of Electronics & Telecommunication, G. H. R. C. E. M., Amravati. S. G. B. A. U., Amravati, India.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: Many of the people give the solution for the making multiplier and accumulator (MAC) section which is use in Digital signal processor for doing addition and multiplication by using number of algorithm and architecture. Multiple constant multiplications (MCM) which dominates the complexity of many digital signal processing systems. This paper gives the solution which reduces the complexity of multiple constant multiplications. With reducing the gate area and required power also. For that in this paper use the shift add architecture and digit serial addition concept. Which is provide the better solution in the form of area and power as compared with multiple Constant multiplier using Common Sub-expression Elimination (CSE) algorithm with high efficiency.

Keywords: MAC, CSE, MCM.

Corresponding Author: MR. SHAILESH S. NICHAT

Access Online On:

www.ijpret.com

How to Cite This Article:

Shailesh Nichat, IJPRET, 2014; Volume 2 (9): 216-225



PAPER-QR CODE

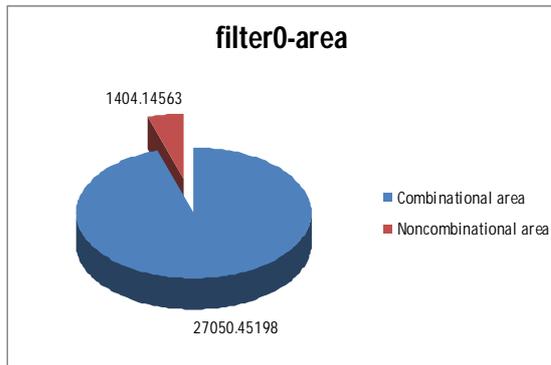
INTRODUCTION

Finite Impulse Response (FIR) filter are of great importance in digital signal processing (DSP) systems since their characteristics in liner phase and feed forward implementation make them very useful for building stable high performance filters. The direct and transposed form FIR filters implementations are mostly use for filtering signal. Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency. The multiplier block of the digital FIR filter in input is realized, has significant on the complexity and performance of the design because of constant multiplications are required. This is generally known as the multiple constant Multiplication (MCM) operation. To do the multiple constant Multiplication (MCM) in MAC block in filter there is many sources available but they are not sufficient to give the better solution. In this paper address this problem by using Shift/add and serial pipelining concept.

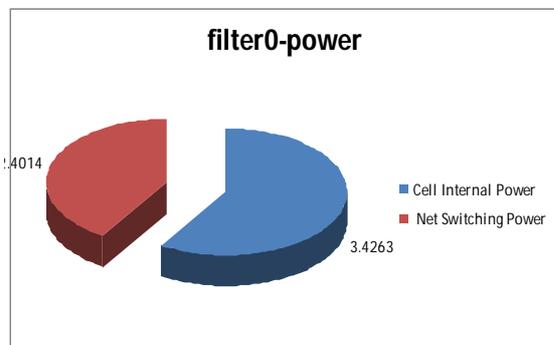
1.1 Direct Form Filter 0

In this experiment first operation perform on the direct form filter. To see the performance of direct form filter, in direct form filter the Latch is connected in upper side with (0-7) input signal. In this experiment eighteen point filter is use. This length can be increase for higher application. The MAC unit is use for addition and subtraction operation. This set of functions implements arbitrary order recursive filter using Direct Form structure. The filters are implemented as cascade of second order Biquard section. These functions provide a slight memory saving. After executing this filter it gives some result which is given in below in the form of pi-chart.

Only floating point data is supported. This function operates on block of input and output data and each call to the function processes block size sample through the filter. The alternate Direct Form only needs N latch units, where N is the order of the filter. This structure is obtained by reversing the order of the numerator and denominator sections of direct form, since they are in fact two linear systems, The disadvantage is that direct form increases the possibility of arithmetic overflow for filter of high Q or resonance. It has been shown that as Q increases, the round-off noise of both direct form topologies increases without bounds. This is because, conceptually, the signal is first passed through an all-pole filter (which normally boosts gain at the resonant frequencies) before the result of that is saturated, then passed through an all-zero filter. Here figure 'a' shows the area which is consumed by filter called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)



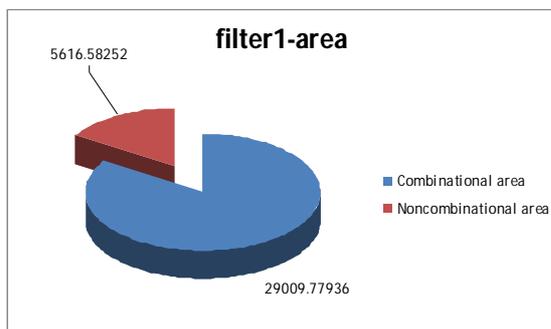
(b)

Figure 1.1: Direct Form FIR Filter. Fig (a) Specific area use by filter. Fig (b) Power use by filter.

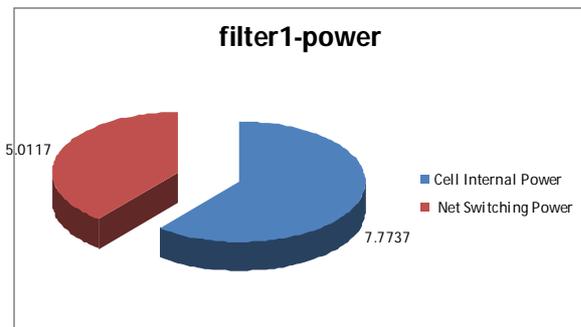
1.2 Transpose Form Filter 1.

For second experiment here use Transpose form filter. In direct form filter changes in the structure is not possible. If change the position of Latch in Direct form it gives wrong output. For execution of proper operation of filter latch is connected to the bottom. This structure is called Transpose form filter. The region to use of Transpose form filter is here any change can possible than direct form filter. Here latch have 0-31 input from adder block. One clock and one reset button is connected to all latch simultaneously so all latches execute at same time. The direct form has one huge addition at the output, which maps very well to the MAC operations on a DSP processor but may be a problem if implemented in hardware. The transposed structure has many small additions separated by delay elements. This might work better in a FPGA or ASIC implementation. The direct form FIR filter needs extra pipeline registers between the adders to reduce the delay of the adder tree and to achieve high throughput. The FIR filter

with transposed structure has registers between the adders and can achieve high throughput without adding any extra pipeline registers. Transposed form is self pipelined with the cycle period the delay of an adder and a multiplier. But it has more area than directed form. You can add delay even in directed form or transposed form to make the design faster which resulted in mixed form. Here figure 'a' shows the area which is consumed by filter called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)



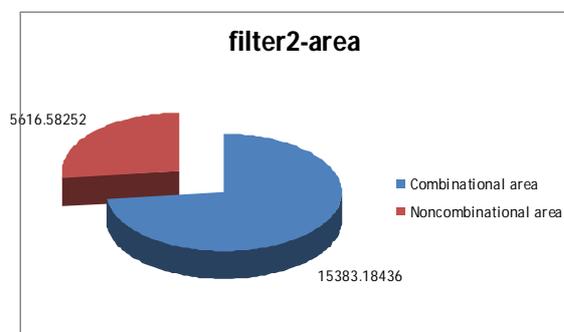
(b)

Figure 1.2: Transpose Form Filter 1. Fig (a) Specific area use by filter. Fig (b) Power use by filter.

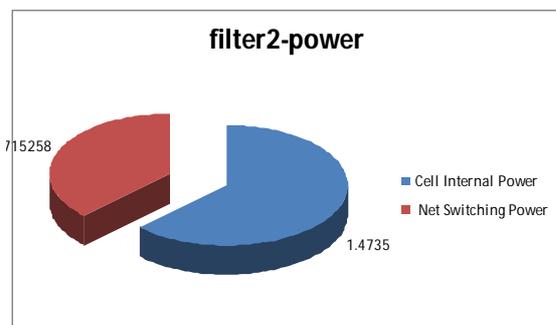
1.3 Filter 2

The objective of this project is to go through a design cycle from initial conception to simulation. In this case, it has been taken several steps further and synthesis as well as place & route was also achieved. The goal is to design and simulate an 8-by-8 bit shift/add multiplier. The result is a completely synthesized 8-by-8 bit and 32-by-32 bit shift/add multiplier with

various design options for speed and area. In this filter the shift add architecture is used to generate the coefficient, this coefficient is repeated in a sequence so here this repeated coefficient is used again so combinational area gets reduced. If the filter is an 18-point filter then it means it has 9 points or 9 coefficients are repeated. So these shift/add architectures provide the facility to reuse the coefficient. Because of that reuse concept the 9 multiplier blocks get directly reduced so the complexity of the filter gets reduced and the combinational area required for the filter also gets reduced. Because of less circuitry, the power required for the filters is also less. This is a similar structure like the transpose form filter, only the shift/add structure improves the area and power consumption of the filter. Here figure 'a' shows the area which is consumed by the filter, called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)

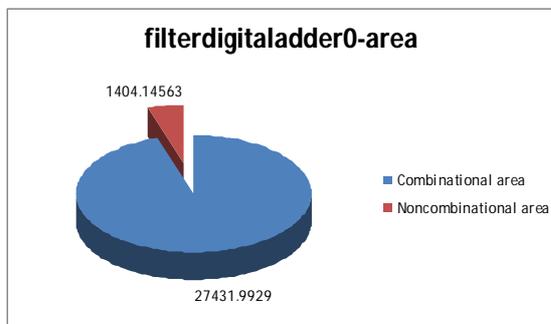


(b)

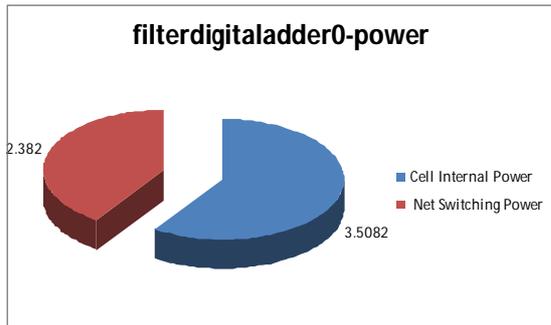
Figure 1.3: Filter 2. Fig (a) Specific area use by filter. Fig (b) Power use by filter.

1.4 Filter Digital Adder 0

This is a same as Direct form filter which is describe in first section here only the change is done in structure is pipelining concept which is reduce the area as well as power consumption of MCM block. Pipelining leads to a reduction in the critical path Either increases the clock speed (or sampling speed) or reduces the power consumption at same speed in a DSP system. It's a Direct form FIR filter where make change in adder block Multiple constant multiplication (MCM) is an efficient way of implementing several constant multiplications with the same input data. The input is applying in pipelining format that one by one sequence. If this operation is perform in parallel input concept that time addition is done very fast in MCM block but it required more power and more area in MCM block. By using pipelining concept input is apply to adder block one by one so the speed of operation of MCM block get little slow but it save the area and power which is required for MCM block. The number of adders and subtracters is reduced resulting in a low complexity implementation. However, for digit-serial arithmetic a shift requires a flip-flop, and, hence, the number of shifts should be taken into consideration as well. In this work we investigate the area, speed, power trade-offs for implementation of FIR filters using MCM and digit-serial arithmetic. We also introduce an algorithm for reducing both the number of adders and subtracters as well as the number of shifts. Here figure 'a' shows the area which is consumed by filter called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)

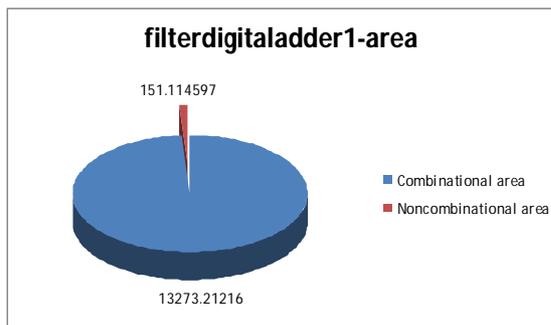


(b)

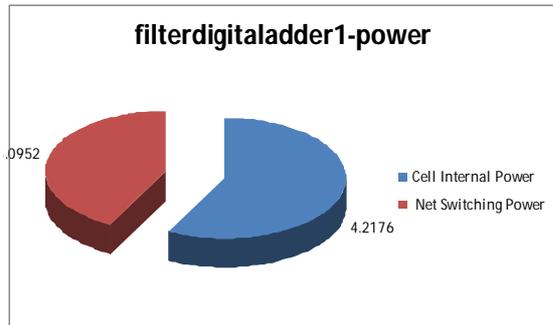
Figure 1.4: Filter Digital Adder 0. Fig (a) Specific area use by filter. Fig (b) Power use by filter.

1.5 Filter Digital Adder 1

It is a Transpose form filter which is Remove the latches make change in adder block Most work on implementation of digit-serial FIR filters has focused on implementation in FPGAs and without using multiplier blocks. However, in the digit-size trade-off in implementation of digit-serial transposed form FIR filters using multiplier blocks was studied. One of the best MCM algorithms in terms of number of adders, If some real-time application requires a faster input rate (sample rate), then this direct-form structure cannot be used! In this case, the critical path can be reduced by either pipelining or parallel processing Pipelining reduce the effective critical path by introducing pipelining Here figure 'a' shows the area which is consumed by filter called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)

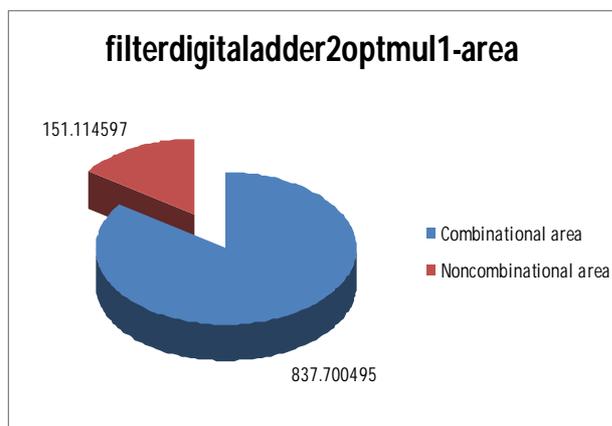


(b)

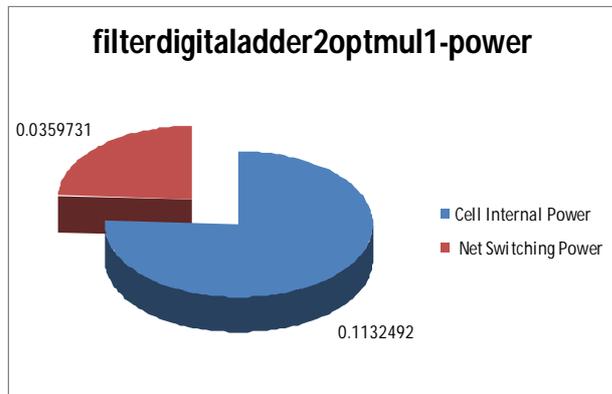
Figure 1.5: Filter Digital Adder 1: Fig (a) Specific area use by filter. Fig (b) Power use by filter.

1.6 Filter Digital Adder 2 optimal 1

In this experiment shift/add structure and pipelining concept both use simultaneously for better result. High-performance digital filters are all important to the execution of digital signal processing systems. The speed of a filter realization counts not alone on the potentialities of the hardware platform employed, but as well on the computational structure of the code. In pipeline processing, any operation on a long critical path is broken into levels of smaller, quicker operations, with registers between levels, so as to get a smaller critical path delay. The result is a higher operating frequency and a higher throughput. In a feedback system, In this structure addition of incoming serial data bit is added. This is final result Transpose form FIR filter which is first use the shift add architecture to reduce the multiplication and then apply the pipelining concept on it . Here figure 'a' shows the area which is consumed by filter called combinational and non-combinational area. Here combinational area is greater than the non-combinational area. Figure 'b' shows the internal power required for the execution.



(a)



(b)

Figure 1.6: Filter Digital Adder 2 optimal 1: Fig (a) Specific area use by filter. Fig (b) Power use by filter.

CONCLUSION

In this paper by seeing result table understand that implementation of digit-serial FIR filters using shift add architecture and digit serial addition concept multiple constant multiplication (MCM) can done easiest way with saving the area and power. Some conclusions regarding design guidelines for low power digit-serial multiplier blocks can be deduced. The actual complexity in terms of adder cost and number of shifts is not the main factor determining the power consumption. Instead the adder depth, as for parallel arithmetic, is a main contributor. Hence, an algorithm with low adder depth should be used. Furthermore, the shifts prevent glitch propagation through subsequent adders. For even coefficients the shifts can be placed either before or after the final additions. Hence, a heuristic for placing the shifts would be also useful. After the shift concept by using the digit serial addition it improve the performance of the multiplier and accumulator (MAC) section which is use in Digital signal processor for doing addition and multiplication.

ACKNOWLEDGEMENT

The author sincerely thanks to Mr. Shrikant. J. Honade, Assistant Professor, G.H.R.C.E.M., Amravati, Prof. N. N. Mandaogade, Head of E & TC Department G.H.R.C.E.M Amravati and Prof. Ashish. S. Bhople, Assistant Professor of E & TC Department, I.B.S.S. C. O. E Amravati for their valuable suggestion, criticism and time to time encouragement.

The author also expresses his warm appreciation to his Brother, Nilesh Nichat for his help in editing this article.

REFERENCES

1. R. Hartley and P. Corbett. "Digit-Serial Processing Techniques". IEEE TCAS II, 37(6):707-719, 1990.
2. L. Aksoy, E. Costa, P. Flores, and J. Monteiro. Exact and Approximate Algorithms for the Optimization of Area and Delay in Multiple Constant Multiplications. IEEE TCAD, 27(6):1013-1026, 2008.
3. L. Aksoy, E. Gunes, and P. Flores. Search Algorithms for the Multiple Constant Multiplications Problem: Exact and Approximate. Elsevier Journal on Microprocessors and Microsystems, 34:151-162, 2010.
4. L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro. Optimization of Area in Digit-Serial Multiple Constant Multiplications at Gate-Level. In ISCAS, to appear, 2011
5. I.C. Park and H.-J. Kang, "Digital filter synthesis based on minimal signed digit representation," in Proc. DAC, 2001, pp. 468-473.