Low Power and Low Area Digital FIR Filter Using Different Multipliers and Adders

T Ramesh Reddy, *M. Tech Student*, and Dr. K. Soundara Rajan, *Professor*
Department of Electronics & Communication Engineering
Jawaharlal Nehru Technological University
Anantapur, India

Abstract-This paper implements low power digital Finite Impulse Response (FIR) filter relying on radix four booth multiplier, serial multiplier and serial adder and shift add multiplier. In this paper we consider multiplier and accumulate FIR filter architecture and folded transform of linear phase FIR filter. These low power multipliers and low power adders are used to reduce dynamic power consumption of digital FIR filter.

I. INTRODUCTION

Digital signal processing (DSP) is used in wide range of applications such as telephone, radio, video etc. Most of DSP computations involve the use of multiply accumulate operations and therefore the design of fast and efficient multiplier imperative. More ever, the demand for portable applications of DSP architectures has dictated the need for low power designs [1]. Digital Finite Impulse Response filter has a lot of arithmetic operations. In general, arithmetic operation modules such as adder and multiplier modules, consume much power, energy, and circuit area. Input bit width of the modules is quite important design parameter for low power. The power digital FIR filter circuit is reduced by optimization of taps and bit width of input signal and filters coefficients [2]. The dynamic switching power consumption of digital FIR filter is reduced by using data transition power diminution technique. This technique is used on adders, booth multipliers and applied for filters to eliminate power consumption due to unwanted data transitions [3]. In [4] they presented a multipliers technique, based on add and shift method and common sub expression elimination for low area, low power and high speed implementation of FIR filters. Finite impulse response filters are widely used in various DSP applications. Block processing can be applied to digital FIR filters to either increase effective throughput or reduce the power consumption of filter [5]. In [6] they proposed pipelined variable precision gating scheme to reduce power consumption of digital FIR filter. This technique uses clock gating to registers in both data flow direction and vertical to data flow direction. The rest of paper is structured as follows. Section2 gives summary of FIR filter theory, and section3 presents the architectures used in our implementation. Section4 gives comparison of implementing architectures. Section5 provides conclusion of the paper.

II. FINITE IMPULSE RESPONSE FILTER THEORY

Digital filters are very important part of digital signal processing. Filters have two uses, one is signal separation and other is signal restoration. Signal separation is needed when the signal has been contaminated with noise or other signals. Signal restoration is used when the signal has been distorted in some way. The most common digital filter is linear time invariant filter. In general filtering is described by simple convolution operation as $y = x * f$ where x is input signal, y is convolved output and f is filter impulse response.

$$y[n] = x[n] * f[n] = \sum_{k=0} f(k)x(n-k)$$

$$= \sum_{k=0} x(k)f(n-k) \qquad (1)$$

Digital filters are two types: Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. The filters designed by using finite number of samples of impulse response are called Finite Impulse Response filters. The filters designed by considering all infinite samples of impulse response are called Infinite Impulse Response Filters. The digital filters are commonly linear time invariant filters.

The straight forward way of implementing LTI Finite Impulse Response filter is finite convolution of input series *x[n]* with impulse response coefficients is given by

$$y[n] = x[n] * f[n] \qquad (2)$$

$$y[n] = \sum_{k=0}^{L-1} f[k]x[n-k] \qquad (3)$$

Where L is the length of FIR filter, $h[n]$ is filters impulse response coefficients, $x[n]$ is input sequence and $y[n]$ is output of FIR filter. The above equations can also expressed in Z domain as

$$Y[Z] = X[Z]H[Z] \qquad (4)$$

Where $H[Z]$ is transfer function of FIR filter in Z domain and is given by

$$H[Z] = \sum_{k=0}^{L-1} h[k]\, Z^{-k} \qquad (5)$$

III. FIR FILT ER IMPLEMENTATION

In this paper multiplier and accumulator (MAC) architecture and linear phase folded architecture are considered

3.1 MAC Finite Impulse Response filter based on radix4 booth multiplier

The linear time invariant FIR filter of order L-1 is shown in fig.1. It employs a collection of multipliers, adders and tapped delay lines. The weight corresponds to multipliers are filter coefficients and are referred ass filter coefficients. Because of tapped delay line structure, the FIR filter is also called as transversal filter.

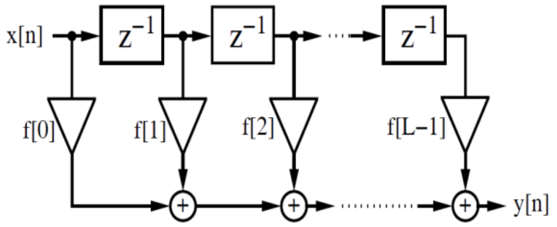

Fig1: Transversal FIR filter

A multiplier has two stages. In the first stage, partial products are generated by the encoder and partial product generator (PPG) and are summed by the compressors. In the second stage, the two final products are added to form the final product through final adder. The block diagram of multiplier is shown in fig.2.

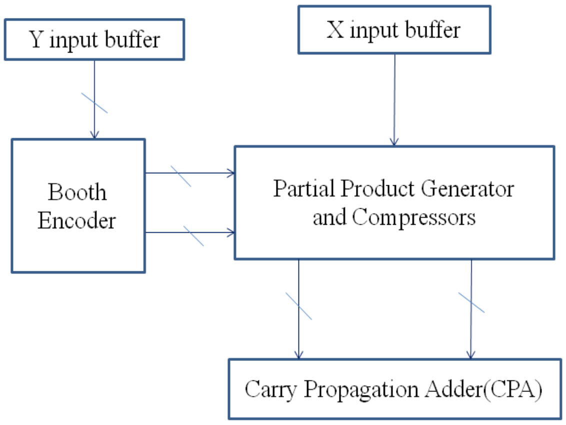
Y input buffer
X input buffer
Booth Encoder
Partial Product Generator and Compressors
Carry Propagation Adder(CPA)

Fig2: Block diagram of multiplier

The multiplier consists of booth encoder block, compression block, and adder blocks. x and y are input buffers. The multiplier y is applied to the booth encoder and y is recoded by the booth encoder. x is the multiplicand. PPG and compressors are the major part of multiplier. Carry propagation adder adds sum and carry vectors which are coming from compressor block. The steps involved in radix4 booth multiplication algorithm are described below.

1. Pad least significant bit of multiplier with zero.
2. Pad most significant bit of multiplier with two zeros (if multiplier has even number of bits) or with one zero (if multiplier has odd number of bits).
3. Divide multiplier in to overlapping groups of three bits.
4. Determine partial product scale factor from the booth encoding truth table.
5. Compute partial products.
6. Sum partial products.

Table1. Booth encoding truth table

y_{2i+1}	y_{2i}	y_{2i-1}	Booth op	Direction	Shift	Addition.
0	0	0	0x	0	0	0
0	0	1	1x	0	-	1
0	1	0	1x	0	-	1
0	1	1	2x	0	1	0
1	0	0	-2x	1	1	0
1	0	1	-1x	1	-	1
1	1	0	-1x	1	-	1
1	1	1	-0x	1	0	0

From the above table, booth multiplier has three operations indicated as direction, shift and addition. Direction indicates whether the multiplicand is positive or negative. If direction is zero then the multiplicand will be positive otherwise multiplicand will be negative. Shift indicates whether the multiplication operation involves shifting operation or not. Addition indicates whether the multiplicand was added to partial product or not. The expressions for these three operations are obtained from the above truth table and are listed below.

Direction, $D_{2i} = y_{2i+1}$ (5)

Shift, $S_{2i} = y_{2i+1} \oplus y_{2i}$ (6)

Addition, $A_{2i} = y_{2i-1} \oplus y_{2i}$ (7)

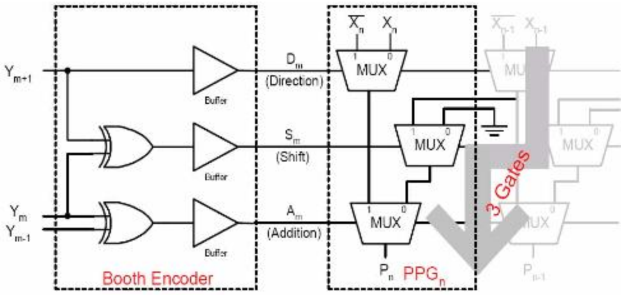

Fig3: Booth encoder and PP where m=2i.

The booth encoder was constructed by using two xor gates. Partial product generator was implemented using three multiplexers and an inverter. Careful optimization partial product generation can lead to hardware reduction. In the general multiplication 8*8, eight partial products and seven adders are required. But in the case of booth multiplier the number of partial products required for multiplication is reduced. Hence the number of adders and the power consumption are also reduced.

3.2 Linear phase folding architecture FIR filter based on radix4 booth multiplier

The symmetrical architecture can be used to implement linear phase FIR filter. Because of the symmetrical architecture the multiplier operations are reduced.

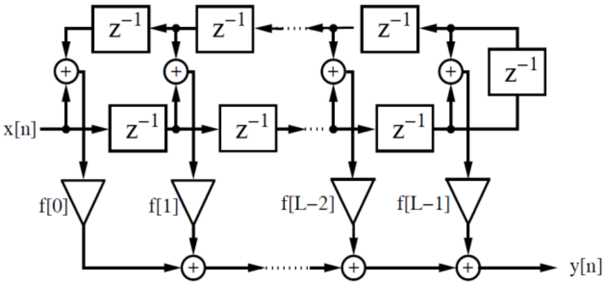

Fig4: linear phase FIR filter

Comparing fig1 and fig4, the number of multipliers reduced half because of the symmetrical architecture. But number of adders remains same and the basic model of this architecture is shown in fig4. For a linear phase FIR filter of order N-1, $h(n) = \pm h(N-1-n)$ where $h(n)$ is filter impulse response coefficients. The folded architecture provides a trade off between hardware speed the area complexity. The folding transformation can be used to implement time multiplexed architectures using less area. Because folding transformation the power consumption also reduced. The fig.5 shows the linear phase folding architecture.

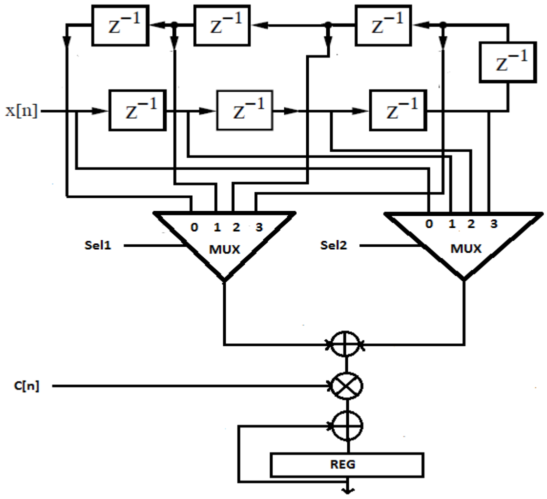

Fig5: folding architecture of linear phase FIR filter.

3.3 MAC Finite Impulse Response filter relying on serial multiplier and serial adder

Digit-serial implementation styles are best suited for implementation of digital signal processing systems. Digit-serial multipliers are obtained using traditional unfolding techniques cannot be pipelined beyond a certain level because of the presence feedback loops. To implement multiplier and accumulate FIR, we use low power serial multipliers and serial adders. Consider a serial multiplier shown in fig.6 where the coefficient word length is four bits.

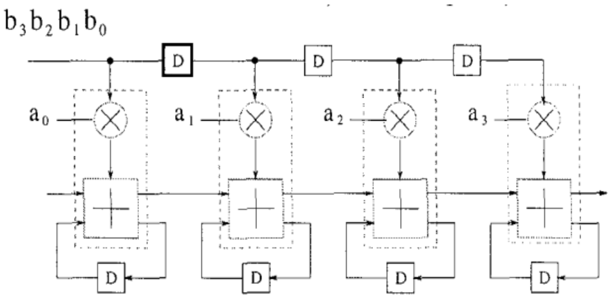

Fig6: low power serial multiplier

This architecture consists of four full adders, four multipliers and some delay elements. In this multiplier, the carry out signal of every adder is fed back after a delay to the carry in signal of the same adder. Therefore the critical path of this architecture of contains W full adder delays. Where W is the word length. The traditional approach for designing the structure involves unfolding the structure by a factor N. where N is equal to digit size. Which can be further reduced to N full adder delay after pipelining. That reduction is not possible here because of presence of feedback loops.

3.4 Finite Impulse Response Filter relaying on shift add multiplier

The shift and add method is used to implement FIR filter. In this we perform all our optimization in multiplier. Here the multiplications are divided into additions and shifts. Hence the complexity of multiplication is reduced.

The coefficients in most of digital signal processing applications for the multiply accumulate operation are constants. In this method, first we arrange decimal coefficients according to positive and negative powers of two. Because of this, the hardware complexity of finite impulse response filter and the power consumption will be reduced. The graph form of this method is shown in fig.7. Another form is, coefficients changed to integer by multiplying the coefficients with multiple powers of 10, then arranging the coefficients according positive and negative powers of two.

Design:

C=3.75

$C=2^{1}+2^{-1}+2^{-2}+2^{0}$

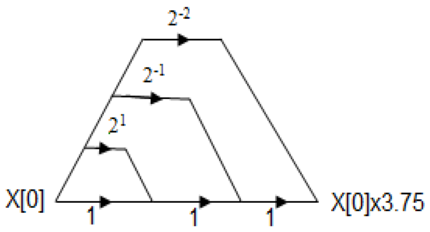

Fig7: Graph form of shift add method

IV RESULTS AND COMPARISIONS

Designs equipped to 8 bit adders, 8 bit multipliers and are accomplished via VHDL hardware description language using Xilinx ISE software and synthesized using Xilinx tools. Power is analyzed using Xilinx Xpower analyzer. Tables I and II shows the power consumption comparison of digital FIR filter which uses booth multiplier, shift add multiplier and serial multiplier and serial adder.

Table I: power consumption of 8 taps,16 bit coefficients, 8 bit input

Freq	Booth without DPDT(mw)	Booth with DPDT using reg(mw)	Booth with DPDT using and gate(mw)
25	469	326	868
50	879	596	669
75	1297	881	1018
100	1703	1137	1283

Table II: Power consumption of proposed filters

Proposed filter	25 MHZ	50 MHZ	75 MHZ	100 MHZ
MAC_booth(mw)	100	140	180	210
Linear phase folding booth(mw)	110	150	190	230
Shift add(mw)	90	120	140	110
Serial multiplier ad serial adder	112	126	141	155

V. CONCLUSION

In this paper a low power and low area digital FIR filter is presented. For reduce power consumption and area we using radix4 booth multiplier, low power serial multiplier and serial adder, shift add multiplier and folding transformation. These filters are compared for power. The proposed FIR filters have been synthesized and implemented using Xilinx tools and power analyzed using Xilinx Xpower analyzer.

ACKNOWLEDGMENT

T Ramesh Reddy would like to thank Prof. K. Soundara Rajan, who had been guiding through out to complete the work successfully, and would also like to thank the HOD, ECE Department and other Professors for extending their help & support in giving technical ideas about the paper and motivating to complete the work effectively & successfully.

REFERENCES

[1] Yun-nanChang, Janardhan H. SatyanarayanaKeshab K.Parhi"LOW POWER DIGIT-SERIAL MULTIPLIERS",1997 IEEE International symposium on circuits and systems,june i3-12,1997

[2] Kousuke TARUMI, Akihiko HYODO, Masanori MUROYAMA, Hiroto YASUURA, "A design method for a low power digital FIR Filter in digital wireless communication systems," 2004.

[3] A. Senthilkumar, A.M. Natarajan, S.Subha "Design and Implementation of Low Power Digital FIR Filters relying on Data Transition Power Diminution Technique" DSP Journal,Volume 8, pp. 21-29, 2008.

[4] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner, "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", IEEE, 2006.

[5] Jin-Gyun Chung, Keshab K. Parhi "Frequency Spectrum Based Low-Area Low-Power Parallel FIR Filter Design" EURASIP Journal on Applied Signal Processing 2002, vol. 31, pp. 944–953.

[6] A. Senthilkumar, 2A.M. Natarajan, "FPGA Implementation of Power Aware FIR Filter Using Reduced Transition Pipelined Variable Precision Gating," Journal of Computer Science , pp. 87-94, 2008.

[7] Shibi Thankachan, "64 x 64 Bit Multiplier Using Pass Logic",2006.

[8] Uwe Meyer-Baese, "Digital Signal with Field Programmable gate Arrays", Springer-Verlag Berlin Heidelberg 2007.

[9] Ronak Bajaj, Saransh Chhabra, Sreehari Veeramachaneni, M B Srinivas, "A Novel, Low-Power Array Multiplier Architecture",2002.

[10] AHMED F. SHALASH, KESHAB K. PARHI "Power Efficient Folding of Pipelined LMS Adaptive Filters with Applications" Journal of VLSI Signal Processing, pp. 199–213, 2000.