Design of Novel FIR Filter Using Add and Shift Multiplier and Carry Save Adder

A. Arun[1], R.Yasodha[2].
PG Scholar[1], Assistant Professor[2]
Shri Andal Alagar College of Engineering, TN, India

Abstract- This project investigates the implementation of a low power FIR filter using Add and Shift Multiplier and Carry Save Adder. This method is used to reduce the dynamic power consumption, Delay and Area of a low power FIR filter. This method include Modified Booth Encoding Algorithm combined with Spurious Power Suppression Technique, Low Power Digital Serial Multiplier along with carry look ahead adder, shift and add multiplier. This proposed FIR filter was synthesized and implemented using Xilinx ISE V7.1 and also the power is analyzed using Xilinx XPoweranalyzer.

I. INTRODUCTION

The impulse response of the filter can be either finite or infinite. The methods for designing and implementing of these two filter classes differ considerably. Finite impulse response (FIR) filters are digital filters whose response to a unit impulse (unit sample function) is finite in duration. This is in contrast to infinite impulse response (IIR) filters whose response to a unit impulse (unit sample function) is infinite in duration. FIR and IIR filters each have advantages and disadvantages. FIR filters can be implemented using either recursive or non-recursive techniques, but usually non-recursive techniques are used. FIR filters are widely used in various DSP applications. In some applications, the FIR filter circuit must be able to operate at high sample rates, while in other applications the FIR filter circuit must be a low-power circuit operating at moderate sample rates.

The low-power or low-area techniques developed specifically for digital filters. Parallel (or block) processing can be applied to digital FIR filters to either increase the effective throughput or reduce the power consumption of the original filter. Traditionally, the application of parallel processing to an FIR filter involves the replication of the hardware units that exist in the original filter. The topology of the multiplier circuit also affects the resultant power consumption. They extensively use a modified common sub expression elimination algorithm to reduce the number of adders. They have proposed a novel approach for a design method of a low power digital baseband processing. Their approach is to optimize the bit-width of each filter coefficient. They define the problem to find optimized bit-width of each filter coefficient. This project presents the method to reduce dynamic switching power of a fir filter using data transition power diminution technique (DPDT). This technique is used on adders, booth multipliers. In this research proposes a pipelined variable precision gating scheme to improve the power awareness of the system.

ICGPC 2014
St.Peter's University, TN, India.

This research illustrates this technique to clock gating to registers in both data flow direction and vertical to data flow direction within the individual pipeline stage based on the input data precision. Finally this project provides the conclusion of the project.

II. METHODOLOGY

A digital filter gives a digital output and consists of digital components. In a digital filtering application, software running on a DSP applications and reads the input samples from an A/D converter. It performs the mathematical manipulations dictated by the required filter type and output the result as D/A converter. An analog filter operates directly on the analog inputs and is built entirely with analog components such as resistors, capacitors, and inductors. There are many types of filter used in DSP applications, but the most commonly used filters are lowpass, highpass, bandpass, and bandstop. A low pass filter allows only low frequency signals (below some specified cutoff) through its output and it can be used to eliminate high frequencies. A low pass filter is small and for limiting the upper most range of frequencies in an audio signal.

A high pass filter does the opposite of low pass filter by rejecting the frequency components below some threshold. An example high pass application is cutting out the audible 60Hz AC power "hum", which can be picked up as noise accompanying almost any signal in the U.S. The designer of a cell phone or any other sort of wireless transmitter would typically place an analog bandpass filter in its output RF stage to ensure only output signals. Engineers mostly use bandstop filters, because this filter passes both low and high frequencies to block a predefined range of frequencies in the middle. The Frequency response Simple filters are usually defined by their responses to the individual frequency components that constitute the input signal.

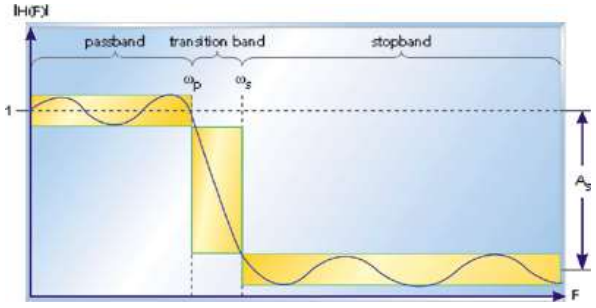

Fig 1: Response of a Low Pass Filter to Various Input Frequencies

The above figure shows the frequency response of low pass digital FIR filter. In this frequency response, $\omega_s$denotes the beginning of the stopband frequency and ω_p denotes the ending of the passband frequencies and A_s denotes the amount of attenuation in the stopband frequency. The stopband frequency ω_s and the passband frequency ω_p will fall within the transition and are attenuated to some degree. Digital filters are mostly used to modify the attributes the signal in the time domain or frequency domain and most commonly used in the Linear Time Invariant. A Linear Time Invariant interacts with the signal and the process is called Linear Convolution.

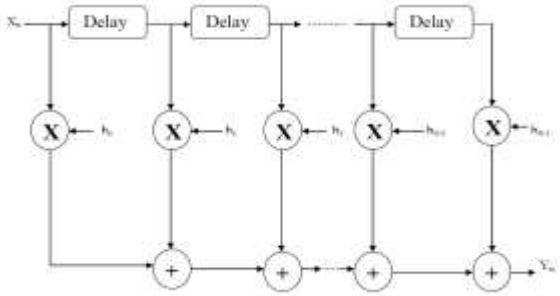

Fig 2: Direct Form FIR Filter

The above figure shows the structure of a direct form FIR filter with length N. The input of the FIR filter X_n is given to the delays of the direct form structure. In this structure the $h_k$values are the coefficients and are designed to the multiplication of the samples in the Filter. The output at time n is the summation of the delayed samples multiplied by the appropriate samples. The process of selecting the filter coefficients and the length of the filter is called Filter Design.

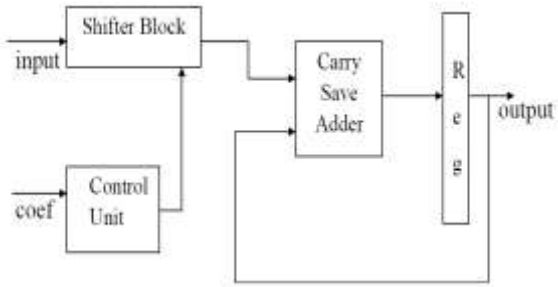

Fig. 3: Block Diagram of low Power Multiplexer base shift and Add Multiplier

i) Shifter Block:

The block diagram of low power multiplexer based shift and add multiplier is shown in the above figure. The input of the shifter block is a 8-bit vector in the low power multiplexer. The output is the shifted version of the input with the amount of shift defined from 0 to 7. The shifter block is a digital circuit that shifts a data word in sequence. The circuit consists of three individual barrel shifters. The first barrel shifter has only one "0" connected to one of the multiplexers (bottom left corner), while the second has two and the third has four. For larger vectors, just keep doubling the number of '0" inputs. If shift "001", for example, then only the first barrel shifter should cause a shift; on the other hand, if shift "111", then all barrels should cause a shift.

The final shifter unit will perform the shifting operation after all the intermediate additions are done. The diagram of barrel shifter is shown below.

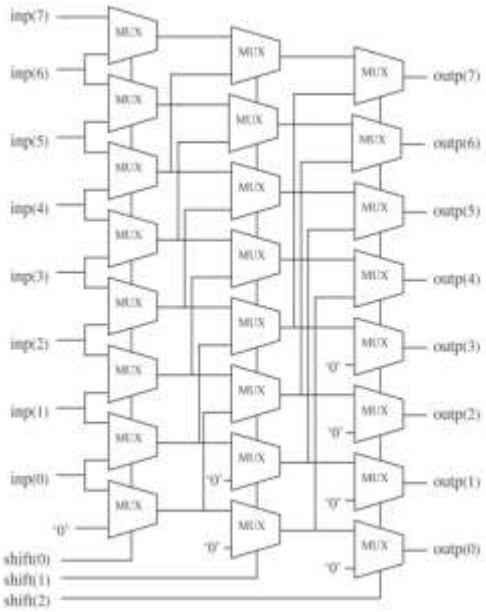

Fig 4: Shifter Block (Barrel Shifter)

ii) Control Unit:

The control unit performs the flow of data through the processor and coordinates the activities of the other unit in the FIR filter. The control unit receives the coefficients of external instructions or commands which converts into a sequence of control signal to implement a sequence of register transfer level operations. The examples of devices that require the control unit are central processing units (CPUs) and graphics processing units (GPUs).

iii) Carry Save Adder:

The main function of the adder block is to add two bytes in the system. This adder block is designed structurally which accepts two inputs and one carry output and one sum output. In existing system normal adder is used to reduce the area and power of the FIR filter, but the proposed system uses carry save adder.

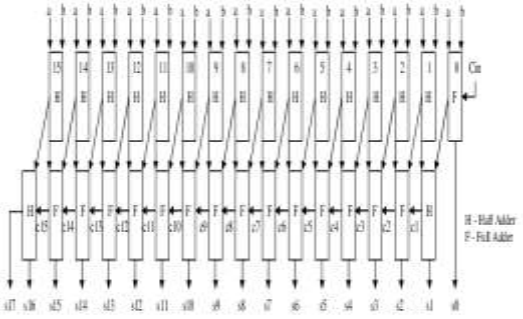

Fig 5: 16-bit Carry Save Adder

The carry save adder is faster than that of the ripple carry adder in the design. The output of the barrel shifter and the feedback of the register is given to the input of the carry save adder in the FIR filter. The diagram of carry save adder is shown below.

iv) Multiplier

Shift-and-add multiplication is similar to the multiplication performed by two numbers. This method adds the multiplicand X to itself Y times, where Y denotes the multiplier. To multiply two numbers, the algorithm is to take the digits of the multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results.

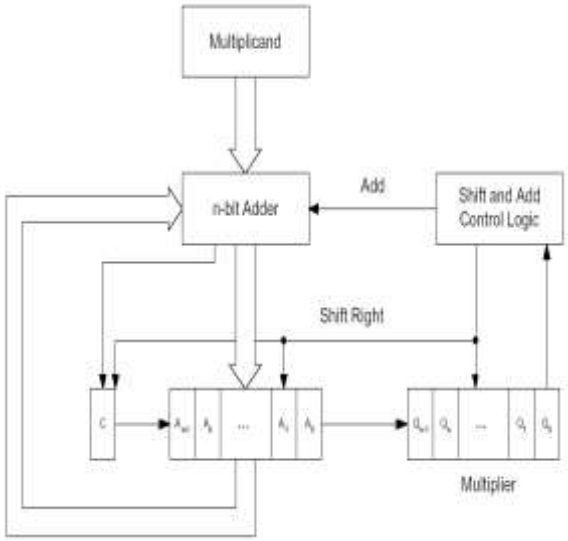

Fig 6: Add and Shift Multiplier

In this case of binary multiplication, since the digits are 0 and 1, each step of the multiplication is simple. If the multiplier digit is 1, a copy of the multiplicand (1×multiplicand) is placed in the proper positions; if the multiplier digit is 0, a number of 0 digits (0×multiplicand) are placed in the proper positions. Build-in multipliers are used to fulfill the requirements of high-speed operations, therefore this number of build-in multipliers is one of the constraints to the number of taps of FIR filter.

III. EXPERIMENTAL RESULTS

In order to evaluate the result of FIR filter, carry save adder is to reduce the power, area and delay by using shift and add multiplier.

Table 1: Power Consumption of Proposed Filter

Proposed Filter	25 MHz	50 MHz	75 MHz	100 MHz
Proposed Method (mw)	38	44	50	56
FIR Base(8 taps,8 bits) (mw)	130	190	250	300

Table 2: Comparison of Slices, FFs, LUTs

Proposed Filter	Slices	FFs	LUTs	Latch
Proposed Method	21	3	24	3
Existing Method	23	3	41	3

IV. CONCLUSION

In this proposed system, the FIR filter using shift and add multiplier and carry save adder is used to reduce the area, power and delay of the system. This filter was compared with the area and power consumption of other common implementation of the filter. The proposed FIR filter was synthesized and implemented using Xilinx and the power is analyzed using Xpoweranalyzer.

REFERENCES

[1] AbdSamadBenkrid and Khaled Benkrid, "Novel Area-Efficient FPGA Architectures for FIR Filtering With Symmetric Signal Extension" 2009.

[2] Ahmed F. Shalash, Keshab K. Parhi "Power Efficient Folding of Pipelined LMS Adaptive Filters with Applications" Journal of VLSI Signal Processing, 2000.

[3] AnupHosangadi and Ryan Kastner "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method" 2006.

[4] A.RenukaNarasimha, K.Rajasekhar,A.Sujana Rani, " Implementation of Low Area and Power Efficient Architectures for Digital FIR Filters" August 2012.

[5] BahmanRashidi, and Majid Pourormazd, "Low Power FPGA Implementation of Digital FIR Filter Based on Low Power Multiplexer" April 2013.

[6] C. N.Marimuthu, P. Thangaraj, "Low Power High Performance Multiplier ICGST-PDCS", December 2008.

[7] I. C. Park and H. J. Kang, "FIR filter synthesis algorithms for minimizing the delay and the number of adders," pp. 770–777, Aug. 2001.

[8] Jin-Gyun Chung, Keshab K. Parhi "Frequency Spectrum Based Low-Area Low-Power Parallel FIR Filter Design" EURASIP Journal on Applied Signal Processing 2002.

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

[10] M. SreeDivya and G. Kiran Kumar "Implementation of Low Power Digital FIR Filter Design Based on Low Power Multipliers and Adders", 2012.

[11] Ronak Bajaj, SaranshChhabra, SreehariVeeramachaneni, M B Srinivas, "A Novel, Low-Power Array Multiplier Architecture", 2002.

[12] Senthilkumar and A. M. Natarajan, "FPGA implementation of power aware FIR filter using reduced transition pipelined variable precision gating," Journal of Computer Science, pp. 87-94, 2008.

[13] ShahnamMirzaei, AnupHosangadi, Ryan Kastner, "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", IEEE, 2006.

[14] T Ramesh Reddy, Dr. K. SoundaraRajan, "Low-area and power Digital FIR Filter Implementations", 2012.

[15] W. Brian and M. Barr, "Introduction to digital filters," Embedded System Programming, December 2002.