Minimum Energy Tracking for Digital Arithmetic and Logical Units

Rakhi.P.G[1] Karthikeyini.C[2]
[1]P.G. Student [2]Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract— In modern world there is a need for low power digital end design and area efficient low power, high performance in Digital signal Processing system. Power minimization is one of the important factors in today VLSI design methodologies because of two main reasons one is the operating life of the battery in portable devices and mobiles. The second reason is due to the increasing number of transistors on a single chip results in high power dissipation and it can lead to reliability. Recent advances in mobile computing and multimedia applications demand high-performance and low-power VLSI digital signal processing (DSP) systems. One of the most widely used operations in DSP is finite-impulse response (FIR) filtering. In the integrated circuit design the packaging problems is a major element to minimize the energy consumption in system. Hence it is important to track the system continuously with respect to time, the minimum operating voltage of digital circuits which in turn depends upon the systems working logics with conventional and logical design. In this regard, as a preliminary work 32 Tap FIR filter that includes adder, multiplier and a D flip flop as a delay element was designed and tested with the parameters such as delay, power and number of flip flops. The FIR filters were implemented using 8 bit input word length and 8-bit coded coefficients. This design presents the conventional 32 Tap Finite Impulse Response filter and it is synthesized in Xilinx design tool.

Key words: Finite Impulse Response, Filter, Multiplier

I. INTRODUCTION

Digital filtering techniques is used to suppress noise, enhance signal in selected frequency ranges, constrain bandwidth, remove or attenuate specific frequencies and other special operations. Digital filters are classified into finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR digital filters can have exactly linear phase response and a very regular architecture, and suffer less from the effects of finite word length as compared with IIR digital filters. Area, delay (performance) and power are the three important design constraints for designing an embedded real-time digital signal processing systems. The area constraint is imposed primarily by considerations of cost.

Area efficient implementation results in a smaller die size and hence becomes more cost effective. It also enables integrating more functionality on a single chip. The performance requirements of a system are driven by its data processing needs. For DSP systems, throughput is the primary performance criterion. The performance constraint is thus dependent on the rate at which the input signals are sampled and on the complexity of processing to be performed. Low power dissipation is a key requirement for portable, battery operated systems as it extends battery life.

For the requirement of high-speed and low-power applications, the development and implementation of high-speed FIR digital filters need both increased parallelism and reduced complexity in order to meet both sampling rate and power dissipation goals. In this thesis, FIR filter is designed based on RNS to achieve high speed operation. Bottom-up design flow is used for maximum circuit performance, minimum design size, and minimum high-volume production cost.

II. FIR FILTER

Digital filters are used in a wide variety of signal processing applications, such as spectrum analysis, digital image processing, and pattern recognition. Digital filters eliminate a number of problems associated with their classical analog counterparts, and thus are often used in place of analog filters. The most common digital filters belong to the class of discrete-time LTI (linear time invariant) systems, which are characterized by the properties of causality, reusability, and stability [1]. They can be characterized in the time domain by the unit-impulse response and in the z-transform domain by the transfer function. A unit-impulse response sequence of a causal LTI system can be either finite or infinite in duration. This property determines their classification as either a finite impulse response (FIR) or an infinite impulse response (IIR) system. To illustrate this, consider the most general case of a discrete time LTI system with the input sequence denoted by $x(kT)$and the resulting output sequence $y(kT)$ given by:

$$y(kT) = \sum_{m=0}^{M-1} b_m x((k-m)T) - \sum_{n=1}^{N-1} a_n y((k-n)T) \qquad (1)$$

The corresponding transfer function in the Z-domain is given by:

$$\hat{H}(z) = \frac{\hat{Y}(z)}{\hat{X}(z)} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{1 + \sum_{n=1}^{N-1} a_n z^{-n}} \qquad (2)$$

If at least one denominator coefficient $a_n$is nonzero, then system is recursive (its current output depends on previous output values), and as a result its impulse response is of infinite duration (IIR system). If all denominator coefficients are zero (polynomial of order 0), the corresponding system is non-recursive (FIR system), and its impulse response is of finite duration. The transfer function of Eq. (2) in this case becomes a polynomial of finite order M-1:

$$\hat{H}(z) = \frac{\hat{Y}(z)}{\hat{X}(z)} = \sum_{m=0}^{M-1} b_m z^{-m} \qquad (3)$$

The corresponding FIR difference equation in time domain is:

$$y(kT) = \sum_{m=0}^{M-1} b_m x((k-m)T) \qquad (4)$$

As with analog filter design, the general shape of the frequency response is often the criteria in discrete filter

design. Recall the frequency response for continuous-time systems was obtained by evaluating the transfer function on the $j\omega$ axis, similarly for the discrete case the transfer function in z is evaluated over the unit circle. In this case substitute $z = \exp(j\omega/\omega_s)$ where ω_s is the sampling frequency in radians per second. Therefore, the frequency response of an FIR filter is given by

$$\hat{H}(z)\Big|_{z=\exp\left(j\frac{\omega}{\omega_s}\right)} = \sum_{m=0}^{M-1} b_m \exp\left(-j\frac{m\omega}{\omega_s}\right) \quad (5)$$

Note that even though the time domain is discrete, the frequency response is continuous (defined for all ω); however, it is periodic with period ω_s due to the periodic behavior of the complex exponential and consistent with the concept of aliasing.

The design of digital filters involves determining the filter order (M) and computing the values of the coefficients (b_i's in the above equations) to achieve the desired filter response. The desired response can be specified in the frequency domain in terms of the magnitude response and/or the phase response. It can also be specified in terms the impulse response. Once filter coefficients are computed, the filter performance must be analyzed to verify the filter meets the specifications.

III. DESIGN

A. *Conventional FIR filter*

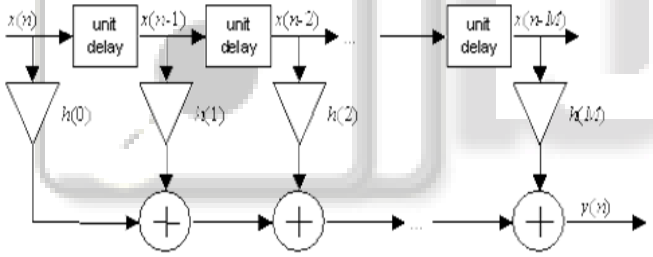

Fig. 1: N Stage FIR filter design

The signal flow diagram shown in figure 1 illustrates the algorithm and suggests an architecture using the elements created above. The word delays are inserted for the 'Z' delay blocks. The multiplications shown in the flow graph correspond to the serial by parallel multipliers with their Parallel inputs programmed with the value of the associated coefficient [3]. A separate word delay and multiplier are used for each tap in the filter.

B. *Standard 3-Tap FIR Filter*

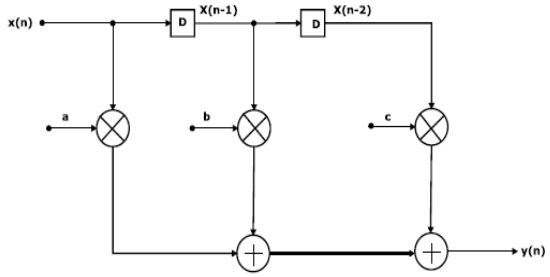

Fig. 2: 3 Stage standard FIR filter design

In standard 3 tap FIR filter the delay element D flip flop is added in the initial data path before the multiplication operation.

C. *Transposed FIR Filter*

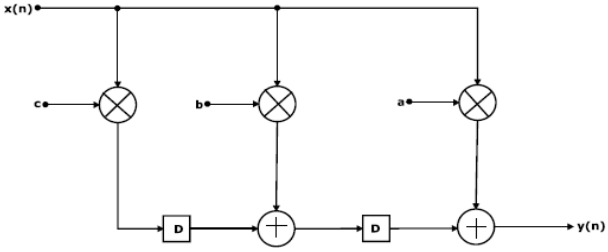

Fig. 3: 3 Stage Transposed FIR filter design

In standard 3 tap FIR filter the delay element D flip flop is added in the initial data path before the multiplication operation.

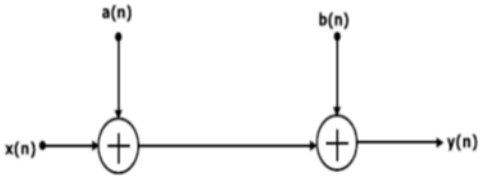

Fig. 4: Data path

The speed of the filter is defined as the rate at which input samples can be processed. To increase the speed it is necessary to reduce the critical path between input and output. The critical path is defined to be the path with the longest computation time among all paths that contain zero delays. For 3-tap FIR filter, the critical path delay is (Tm+2Ta).

Pipelining reduces the effective critical path by introducing pipelining latches along the data path. The critical path is now reduced from T_m+2T_a to $T_m + T_a$ shown in fig (2 and 3). In this arrangement while the left adder initiates the computation of the current iteration the right adder is completing the computation of the previous iteration result.

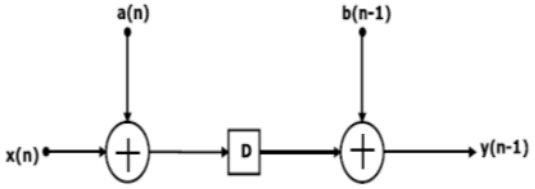

Fig. 5: 2-level pipelined structure

D. *2's Complement*

To calculate the 2's complement of an integer, invert the binary equivalent of the number by changing all of the ones to zeroes and all of the zeroes to ones (also called 1's complement), and then add one to the obtained result. It results in 2's complement. The 1's and 2's Complement of the following examples are described. The reason we tend to use 2's complement numbers is that traditional binary addition works exactly the same as long as the result is in the range of representation. We just lop off any extra bits. For the following problems add the values as if they were unsigned numbers, only keeping the first four digits.

1) Example binary input: 10000000

1's complement of 10000000 is represented by 11111111 – 10000000 = 01111111 (Or)

1's complement of 10000000 = 01111111

2's complement of 10000000 = 01111111+1, The result shows digit as 10000000.

Another FIR filter known as transpose or data-

broadcast structured shown in fig 4. The critical path of the filter of fig 5 can be reduced without introducing any pipelining latches by transposing structure. Now the propagation delay is (Tm + Ta). Figure 6 shows the 32 Tap FIR filter [4] design with tree structure, with combined adder design with stage by stage design. The conventional design is based on simple and energy efficient design. Hence for implementing the filter in arithmetic and logical design it is very useful to synchronize the input X_{in} to output Y(n).

Design consists of Delay Flip flop design for single delay element and stage by stage the signal is processed by adding the multiplied coefficients and input data. The coefficients which are in negative then it will be converted to 2's complement and result will be added. Each stage is delayed and processed. When designing the digital integrated circuits with a (hardware description language) VHDL or verilog, the designs are converted at a higher level of abstraction than logic families or logic gate level model. In HDLs the designer declares the registers. The code will be converted and describes the internal combination logic by using constructs that are familiar from programming languages such as if-then-else and arithmetic operations. This level is known as register-transfer level.

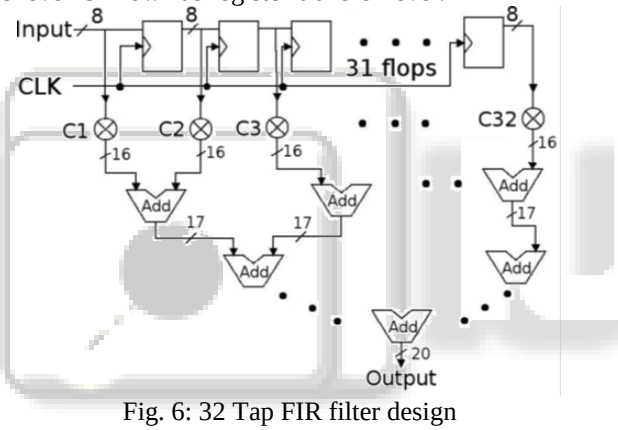

Fig. 6: 32 Tap FIR filter design

IV. XILINX ISE

Xilinx ISE (Integrated Software Environment) is a software device which produced by synthesis and analysis of HDL designs, it can perform timing analysis which can be perform enabling the developer to synthesize designs, to inspect RTL diagrams, simulate a proposal's reaction to altered stimuli, and configure the target device with the programmer. The Xilinx ISE is a complete FPGA/CPLD programmable logic design suite providing[8]: - Programmable specification of logic through representation of capture or Verilog/VHDL - To Synthesis and Place & Route of identified logic for various Xilinx FPGAs and CPLDs - To provide the timing simulation and Functional simulation - To Copy of structure data into goal device through communications cable The Xilinx ISE Design Suite offers an integrated flow with the ISE Simulator (ISim) which can allows simulations to launch directly from the Project Navigator (ISE). All simulation commands that prepare the ISim simulation are generated by ISE Project. The Xilinx ISE Simulator is a Hardware Description Language (HDL) simulator that allows the computer operator to execute behavioral, functional and timing simulations for VHDL, Verilog and mixed-language designs.

The I Sim Graphical User Interface comprises the toolbars, wave window, status bar and the panels. From the main window we can vision the simulation visible shares of the design add and vision signals in the wave window, employ I Sim instructions to run the simulation program and to observe the design, and debug necessary. Xilinx offers power approximation tools like Xilinx Power Estimator (XPE) and Xilinx Power Analyzer (XPA). For the application of pre-implementation device, in early stages XPE can be used in to design cycle after the RTL explanation of the design is partial. After completion of implement part the XPA device can be used to more accurate approximations and power enquiry. XPE shows tabs for each type of module in device architecture. Depending upon The number of tabs showed in XPE will differ provisional on the device architecture.

V. RESULTS AND DISCUSSION

FIR	Flip flops	Power (w)	Gate Delay (ns)
4 Tap	55	0.034	4.040
8 Tap	96	3.517	2.910
32 Tap	248	3.520	2.910

Table 1: Comparison Table

The comparisons result shows the conventional FIR filter construction based on the power, area and gate delay. The transposed FIR filter is designed by using RTL design flow with the help of VHDL language to verify and synthesis the functionality of digital filter. Since result shows the gate delay of 32 Tap FIR Filter and 8 Tap FIR filters both same, even though the internal flip flop ranges are varied. From comparison power result shows the 0.003 watts difference between 8 Tap filter and 32 Tap filter.

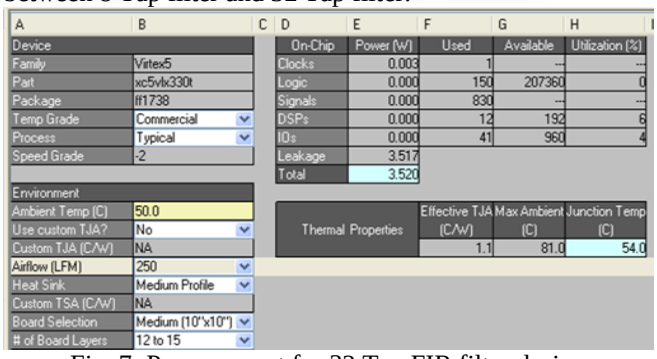

Fig. 7: Power report for 32 Tap FIR filter design

```
Timing constraint: Default OFFSET OUT AFTER for Clock 'Clk'
  Total number of paths / destination ports: 24 / 24
--------------------------------------------------------------------------------
Offset:              2.910ns (Levels of Logic = 1)
  Source:            Madd_A141 (DSP)
  Destination:       Yout<23> (PAD)
  Source Clock:      Clk rising

  Data Path: Madd_A141 to Yout<23>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     DSP48E:CLK->P23       1   0.480    0.286  Madd_A141 (Yout_23_OBUF)
     OBUF:I->O                 2.144           Yout_23_OBUF (Yout<23>)
    ----------------------------------------
    Total                      2.910ns (2.624ns logic, 0.286ns route)
                                       (90.2% logic, 9.8% route)

================================================================================
```

Fig. 8: Gate delay for 32 Tap FIR filter design

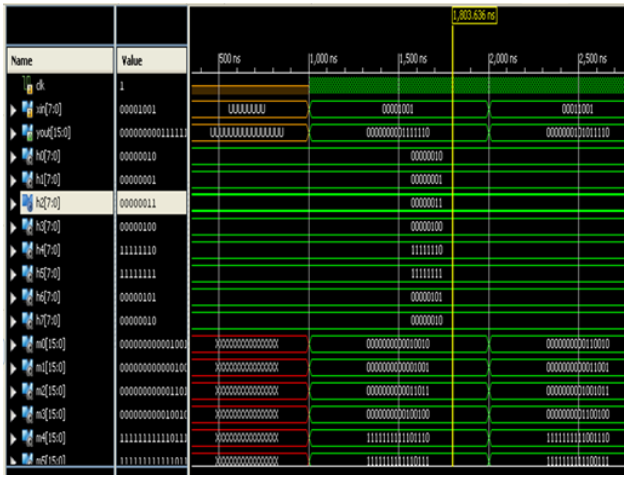

Fig. 9: Output result for 8 Tap FIR filter design

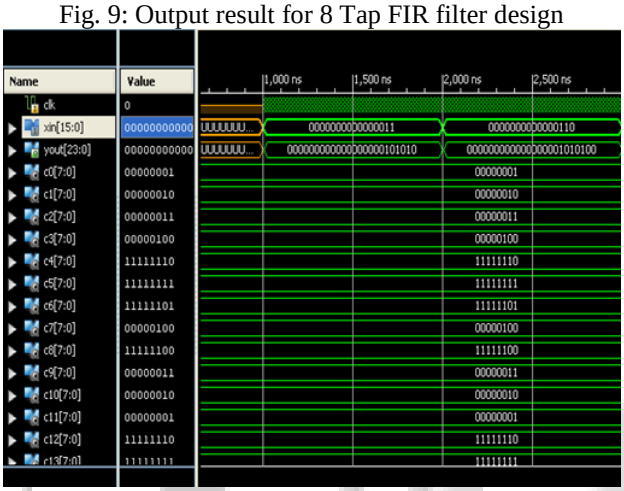

Fig. 10: Output result for 32 Tap FIR filter design

The figure 8 shows the gate delay 2.910ns and figure 9 & figure 10 shows the data processed varies with respect to clock signal and data input signal as well as input coefficients. The multiplier blocks and adder blocks are constructed using the signal coefficients. The signed bit representation of each bit is complemented by using 2's complement method.

Figure 6 shows a 32-tap FIR filter. It consists of 32 8-bit multipliers, 31 adders ranging from 16 to 19 bit, and 31 single unit delay elements, which totals to 21020 gates. The power consumed by the FIR filter was modeled by testing the multiplier under several input combinations. The first input combination was having the multiplier fixed to 0×01 and the multiplicand swinging between 0×19 and 0×00 on every clock cycle. This refers to be swinging input case in Table I and is used to estimate the maximum possible power consumption of FIR structure.

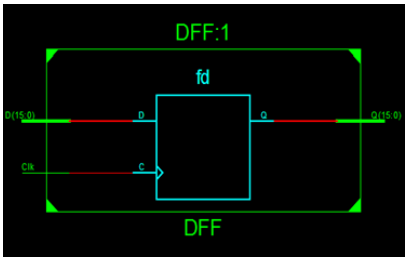

Fig. 11: RTL View –D Flip Flop Design

Digital filters are widely used in Digital communications and audio/video processing. In particular, finite impulse response (FIR) filters are used for their ease of implementation and stability.

VI. Conclusion

A low-power, high-throughput, and low-area implementation of 4, 8 and 32 TAP FIR filter based on transposed structure with a comparison of all stages is discussed here. The delay elements in-between the multiplier and adder are constructed based on D flip flop design. The total FIR design is designed up to 32 tap FIR filter with a delay 2.910ns and power 3.50W designed in Xilinx 14.2 suite. The performance of the processor depends on the data-path which in turn depends on the individual units such as the adder. Thus there is a necessity to make a minimized structure to improve the performance of the adder. It is very well known that the FIR filter consists of Delay elements, Multipliers and Adders. Because of usage of Multipliers in conventional design methodology gives rise to 2 demerits that are: Increase in Area and Increase in the Delay which ultimately results in low performance (Less speed). To reduce the number of adders and multipliers, array multipliers can be used. Thus the performance of FIR filter is improved.

References

[1] Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010,PP(99):1~5

[2] M. Seok et al., "The Phoenix processor: A 30 pW platform for sensor applications," in Proc. IEEE Symp. VLSI Circuits, Jun. 2008, pp. 188–189.

[3] J.G. Proakis and D.G. Manolakis, Digital Signal Processing-Principles,Algorithms and Applications New Delhi: Prentice-Hall, 2000

[4] Sagar Venkatesh Gubbi And Bharadwaj Amrutur, "All digital energy sensing for minimum energy tracking" IEEE transactions on very large scale integration (VLSI) systems, vol. 23, no. 4, April 2015

[5] Use Meyer-Baese.Digitalsignal processing with FPGA [M].Beijing:Tsinghua University Press, 2006: pp-50~51.

[6] Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010,PP(99):1~5.

[7] Chao Cheng and Keshab K Parhi. Low-Cost Parallel FIR Filter Structures With 2-Stage Parallelism[J].IEEE Transactions on Circuits and Systems I: Regular 2007,54(2):280~290.

[8] Tearney G J and Bouma B E. Real-Time FPGA Processing for High-Speed Optical Frequency Domain,Imaging [J]. IEEE Transactions on Medical Imaging, 2009,28(9):1468~1472.

[9] M. Potkonjak, M. B. Srivastava, and A. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," IEEE Trans. Computer- Aided Design Integr. Circuits Syst., vol. 15, no. 2, pp. 151–165, Feb. 1996.

