International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

EFFICIENT SHIFT ADD IMPLEMENTATION OF FIR FILTER USING VARIABLE PARTITION HYBRID FORM STRUCTURE

Arunraj.M[1], Jayaprasanth.P[2], Ragul.G[3], Rahul.R[4]

[1,2,3,4]*Student, Department of Electronics and Communication, Adhiparasakthi Engineering College, Melmaruvathur, Tamilnadu, India*

Abstract - *Critical delay is major drawback in today's electronic designs. Components which are widely used in design, such as shift add and prefix adder in filter, should consume as little power as possible. Prefix adder are an efficient way of describing and generating certain sequences in hardware implementations. Hence it seen that this structure greatly reduce the delay, and would be especially beneficial for a structure with large number of pins. The proposed method significantly reduces dynamic power dissipation, simplifies the design process for single sequence of output generation, and eliminate the need of some hardware. The achievable rate and power reduction to improve the performance in filter architecture by implementing shift add. The Prefix adder achieves substantial reduction on the power consumption by reducing the gate count and dynamic power dissipation.*

Key Words: **Shift add, Prefix adder, Variable partition, Hybrid form structure, Efficient.**

1. INTRODUCTION

Performed a detailed complexity analysis in terms of the hardware and time consumed by the hybrid form structures. To have a more efficient implementation, a variable size partitioning approach is proposed in this project. 2 The guaranteed stability and linear phase response of finite impulse response (FIR) filters have made it a popular candidate for several digital signal processing (DSP) applications. The area, time and power consumption of an FIR filter are largely dominated by the complexity of multiplications. Several attempts have therefore been made to reduce this complexity by multiplier-less FIR filter implementation, where the multiplication operations are realized by optimized shift-and-add based networks.

1.1 Filter

In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying). The impulse response (that is, the output in response to a Kronecker delta input) of an Nth-order discrete-time FIR filter lasts exactly N + 1 samples (from first nonzero element through last nonzero element) before it then settles to zero.

1.2 Properties of filter

An FIR filter is designed by finding the coefficients and filter order that meet certain specifications, which can be in the time domain (e.g. a matched filter) and/or the frequency domain (most common). Matched filters perform a cross-correlation between the input signal and a known pulse shape. The FIR convolution is a cross-correlation between the input signal and a time-reversed copy of the impulse response. Therefore, the matched filter's impulse response is "designed" by sampling the known pulse-shape and using those samples in reverse order as the coefficients of the filter.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cut offs are needed. However, many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

1.3 Advantage of filter

The main challenging areas in VLSI are performance, cost, testing, area, reliability and power. These demands for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation for circuit implementation. Linear feedback shift registers (LFSR's) are an efficient way of describing and generating certain sequences in hardware implementations. A linear feedback shift register is composed of a shift register R which contains a sequence of bits and a feedback function f which is the bit sum (XOR) of a subset of the entries of the shift register. A design must contain these enable conditions in order to use and benefit from clock gating.

This clock gating process can also save significant die area as well as power, since it removes large numbers of mux and replaces them with clock gating logic. This clock gating logic is generally in the form of "integrated clock gating" (ICG) cells. However, the clock gating logic will change the clock

tree structure, since the clock gating logic will sit in the clock tree.

2. MODULE DESCRIPTION

In the window design method, one first designs an ideal IIR filter and then truncates the infinite impulse response by multiplying it with a finite length window function. The result is a finite impulse response filter whose frequency response is modified from that of the IIR filter. Multiplying the infinite impulse by the window function in the time domain results in the frequency response of the IIR being convolved with the Fourier transform (or DTFT) of the window function. If the window's main lobe is narrow, the composite frequency response remains close to that of the ideal IIR filter.

The ideal response is usually rectangular, and the corresponding IIR is a sinc function. The result of the frequency domain convolution is that the edges of the rectangle are tapered, and ripples appear in the passband and stopband. Working backward, one can specify the slope (or width) of the tapered region (transition band) and the height of the ripples, and thereby derive the frequency domain parameters of an appropriate window function.

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Clock gating saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred Clock gating works by taking the enable conditions attached to registers, and uses them to gate the clocks. A design must contain these enable conditions in order to use and benefit from clock gating. This clock gating process can also save significant die area as well as power, since it removes large numbers of mux and replaces them with clock gating logic. This clock gating logic is generally in the form of "integrated clock gating" (ICG) cells. However, the clock gating logic will change the clock tree structure, since the clock gating logic will sit in the clock tree

2.1 PARKS-MCCLELLAN METHOD

The Remez exchange algorithm is commonly used to find an optimal equiripple set of coefficients. Here the user specifies a desired frequency response, a weighting function for errors from this response, and a filter order N. The algorithm then finds the set of coefficients that minimize the maximum deviation from the ideal. Intuitively, this finds the filter that is as close as you can get to the response given that you can use only coefficients. This method is particularly easy in practice since at least one text includes a program that takes the desired filter and N, and returns the optimum coefficients.

Equi ripple FIR filters can be designed using the FFT algorithms well. The algorithm is iterative in nature. You simply compute the DFT of an initial filter design that you have using the FFT algorithm (if you don't have an initial estimate you can start with h[n]=delta[n]). In the Fourier domain or FFT domain you correct the frequency response according to your desired specs and compute the inverse FFT. In time-domain you retain only N of the coefficients (force the other coefficients to zero). Compute the FFT once again. Correct the frequency response according to specs.

2.2 DIGITAL MULTIPLICATON

A multiplication by a fixed-point constant can be done "multiplierless" using additions (or subtractions) and shifts only. This is relevant for hardware implementation to avoid costly multipliers, but may also be beneficial for software implementations, for example, for embedded processors. As a simple example, y = 5x can be computed as y = (x<<2) + X. Such a solution is called a multiplier block. For a given constant c, the problem is to find a multiplier block with the least number of adds/subtracts. This problem and two extensions are visualized 27 on the right. We have developed algorithms and online generators for each of the problems. Finding an optimal solution for the problems is NP complete [1]. Thus our algorithms find only a close-to-optimal solution. Note that the three problems are related to, but fundamentally different from the adder chain problem discussed in detail.

2.3 SCM

A straight forward way of multiplying by a given constant c using add/subtracts and shifts only can be read off from the bit representation of c. We call it the binary method. The well-known CSD (canonical signed digit) recoding reduces the number of add/subtracts required. The smallest example where CSD fails to produce the optimal solution is c=45 (CSD is above, the optimal below): The optimal solution is only known for constants of bitwidth b<= 19 [1].

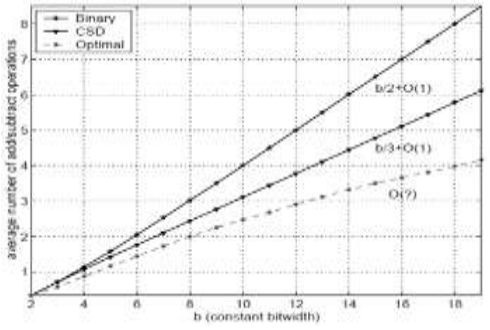

Fig-1 SCM

schematic diagram of 4 Stage parallel linear feedback shift register designed in the Xilinx suite. In this schematic the Integrated clock gating is given as the input to pparallel LFSR. So, the consumption of clock pulse gets reduced using Control logic. The common clock is given by dividing the clock pulse as input with the help of Integrated Clock Gating (IGC) to the 4 Stage parallel LFSR.

2.4 CSE

Subexpression elimination can be applied to a set of constant multipliers that multiply a common variable. The multiple constant multiplication (MCM) problem determines how subexpression elimination can be applied to the set of constant multipliers so that the number of shifts and additions required for implementation is minimized. A general framework and algorithm for solving this problem has been presented in, One of the attractive features of this algorithm is its versatility and adaptability to various forms of the MCM 30 problem. The algorithm uses an iterative matching process that consists of 5 basic steps.

A generic flow of FIR filter design and implementation can be divided into three stages: finding filter order and coefficients, coefficient quantization, and hardware optimization, in the first stage, the filter order and the corresponding coefficients of infinite precision are determined to satisfy the specification of the frequency response. Then, the coefficients are quantized to finite bit accuracy. Finally, various optimization approaches such as CSE are used to minimize the area cost of hardware implementations. Most prior FIR filter implementations focus on the hardware optimization stage.

3 PROPOSED SYSTEM

3.1 Sklansky adder

The Sklansky Adder is not as heavily impacted by wire delay in comparison to similar prefix adder such as Kogge-Stone. We assume the critical path in an adder is determined by the time required to pass the carry-bit from the least significant bit to most significant bit, illustrates the critical path of Sklansky adder

The "Sklansky's adder" builds recursively 2-bit adders then 4-bit adders, 8-bit adders, 16-bit adder and so on by abutting each time two smaller adders. The architecture is simple and regular, but suffers from fan-out problems. Besides in some cases it is possible to use less cells with the same addition delay. The Sklansky or divide-and-conquer tree reduces the delay to stages by computing intermediate prefixes along with the large group prefixes. This comes at the expense of fanouts that double at each level.

The gates fanout to, respectively. These high fanouts cause poor performance on wide adders unless the high fanout gates are appropriately sized, or the critical signals are buffered before being used for the intermediate prefixes. Transistor sizing can cut into the regularity of the layout because multiple sizes of each cell are required although the larger gates can spread into adjacent columns.

Important of Sklansky adder is to reduce critical path when number of bits becomes large. It is used to calculate the sum very fast.Adding a large number of input together. Usually, a very fast carry-look ahead or carry-select adder is used for this last stage, but Sklansky adder in order to obtain the optimal performance.

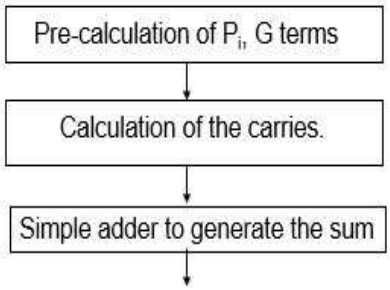

Fig-2 Algorithm for sklansky adder.

4. DESIGN SPECIFICATION

The controller is the control unit of adder and D-flipflop. It receives a RESET and CLK commands all other modules untils the results is obtained and it outputs is displayed. The complete adder is simply a top-level module which instantiates all the lower level functional modules describes. The associated VHDL source code is included in next section. The test bench is designed such that it can provided a serial of stimuli to be added the result is obtained. The simulation result files contains operand1 and operand2 sent to the adder and result received. In order to quickly validated the results. Timing is of great concern in any digital system. In this case there are two ways of looking at the timing issues. The first is the timing requirements of generating the fastest clock while respecting each individual timing requirement such that the functionality is not compromised.

The second is the overall latency of the operation. Upon locating he critical path, a new adder design was used to reduce the delay

This corresponding to the input of the 8-bit sklansky adder module to the carry-out which selects the appropriate results of the next stage of adder. As expected, this delay is shorter than the previous adder design and thus faster clock frequency is realised.

For the purpose of this project, a very little constraints were imposed on the design to be synthesized. Neither pin out, stringent timing, synthesis effort level nor was area optimized specified. The target devise was selectrd along with the source VHDL file and constraints file.

Place and route was performed using Xilinx design manager. The net list file created during the synthesis phase was imported into the tool which routed the design the appropriated target and generated a bit files which is used to physically program the device.

However, power optimization software packages can be used to apply the concepts of clock gating in order to reduce the power consumption of the circuit. For example, Xilinx has an option called "Intelligent Clock Gating" which uses the clock enable pin in a slice to neutralize superfluous switching activity. The technique is different from the classic clock gating discussed in this article because Intelligent Clock Gating doesn't actually create new clocks. Instead, Xilinx's technique uses clock enable pins of slices to disable registers that don't contribute to the circuit's operation for a given clock cycle.

Since the DFF shown in is sensitive to the positive edge of the clock, if the en signal too comes from devices that change state at the rising edge of the clock. In Figure 6.4, the clock signal, ck, goes from low to high at t=t1; some time later, the en signal transitions to high at t=t2. The time difference t2-t1 corresponds to the delay of the circuitry that produces the en signal. In this case, t2-t1will correspond to the delay of the FFs that store this particular state of the FSM plus the delay of the combinational circuit that generates the en signal from the Finite State Machine state(FSM).

Hence, the transitions of en will occur some time after the rising edge of ck. Let's use the above example waveforms to find a circuit that can generate an appropriate gated clock, gck, for From t2 to t4, the en signal is high and gck must be equal to ck. What if en is logic low? Should gck be high or low in this case? First, we assume that, for en=0, gck is set to low. Then, the red waveform shown in Figure 2.5 is obatined. To generate this waveform, replace the unknown circuit of with an AND gate.

5. RESULTS

The implementation of multiplication operation with shift and addition allows us to obtain an optimal performance. The coefficient quantization tends to decrease with increasing propagation length, rendering it as an attractive solution. From the below diagram represent the schematic block of the FIR filter.

In the block consists of the inputs of data, clock, reset, and the output was finalout. Using clockpulse input can be entered and produced the sequence of output. The choice of particular adder depends on the number of stages of logic, the number of logics gates, maximum fan out on each gate and amount of wiring between the stages, after the addition output is displayed.

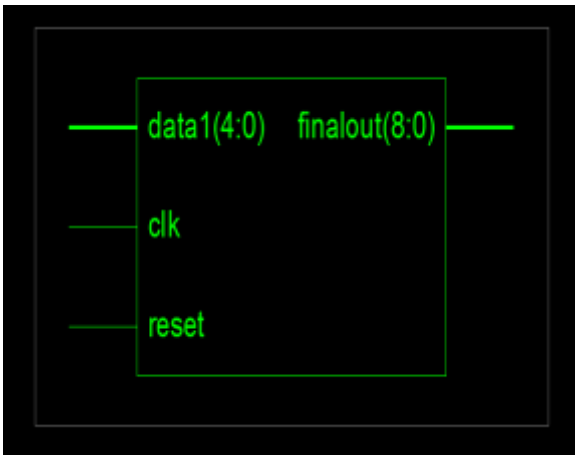

Fig-3 Block diagram for fir filter

The below the diagram is represent the RTL schematic diagram for the sklansky adder. This adder reduce the both power, area consumption. The focus in the present work is to establish the concept of sklansky adder implementation, therefore default adder in FPGA in hardware implementation.

The proposed adder scheme is verified by simulating a 8x8 adder in Xilinx Design suite ISE 9.2. The simulation have been performed using integrated software environment included in the design suit.

The scheme is mapped onto hardware using The RTL schematic view of the design showing the various blocks for 8x8 adder.

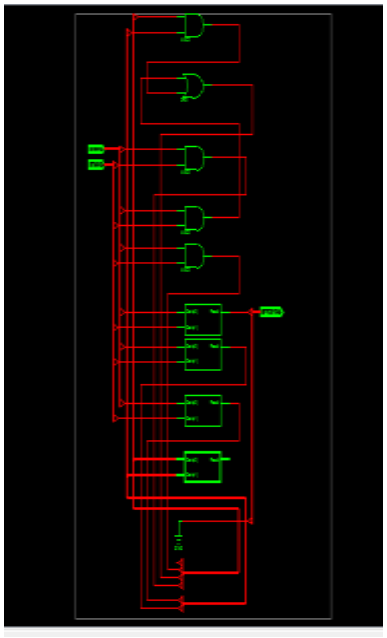

Fig-4 HDL schematic for adder

Further then the shifted vector go through an adder tree, each adder having two 8 bit vectors as inputs and giving back an 9 bit output. The shifters facilitate the shifting operation unlike conventional sequential adder where the data is shifted one bit position per cycle. Thus the partial product generation becomes faster and results in improved speed performance of the multiplication process.

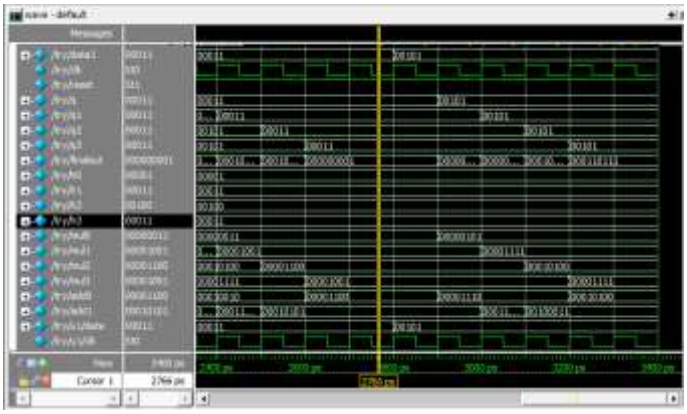

Fig- 5 Output waveform for proped system

The adder circuit is implemented in the form of a binary tree where each added is 8 bit wide as each of the partial product terms are of 9 bits due to the extension of the sign bit which has to be taken into consideration for the multiplication of signed numbers. The adder circuit can be implemented using various techniques available. The default adder of FPGA is used to test the proposed method.

```
=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Multipliers                                      : 2
 5x2-bit multiplier                               : 2
# Adders/Subtractors                               : 2
 8-bit adder                                      : 2
# Registers                                        : 20
 Flip-Flops                                       : 20

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

Fig-6 HDL synthesis of report

Therefore, the final result can be showed. The line indicated the input of the signal, second line is indicated the clk of the signal, and output of each d-filpflop is showed one by one.

The end of the result of wave is named as he finalout, which produced the sequence of 9-bit output. The sequence of outputs is produced by the corresponding input. The graphical design view, obtained in the Xilinx Platform Studio (XPS) for the target Multiplier device where the design is finally ported. Xilinx Zynq 702 System on Chip is a general purpose board having advantage of parallel units 9 (configurable logic blocks (CLB's)) as well as sequential unit (ARM Cortex A9 Duel core processor).

Embedded Design Kit (EDK) is used to model the multiplier in VHDL language. Further, to communicate with the on-chip processor Software development kit (SDK) is used which serves as a medium to give input to the multiplier and collect the final output. Finally the design is mapped on the FPGA.

From the structure greatly reduces the delay, and would be especially beneficial for a structure with large number of inputs. This advantage is, however, obtained at the expense of large area and a complex structure.

Using Xilinx, can be identified the number of multiplier and subtractor, Adder, registers, flip-flop are used. From the project, reduce the multiplier and adders.

The final output waveform of 4 stage parallel linear feedback shift register obtained in ModelSim 6.4a. The output of LFSR is given as 4bit in which the corresponding output is obtained by shifting the bit in parallel stage and clock pulse is consumed with the help of clock gating. By the help of clock gating input, power dissipation occurred in flipflop gets reduced in such a way that the performance of the circuit in increased. By giving common clock input with

the help of Integrated Clock Gating (IGC) the performance is improved by reducing power consumption.

6. CONCLUSION

A high processing SAD architecture is presented for Integer Motion Estimation for HEVC encoder. A group of 16 processing unit is used to calculate SAD of different block sizes from 4x4 to 64x64. Our proposed architecture takes 50ms clock speed for calculation of 316 square and 340 SMP blocks. Our scheme uses both pipelined and parallel processing to increase the performance and reduce computation time. The proposed method has better speed compare to related work and also consumes lesser hardware. The proposed method uses both parallel and pipeline approaches to improve the real-time performance. The proposed method has 1.36 times and 2.5 times higher speed than the previous work and best suited to real-time applications

REFERENCES

[1] L. Aksoy, P. Flores, and J. Monteiro, "A tutorial on multiplierless design of FIR filters: Algorithms and architectures," Circuits, Syst., Signal Process., vol. 33, no. 6, pp. 1689–1719, 2014.

[2] K. Azadet and C. J. Nicole, "Low-power equalizer architectures for high speed modems," IEEE Commun. Mag., vol. 36, no. 10, pp. 118–126, Oct. 1998.

[3] A. Belghadr and G. Jaberipur, "FIR filter realization via deferred endaround carry modular addition," IEEE Trans. Circuits Syst. I, Reg. Papers, to be published, doi: 10.1109/TCSI.2018.2798595.

[4] J. Chen, C.-H. Chang, J. Ding, R. Qiao, and M. Faust, "Tap delayandaccumulate cost aware coefficient synthesis algorithm for the design of areapower efficient FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 2, pp. 712–722, Feb. 2018.

[5] J. Chen, C. H. Chang, F. Feng, W. Ding, and J. Ding, "Novel design algorithm for low complexity programmable FIR filters based on extended double base number system," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 1, pp. 224–233, Jan. 2015.

[6] J. Ding, J. Chen, and C.-H. Chang, "A new paradigm of common subexpression elimination by unification of addition and subtraction," Microprocess. Microsyst., vol. 35, no. 10, pp. 1605–1617, 2016.

[7] J. Chen, J. Tan, C.-H. Chang, and F. Feng, "A new cost-aware sensitivity driven algorithm for the design of FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 64, no. 6, pp. 1588–1598, Jun. 2017.

[8] M. Faust and C.-H. Chang, "Optimization of structural adders in fixed coefficient transposed direct form FIR filters," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2009, pp. 2185–2188.

[9] M. Faust and C.-H. Chang, "Optimization of structural adders in fixed coefficient transposed direct form FIR filters," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2009, pp. 2185–2188.

[10] M. Faust, M. Kumm, C.-H. Chang, and P. Zipf, "Efficient structural adder pipelining in transposed form FIR filters," in Proc. IEEE Int. Conf. Digit. Signal Process. (DSP), Jun. 2015, pp. 311–314.