



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume3, Issue3)

Available online at www.ijariit.com

Design and Implementation of Fir Filter Using Retiming Technique

R. Dhivya

Velalar College of Engg. & Technology Erode
dhivvaraja36@gmail.com

Dr. P. Brindha

Velalar College of Engg. & Technology Erode
brindhaten@gmail.com

Abstract: Finite Impulse Response (FIR) Filter can be designed by the provision of specifications which are for a particular application requirement. An efficient FIR filter is designed using register reduction retiming technique. Also, an optimization environment is designed such that filter components of post retimed circuit such as adder and multiplier are upgraded depending on the Very Large Scale Integration (VLSI) design metrics such as area, speed, and power. In conventional cut-set retiming of FIR filter, higher critical path delay and latency occur due to unwanted pipelining. They require higher clock period. So overcome, FIR filter it's designed using Novel node – splitting and Node merging techniques. These schemes reduce the critical path delay by 50% and latency by 60%. In this paper node merging method is simulated and its performance parameters such as power, delay, critical path, latency, clock period and a number of registers are analysed. In Data Flow Graph (DFG) retiming of the digital circuit is used to reduce the propagation delay and critical path. The reduced propagation delay, in turn, reduces pipeline overheads. Thus, cut-set retiming can be made more efficient without increasing the register complexity and latency. The FIR architecture for node merging has to be designed and its performance is to be analysed. The proposed FIR filter using retiming technique are simulated using MODELSIM 10.1b and power analysis of the proposed work is done using Xilinx ISE 9.2i. In this proposed work the layout will be developed using AARON 9.03 tool.

Keywords: FIR filter, Retiming Technique, Critical Path Delay, Cut-Set Retiming.

I. INTRODUCTION

The FIR means “Finite Impulse Response”. The FIR filters are of the non-recursive type whereby the present output sample depends on the present input sample and previous input sample. Recursive is a type of filter which reuses one or more of its outputs as an input. In signal processing, a 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 FIR filter’s ability to provide stable linear- phase behaviour has made it gain acceptance in wide kind of signal processing applications. Since FIR filters have a capability of no phase distortion, they are considered as important. The frequency response of FIR filter is based on the value of coefficients or taps. The impulse response of a FIR digital filter is of finite duration.

The FIR filters are Bounded Input and Bounded Output (BIBO) stable and are easy to design. FIR filters are also less sensitive to filter coefficient quantization errors. This is very much required while implementing FIR filters on digital signal processors. Since the signal processing applications are iterative and recursive, high-performance FIR filters are very much needed. Here, initially, the operating frequency of FIR filters is improved by register minimization retiming without compromising on register count. In retiming, registers are moved over nodes in the logic network in such a way that functionality of the logic network is maintained the same. However, after retiming, the number of registers can be slightly increased for certain design cases.

II. RELATED WORK

A. FULLADDER

A full adder adds binary numbers and record for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and C_{in} . A and B are the operands, and C_{in} is the bit carried in from the next less significant stage.

TABLE I TRUTH TABLE OF FULL ADDER

INPUTS			OUTPUTS	
A	B	C _{in}	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table I shows the truth table of full adder circuit in which there are three inputs A, B and C_{in} and two outputs C_{out} and Sum (S).

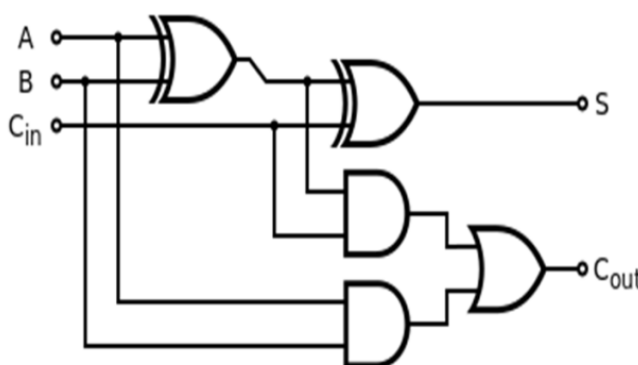


Fig. 1. Logic diagram of full adder

Fig. 1 shows the basic logic involves in a full adder. Two XOR gates, two AND gates and one OR gate is used.

B. WALLACE TREE MULTIPLIER

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two numbers. The Wallace tree has three steps:

- a) Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, springy results. Depending on the position of the multiplied bits, the wires carry different weights.
- b) Reduce the number of partial products to two by covers of full and half adders
- c) Group the wires in two numbers, and add them with a conventional adder. A typical Wallace tree architecture is shown in the figure below Fig. 2. In the diagram, AB0-AB7 represents the partial products the partial products.

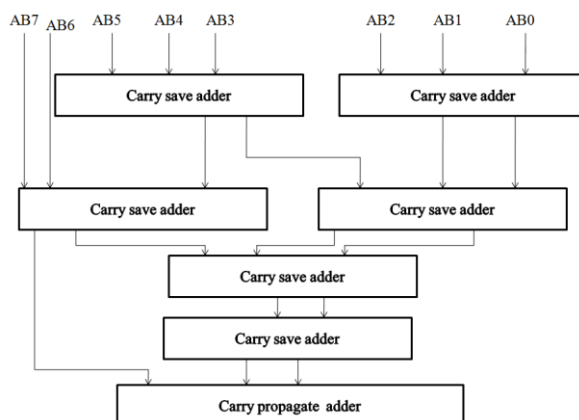


Fig. 2. Wallace tree multiplier

The Wallace-tree multiplier is popularly used due to its lower propagation delay. The multiplication scheme of 8x8 Wallace-tree multiplier is shown by the dot diagram in the figure below Fig. 3. It performs successive carry-save reduction of three bits to two bits by a 3-2 counter (or carry-save adder) so as to reduce the tree height finally to 2 (one row consisting of ‘carry’ bits and the other

row comprising of ‘sum’ bits). The sum word and the carry word are finally added together by the vector-merging adder to compute the result of the multiplication.

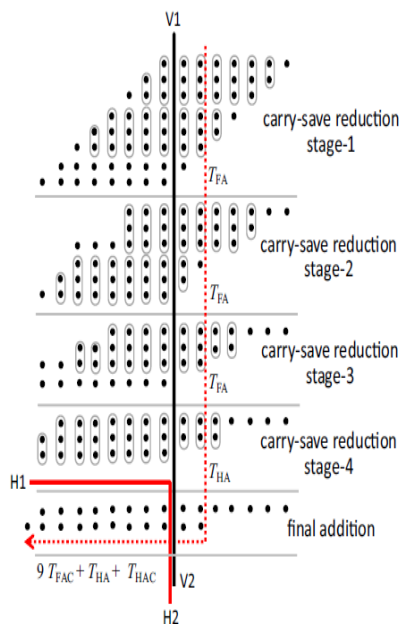


Fig. 3. Dot diagram of 8 ×8 basic Wallace tree multiplier

III. PROPOSED RETIMING TECHNIQUE

The proposed technique for the reduction of critical path by a grouping of retiming with node splitting and node merging. Besides, the application of the proposed system in FIR filter and simple examples of recursive computation.

Digital Signal Processing (DSP) deals with the operation of digital signals using complex signal processing systems built from basic building blocks like filters. The proposed work performance in the FIR filter in added parts. This work evaluates the performance of the improved design in terms of delay and power. So they need to design high-speed adders. The ripple carry adder part is replaced with the standard and improved carry select adder and the FIR Filter is implemented and the performance of the design is evaluated in terms of delay and power.

A. RETIMING

A critical path operation through a combinational cloud in a feed-forward system can be broken by the addition of pipeline registers. Pipelining and retiming are two different aspects of digital design. In pipelining, extra registers are added that change the transfer function of the system, whereas, in retiming, registers are relocated in a design to optimize the desired point. Digital filters are classified as the most common blocks in signal processing applications that can be represented by a synchronous data-flow graph (DFG). Applying retiming techniques on the synchronous data flow graphs lead rapid digital circuits. It's a technique which is often used to reduce the clock period by the relocation of delay without changing the functionality of the circuit.

Retiming can either shift the existing registers in the design or can be followed by pipelining, where the designer places a number of registers on a cut-set line and then applies a retiming conversion to place these registers at appropriate edges to minimize the critical path while keeping all the other objectives as the resulting purpose. Retiming is the employed to optimize a set of design objectives. These usually involve differing design solutions, for example maximizing performance may result in an increase in area. Retiming also assists in maximizing the testability of the design and to increase the performance of field programmable gate arrays (FPGAs) using registers pooling. A known data flow graph (DFG) can be retimed using cut-set or delay transmit approaches.

Retiming is a transformation technique used to change the delay factor in a circuit without affecting the input/output characteristics of the circuit. For example, consider the IIR filter in Fig. 4(a). This filter is described by

$$\begin{aligned}
 w(n) &= ay(n-1) + by(n-2) \\
 y(n) &= w(n-1) + x(n) \\
 &= ay(n-2) + by(n-3) + x(n)
 \end{aligned}$$

The filter in fig 2.1 (b) is described by

$$\begin{aligned}
 w_1(n) &= ay(n-1) \\
 w_2(n) &= by(n-2) \\
 y(n) &= w_1(n-1) + w_2(n-1) + x(n) \\
 &= ay(n-2) + by(n-3) + x(n)
 \end{aligned}$$

Although the filter in Fig. 4(a) and Fig. 4(b) have delays at different locations, these filters have the same input/output characteristics. These filters can be derived from one another using retiming.

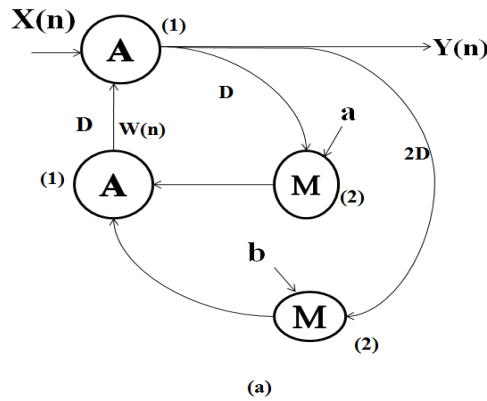


Fig. 4.(a) A DFG.

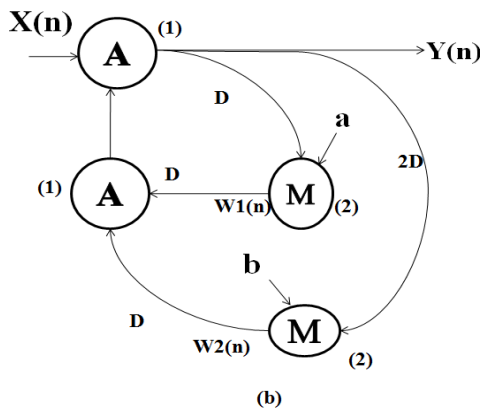


Fig. 4.(b) The retimed DFG obtained using $r(1) = 0, r(2) = 1, r(3) = 0,$ and $r(4) = 0$.

Retiming has many solicitations in synchronous circuit design. These applications include reducing the clock period of the circuit, reducing the number of registers in the circuit, reducing the power consumption of the circuit and logic synthesis. The retiming is to reduce the clock period and number of registers.

Retiming can be used to increase the clock speed of a circuit by reducing the computation time of the critical path. The critical path is defined to be the path with the longest computation time among all paths that contain zero delays and computation time of the critical forward is the lower bounce on the clock period of the circuit. The critical path of the filter in Fig. 4(a) passes through 1 multiplier and 1 adder and has a computation time of 3 u.t. The retimed filter has a critical path that passes through 2 adders and has a computation time of 2 u.t. By retiming the filter in Fig. 4(b), the clock period has been reduced from 3 u.t. or 33%.

The filter uses the 4 registers while another filter in uses 5 registers. Since retiming can affect the clock period and the number of registers, it is sometimes desirable to take both of these parameters into account.

Retiming can be used to reduce the power consumption of a circuit, by reducing the switching action, which can lead to dynamic power dissipation in static CMOS circuit. Placing registers at the inputs of nodes with large capacitances can reduce the switching actions at these nodes, which can be lead to low power solutions.

B. Cut-Set Retiming

Cut-set retiming is a technique to retime a data flow graph by applying a suitable cut-set line. A suitable cut-set is generally a set of forwarding and backward edges within a DFG intersected by a cut-set line such that if these edges are uncovered away from the graph, the graph becomes disjoint. The cut-set retiming involves transferring a number of delays from edges of the same path across a cut-set line of a DFG to all edges of differing direction across the same line. These transfers of delays do not affect the transfer function of the DFG.

Cut-set retiming only affects the weights of cut-set in the cut-set. If the two disconnected subgraphs are labeled G_1 & G_2 then cut set retiming consist of adding k delay to each edge from G_1 to G_2 and removing k delay from each edge from G_2 to G_1 .

Feasible solution:

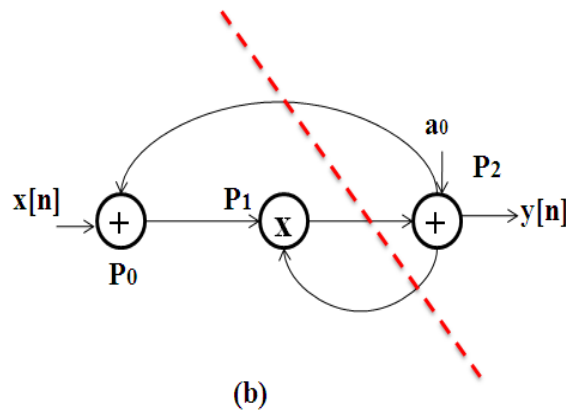


Fig.5 (b) DFG with cut-set line

Cut-set retiming can be applied to a direct-form FIR filter to get a transposed direct-form (TDF) account. The TDF is formed by initial reversing the direction of addition, sequentially placing cut-set lines on every delay edge. Each cut-set line cuts two edges of the DFG, one in the forward and the other in the backward direction. Then retiming moves the registers from the forward edge to the backward edge. This transforms the filter from DF to TDF. This form breaks the critical path by placing a register by every addition process.

C. RETIMING OF AN INFINITE IMPULSE RESPONSE FILER

The FIR filter uses splitting and merging technique combined with retiming helps to reduce the critical path delay. The DFG of FIR filter of length N = 4 is shown in Fig. 6(a) where the multiplication nodes, as well as addition nodes, are split into two parts of closely equal delays.

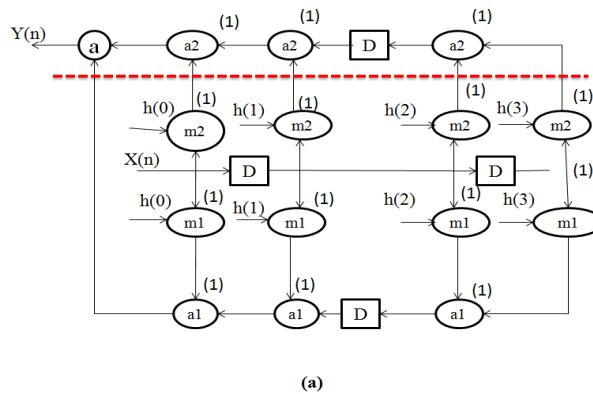


Fig. 6 (a) DFG of Direct- Form configuration of FIR filter of length N = 4

The feed-forward cut-set is shown by the horizontal dashed line in Fig. 6(a); and the resulting retimed DFG is shown in Fig. 6(b). Two pairs of (m1, a1) connected nodes (in gray area) in the DFG of Fig. 6(b) could be merged to form nodes N₁ and N₂.

Assuming that the multiplication nodes and adder nodes are split into two parts of practically equal propagation delay of

$$1 \text{ u.t.} = (1/2) T_{\text{MULT}} \text{ or } (1/2) T_{\text{ADD}}$$

where T_{ADD} is (~2L) bit addition time, while T_{MULT} is the time required for multiplication of two L-bit words, the delay of N₁ and N₂ nodes can be found to be nearly 1 + Δ. The critical path of the retimed FIR filter is the same as the delay of N₃ node, which amounts to

$$T_{\text{FIR-RT}} = \sim (1/2) T_{\text{MULT}} + 3\Delta$$

Where Δ is small compared with T_{MULT}, the proposed retiming combined with node splitting and node merging can result in substantial reduction of the critical path over the lowest achievable critical path (one multiplication time, T_{MULT}) by the conventional retiming.

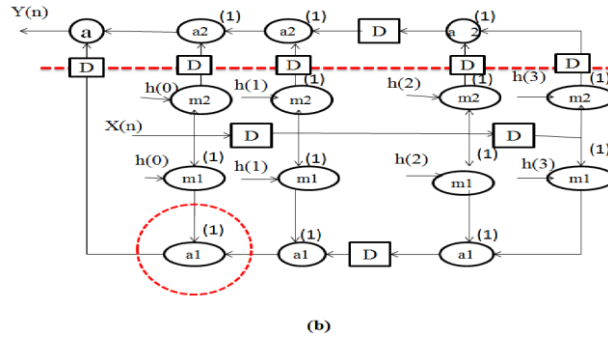


Fig. 6 (b) Splitting of nodes of the DFG

RESULTS AND DISCUSSION

The waveform of the existing and proposed method of the finite impulse response filter using a retiming technique which is simulated using MODELSIM and analysis using XILINX is shown in waveform below.

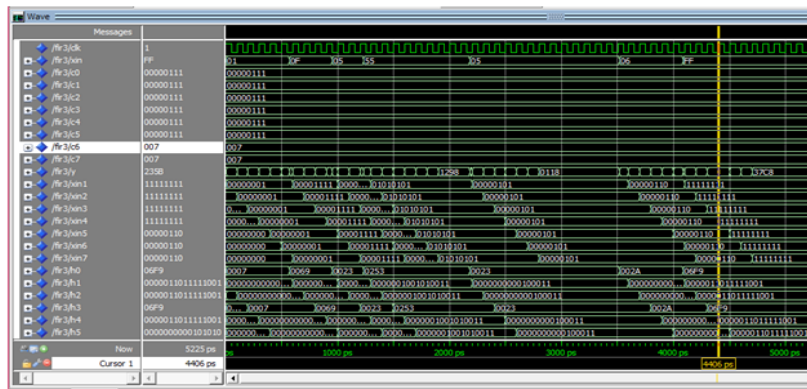


Fig. 7. Waveform of existing method

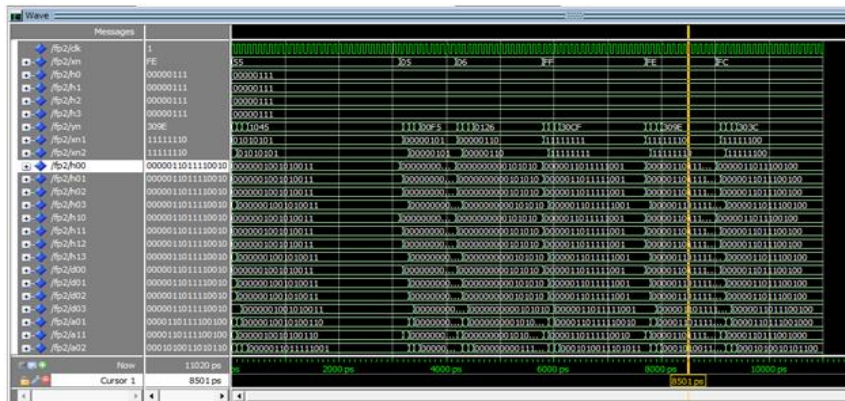


Fig.8. Waveform of proposed method

The comparison of existing method and proposed method of a number of registers, the number of LUTs, delay and power are shown in the table below.

TABLE II
COMPARISON OF EXISTING SYSTEM AND PROPOSED SYSTEM

Systems	No. of Register	No. of LUTs	Delay in (ns)	Power in (mW)
Existing	48	1489	29.03	74.2
Proposed	40	1281	18.77	72.2

CONCLUSION

Retiming is main and powerful technique to digital circuit performance optimization. Retiming decreases the critical delays of a circuit by varying its location so that combinational path delays are balanced across the entire design. In FIR filters the retiming in combination with node splitting and node merging offers nearly 50%–60% reduction of the critical path with a secondary area overhead. The techniques presented in this project can be applied during retiming of the DFGs involving multiply–add operations and adder-trees to decrease pipelining overhead and critical path delay. The proposed retiming techniques for fixed-point circuits can be extended to floating-point circuits. The proposed method area and delay have been reduced compared to the existing method. Both designs are synthesized in Xilinx ISE 9.2i FPGA (field programmable gate array), MODELSIM 10.1b Tool and also analyze the power, delay, number of the register, clock period and power consumption. Finally, the layout will be developed using AARON 9.3 Tool.

REFERENCES

- [1] Chao L.F and ShaE.H.M (1997), ‘Scheduling data flow graphs via retiming and unfolding’, IEEE Trans. Parallel Distrib. Syst., vol. 8, no. 12, pp. 1257–1264
- [2]Leiserson C. E. and J. B. Saxe (1991), ‘Retiming synchronous circuitry’, Algorithmica, vol. 6, no. 1-6, pp. 5–35.
- [3]Meher P. K. (2015), ‘Seamless pipelining of DSP circuits’, J. Circuit Syst.Signal Process. IEEE Trans. Circuit Syst. I, Reg. Papers, vol. 67, no. 5, pp. 778–788.
- [4]Meher P. K and Park S. Y. (2014), ‘Critical path analysis and low-complexity implementation of the LMS adaptive algorithm’, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 3, pp. 778–788.
- [5]Monteiro J, Devadas S, and Ghosh A. (1996), ‘Retiming sequential circuits for low power’, Int. J. High-Speed Electron. Syst., vol. 7, no. 2, pp. 323–340,
- [6]Parhi K. K. (2007), ‘VLSI Digital Signal Processing Systems Design and Implementation’, New York, NY, USA: Wiley.
- [7]Sudha .S .A, Marimuthu C. N. (2014), ‘Design of Area Delay-Power Efficient Adaptive Filter using Wallace Tree Multiplier’, Vol. 2, Issue 4.
- [8]Shenoy N. and Rudell R. (1994), ‘Efficient implementation of retiring’, in Proc. IEEE/ACM Int. Conf. Comput. Aided Design, pp. 226–233.
- [9] Yu Pan and Pramod Kumar Meher (2014), ‘Bit-Level Optimization of Adder-Trees for Multiple Constant Multiplications for Efficient FIR Filter Implementation’, IEEE trans. regular papers, vol. 61, no. 2.