



Research article

Different retiming transformation technique to design optimized low power VLSI architecture

Jalaja S^{1,*} and Vijaya Prakash A M²

¹ Department of Electronics and Communication, Research Scholar, Visvesvaraya Technological University, Bangalore Institute of Technology, Bangalore, India

² Department of Electronics and Communication, Bangalore Institute of Technology, Bangalore, India

* **Correspondence:** Email: jalajabit@gmail.com; Tel: +919845100604.

Abstract: A different method for designing low power retime architecture is presented in this paper. The modified retiming transformation techniques approach to reduce the dynamic power consumption of the digital circuit, without compromising the output results. In this paper, retiming transformation is extended in two-ways to reduce the power consumption of the design. Graphical Circular Retiming and Tabular Shift Retiming are the two methods used to realize how the registers are mapped to reduce the glitching power. Proposed transformation technique delay value is placed in the form of metric and verified without sacrificing the functionality. Proposed transformation technique is applied to FIR filter to analyze the simulation and synthesis results as proofs of this concept. Experimental results are compared with the conventional retiming transformation technique with the same operating frequency, and with the significant reduction in dynamic power of FIR filter.

Keywords: graphical retiming transformation; tabular retiming transformation; reduction in dynamic power

1. Introduction

Reduction in power is very much essential in new smart electronics devices. Modern tools provide many technique to reduce dynamic and leakage power. So far dynamic power reduction is done with logic restructuring and synthesis, gate sizing, bus multiplexing, retiming and so on. In this paper, a different form of retiming transformation is proposed to reduce the power consumption of the design. These technique is proposed based on directed graphs, algebraic method and recurrence equations. Proposed technique is applied to FIR filter to compare the power consumption and hardware block of the existing design. The conventional equation for an n-tap FIR filter is represented as,

$$y_n = \sum_{k=0}^{n-1} a_k x_n \quad (1.1)$$

Where x_n is the input data, y_n is the filtered output and a_i is the coefficient of the filter. A set of input data samples are fed to x_n and their respective tap input data samples are given by $x[n - k]$. The structure of conventional transposed FIR filter [13,14] is shown in Figure 1.

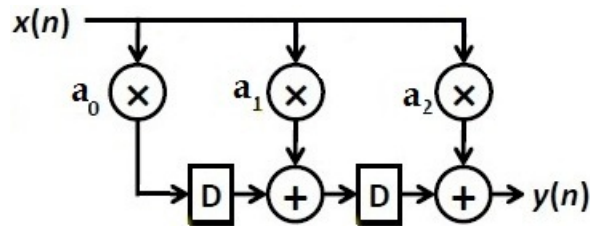


Figure 1. Structure of transposed FIR filter.

Infinite impulse response filter (IIR) and finite impulse response filter (FIR) are the two different types of digital filters in signal processing. Compared to IIR filter design, FIR filter is the simplest one with several advantages. Linear phase finite impulse response filter is used in many DSP processors for rapid processing and higher efficiency. Basic hardware blocks required to design FIR filter are adder, delay lines, and multipliers. Many researchers had proposed different techniques and architectures to design FIR filter multiplication block. However, some trade off with respect to area, delay and power dissipation were found with earlier state-of-the-art research. Graphical Circular Retiming and Tabular Shift Retiming transformations were applied to FIR filter multiplication block to reduce the power consumption of the design. The convolution of two input data signals is given by

$$y_n = a_i * x_n \quad (1.2)$$

Many researchers had proposed different forms of optimized multiple constant multiplication (MCM) to design FIR filter. The product-accumulation [13] was implemented using integrated MCM block with structural adder to design FIR filter in which it was optimized using retiming transformation technique. Incremental delay of their corresponding product accumulation was significantly reduced, with the penalty of small area [14] and this technique was used to analyze the delay of transposed FIR filters. The transposed FIR filter MCM scheme [3] was implemented to reduce the complexity of the design. Retimed data flow graph transpose FIR filter type II configuration was also highlighted to minimize the register complexity. Mathematical formulation for block transpose FIR filter was also derived. In MCM, block minimization problem had been overcome with mixed integer programming based algorithm [15] and different filter order. The performance of the results were analyzed in terms power and the cell area. Different types of multiplication methods [1] were used to design 16 order FIR filter to reduce the complexity of the design. For different frequency, energy delay product (EDP), the results were compared with the existing design. Distributed arithmetic high throughput reconfigurable FIR filter [13] was designed to reduce hardware cost. The registers were shared by distributed arithmetic for different bit slice. The

realization of MCM block [9] was addressed to minimize the adders/subtractors block of FIR filter design. The research work [9] showed better result compared to conventional implementation of MCM block. Ring topology based FIR filter [2] was modeled for neural network application. In this work author used modified retiming serial multiplier to achieve low power consumption and two different types of adders were used to increase the performance. In [10] the researcher introduced a new algorithm to realize the FIR Filter to reduce the hardware complexity of the design. It was used to remove the power line interference (PLI) from electrocardiogram (ECG) signal. Approximate synthesis technique was implemented in [14] to achieve energy efficiency in FIR filter design. Sub expression elimination algorithm was used to design FIR filter and power reduction was achieved using approximate computing. The Programmable FIR filter was designed using common sub expression for multiplier less block. [6] Low complexity circuit was obtained based on extended double base number system. FIR filter as well as Lattice FIR filter was redesigned using node splitting and node merging retiming technique in [11] and performance of the filter was enhanced after retiming transformation without affecting the area complexity. Register transfer level (RTL) register retiming reduces the critical path delay compared to conventional retiming [12] and RTL retiming was applied to control logic, of multiplexers to improve the clock frequency. Based on above survey, different Retiming transformation techniques are introduced in this paper to improve the performance level of FIR filter.

2. Graphical and tabular retiming transformation

Leiserson and Saxe 1983 [4,5] proposed the retiming transformation technique, in which delay elements move from input path to output path without sacrificing the input/output functionality of the circuit. Clock period minimization algorithm was proposed [4,5] and implemented to synchronous design to increase the speed. Retiming transformation can also be applied to high switching activity factor node to reduce power consumption of the design [7] (Monteiro et al. 1993). In CMOS technology, large amount of power is consumed due to dynamic power, further it is subdivided into short circuit power, glitch power and switched power dissipation. The power consumption of a CMOS digital circuit is formulated in equation (3):

$$P = \alpha C_i V_{dd}^2 f_i + I_{sc} V_{dd} f_i + I_{leak} V_{dd} \quad (2.1)$$

Where C_i is output gate capacitance, V_{dd} is the supply voltage, f_i is the gate frequency, α is the switching activity factor, I_{sc} is the short circuit current during switching and I_{leak} indicates leakage current. The pictorial representation of retiming transformation is discussed in detail [8] (Parhi 1999). Figure 2 shows the Tabular Shift multiplication, in which filter coefficient is folded and shifted one position left and multiplied with input signal. Each output vertices are partitioned to insert the registers 'D', and to implement the retiming transformation. To minimize the power consumption of the filter, the flip-flops are shifted towards the product of shifted filter coefficient and input signal and is as shown in Figure 3. Once the filter tap-length is increased, large multiplication steps are required to perform FIR filter operation. Therefore, flipflops are shifted in the diagonal form to reduce the glitch that occurs due to undesirable signal transition. The synthesis result shows that higher the taplength the lower is the area. This is due to the adjustment in delay after applying tabular retiming transformation. A 4-tap FIR filter multiplication block is implemented using Graphical Shift

multiplication method as shown in Figure 4. Each output samples are calculated with the sum of individual product terms of input signal and filter coefficient in the form of shifted folded sequence where each output samples are represented as $y_0, y_1 \dots y_n$. In Figure 5(a) a diagonal cut-set line is applied to divide the graph into two subgraph to move the registers and to distribute the delay. In the DFG of Figure 5(b) desired number of registers are added to all product samples like m_1, m_2, m_3 and m_4 . Then systematically the registers are moved based on node transfer theorem and its multiplication example as shown in Figure 8 analyzes area-delay-power value.

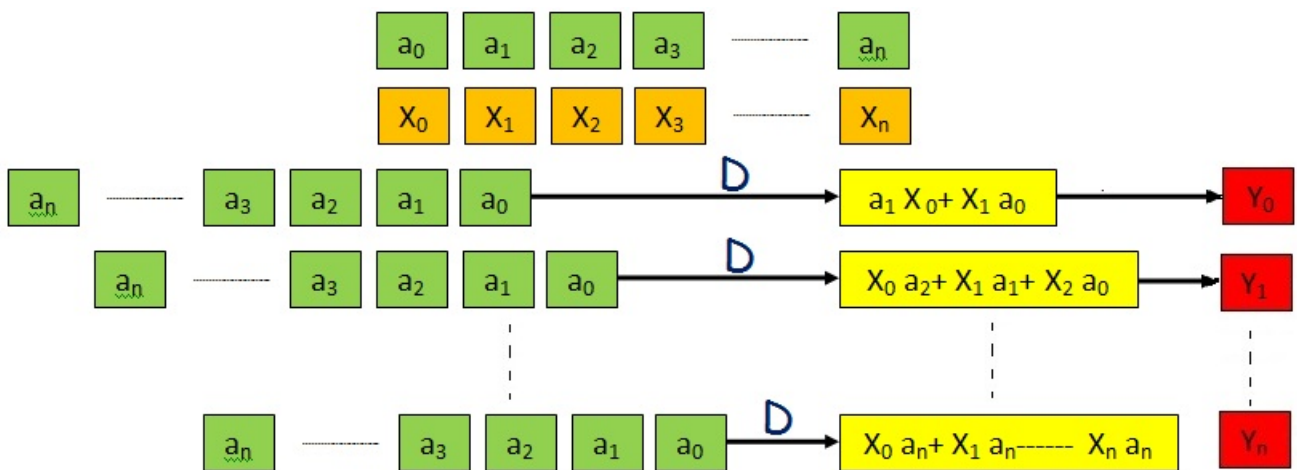


Figure 2. Tabular shift multiplication.

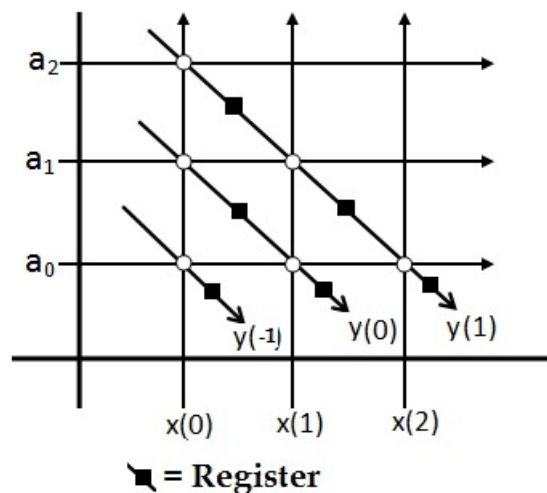


Figure 3. Tabular shift retiming transformation.

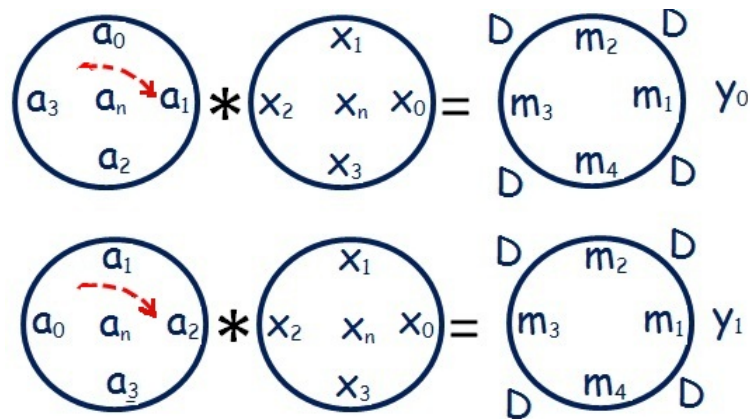


Figure 4. Graphical shift multiplication.

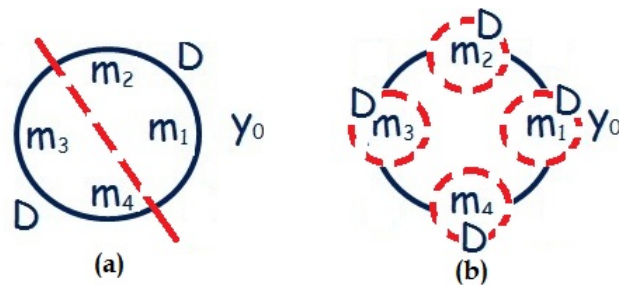


Figure 5. Graphical shift retiming transformation.

3. Analysis of proposed retiming transformation

Let us consider the circuit graph $G = \langle V, E, d, w \rangle$ consists of a set of n vertices and a set of edges with the following additional notations:

- V = Set of nodes (vertices) of graph G and it indicates the data operations.
- E = Set of directed edges of G .
- Vertex u to vertex v of the directed edge is denoted as $e:u \rightarrow v$.
- $w(u,v)$ is the number flip-flops and also referred as the weight of the edge.
- $d(v)$ is gate delay of vertex v . The two edges U and V are retimed to obtain $w_r(e)$.
- D represents the registers on an edge.

Retiming properties are summarized with the following equation and data flow graph diagram is shown in Figure 6. After retiming edge weighting w_r defined for an edge $u \xrightarrow{e} v$ in G , can be written as follows:

$$w_r(e) = w(e) + r(U) - r(V) \quad (3.1)$$

For example, Figures 7 and 8 show pictorial representations of data blocks and its corresponding cutset line. Fig. 7 consists of set of objects $V=\{v1, v2, v3, v4\}$ called vertices and another set, $E=\{e1, e2, e3\}$ of elements called edges. Initially each vertex is placed in separate cutset line $c1$ through $c5$ which is shown in Fig. 7. The primary goal of register allocation to $c1 - c5$ line is to

minimize the power consumption of the design. On the next iteration all the register whose representative vertices is further retimed as shown in Fig. 8. The multiplier block takes longer computation than adder block. As a result delay and power consumption are more for multiplier block compared to adder block. The two multiplications can share the two-stage pipeline multiplier, and this computational cloud exactly divides into two stages to reduce the complexity of the design. Further to reduce the power consumption of the design, all the registers moved to high switching activity of the nets. Combinational cloud is optimized using node transfer theorem to reduce the power consumption.

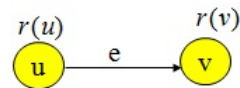


Figure 6. Retime signal flow graph.

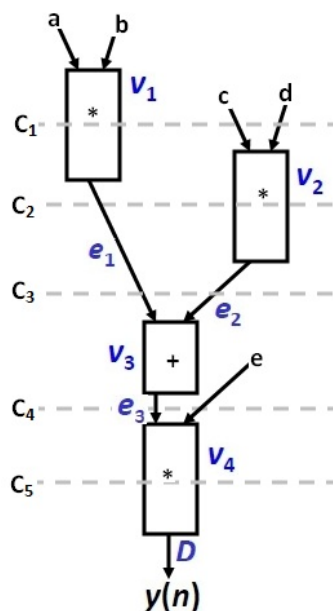


Figure 7. Example of data flow graph.

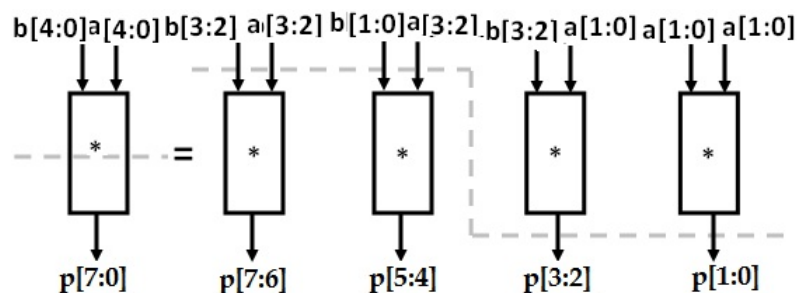


Figure 8. Retime multiplication block.

The Tabular Shift Retiming equation is formulated in Fig. 3. Resultant multiplication output values are stored in $Y_0 Y_1 \dots Y_n$ and represented in the DFG as shown in Figure 9. Let r be retiming and then for any path $u \xrightarrow{p} v$ in G , form the retiming equation (5) as follows.

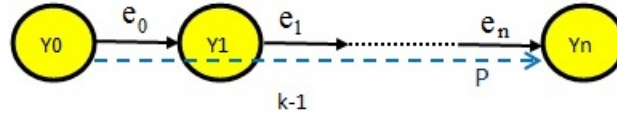


Figure 9. Tabular shift retime signal flow graph.

$$\begin{aligned}
 w_r(p) &= \sum_{i=0}^{k-1} w_r(e_i) \\
 &= \sum_{i=0}^{k-1} (w(e_i) + r(Y_{i+1}) - r(Y_i)) \\
 &= \sum_{i=0}^{k-1} w(e_i) + \left(\sum_{i=0}^{k-1} r(Y_{i+1}) - \sum_{i=0}^{k-1} r(Y_i) \right) \\
 &= w(p) + (r(Y_n) - r(Y_0))
 \end{aligned} \tag{3.2}$$

Similarly for the Graphical Shift Retiming feasible solution equation (6) and (7) obtained from Fig. 4 as follows:

$$\begin{aligned}
 w_r(G_{m1m2} \rightarrow G_{m3m4}) &= w(G_{m1m2} \rightarrow G_{m3m4}) + (r(G_{m3m4}) - r(G_{m1m2})) \\
 &= w(G_{m1m2} \rightarrow G_{m3m4}) + k
 \end{aligned} \tag{3.3}$$

$$\begin{aligned}
 w_r(G_{m3m4} \rightarrow G_{m1m2}) &= w(G_{m3m4} \rightarrow G_{m1m2}) + (r(G_{m1m2}) - r(G_{m3m4})) \\
 &= w(G_{m3m4} \rightarrow G_{m1m2}) - k
 \end{aligned} \tag{3.4}$$

Where range of k lies between-

$$-\min[w(G_{m1m2} \rightarrow G_{m3m4})] \leq k \leq \min[w(G_{m3m4} \rightarrow G_{m1m2})]$$

4. Retiming for low power

Once the FFs are moved from one location to another location, it affects the switching activity on net as well as capacitance. The product of the load capacitance on every node and switching activity factor of the circuit represents the power dissipation. Placing the register at the high switching activity node can reduce the power dissipation of the design. In this proposed work for conventional transposed FIR filter, average switching activity factor is estimated. Tabular Shift Retiming and Graphical Shift Retiming are applied to the same filter to reduce the glitching power. Power dissipation of the circuit is significantly affected from number of registers in a circuit. Register added to each multiplier fan-out nodes to increase the performance of the design. Consider the example in Figure 10 which consist of

c1, c2, c3, c4 and c5 complex combinational blocks. For high fan-out node c5 as shown in Fig. 10 (a), is inserted with register to achieve low power results as shown in Fig. 10(b).

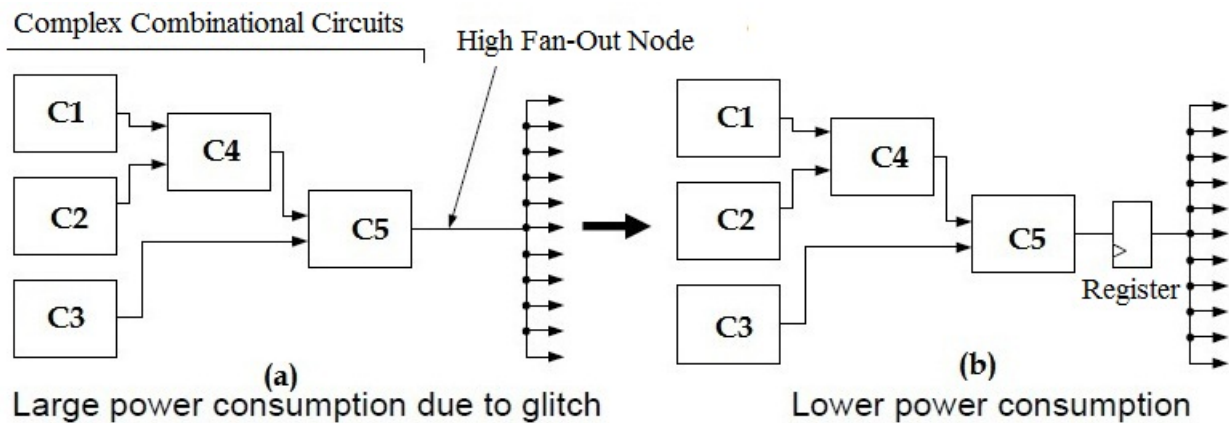


Figure 10. Retiming for low power.

5. Transposed FIR filter

Hardware implementation of Transposed FIR filter is beneficial compared to direct form structure. Transposed FIR structure multiplier block is retimed with different transformation technique. Inside the multiplier block, several cutset lines are contained in which products are generated at each stage and are retimed to reduce the power consumption of the FIR filter. Many researchers have proposed many techniques to reduce the complexity of multiplier block. In this paper, proposed optimization technique is applied to multiplier block. Feed-forward cutset is applied to conventional transposed FIR filter as shown in Fig. 2 and Retime is applied to multiplier as well as adder block in different ways.

Consider the 3-tap FIR filter example to implement retiming transformation, where red dotted lines indicate the cutset line as shown in Figure 11. Every new data sample has to be left shifted and multiplied by its coefficient a_i , to compute output data for one single input; it requires three multiplier and two adders block. For higher tap length the number of multiplication block are more. For n number tap length the complexity of the design increases as well as performance is also degraded. So in order enhance these two feature different method of retiming transformation is applied to filter design. Tabular Shift and Graphical Shift Retiming technique replace every register in a high fan-out node to reduce power consumption. Here different form of retiming transformation is applied to bit addition and bit multiplication concurrently. Cut-set line applied to each computational unit to implement the retiming transformation as shown in Fig. 11. Synthesis results shows better performance result compared to existing FIR filter [11,13].

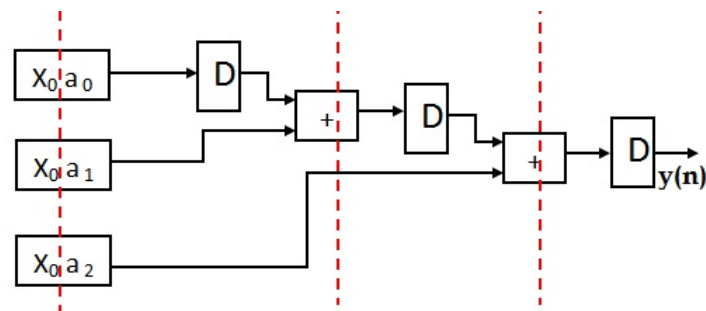


Figure 11. Modified 3-tap cutset Transposed FIR filter.

6. ASIC implementation of synthesis result

The proposed paper discusses the ASIC (Application Specific Integrated Circuit) based implementation of retiming transformation method to design FIR filter. Proposed structure is synthesized using cadence compiler 45 nm to obtain delay, area and power consumption of the design from an ASIC perspective. The different forms of positioning the flip-flops to FIR filter and its performance results are summarized in Tables 1, 2, 3, 4 and 5. Obtained synthesis result graphs are plotted to analyze the performance of power consumption and Power Delay Product as shown in Figures 12 and 13. The calculated Area-Delay-Product (ADP) and Power-Delay-Product (PDP) of the design are compared with the existing design. Product Accum. (Accumulation) design structure [13] is compared with proposed design, which has 72.9% reduction in terms of PDP and 54.66% in terms of ADP. The synthesis comparison result in terms of delay, area and power consumption listed in table for different tap length (N) 25, 60, 80 and 108 with block-size (L) 16. Table V shows the comparison result of existing node merging and node splitting retiming (node S-M) technique [11] with proposed method. For 128-tap length FIR filter Graphical Shift Mult. (Multiplication) Retiming consumes less ADP compared to state-of-the-art design of [11].

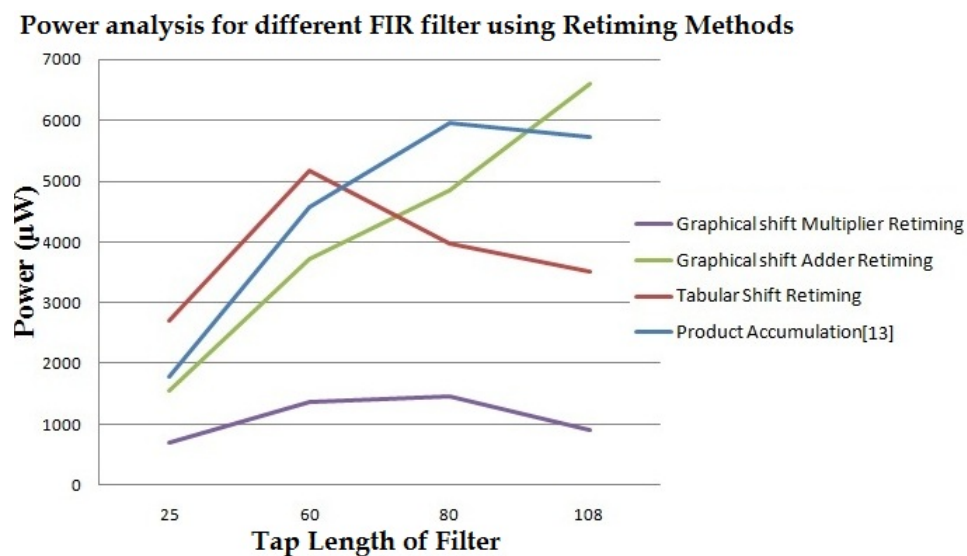


Figure 12. Power performance graph for different tap-length.

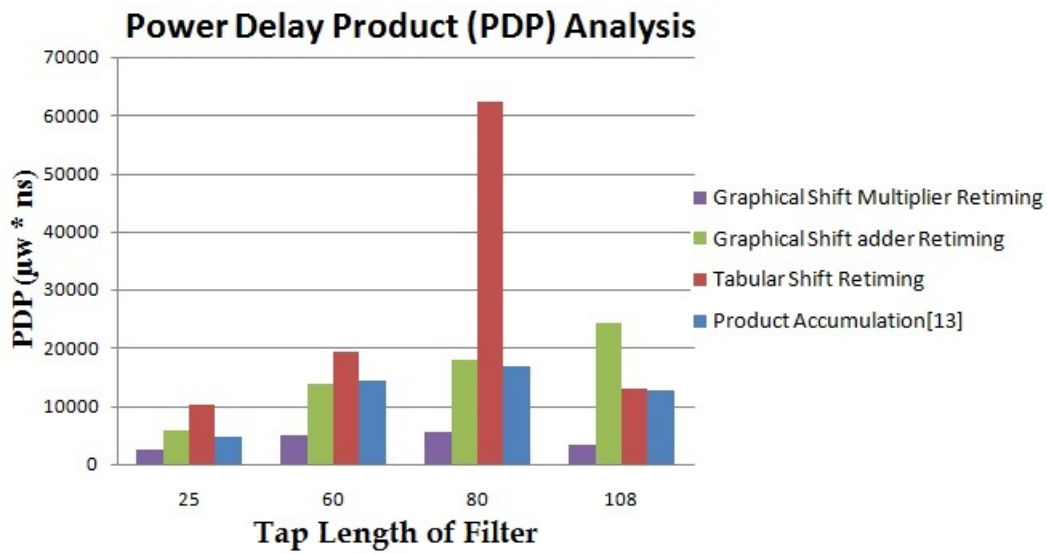


Figure 13. PDP analysis graph for different tap-lengths.

Table 1. Retime filter synthesis result for L=16 and N=25.

Method	Product Accu. [13]	Tabular Shift Retiming	Graphical Shift adder Retiming	Graphical Shift Mult. Retiming
Power (μW)	1792.6	2706.204	1548.03	699.983
area (μm^2)	14028	41564	37501	8824
CPD (ns)	2.65	3.72	3.67	3.74
ADP ($\mu\text{m}^2 \cdot \text{ns}$)	37174.2	154618.08	137628.67	33001.76
PDP ($\mu\text{W} \cdot \text{ns}$)	4750.3	10072.47	5681.16	2617.93

Table 2. Retime Filter synthesis result for L=16 and N=60.

Method	Product Accu. [13]	Tabular Shift Retiming	Graphical Shift adder Retiming	Graphical Shift Mult. Retiming
Power (μW)	4584.4	5170.572	3722.43	1354.005
area (μm^2)	35099	78289	89608	17091
CPD (ns)	3.12	3.72	3.67	3.74
ADP ($\mu\text{m}^2*\text{ns}$)	109508.9	244418.258	328861.36	63920.34
PDP ($\mu\text{W}*\text{ns}$)	14303.3	19244.868	13683.65	5063.96

Table 3. Retime Filter synthesis result for L=16 and N=80.

Method	Product Accu. [13]	Tabular Shift Retiming	Graphical Shift adder Retiming	Graphical Shift Mult. Retiming
Power (μW)	5952.4	3981.29	4862.595	1450.421
area (μm^2)	45518	6874	119608	18366
CPD (ns)	2.81	3.72	6.32	3.74
ADP ($\mu\text{m}^2*\text{ns}$)	127905.6	25585.028	439679.008	68688.84
PDP ($\mu\text{W}*\text{ns}$)	16726.2	62254.172	17874.899	5424.5708

Table 4. Retime Filter synthesis result for L=16 and N=108.

Method	Product Accu. [13]	Tabular Shift Retiming	Graphical Shift adder Retiming	Graphical Shift Mult. Retiming
Power (μW)	5717.3	3511.26	6608.149	911.377
area (μm^2)	43867	53615	161828	11699
CPD (ns)	2.2	3.72	6.32	3.74
ADP ($\mu\text{m}^2*\text{ns}$)	96507.4	199555.03	593908.76	43754.26
PDP ($\mu\text{W}*\text{ns}$)	12578.1	13068.91	24291.55	3408.55

Table 5. Retime Filter synthesis result for L=16 and N=128.

Method	Retiming with S-M [11]	Tabular Shift Retiming	Graphical Shift adder Retiming	Graphical Shift Mult. Retiming
Power (μW)	-	4597.17	8036.41	1194.44
CPD (ns)	2.3	3.72	3.67	3.74
area (μm^2)	389971.8	70308	191672	15294
ADP ($\mu\text{m}^2*\text{ns}$)	896935.1	261545.76	704586.27	57199.56
PDP ($\mu\text{W}*\text{ns}$)	-	17101.47	29493.62	57199.56

7. Conclusion

In this proposed design, a different method of retiming transformation is applied for digital design. The technique described in this paper is experimented on a FIR filter. Suitable projection of retiming transformation is applied to feed forward cut set to analyze the performance of area-delay-power and to reduce the computational complexity of the existing design. Significant reduction in dynamic power is obtained compared to conventional retiming transformation. Synthesis result of proposed design has achieved power saving of 84% compared to product accumulation method. The proposed retiming transformations have trade off in terms of area delay and power consumption with the requirement of the different application environment.

Conflict of interest

The authors declare no conflicts of interest in this paper.

References

1. Mittal A, Nandi A and Yadav D (2017) Comparative study of 16-order FIR filter design using different multiplication techniques. *IET Circ Device Syst* 11: 196–200.
2. Rashidi B (2013) High performance and low-power finite impulse response filter based on ring topology with modified retiming serial multiplier on FPGA. *IET Signal Process* 7: 743–753.
3. Mohanty BK and Meher PK (2016) A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications. *IEEE Transactions on Very Large Scale Integration Systems* 24: 444–452.
4. Leiserson CE, Rose FM and Saxe JB (1983) Optimizing synchronous circuitry by retiming. *3rd Caltech conference on VLSI*, pp. 87–116, Springer, Berlin, Heidelberg.
5. Leiserson CE and Saxe (1991) Retiming synchronous circuitry. *Algorithmica* 6: 5–35.
6. Chen JJ, Chip-Hong C, Feng F, et al. (2015) Novel Design Algorithm for Low Complexity Programmable FIR Filters Based on Extended Double Base Number System. *IEEE T Circuits-I* 62: 224–233.
7. Monteiro JC, Devadas S and Ghosh A (1993) Retiming sequential circuits for low power. *Proceedings of the IEEE/ACM International conference on Computer-Aided Design*, 398–402.
8. Parhi KK (2007) VLSI digital signal processing systems: design and implementation. *John Wiley and Sons*.
9. Aksoy L, Flores PF and Monteiro JC (2014) Efficient Design of FIR Filters Using Hybrid Multiple Constant Multiplications on FPGA. *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, 42–47.
10. Meidani M and Mashoufi B (2016) Introducing new algorithms for realising an FIR filter with less hardware in order to eliminate power line interference from the ECG signal. *IET Signal Process* 10: 709–716.

11. Meher PK (2016) On Efficient Retiming of Fixed Point Circuits. *IEEE Transactions on Very Large Scale Integration Systems* 24: 1257–1265.
12. Park SY and Meher PK (2014) Efficient FPGA and ASIC Realizations of a DA-Based Reconfigurable FIR Digital Filter. *IEEE T Circuits-II* 61: 511–515.
13. Lou X, Yu YJ and Meher PK (2016) Analysis and Optimization of Product-Accumulation Section for Efficient Implementation of FIR Filters. *IEEE T Circuits-II* 63: 1701–1713.
14. Kang Y, Kim J and Kang S (2016) Novel Approximate Synthesis Flow for Energy-efficient FIR Filter. *2016 IEEE 34th International Conference on Computer Design (ICCD)*, 96–102.
15. Pan Y and Meher PK (2014) Bit-Level Optimization of Adder-Trees for Multiple Constant Multiplications for Efficient FIR Filter Implementation. *IEEE T Circuits-I* 61: 455–462.



AIMS Press

©2018 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)