

Memory, Area and Power Optimization of Digital Circuits

Laxmi Gupta

Electronics and Communication Department
Jaypee Institute of Information Technology
Noida, Uttar Pradesh, India

Ankita Bharti

Electronics and communication Department
Jaypee Institute of Information Technology
Noida, Uttar Pradesh, India

Abstract- In this paper we illustrate the retiming technique to reduce the iteration period and to minimize the registers. So by this technique computation time is reduced for processors and fast speed is achieved. This is used in real time implementation to optimize performance. Shortest path algorithm such as Bellman ford and Floyd Warshall are used in retiming. These retiming techniques are explained in quantitative manner and the results are given thereafter. Retiming helps in reducing switching time. Minimization of the registers through retiming can help in reducing memory requirement and also reduce the requirement of area. And thereafter power consumption is also reduced due to less area and less switching time as given by the equation of dynamic power.

Keywords- Bellman Ford, constraint graph, floyd warshall, feasibility constraint, IIR, Linear programming.

I. INTRODUCTION

This paper illustrates retiming of a 3 tap filter. Retiming is a technique where the structural location of latches or registers in a digital circuit moves to improve its performance such as area and power characteristics in such a way that it preserves its functional behavior at its output.

The technique uses a direct flow graph where each vertex represents asynchronous combinational blocks and edges represents a series of registers or latches (the number of registers or latches can be zero). Each vertex has a value corresponding to the delay through the combinational circuit. Now after doing this we can optimize the circuit by pushing registers from output to input and vice versa. Two operations can be used:

(1) Deleting a register from each input of a

vertex while adding a register to all outputs (2) Conversely adding a register to each input of vertex and deleting a register from all outputs. In each case if all the rules are followed then the circuit will have same functional behavior as it did before this retiming technique. Retiming is done through Linear programming technique using branch and bound algorithms such as Bellman Ford and Floyd Warshall. But Bellman Ford algorithm is used in this paper. This technique is applied after converting the tap filter into a DFG graph. This shows results in area power and time minimization of VLSI circuits consisting of millions of gates.

The initial step of formulating the retiming problem as described by Leiserson and Saxe is as follows[3]. Given a direct flow graph $G = (v,e)$ whose each vertices represent logic gates or combinational delay elements present in a circuit, assume there is a direct edge $e = (u,v)$ between two elements that are connected directly or by one or more registers. Let us suppose the weight of each edges $w(e)$ is the number of registers present along the edge in the initial circuit which is to be given. Let us consider the $p(v)$ propagation delay through each vertex is v . The aim in retiming is to compute an integer lag value $l(v)$ for each vertex such that the retimed weight of the circuit is $w_r(e) = w(e) + l(v) - l(u)$ of every edge and this update weight is non-negative. This shows that this preserves the output functionality of the circuit. The most common use of retiming technique is to minimize the clock period. A simple technique to optimize the clock period is to search for the minimum feasible period. The feasibility of clock period T can be checked in several ways. The linear program given below is only feasible if and only if clock period T is a feasible clock period [1] [2]. Let us consider $w(u,v)$ is the minimum number of registers along any path from u to v with $w(u,v)$ register. The limitations of this approach arise due to size of the W and D matrices. Xilinx ISE software is generally used for Register Retiming.

Given $w(e)$, $w(u,v)$, $D(u,v)$ and a target clock period T

$$r(v) : V \rightarrow Z \quad (1)$$

$$r(u) - r(v) \leq w(e) \quad (2)$$

$$r(u) - r(v) \leq w(u,v) - 1$$

$$\text{if } D(u,v) > c$$

II. PROPOSED IIR FILTER

The proposed IIR filter represents a 3 tap filter which is shown in fig.1. This IIR Filter consist some adder and multiplier both. Each adder and multiplier has some computational time. For adder the computational time is $1u.t$ and for multiplier it is $2u.t$. For retiming of this IIR filter we have to firstly design its DFG. A DFG of this IIR filter is shown in fig.2[3]. This contains nodes and edges. Each node is made of either multiplier or adder and contains some computational time and each edge consists of delay which represents its registers. For an instance consider the proposed 3 tap IIR.

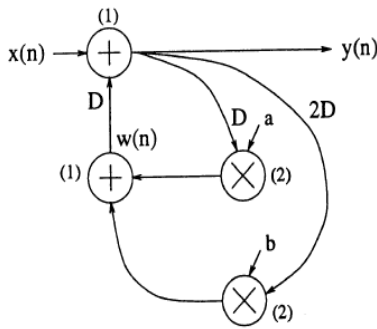


Fig.1 Proposed 3-Tap IIR Filter

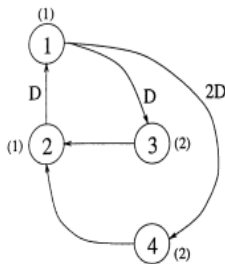


Fig.2 DFG(direct flow graph) of proposed Filter

Retiming can have many applications such as reducing computational time of critical path, reducing number of registers and power optimization. Algorithms used to reduce critical path of graphs by and to reduce registers required is given in following sections.

III. RETIMING

A. Calculation of Matrices

Retiming is done a original graph to obtain a retimed graph. shown in fig (3). $w(e)$ is the weight of the edges of original graph representing delays between two nodes in the filter Retiming formulation is based on this equation.

$$w_r(e) = w(e) + r(v) - r(u) \quad (3)$$

Minimum feasible clock period which is the time of longest path between input and output with no delays is given by-

$$\Phi(G) = \max \{t_p, w_p=0\} \quad (4)$$

$W(U,V)$ and $D(U,V)$ are calculated in the algorithm:

In first step we calculate $M = T_{max} \times n$ where T_{max} is largest computational time of a node in the graph and n is number of nodes. In the new graph weights are replaces using following equation-

$$w'(e) = Mw(e) - t(u) \quad (5)$$

In the initial step of formulating retiming of IIR filter we consider $M=8$ as $\max=2$ and $n=4$. The new updated weight

DFG of the proposed IIR filter is shown below in Fig.3.

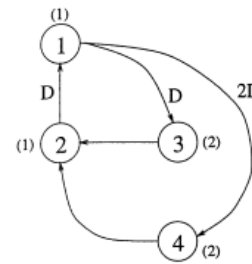


Fig.3 DFG of G' (graph used to find W and D matrices)

In second step S matrix is given below is made by calculating shortest path by Floyd Warshall Algorithm. Nodes are represented by matrix indices.

$$S'_{UV} = \begin{bmatrix} 12 & 5 & 7 & 15 \\ 7 & 12 & 14 & 22 \\ 5 & -2 & 12 & 20 \\ 5 & -2 & 12 & 20 \end{bmatrix}$$

If $U \neq V$ then

$$W(U,V) = \left\lfloor \frac{S'_{UV}}{M} \right\rfloor \quad (6)$$

If $U = V$ then $W(U,V) = 0$. $W(U,V)$ matrix is given by :-

$$W(U,V) = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 3 \\ 1 & 0 & 0 & 3 \\ 1 & 0 & 2 & 0 \end{bmatrix}$$

After this $D(U,V)$ is calculated by given equation.

$$D(U,V) = MW(U,V) - S'_{UV} + t(v) \quad (7)$$

$$D(U,V) = \begin{bmatrix} 1 & 4 & 3 & 3 \\ 2 & 1 & 4 & 4 \\ 4 & 3 & 2 & 6 \\ 4 & 3 & 6 & 2 \end{bmatrix}$$

A. OPTIMIZING THE CLOCK PERIOD AND DRAWING CONSTRAINT GRAPH.WITH RETIMED DFG

The values are obtained to test whether any retiming solution exists or not with a clock period as desired. Given clock period C there is a retiming solution such that $\Phi(G_r) \leq C$ if the constraints hold which are:

1. Feasibility constraint : $r(u) - r(v) \leq w$
2. Critical path constraint : $r(u) - r(v) \leq W(U,V) - 1$

Feasibility constraint ensures number of delays in the path to be non- negative and critical path constraint ensures that $\Phi_G \leq C$. These equations are formed and a feasible solution $C=3$ is chosen. The constrained graph for $C= 3$ is given below in fig.4

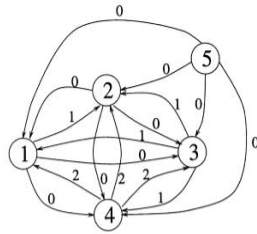
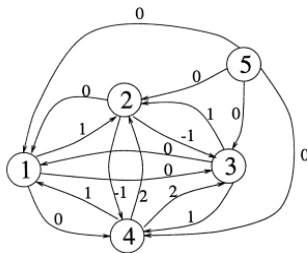


Fig.4 Constraint graph for C=3

But C=3 is originally given for the graph so we can keep our clock period less than equal to 3. So we choose 2. Again inequalities are formed. Constraint graph for 12 inequalities are formed and by using Bellman Ford and Floyd Warshall algorithms we can determine that following graph has no negative cycles. Considering node 5 as source node retiming solution is obtained by Bellman Ford and Floyd Warshall. The constraint graph for C = 2 is shown below in fig.5



Fig(5) constraint graph for c=2

IV. RETIMING FOR REGISTER MINIMIZATION

A. Introduction

This section explains the algorithm for minimizing the number of registers by retiming the and also for optimization of cost constraint. For any node:-

$$R_v = \max_{v \rightarrow u} \{w_e\} \quad (8)$$

This is maximum requirement for a node V and minimum cost of registers is given by

$$COST = \sum R_v \quad (9)$$

B. Constraints :

$\sum R_v$ is subject to three constraints which are given below:

- I. Fan out constraint: $R_v \geq w_r(e)$ for all edges and nodes v.
- II. Feasibility constraint $r(U) - r(V) \leq w_e$ for every edges.
- III. Clock period constraint $r(U) - r(V) \leq W(U,V) - 1$

This type of solution can be done through LP techniques but we introduce a gadget. This gadget represents a FANOUT node. This node is called a Dummy Node with computation time $t(u)=0$. Dummy edges are introduced with weights of

edges $w(e_i) = w_{max} - w(e_i)$. w_{max} is maximum weight of all edges. Each edges in this DFG is associated with a breadth k. T is breadth is nothing but a number used so that the device properly models the memory required by the edges $e_i, 1 \leq i \leq m$. The breadth of each edges is $\frac{1}{m}$.

IV. RESULTS

The retimed graph has $\Phi(G_r)=2$ by redistributing the delays through retiming algorithm as shown in fig(6). As shown in fig.(6) that it reduces the critical path also. As critical path is reduce than it also reduce the dynamic power by the formula $(\alpha.f.C.Vdd^2)$. Since in VLSI industries the circuits consist- ing of millions of gates hence it is the big task to reduce the area and power. Thus by this technique we can reduce the area and power and make it feasible.

Original graph has requirement of 11 registers but retimed DFG has only 7 registers requirement as shown in fig (7). By the reduction in register through this technique we can also minimize memory and saving the cost and also make the system faster. Hence it became economica-lly feasible.

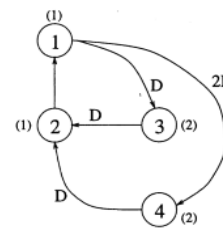


Fig (6) Retimed graph of proposed IIR Filter

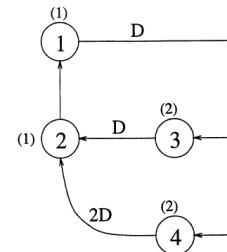


Fig (7) Minimization of registers

V. CONCLUSION

The proposed work is used to reduce the critical path, clock period, no of storage, delay elements and power consumption of synchronous as well as combinational circuit (digital circuits)[1]. To obtain retiming we used shortest path algorithm. Retiming technique can also be used as a preprocessing step for folding and also for computation of round off noise. All possible retiming solution can also be obtained from exhaustive retiming approaches.

VI. ACKNOWLEDGEMENT

The authors are grateful to Jaypee Institute Of Information Technology for providing resources at the department of Electronic and Communication Engineering to enable them to carry out this research work.

REFERENCES

- [1] S. Jalaja and A. M. V. Prakash, " Design of low power based VLSI architecture for constant multiplier and high speed implementation using retiming technique," 2016 International Conference on Microelectronics, Computing and communication (*microcom*), Durgapur, 2016, pp. 1-6.
- [2] G. Venkataramani and Y. Gu, "System-Level Retiming and Pipelining," 2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines, Boston, MA, 2014, pp. 80-87.
- [3] PARHI, K. K. (1999). *VLSI digital signal processing systems*. Canada John Wiley.