AVLSI DSP DESIGN AND IMPLEMENTATION OF ALL POLE LATTICE FILTER USING RETIMING METHODOLOGY

[1]PURU GUPTA & [2]TARUN KUMAR RAWAT

Dept. of Electronics and Communication Engineering,
Netaji Subhas Institute of Technology -Dwarka Sector 3, New Delhi, India
Email: purugupta5000@gmail.com, tarundsp@gmail.com

Abstract: All pole lattice filters are used in a variety of signal processing applications that is speech processing, adaptive filters and various other applications. The implementation of lattice filter requires more clock period hence low speed. There are various transformation technique present for design of high-speed or low-area or low-power implementations. This paper presents design of high-speed (smaller clock period) implementation of 8th order all pole lattice filter using the methodology named as Retiming. Retiming reduces the clock period of the circuit, reducing the number of registers in the circuit, reducing the power consumption of the circuit. Therefore, retiming has been used to reduce the clock period of all pole lattice filters and it increases the speed of the system.

Keywords: *Data Flow Graph, Iteration, Critical Path.*

1. INTRODUCTION

Digital signal processing (DSP) is used in numerous applications such as video compression, digital set-top box, cable modems, digital versatile disk, wireless communications, digital radio, digital still and speech processing, radar imaging etc. The field of DSP has always been driven by the advances in DSP applications and in scaled very-large-scale-integrated (VLSI) technologies. Therefore at any given time, DSP applications impose several challenges on the implementations of the DSP systems. These implementations must satisfy the enforced sampling rate constraints of the real-time DSP applications and must require less space and power consumption [3].There are various methodologies needed to design custom or semi-custom VLSI circuits for these applications. DSP computation is different from general-purpose computation. In DSP computation, the same program is executed repetitively on an infinite time series. The non-terminating nature can be exploited to design more efficient DSP systems by exploiting the dependency of tasks both within an iteration and among multiple iterations [1]. Furthermore, long critical paths [1][3] in DSP algorithms limit the performance of DSP systems. These algorithms need to be transformed for design of high-speed or low-area or low-power implementations. DSP Algorithms are used in various real-time applications with different sampling rate requirements that can vary from about 20KHZ in speech applications to over 500 MHz in radar and HD television applications. The computation requirement of a video compression system for HDTV can range from 10 to 100 giga operations per second. The dramatically different sample rate and computation requirements necessitate different architecture considerations for implementations of DSP algorithms [3]. There are several high-level architectural transformations mentioned as follows [3]: Pipelining and Parallel processing, Retiming, Un-Folding, Folding, Systolic array design methodology.

In this paper we have presented retiming technique to implement 8^{th} order all pole lattice filter with higher speed. Section II, III presents about the retiming and its algorithm. Section IV and V presents the all pole lattice filter and implementation of it using retiming. Section VI and VII presents the results.

2. RETIMING

Retiming [3][4][6] is a transformation technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit. For example, consider Data Flow Graph (DFG) [1] [3] of IIR filter in Fig. 4. This filter is described by:

$$w(n) = ay(n-1) + by(n-2) \qquad (1)$$

$$y(n) = ay(n-2) + by(n-3) + x(n) \qquad (2)$$

The retimed filter in Fig. 5 is described by:

$$w_1(n) = ay(n\text{-}1) \text{ and } w_2(n) = by(n\text{-}2) \qquad (3)$$

$$y(n) = ay(n-2) + by(n-3) + x(n) \qquad (4)$$

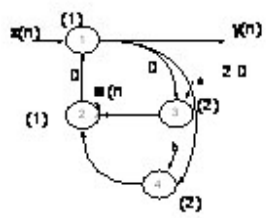

Fig. 4: DFG of an IIR filter

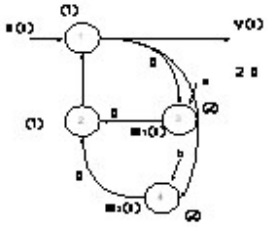

Fig. 5: Retimed DFG of an IIR filter

It is clear from Fig. 4, Fig. 5 and equation (1) to (4) that retiming sustain the input/output relationship. Retiming maps a circuit G to retimed circuit G_r . A retiming solution is characterized by a value $r(V)$ for each node V in the graph. Let $w_r(e)$ denote the weight of the edge e in the retimed graph G_r. The weight of the edge $U \rightarrow V$ in the retimed graph is computed from the weight of the edge in the original graph using: $w_r(e) = w(e) + r(V) - r(U)$. The retiming values r(1) = 0, r(2) = 1, r(3) = 0, and r(4) = 0 can be used to obtain the retimed DFG shown in Fig. 5 from the original DFG shown in Fig. 4. The critical path in Fig. 4 is 3 u.t. for path $3 \rightarrow 1$ and in Fig. 5 the critical path has been reduced to 2 u.t. for path $2 \rightarrow 1$. There is an algorithm to find retiming solution $r(V)$ explained in detail in section III. Retiming is done to meet the following: (i) Clock period minimization, (ii) Register minimization. In this paper we are performing clock period minimization of 8th Order All Pole Lattice Filter using retiming.

3. ALGORITHM FOR RETIMING

In this section, an algorithm for Retiming a DFG is described [3][4]. For a circuit G, the minimum feasible clock period is the computation time of the critical path, which is the path with the longest computation time among all paths with no delays. Mathematically, the minimum feasible clock period, $\Phi(G)$, is defined as $\Phi(G) = max\{t(p) : w(p) = 0\}$, Where $t(p)$ is the computation time of the corresponding path p which has delay i.e. w(p) equal to zero. The algorithm presented in this section finds a retiming solution r_0 such that $\Phi(G_0) \Leftarrow \Phi(G_r)$ for any other retiming solution r.The two quantities $W(U, V)$ and $D(U, V)$ are used in this algorithm. The quantity $W(U, V)$ is the minimum number of registers on any path from node U to node V and $D(U, V)$ is the maximum computation time among all paths from U to V with weight $W(U, V)$. Formally,

$$W(U, V) = \min\{w(p) : U \overrightarrow{p} V\} \quad (5)$$

$$D(U, V) = \max\{t(p) : U \overrightarrow{p} V \text{ and } w(p) = W(U, V)\} \quad (6)$$

3.1 Calculation of S(U, V), W (U, V) and D(U, V)

The Following steps can be used to compute S(U, V), $W(U, V)$ and $D(U, V)$:

1. Let $M = t_{max}$, where t_{max} is the maximum computation time of the nodes in G and n is the number of nodes in G.
2. Form a new graph G' which is the same as G except the edge weights are replaced by $w'(e) = M\, w(e) - t(U)$ for all edges $U \rightarrow V$.
3. Solve for all pair shortest path problem on G' by using Floyd Warshall algorithm [3]. Let $S'(U,V)$ be the shortest path form $U \rightarrow V$.
4. If $U = V$, then $W(U,V) = \lceil S(U, V)/M \rceil$ and $D(U, V) = M\, W(U, V) - SU\, V + t(V)$. If $U = V$, then $W(U, V) = 0$ and $D(U, V) = t(U)$, where operation $\lceil x \rceil$ is the floor of x, which is the smallest integer greater than or equal to x.

3.2 Obtaining inequalities

The values of $W(U,V)$ and $D(U,V)$ are used to determine if there is a retiming solution that can achieve a desired clock period. Given a desired clock period c, there is a feasible retiming solution r such that $\Phi(G_r) \leq c$ if the following constraints hold:(i) Feasibility constraint: $r(U) - r(V) \leq w(e)$ for every edge $U \overrightarrow{e} V$ of G, (ii) Critical path constraint: $r(U) - r(V) \leq W(U, V) - 1$ for all the vertices U,V in G such that $D(U, V) > c$.

The constraints above give the set of M inequalities in N variables, where each inequality has the form $r_i - r_j \leq k$ for integer values of k, one of the shortest path exists and to find a solution if one does indeed exist. This is done using separate procedure described in next sub-section.

3.3 Solving systems of inequalities

5. Steps to draw a constraint graph:

- Draw the node i for each of the N variables $r_i, i = 1, 2, \cdots, N$.
- Draw the node N + 1.
- For each inequality $r_i - r_j \leq k$, draw the edge $j \rightarrow i$ from node j to node i with length k.

- For each node i, i = 1,2,..,n, draw the edge $N + 1 \rightarrow i$ from the node N + 1 to the node i with length 0.

6. Steps to Solve constraint graph using a shortest path algorithm.

- Solve for all pair shortest path problem on *G* by using Floyd Warshall algorithm [3].
- The system of inequalities has a solution if and only if the constraint graph contains no negative cycles. if a solution exists, one solution is where r_i is the minimum-length path from the node N + 1 to the node i.

The solution gives the values of retiming values and hence the weight of the edge $U \rightarrow V$ in the retimed graph is computed from the weight of the edge in the original graph using:

$$w_r(e) = w(e) + r(V) - r(U) \qquad (7)$$

4. LATTICE FILTER

Lattice filters [8][9] are used in a variety of signal processing applications. These includes: (i) The lattice filter is extensively used in digital speech processing and in implementation of adaptive filter [10], (ii) The lattice filter has an important application on lines used by broadcasters for stereo audio feeds. Lattice filter has an intrinsically balanced topology. This is useful when used with landlines which invariably use a balanced format. Many other types of filter section are intrinsically unbalanced and have to be transformed into a balanced implementation in these applications which increases the component count. This is not required in the case of lattice filters. It is a preferred form of realization over other FIR or IIR filter structures because in speech analysis and in speech synthesis the small number of coefficients allows a large number of formants to be modeled in real-time. Lattice filters exists in three forms: (i) All-Zeros lattice is the FIR filter representation of the lattice filter, (ii) All-Pole lattice filter is the IIR filter representation, (iii) Pole-Zero lattice ladder filter. In this paper we are implementing all pole lattice filter of 8th order using retiming in which signals are delayed before they are added to the input. Comb filters may be implemented in discrete time or continuous time.

4.1 All Pole Lattice Filter

A lattice structure for an IIR filter is restricted to an all-pole system function. Similarly, the general structure of a All-Pole lattice filter of order N has a lattice structure with N stages as shown in shown in Fig. 6.

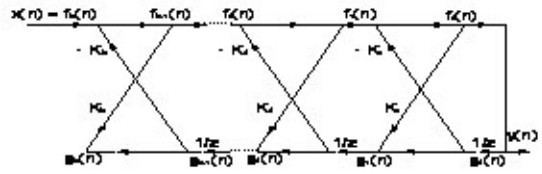

Figure 6. All Pole Lattice Filter of order N

$$f_N(n) = x(n) \qquad (8)$$

$$f_{m-1}(n) = f_m(n) - K_m g_{m-1}(n-1), \quad m = N, N-1, \cdots (9)$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1), \quad m = N, N-1, \cdots, 1 \ (10)$$

$$y(n) = f_0(n) = g_0(n) \qquad (11)$$

5. IMPLEMENTATION OF ALL POLE LATTICE FILTER USING RETIMING

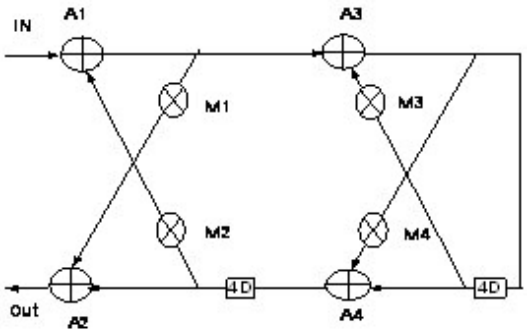

Figure 7. 8-th order All Pole Lattice Filter.

All-Pole lattice filter is shown in the Fig. 7. The DFG for the same is given in Fig. 8.

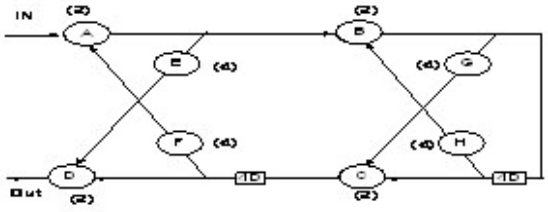

Figure 8. . DFG of 8-th order All Pole Lattice Filter

For the 8th order 4 stage pipelined all-pole lattice filter as shown in Fig. 9, the addition and multiplication require 2 and 4 u.t., respectively [3, Page 143, Problem No 11]. The Critical path [1][3] ($F \rightarrow A \rightarrow B \rightarrow G \rightarrow C$) of the all pole lattice filter shown in Fig. 9 is 14 u.t. Hence, the clock period is 14 u.t. because critical path represents clock period of any system. This is quite high and slows down the system. So, the filter has to be transformed to attain the minimum clock period possible. We cann ot chose clock period less than 4 u.t. because as the per the DFG shown in Fig. 9 of all pole lattice filter have maximum node execution

time as 4 u.t. and clock period cannot be less than it. So in this paper reduction of the clock period till 4 u.t. has been shown. The technique used here is retiming which allows the DSP program to be implemented with reduced clock period. As per the retiming algorithm following are the steps shown to find the retiming solution:

7. tmax for the DFG of all pole lattice filter is 4 u.t. and the number of nodes n = 8, So M = 4 ∗ 8= 32.

8. Form a new graph G' shown in Fig. 9. which is the same as G except the edge weights are replaced by w'(e) = M w(e) – t(U) for all edges U→ V. Calculation of w'(e) for all paths of DFG:

(i) $A \rightarrow B\ i.e.\ w(e) = 32 * 0 - 2 = -2.$

(ii) $B \rightarrow H\ i.e.\ w(e) = 32 * 4 - 2 = 126.$

(iii) $B \rightarrow C\ i.e.\ w(e) = 32 * 4 - 2 = 126.$

(iv) $B \rightarrow G\ i.e.\ w(e) = 32 * 0 - 2 = -2.$

(v) $G \rightarrow C\ i.e.\ w(e) = 32 * 0 - 4 = -4.$

(vi) $H \rightarrow B\ i.e.\ w(e) = 32 * 0 - 4 = -4.$

(vii) $C \rightarrow D\ i.e.\ w(e) = 32 * 4 - 2 = 126.$

(viii) $C \rightarrow F\ i.e.\ w(e) = 32 * 4 - 2 = 126.$

(ix) $A \rightarrow E\ i.e.\ w(e) = 32 * 0 - 2 = -2.$

(x) $E \rightarrow D\ i.e.\ w(e) = 32 * 0 - 4 = -4.$

(xi) $E \rightarrow D\ i.e.\ w(e) = 32 * 0 - 4 = -4.$

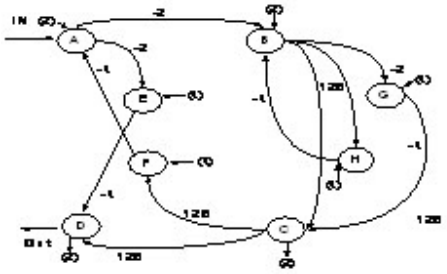

Figure 9. Modified Graph of DFG in fig. 9.

9. Solve for all pair shortest path problem on G' by using Floyd Warshall algorithm [3]. Let S'(U,V) be the shortest path form $U \overrightarrow{e} V$, also solve W(U,V) and D(U,V) as per the algorithm. The tables are given as follows:

S'(U,V)	A	B	C	D	E	F	G	H
A	114	-2	-8	-6	-2	118	-4	24
B	116	114	-6	120	114	120	-2	126
C	122	120	114	126	120	126	118	246
D	∞	∞	∞	∞	∞	∞	∞	∞
E	∞	∞	∞	-4	∞	∞	∞	∞
F	-4	-6	-12	-10	-6	114	-8	120
G	118	116	-4	122	116	122	114	242
H	112	-4	-10	116	110	116	-6	122

Table I: Values of S' (U, V) , W(U,V) and D(U,V) for the Data Flow Graphs of All-Pole Lattice Filters

W(U,V)	A	B	C	D	E	F	G	H
A	0	0	0	0	0	4	0	4
B	4	0	0	4	4	4	0	4
C	4	4	0	4	4	4	4	8
D	∞	∞	∞	0	∞	∞	∞	∞
E	∞	∞	∞	0	0	∞	∞	∞
F	0	0	0	0	0	0	0	4
G	4	4	0	4	4	4	0	8
H	4	0	0	4	4	4	0	4

	A	B	C	D	E	F	G	H
A	2	4	10	8	6	14	8	8
B	14	2	8	10	18	12	6	6
C	8	10	2	4	12	6	14	14
D	∞	∞	∞	2	∞	∞	∞	∞
E	∞	∞	∞	6	4	∞	∞	∞
F	6	8	14	12	10	4	12	12
G	12	14	6	8	16	10	4	14
H	18	6	12	14	22	16	10	4

4. The minimum feasible clock period *c* can be chosen as 4 u.t. We cannot chose clock period less than 4 because as the per the DFG of all pole lattice filter the maximum node execution time is 4 unit and clock period cannot be less than it. For DFG in Fig. 1, if *c* chosen to be 4, the inequalities $r(U) - r(V) \le w(e)$ for every edge $U \overrightarrow{e} V$ are: $r(A) - r(B) \le 0$, $r(A) - r(E) \le 0$, $r(B) - r(C) \le 4$, $r(B) - r(G) \le 0$, $r(B) - r(H) \le 4$, $r(C) - r(D) \le 4$, $r(C) - r(F) \le 4$, $r(E) - r(D) \le 0$, $r(F) - r(A) \le 0$, $r(G) - r(C) \le 0$, $r(H) - r(B) \le 0$.

And the inequalities $r(U) - r(V) \le W(U, V) - 1$ for all the vertices *U, V* in G such that $D(U, V) > 4$ *are :*

$r(A) - r(C) \le -1$, $r(A) - r(D) \le -1$, $r(A) - r(E) \le -1$, $r(A) - r(F) \le 3$, $r(A) - r(G) \le 3$, $r(A) - r(H) \le -1$,

$r(B) - r(A) \le 3$, $r(B) - r(C) \le 3$, $r(B) - r(D) \le 3$, $r(B) - r(E) \le 3$, $r(B) - r(F) \le -1$, $r(B) - r(G) \le 3$, $r(B) - r(H) \le 3$,

$r(C) - r(A) \le 3$, $r(C) - r(B) \le 3$, $r(C) - r(E) \le 3$, $r(C) - r(F) \le 7$, $r(C) - r(G) \le 3$, $r(C) - r(H) \le 7$,

$r(D) - r(A) \leq \infty$, $r(D) - r(B) \leq \infty$, $r(D) - r(C) \leq \infty$, $r(D) - r(E) \leq \infty$, $r(D) - r(F) \leq \infty$, $r(D) - r(G) \leq \infty$, $r(D) - r(H) \leq \infty$,

$r(E) - r(A) \leq \infty$, $r(E) - r(B) \leq \infty$, $r(E) - r(C) \leq \infty$, $r(E) - r(D) \leq -1$, $r(E) - r(F) \leq \infty$, $r(E) - r(G) \leq \infty$, $r(E) - r(H) \leq \infty$,

$r(F) - r(A) \leq -1$, $r(F) - r(B) \leq -1$, $r(F) - r(C) \leq -1$, $r(F) - r(D) \leq -1$, $r(F) - r(E) \leq -1$, $r(F) - r(G) \leq -1$, $r(F) - r(H) \leq 3$,

$r(G) - r(A) \leq 3$, $r(G) - r(B) \leq 3$, $r(G) - r(C) \leq -1$, $r(G) - r(D) \leq 3$, $r(G) - r(E) \leq 3$, $r(G) - r(F) \leq 3$, $r(G) - r(H) \leq 7$,

$r(H) - r(A) \leq 3$, $r(H) - r(B) \leq -1$, $r(H) - r(C) \leq -1$, $r(H) - r(D) \leq 3$, $r(H) - r(E) \leq 3$, $r(H) - r(F) \leq 3$, $r(H) - r(G) \leq -1$.

Combining the equations in feasibility and critical path constraint section that have identical left-hand sides results in the set of equations:

$r(A) - r(B) \leq 0$, $r(A) - r(C) \leq -1$, $r(A) - r(D) \leq -1$, $r(A) - r(E) \leq -1$, $r(A) - r(F) \leq 3$, $r(A) - r(G) \leq 3$, $r(A) - r(H) \leq -1$,

$r(B) - r(A) \leq 3$, $r(B) - r(C) \leq 3$, $r(B) - r(D) \leq 3$, $r(B) - r(E) \leq 3$, $r(B) - r(F) \leq -1$, $r(B) - r(G) \leq 0$, $r(B) - r(H) \leq 3$,

$r(C) - r(A) \leq 3$, $r(C) - r(B) \leq 3$, $r(C) - r(D) \leq 4$, $r(C) - r(E) \leq 3$, $r(C) - r(F) \leq 4$, $r(C) - r(G) \leq 3$, $r(C) - r(H) \leq 7$,

$r(D) - r(A) \leq \infty$, $r(D) - r(B) \leq \infty$, $r(D) - r(C) \leq \infty$, $r(D) - r(E) \leq \infty$, $r(D) - r(F) \leq \infty$, $r(D) - r(G) \leq \infty$, $r(D) - r(H) \leq \infty$,

$r(E) - r(A) \leq \infty$, $r(E) - r(B) \leq \infty$, $r(E) - r(C) \leq \infty$, $r(E) - r(D) \leq -1$, $r(E) - r(F) \leq \infty$, $r(E) - r(G) \leq \infty$, $r(E) - r(H) \leq \infty$,

$r(F) - r(A) \leq -1$, $r(F) - r(B) \leq -1$, $r(F) - r(C) \leq -1$, $r(F) - r(D) \leq -1$, $r(F) - r(E)$, $r(F) - r(G) \leq -1$, $r(F) - r(H) \leq 3$, $r(G) - r(A) \leq 3$, $r(G) - r(B) \leq 3$, $r(G) - r(C) \leq -1$, $r(G) - r(D) \leq 3$, $r(G) - r(E) \leq 3$, $r(G) - r(F) \leq 3$, $r(G) - r(H) \leq 7$,

$r(H) - r(A) \leq 3$, $r(H) - r(B) \leq -1$, $r(H) - r(C) \leq -1$, $r(H) - r(D) \leq 3$, $r(H) - r(E) \leq 3$, $r(H) - r(F) \leq 3$, $r(H) - r(G) \leq -1$

5. As per the retiming algorithm constraint graph is drawn as shown in Fig. 10 with value of $N = 8$:

Figure 10. Constraint graph of 8th order All-Pole Lattice Filter.

Using shortest path algorithm following table produced on solving the constraint graph shown in Fig. 10:

Table II: Values of S'(U,V) for the Data Flow Graphs of All-Pole Lattice Filters

S'(U,V)	A	B	C	D	E	F	G	H	I
A	2	2	2	∞	∞	-1	2	2	∞
B	0	2	2	∞	∞	-1	2	-1	∞
C	-2	-2	2	∞	∞	-2	-1	-2	∞
D	-2	2	2	∞	-1	-2	2	2	∞
E	-1	2	2	∞	∞	-2	1	2	∞
F	-2	2	2	∞	∞	1	2	1	∞
G	-1	-1	2	∞	∞	-1	2	-2	∞
H	3	3	6	∞	∞	2	6	2	∞
I	-2	-2	0	0	-1	-3	-1	-3	∞

Using Floyd-Warshall algorithm, it can be determined that this graph contains no negative cycles, so the retiming solution for $c = 4$ is the solution to the single-source shortest path problem with the origin at the node I, i.e. $r(A) = -2, r(B) = -2, r(C) = 0, r(D) = 0, r(E) = -1, r(F) = -3, r(G) = -1, r(H) = -3$. These retiming values obtained are used to obtain weight of the edge $U \rightarrow V$ *in* the retimed graph using equation (7) as follows:

(i) $A \rightarrow B$ *i.e.* $w_r(e) = 0 + (-2) - (-2) = 0$.

(ii) $B \rightarrow H$ *i.e.* $w_r(e) = 4 + (-3) - (-2) = 3$.

(iii) $B \rightarrow C$ *i.e.* $w_r(e) = 4 + (0) - (-2) = 6$.

(iv) $B \rightarrow G$ *i.e.* $w_r(e) = 0 + (-1) - (-2) = 1$.

(v) $G \rightarrow C$ *i.e.* $w_r(e) = 0 + (0) - (-1) = 1$.

(vi) $H \rightarrow B$ *i.e.* $w_r(e) = 0 + (-2) - (-3) = 1.$

(vii) $C \rightarrow D$ *i.e.* $w_r(e) = 4 + (0) - (0) = 4.$

(viii) $C \rightarrow F$ *i.e.* $w_r(e) = 4 + (-3) - (0) = 1.$

(ix) $A \rightarrow E$ *i.e.* $w_r(e) = 0 + (-1) - (-2) = 1.$

(x) $E \rightarrow D$ *i.e.* $w_r(e) = 0 + (0) - (-1) = 1.$

(xi) $F \rightarrow A$ *i.e.* $w_r(e) = 0 + (-2) - (-3) = 1.$

The retimed DFG of 8the order All-Pole lattice filter shown in Fig. 11

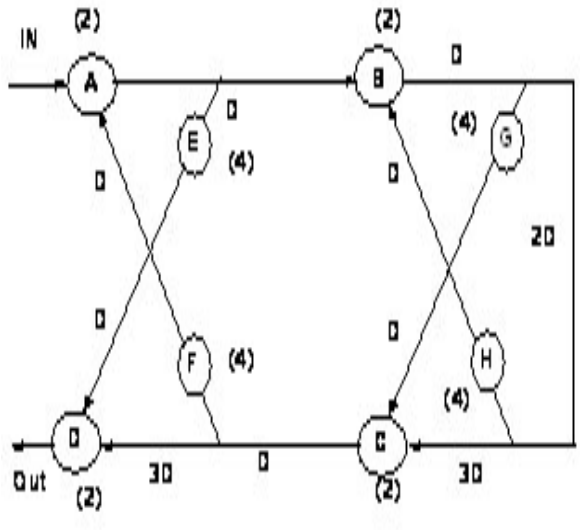

Figure 11. Retimed DFG of 8th order All-Pole Lattice Filter

6. RESULTS

In this paper we had implemented 8th order all-pole lattice filter using retiming technique. The DFG of the same is given in Fig. 9. It has four addition nodes having computation time of 2 u.t. and four multiplication nodes having computation time of 4 u.t. The Critical path ($F \rightarrow A \rightarrow B \rightarrow G \rightarrow C$) of the 8th order all-pole lattice filter is 14 u.t. Hence, the clock period is 14 u.t. Now the critical path and the clock period of 8th order all-pole lattice filter is reduced to 4 u.t. after applying retiming transformation. Critical path is now ($A \rightarrow B$) i.e. 4 u.t. as shown in Fig. 12. As compared to original DFG of all-pole lattice filter in Fig. 9 the retimed DFG in Fig. 12 has critical path reduced to 4 u.t.

7. CONCLUSIONS

We have presented 8th order all-pole lattice filter shown in Fig. 12 with reduced critical path of 4 u.t. Initially the 8-th order all-pole lattice had higher critical path of 14 u.t. and it was leading to very high clock period and less speed. Retiming methodology had be applied on 8-th order all-pole lattice filter to reduce its critical path to 4 u.t. and hence increasing the speed of the filter. Now the retimed filter require less clock period for its processing hence enhancing the speed of it. In this way retiming can be used in various applications of all-pole lattice filters of any order to achieve higher speed.

REFERENCES

[1]. P.Lapsley, J.Bier, A. Shoham, and E.A. Lee, DSP Processor Fundamentals: Architectures and Features, Elsevier, 1993.

[2]. K.K. Parhi and D.G. Messerschmitt, "Static rate-optimal scheduling of iteration data-flow programs via optimum unfolding," IEEE Trans. on Computers,vol. 40, no. 2, pp. 178-195, Feb. 1991

[3]. Keshab K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. Wiley, 2009

[4]. S. Simon, E. Bernard, M. Sauer, and J. Nossek, "A new retiming algorithm for circuit design," in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), (London, England), May 1994.

[5]. L.E. Lucke and K.K. Parhi, "Data-Flow transformations for critical path time reduction in high-level DSP synthesis," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,vol. 12, no. 7, pp. 1063-1068, July 1993.

[6]. L.-F Chao and E. Sha, "Retiming and Unfolding data-flow graphs," in Proc. of 1992 International Conference on Parallel Processing.part II, (St. Charles,IL), pp. 33-40, Aug. 1992.

[7]. K.K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," IEEE Trans. On Circuits and Systems, vol. 38, no. 4, pp. 358-375, April 1991.

[8]. Sanjit K. Mitra, Digital Signal Processing: A computer-Based Approach. McGraw-Hill, Second Edition.

[9]. John G. Proakis and Dimitris G. Manoakis, Digital Signal Processing: Principles, Algorithms, and Applica- tions.PHI, Third Edition.

[10]. Martinez J.I.M., Nakano K., "Cascade lattice IIR adaptive filter structure using simultaneous perturbation method for self-adjusting SHARF algorithm," in SICE Annual Conference, pp. 2156 - 2161,2008.

◈◈◈