ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of Reversible FIR Filter for Speech Signal Processing

Hajara Siddiqui, B.E
M.Tech (VlSI & Embedded Systems),
Adusumilli Vijaya Institute of Technology and Research Centre.

Syed Jilani Pasha, B.E, M.Tech, (Ph.D)
Assistant Professor,
Adusumilli Vijaya Institute of Technology and Research Centre.

Abstract:

Reversible computing is a model of computing where the computational process to some extent is reversible. A necessary condition for reversibility of a computational model is that the relation of the mapping states of transition functions to their successors should at all times be one-to-one. Reversible logic has emerged as an alternate design technique to the conventional logic, resulting in lower power consumption and lesser circuit area. In this paper, an efficient architecture of FIR filter structure is presented. For achieving low power, reversible logic mode of operation is implemented in the design. Area overhead is the tradeoff in the proposed design. From the synthesis results, the proposed low power FIR filter architecture offers power saving when compared to the conventional design. The area overhead is for the proposed architecture.

Keywords:

Reversible logic; FIR filter; Low power; Speech samples

I.INTRODUCTION:

Digital Signal Processing is used in wide range of applications such as radio, television, video etc. Its main basic tool Finite Impulse Response (FIR) digital filters. The filtering requires arithmetic operations. The adder and multiplier module consume much circuit area and power. The complexity of the filter is mainly because of the multiplication operation in FIR filter. For low power design input bit width of the module is quite important. The adders, Wallace, dadda multipliers are applied for filters to eliminate power consumption due to unwanted data transitions [1].

In [2] they presented a multipliers technique, based on add and shift method and common sub expression elimination for low area, low power and high-speed implementation of FIR filters.Finite impulse response filters are widely used in various DSP applications. The general FIR filter can be expressed as the following equation:

$$y(n) = \sum_{k=0}^{N-1} C_k x(n-k) \qquad (1)$$

Where N represents the length of the FIR filter, the kth coefficient and x(n-k) is the input data at time instant n-k. where M is the tap number of the FIR filter, ck are the filter coefficients, and x(n-k) is the input signals. The above equations can also expressed in Z domain as

$$Y(z) = x(z)\,H(z) \qquad (2)$$

Where H(z) is the transfer function of FIR filter in Z domain and is given by

$$H\,(z) = \sum_{k=0}^{L-1} h(k) z^{-1} \qquad (3)$$

An FIR filter of order N is characterized by N+1 coefficients and in general, require N+1 multipliers and N two-input adders [7]. Structures in which the multiplier coefficients are precisely the coefficients of the transfer function are called direct form structures. In digital signal processing, arithmetic operations like multipliers and adders play a major role. Multiplier occupies a larger area in the FIR Filter. The common multipliers used will be Wallace, dadda multipliers. The adders used for comparison are Ripple Carry Adder, Carry look ahead adder, Carry save adder and Carry select adder. All the combinations of the adder and multipliers are used in the design.

Among all the combination the FIR Filter with Carry look ahead adder and Wallace multiplier gives better result when compared to the other combinations.

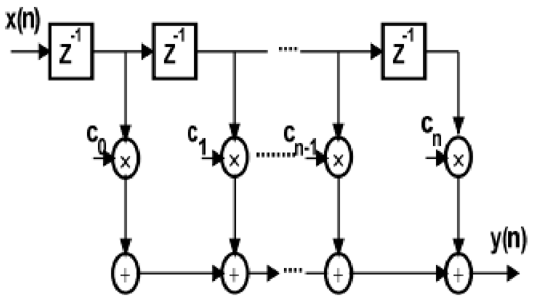


Figure.1 Direct form FIR Filter structure

The organization of the paper is as follows: section II shows the conventional FIR filter design, Section III describes the proposed low power reversible logic based design of FIR filter, simulation Results and discussion are presented in Section IV and finally conclusion is presented in Section V.

II.CONVENTIONAL FIR FILTER DESIGN:

FIR filter consists of multipliers, delay element and adders. The various multipliers and adders are described as follows:

A. Multipliers:

Multiplication involves 2 basic operations: the generation of the partial product and their accumulation. Therefore, there are two possible ways to speed up the multiplication thereby reducing the complexity, and as a result reduces the time needed to accumulate the partial products .Both solutions can be applied simultaneously. Both the multipliers consists of three stages. In the first stage the partial product matrix is formed. The second stage the obtained partial product matrix is reduced to a height of two. In the third stage the two rows are combined by carry propagating adder structure. Both Wallace and Dadda multipliers are used for reducing the partial products. The partial product are reduced as soon as possible in Wallace multiplier.

Dadda multiplier performs the minimum reduction necessary at each level to perform the reduction in the same number of levels as Wallace multiplier. Both the multipliers exhibit similar delay because same number of pseudo adder levels are used to perform the partial product reduction. Wallace multiplier uses smaller carry propagating adder, while Dadda multiplier uses larger carry propagating adder. Dadda's multiplier is the fast multiplier. Wallace and Dadda multipliers are unsigned multipliers while Baugh Wooley multiplier is a signed multiplier. Baugh and Wooley [3] have presented the modifications required to use the signed operands with column compression multipliers.

A 1Wallace Multiplier:

In Wallace tree multiplication, for 4bit multiplicand and multiplier there will be 16 partial products. The partial products are formed by using AND gates. A parallel (n,m) counter is a circuit which has n inputs and produces m outputs. A full adder is an implementation of a (3,2) counter which takes 3 inputs and produces 2 outputs. Similarly a half adder is an implementation of a (2,2) counter which takes 2 inputs and produces 2 outputs. Totally 3 stages of partial product reduction for 4 bit multiplication. If there are p rows of partial products, rows are grouped and the remaining p mod 3 rows are passed to the next stage. These rows are summed using full adders [(3,2) counters] if there are 3 partial products in 1 column and using half adders[(2,2) counter] if there are 2 partial products in 1 column. The resulting sum and carry of the full adder and half adder are passed on to the next stage. Finally in the last stage Full adders are used to obtain the product. The height of the matrix in the jth reduction stage, is given by the following recursive equations [4]

$$w_0 = N \tag{4}$$

$$w_{j+1} = 2.\left\lfloor \frac{w_j}{3} \right\rfloor + w_j \bmod 3 \tag{5}$$

The principle of Wallace tree multiplication can be extended to longer word lengths. Four reduction stages are required with matrix heights of 6, 4, 3 and 2.

A 2Dadda Multiplier:

Dadda multipliers are derived from parallel multipliers presented by Wallace in [5]. In the first stage of the dadda multiplier, partial products are formed by using N2 AND gates. In the second stage, the partial product matrix is reduced to a height of two. Dadda multipliers use a minimal number of (3,2) and (2,2) counters at each level during the compression to achieve the reduction. The reduction procedure for Dadda compression trees is given by the following recursive algorithm [6].

1. Let d_1=2 and d_{j+1}=[1.5.d_j]. d_j is the height of the matrix for the jth stage. Repeat until the largest jth stage is reached in which the original N height matrix contains at least one column which has more than d_j partial products.
2. In the jth stage from the end, place (3,2) and (2,2) counters as required to achieve a reduced matrix. Only columns with more than d_j partial products or which will have more than d_j partial products as they receive carries less significant (3,2) and (2,2) counters are reduced.
3. 3. Let j = j-1 and repeat step 2 until a matrix with a height of two is generated. This occurs when j= 1.
   The 8 x 8 Dadda multiplier requires four reduction levels with matrix heights of 6,4,3 and 2. 64 AND gates, 35 (3,2) counters, 7 (2,2) counters and 14 bit Carry propagate adder are required to form 16-bit product.

B Adders:

B 1 Ripple Carry Adder (RCA):

The basic unit of ripple carry adder is full adder. It can be constructed by connecting full adders in cascaded, with the carry out of the previous 1-bit full adder is given as carry-in to the next 1-bit full adder in the chain. In this cascaded structure, carry out propagates or ripples through the circuit.

Ripple carry adder occupies smaller area on the chip and offers high performance to random input data. The delay of the ripple carry adder depends on the length of the propagation path. Due to this reason, RCA is not suitable for circuits with non-random input operands. In the ripple carry adder, the output is known only after the carry of the previous stage is produced. Thus, the sum of the most significant bit is only available after the carry signal has rippled through the adder from the least significant stage to the most significant stage which is worst case addition. As the result, the final sum and carry bits will be valid after a considerable delay.

The delay associated with RCA is given as:

$$t_{RCA} = nt_{FA} \qquad (6)$$

where n is the number of 1-bit FAs connected in RCA and t_{FA} is the delay associated with 1-bit full adder circuit. This critical delay increases linearly with number of bits (n)[8].

B.2 Carry Look-Ahead Adder (CLA):

The major problem associated with ripple carry adder is its delay which increases with number of bits or depends upon the propagation path from least significant bit to most significant bit. In ripple carry, it is required that carry should be passed through all lower bits to compute the sum for higher bits. Therefore, for fast applications, a better design is required which can be achieved by carry look-ahead adder (CLA). CLA solves the problem of delay in RCA by calculating the carry signal in advance based on the input signal. Therefore, CLA provides lower delay than RCA at the price of more complex hardware and large area on the chip. Generate and propagate logic is used in the CLA. The main advantage of CLA is that carry delay and sum delay are independent of the number of bits one need to add. The disadvantage of the carry look-ahead adder is that the carry logic becomes complicated for more than 4-bits[8].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

B.3 Carry Select Adder (CSL):

In a ripple carry adder, every full adder cell has to wait for the carry in signal before generating carry out, which is time consuming. One way to get rid of this problem is to assume both possible values of the Cin, i.e., Cin = 0 and Cin = 1. After that, it is required to evaluate the result for both possibilities in advance. After knowing the correct value of Cin, the correct result will be chosen with the help of 2:1 MUX. This idea is implemented in the design of carry select adder. Therefore, carry select adder computes two results in parallel. Each result is computed for two different values of carry in. The carry select adder is simple and fast. Addition of two n-bit (a,b) numbers is performed by breaking the input into two blocks. For each carry save block, sum and carry values are propagated for both carry-in =0 and carry-in = 1. The actual carry-out value is then fed into a multiplexer that picks the correct sum and carry-out for the next block. When the two phases of carry-in are equal, the total gate delay is minimal[8].

B.4 Carry Save Adder (CSA):

In carry save adder, carries are saved as partial carries rather than propagated. These partial carries are added to the next operand during the next addition. One can accelerate each addition by postponing the carry propagation. The carry save adder adds up numbers by a series (multiple operand addition), followed by a carry propagate (carry propagate) addition. A carry save adder sums up a partial sum and partial carry from the previous stage as well as operand and produces a new partial sum and partial carry.

III.PROPOSED LOW POWER REVERSIBLE LOGIC BASED DESIGN OF FIR FILTER

The proposed method of low power FIR filter design consists of multipliers, delay elements and adders, realized using reversible logic gates. Reversible gates are circuits in which the number of outputs is equal to the number of inputs and there is a one-to one correspondence between the vector inputs and outputs. It not only helps us to determine the outputs from the inputs but also helps us uniquely recover the inputs from outputs. The following are the important design constraints for reversible logic circuits: Reversible logic circuits should have minimum quantum cost. Reversible logic gates do not allow fanouts. The design can be optimized so as to produce minimum number of garbage outputs. The reversible logic circuits must use minimum number of constant inputs. The reversible logic circuits must use a minimum logic depth or gate levels[8].

A REVERSIBLE GATES:

a. FREDKIN GATE:

Fredkin gate is a conservative gate (Figure.2). Hamming weight of the input vector is same as the hamming weight of the output vector. It has 3 gate inputs and 3 gate outputs(3*3). It has a quantum cost* five. Fredkin gate is used as delay element in the filter design.

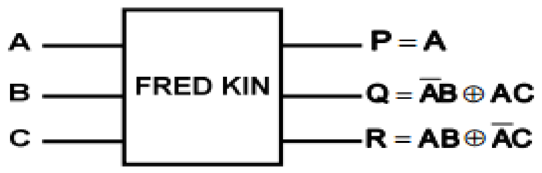


Figure2: Fredkin gate

*Quantum Cost (QC) of any reversible circuit is the total number of 2x2 quantum primitives which are used to form equivalent quantum circuit.

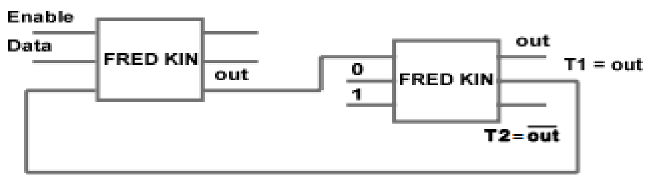


Figure 3: Fredkin gate used as Dlatch

b. PERES GATE:

Peres gate has 3 gate inputs and outputs in case of half adder (Figure.4), Peres gate has 4 inputs and 4 outputs

(4*4) in full adder realization (Figure.5). Its quantum cost is five.

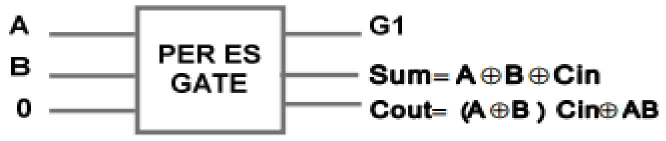


Figure4. Peres gate as half adder

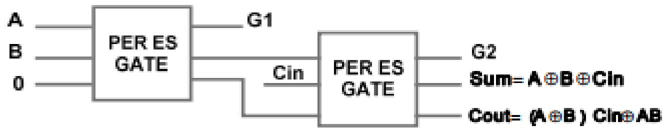


Figure.5 Peres gate as full adder

c. PV GATE:

PV gate has 3 gate inputs and outputs (3*3) S acts as select line , Q and R are the outputs

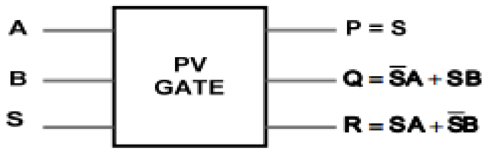


Figure6 : PV gate as mux

d.TOFFOLI GATE:

Toffoli gate has 3 gate inputs and outputs and it is replaced in the position of AND and XOR gates.

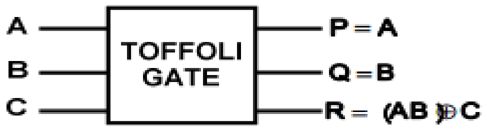


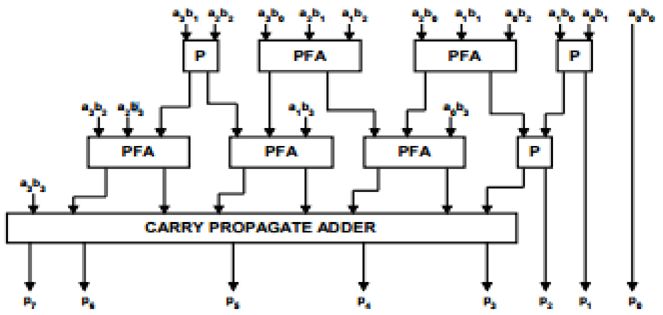
Figure. 7 Toffoli gate as AND and XOR gates

B Reversible Multipliers:

The Conventional Wallace tree and dadda multipliers are realized using reversible logic.

B.1 Reversible Wallace multiplier

For 4 bit multiplication totally there are 3 stages of partial product reduction. If there are p rows of partial products,*[P/3] rows are grouped and the remaining p mod 3 rows are passed to the next stage. These rows are summed using full adders [(3,2) counters] if there are 3 partial products in 1 column and using half adders[(2,2) counter] if there are 2 partial products in 1 column. The resulting sum and carry of the full adder and half adder are passed on to the next stage. Finally in the last stage Full adders are used to obtain the product.The full adder block is replaced by the Peres full adder. Similarly, half adder block is replaced by Peres half adder block.Figure.8 shows the 4x4 reversible Wallace multiplier.

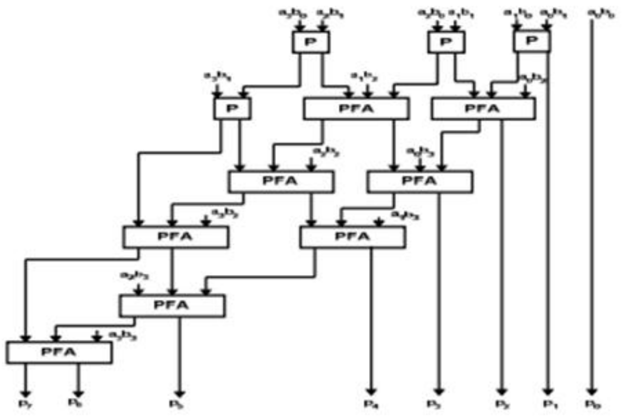


P – Peres reversible gate as half adder, PFA – Peres reversible gate asfull adder

Figure8: Reversible Wallace multiplier

B.2 Reversible Dadda multiplier:

In the Dadda multiplication,[Figure.9] the first stage consists of N2 AND gates, forming the partial products. The successive reduction levels involves the Peresfull adder and Peres half adder blocks leading to the resultant product.



P – Peres reversible gate as half adder, PFA – Peres reversible gate asfull adder

Figure9: Reversible Dadda multiplier

C Reversible Adder:

Among the adders discussed in the conventional design, Carry look ahead adder realization in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

reversible logic gives better result when compared to the Ripple carry adder, Carry select adder and Carry save adder. In the adder blocks, the full adder is replaced by Peres gate. In CLA, the generate and propagate block is replaced by the Peres gate and Toffoli reversible gates, since addition and multiplication operation are present in the corresponding reversible gates. The Figure.10 shows the generate propagate logic (GPL) realization in Carry look ahead adder. The inputs to the GPL are a, b and c and the outputs are p,q,c1.

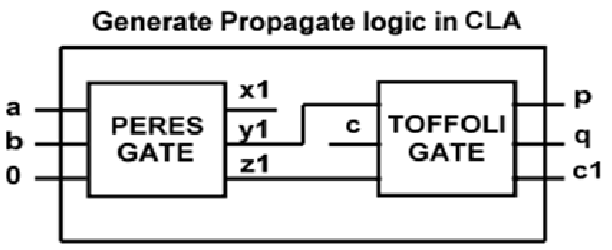


Figure10 : Generate propagate logic in CLA

IV.SIMULATION RESULTS:

All the synthesis and simulation results are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures.

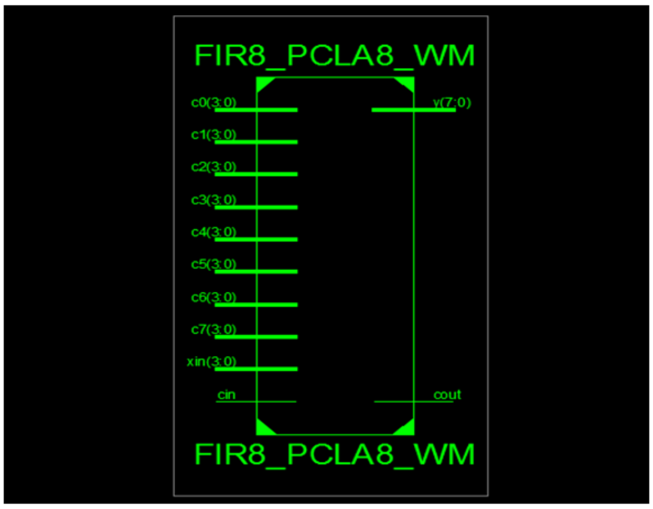


Figure11:RTL schematic of Reversible 8-Tap FIR using Carry Look Ahead adder and Wallace Tree multiplier

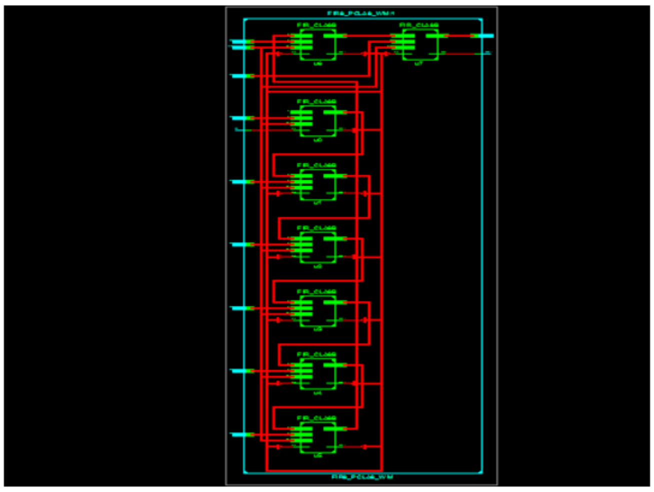


Figure12: RTL sub schematic of Reversible 8-Tap FIR using Carry Look Ahead adder andWallace Tree multiplier

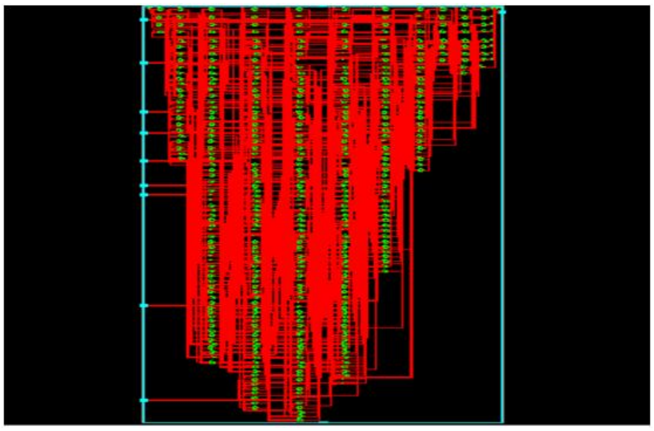
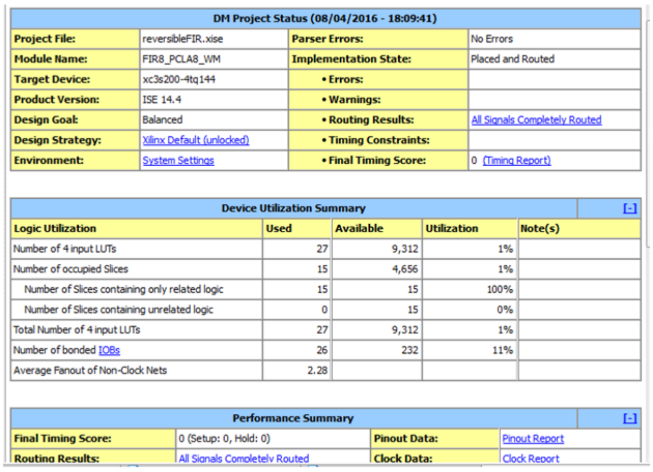


Figure13:Technology schematic of Reversible 8-Tap FIR using Carry Look Ahead adder andWallace Tree multiplier

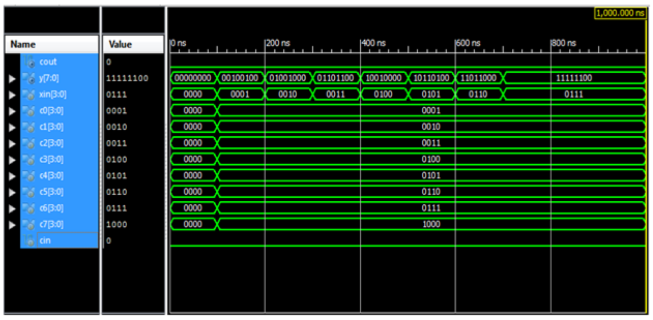


DM Project Status (08/04/2016 - 18:09:41)			
Project File:	reversibleFIR.xise	**Parser Errors:**	No Errors
Module Name:	FIR8_PCLA8_WM	**Implementation State:**	Placed and Routed
Target Device:	xc3s200-4tq144	**• Errors:**	
Product Version:	ISE 14.4	**• Warnings:**	
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	27	9,312	1%	
Number of occupied Slices	15	4,656	1%	
Number of Slices containing only related logic	15	15	100%	
Number of Slices containing unrelated logic	0	15	0%	
Total Number of 4 input LUTs	27	9,312	1%	
Number of bonded IOBs	26	232	11%	
Average Fanout of Non-Clock Nets	2.28			

Performance Summary			[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report

Figure14:Design Summary of Reversible 8-Tap FIR using Carry Look Ahead adder andWallace Treemultiplier



Name	Value
cout	0
y[7:0]	11111100
xin[3:0]	0111
c0[3:0]	0001
c1[3:0]	0010
c2[3:0]	0011
c3[3:0]	0100
c4[3:0]	0101
c5[3:0]	0110
c6[3:0]	0111
c7[3:0]	1000
cin	0

Figure15: Simulation results of Reversible 8-Tap FIR using Carry Look Ahead adder andWallace Treemultiplier

V.CONCLUSION AND FUTURE WORK:

Hence the Design of low power reversible FIR filter architecture is Done. In the proposed method the input data samples and filter coefficients are given as input to the multipliers and adders designed using reversible logic gates. The proposed scheme is compared with the FIR filter realized using normal multipliers and adders and the simulation results, shows that the proposed FIR filter designed using reversible Wallace multiplier and carry look ahead adder gives power saving with small area overhead. The technique can be used in phonetic signal processing system. The proposed work can be extended to adaptive filter.

REFERENCES:

[1]HunsooChoo, Khurram Muhammad, and Kaushik Roy, “Two’s Complement Computation Sharing Multiplier and Its Applications to High Performance DFE,” IEEE Transactions On Signal Processing, Vol. 51, No. 2, Feb. 2003.

[2]Richard Hartley, “Optimization Of Canonic Signed Digit Multipliers For Filter Design,” IEEE International Sympoisum on Circuits and Systems, Vol. 4, Jun 1991.

[3]C.R.Baugh and B.A. Wooley, “A two’s complement parallel array multiplication algorithm,” IEEE Trans. On Computers, vol.22, pp. 1045-1047, 1973.

[4]K.A.C.Bickerstaff, E.E.Swartzlander Jr., and M.J.Schulte, “Analysis of column compression multipliers,” 15th IEEE Symp. On Computer Architecture, pp.33-39, 2001.

[5]C.S.Wallace, “A suggestion for a fast multiplier,” IEEE Trans. On Computers, vol. 13, pp. 14-17, 1964.

[6]E.E. Swartzlander Jr., “Merged arithmetic,” vol.29, pp. 946-950, 1980.

[7]S. White, “Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review,” IEEE ASSP Magazine, July 1989, pp. 4 –19.

[8]KeshabK.Parhi, “VLSI Digital Signal Processing,” Wiley.