Research Journal of Applied Sciences, Engineering and Technology 7(13): 2705-2710, 2014
DOI:10.19026/rjaset.7.589
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 04, 2013 Accepted: September 27, 2013 Published: April 05, 2014

Research Article

Implementation of High Speed FIR Filter: Performance Comparison with Different Parallel Prefix Adders in FPGAs

R. Uma and P. Dhavachelvan
Department of Computer Science, School of Engineering, Pondicherry University, Puducherry, India

Abstract: This study describes the design of high speed FIR filter using parallel prefix adders and factorized multiplier. The fundamental component in constructing any high speed FIR filter consists of adders, multipliers and delay elements. To meet the constraint of high speed performance and low power consumption parallel prefix adders are more suitable. This study focus the design of new Parallel Prefix Adder (PPA) and new multiplier cell called factorized multiplier with minimal depth algorithm and its functional characteristics is compared with the existing architecture in terms of delay and area. The performance evaluation of the proposed PPA and multiplier are examined for the bit sizes of 8, 16, 32 and 64. The coefficient of the filter is obtained through hamming window using MATLAB program. The proposed FIR filter using new PPA and factorized multiplier has been prototyped on XC3S1600EFG320 in Spartan-3E Platform using Integrated Synthesis Environment (ISE) for 90 nm process. Nearly 14% of slice utilization and 34% of speed improvement has been obtained for FIR using new PPA and factorized multiplier.

Keywords: Delay element, factored multiplier, FIR filter, parallel prefix adders, signal processing

INTRODUCTION

The FIR filters are used to remove unwanted signal component using discrete transfer function of the input with a set of filter coefficients. They are widely used in DSP applications like image processing, filtering, decimation and interpolation. The general distinctiveness of a FIR filter is to modify the characteristic of signals in time and frequency domain. The basic concept of the FPGA FIR filter have been excerpted in the literature. It is reported that the FIR filters are implemented in systolic and non-systolic architecture. The core elements in any FIR filters are adders, multipliers and delay elements. Adders are one of the fundamental components in many applications. To implement a high speed FIR filter parallel prefix adders are more suitable than ripple carry, carry save, carry select and carry look-ahead adders (Uma *et al*., 2012).

The high speed parallel prefix adders are always better opted when the need for a high speed circuit exists. As for the literature views, tradeoffs for parallel prefix adders were done among number of logic levels, fan outs and wiring tracks. The conditional-sum addition (Sklansky, 1960) is a fast addition paving a logarithmic speed-up. It has the minimum logic depth ($\log_2 N$) and it needs the least routing tracks. Due to the large fan out, the area and circuit speed is also affected. Kogge-stone adders (Kogge and Stone, 1973) are the fastest prefix tree. The time complexity for carry signal is O (log n), therefore it is considered to be the fastest adder design. This is the most commonly used parallel prefix topology. The main features of this adder are that, it has uniform fan out, exhibits regular structure with minimum logical depth.

The Brent-Kung adder, Richard (1982) is a parallel prefix form carry look-ahead adder. It has a high logic depth. It is considered as one of the better tree adders for minimizing wiring tracks, fan out and gate count and it has the minimum number of nodes possible. Ladner Fischer (Richard and Fischer, 1980) prefix structure requires less implementation area but have unlimited fan out comparatively. Han and Carlson (1987) implemented an hybrid adder derived from the Brent-Kung and Kogge-Stone algorithms. This type of adder gives a good balance between the logic depth and fan-out. Knowles (2001) adder falls between the family of Kogge-Stone and Sklansky. Efficient carry-look ahead adder (Patel and Boussakta, 2007; Sabyasachi and Khatri, 2008) architecture based on the parallel-prefix computation with triple-carry-operator are presented.

The next fundamental module of FIR filter is the high speed multipliers. There are different classes of multipliers exist, among these multipliers, the array multiplier (Sheplie, 2004; Ching, 2005) is the simplest one in terms of area and power consumption. But the delay of the circuit is high since this topology uses full

Corresponding Author: R. Uma, Department of Computer Science, School of Engineering, Pondicherry University, Puducherry, India

adder structures which forms carry chain in different stages of multiplier circuit. This disadvantage can be solved by using carry save (Wallace, 1964) adders by the structure of Wallace Tree. This Wallace Tree multiplier reduces the ripple carry delay in the internal adder circuits, thereby reducing the propagation delay of the circuit. In the reference (Baugh and Wooly, 1973) presents modified booth algorithm with radix-4 cellular array modular multiplier circuit. This type of multiplier reduces the number of iterations by using pipeline structure and direct radix-2 implementation of Montgomery.

This study describes the design of high speed FIR filter using new parallel prefix adders and factorized multiplier to meet the constraint of high speed performance and low power consumption.

MATHEMATICAL MODELING OF PARALLEL PREFIX ADDER

This section presents the implementation and simulation output of proposed Delay-Area efficient PPA and multiplier circuit design using factorization method.

Parallel prefix adders allow more efficient implementation of the carry look-ahead technique. These are nothing but a two level carry look-ahead adders. The addition in PPA is usually expressed in terms of carry generation signal g_j, carry propagation signal p_j, carry signal c_j and sum signal s_j at each bit position ($1 \le j \le n$):

$$g_j = a_j b_j \tag{1}$$

$$p_j = a_j \oplus b_j \tag{2}$$

$$C_j = \begin{cases} g_j & \text{if } j = 1 \\ g_j + p_j c_{j-1} & \text{if } 2 \le j \le n \end{cases} \tag{3}$$

$$s_j = p_j \oplus c_{j-1} \tag{4}$$

The extended consecutive bits carry and propagation are computed as:

$$G_{[j:l]} = \begin{cases} g_j & \text{if } j = l; \text{ if } n \ge j > l \ge 1 \\ G_{[j:m]} + P_{[j:m]} G_{[m-1:l]} \end{cases} \tag{5}$$

$$P_{[j:l]} = \begin{cases} p_j & \text{if } j = l; \text{ if } n \ge j > l \ge 1 \\ P_{[j:m]} P_{[m-1:l]} \end{cases} \tag{6}$$

Figure 1 shows two of basic components: g-p generator, (G, P) and g generators, (G). They are denoted by a black and grey cells respectively.

The proposed adder is the hybrid adder obtained from the combination of Ladner Fischer and Han Carlson. The first two stages of the adder follow Han Carlson adder topology and the remaining stages follow the Ladner Fischer adder. These networks have low gate

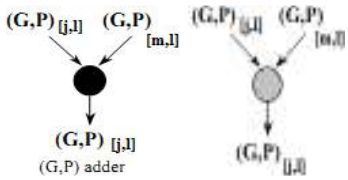
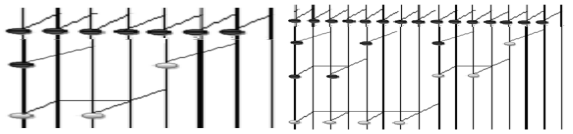
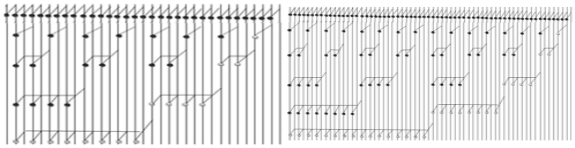


Fig. 1: Black and gray cell



(a) 8-bit proposed PPA (b) 16-bit proposed PPA



(c) 32-bit proposed PPA (d) 64-bit proposed PPA

Fig. 2: Proposed adder for 8, 16, 32 and 64-bit

counts. This adder concentrates on the design of a parallel prefix network with a minimal depth case. The main limitation exists in Ladner Fisher adder is that the lateral fan out of the prefix cells doubles at every levels. Thus additional buffers are used, as this drawback can adversely affect the performance. This can be eliminated through near minimum depth prefix algorithm using (Richard and Fischer, 1980). The best characteristics of both the adder are adopted in order to construct this proposed adder. It has the order of log N and the numbers of nodes are N/2-1 and N/4. The total number of computational nodes are N-1+ ($\log_2$N-2) N/4. This adder gives the most effective performance in the aspects of delay and area. The proposed adder for 8, 16, 32 and 64 bit is depicted in Fig. 2.

The performance evaluation of the proposed PPA and multiplier are examined for the bit sizes of 8, 16, 32 and 64. The target FPGA device chosen for the implementation of these adders has been prototyped on XC3S1600EFG320 in Spartan-3E Platform using Integrated Synthesis Environment (ISE) for 90 nm process. Structural data flow modeling using Verilog HDL was used to model each adder. The optimization targets for these adders are set to speed constraint optimization. Table 1 presents the simulated results of existing adders and proposed adder in terms of path combinational delay and total slice utilization.

PROPOSED MULTIPLIER CIRCUIT USING FACTORIZATION METHOD

The multiplier circuit is implemented using factoring method. A method for multiplying numbers by factoring one of the numbers into smaller parts. For example,

Table 1: Simulated results of existing and proposed parallel prefix adders

	Sklansky		Brent kung		Kogge stone		Riyaz		Sabyasachi das		Proposed	
Size	Delay (ns)	Slices	Delay (ns)	Slice	Delay (ns)	Slices	Delay (ns)	Slices	Delay (ns)	Slices	Delay (ns)	Slices
8	10.0	9	9	8	10.4	16	10.1	12	11.0	8	8.91	9
16	11.0	27	18.4	21	12.8	35	19.4	22	16.4	21	13.90	18
32	20.8	52	23.8	51	14.3	122	34.2	51	27.1	51	18.30	38
64	23.9	121	30.2	122	18.8	274	52.3	117	43.2	105	22.10	80

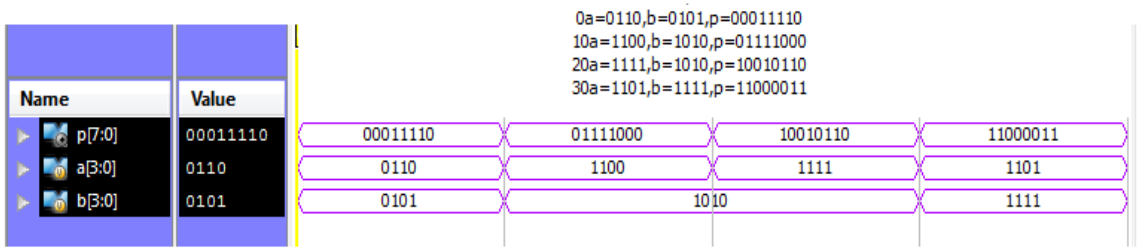


Fig. 3: Simulation result of 8-bit proposed multiplier circuit

Table 2: Pseudo code for implementing 8-bit multiplier design

Pseudo code: //8-bit multiplier cell

```
Begin
   A← input1
   B ← input2
   P← output
   N←3
   tQ, tR, tC← internal computation variable
If (A<=2^N)
      tQ←2^N - A
      tR←{B, 000} //Concatenate the value of B with 3 zeros
      tC←B* tQ
      P = tR - tC
else
      tQ←2^N + A
      tR←{B, 000} //Concatenate the value of B with 3 zeros
      tC←B* tQ
      P = tR + tC
End If
End
```

41×99 = 41x (100-1) = 4100-41 = 4059. For binary multiplication of A and B, factor A into smaller number. In designing 8-bit multiplier cell it is easier to factor the number in terms of 8. This makes the computation process simple by incorporating shift operation. The number of adders used in the proposed multiplier cell will be less by incorporating this factorization method. The pseudo code for implementing this multiplier cell is presented in Table 2 and its simulation result is shown in Fig. 3. The comparison results obtained for array, Wallace, booth multiplier with proposed work is shown in Table 3. From the simulation result it can be observed that the proposed multiplier cell utilizes less slices and critical path delay when compare to the existing multipliers.

For example consider the design of 8-bit multiplier circuit:

A = 0110, B = 0101 for (A<=1000)
0101* (1000 - 0010)
0101*1000 - 0101*0010

A = 1100, B = 1010 for (A>1000)
1010* (1000 + 0100)
1010*1000 + 1010*0100

Implementation of FIR filter:

FIR filter design using various parallel prefix adders: For an N-tap FIR filter with coefficients h (k), whose output is described by:

$$y(n) = h_0 x(n) + h_1 x(n-1) + h_2 x(n-2)...h_{N-1} x(n-N-1) \quad (7)$$

The filter's Z transform is:

$$H(z) = h_0 z^{-0} + h_1 z^{-1} + h_2 z^{-2} + ...h_{N-1} Z^{-N-1}$$

or,

$$H(Z) = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (8)$$

Different filter design has been reported in the literature (Wei *et al*., 2008; Jiang and Bao, 2010). The basic architecture of a linear FIR filter is shown in Fig. 4. The design structure of this filter consists of adders, multipliers, filter coefficients (h_0, h_1-h_{n-1}) and delay elements. The delay element is implemented as D flip-flop. The filter coefficients are stored in ROM. A processing element is implemented with adder and multiplier. The processing element block will be called or used as many number of time as per the requirement. The coefficient of the filter using hamming window is found through MATLAB program. The coefficient may be signed fractional number which must be converted into unsigned binary and scaled to a coefficient width of 5 and it is stored in ROM. The top block of 4-tap FIR is shown in Fig. 5.

Table 3: Comparison of existing multiplier and proposed circuit

Multiplier	Bit size	Slices	Critical delay (ns)	Routing delay	Gate delay	Logic level
Array	8	72	17.800	11.285	6.515	22
	16	147	45.670	23.450	23.678	48
	32	823	88.920	44.670	44.210	96
Wallace	8	90	14.130	10.100	4.120	11
	16	387	37.460	23.670	14.560	36
	32	1565	75.428	40.230	35.120	76
Booth	8	82	13.450	10.340	3.050	14
	16	125	40.560	20.230	20.120	32
	32	670	67.890	33.450	34.120	47
Proposed	8	34	11.118	1.880	10.234	7
multiplier	16	103	20.230	10.120	10.110	22
	32	550	52.120	26.120	25.560	52

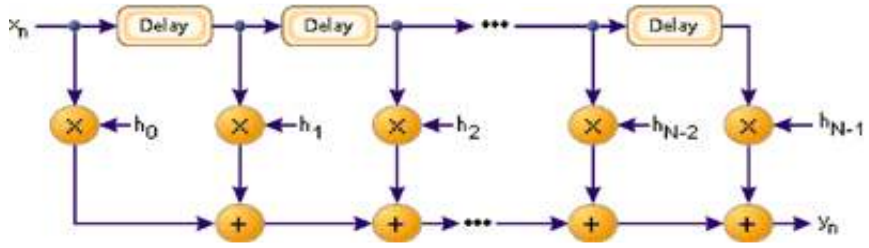


Fig. 4: Basic architecture of a linear FIR filter

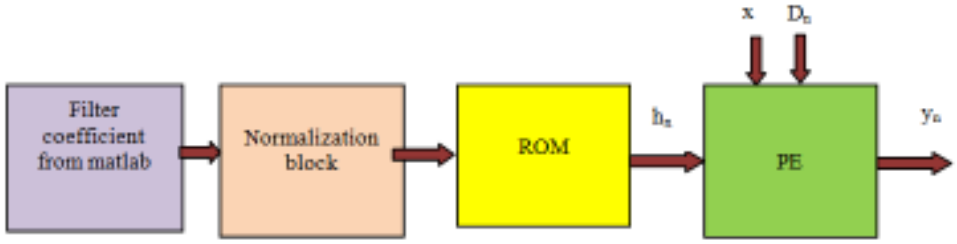


Fig. 5: Top block FIR filter

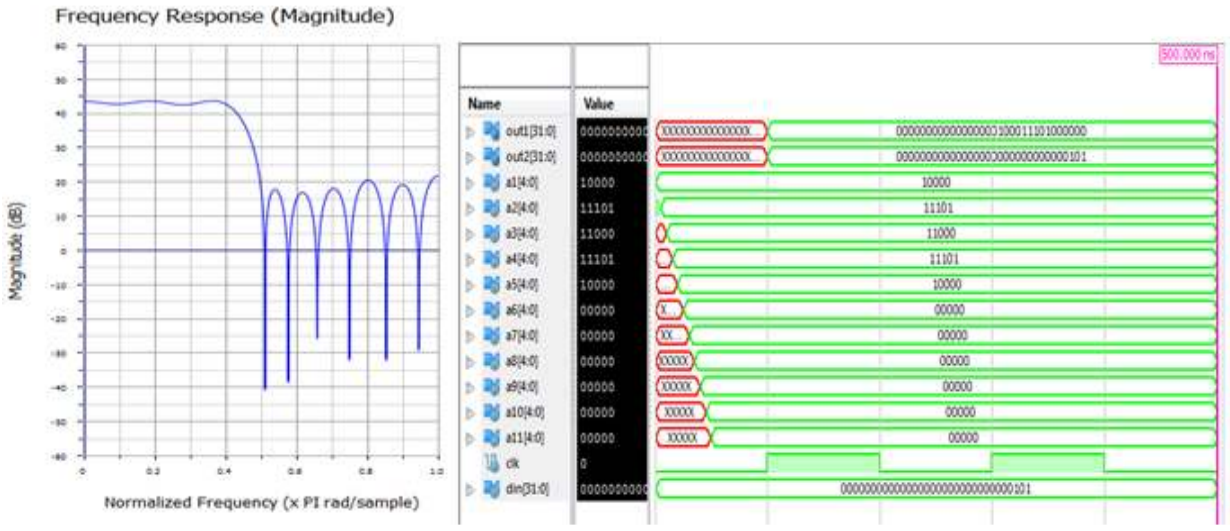


Fig. 6: Simulated result of 4-tap FIR filter

RESULTS AND DISCUSSION

The proposed 4-tap FIR is simulated with different parallel prefix adders using Xilinx 12.1 ISE with constraints fixed for speed and area optimization. The filter coefficients are obtained for the following specification using hamming windowing technique Filter order-4, Type of window-hamming window, The

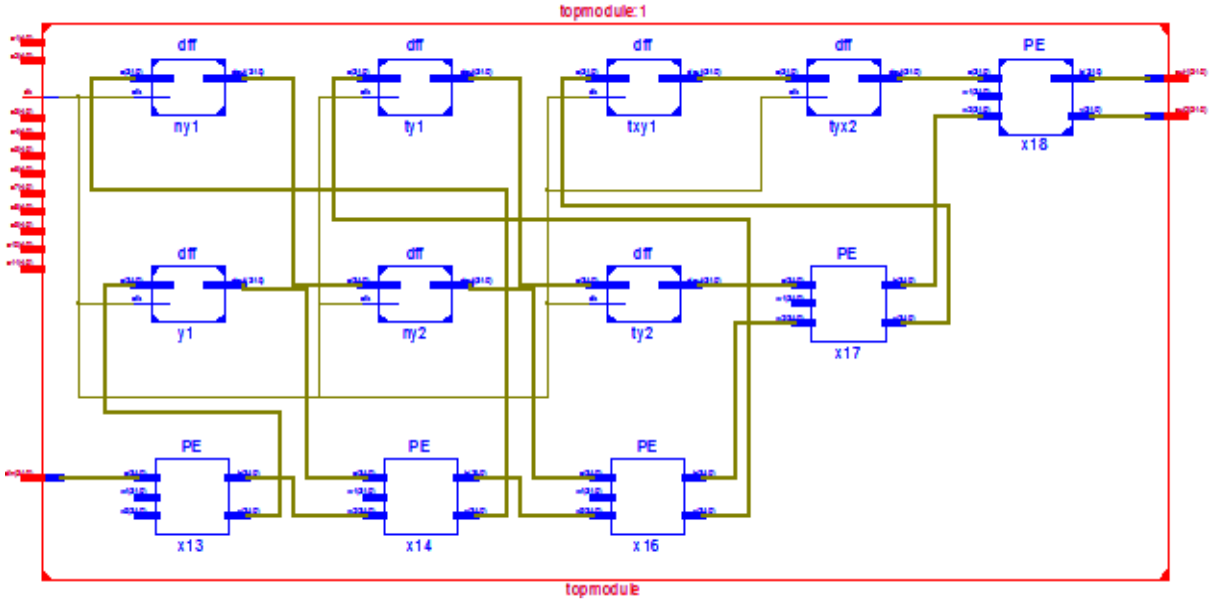


Fig. 7: Synthesis result of 4-tap FIR filter

Table 4: Comparison for FIR filter design using various adder topologies

Filter design with various adder topology	Rise time delay (before clk) ns	Hold on delay (after clk) ns	Slices	Total delay ns
Sklanshy	2.2340	36.570	398	56.540
Brent kung	2.1731	48.440	392	52.731
Kogge stone	3.5100	42.356	800	39.540
Riyaz	2.3450	39.650	450	47.123
Sabyasachi das	2.7560	40.987	456	47.879
Proposed	1.7310	35.600	389	37.137
Filter design with various adder topology	Min. period (ns)	Frequency MHz	Improvement in slices (%)	Improvement in delay (%)
Sklanshy	2.345	270.37	2.261307	34.31730
Brent kung	2.445	286.36	0.765306	29.57274
Kogge stone	2.245	272.37	51.375000	16.07739
Riyaz	2.245	271.12	13.555560	21.19135
Sabyasachi das	2.451	290.16	14.692980	22.43572
Proposed	2.560	292.54		

Min.: Minimum

frequency range: (fs = 42000 (sampling frequency), fc = 10600 (cutoff frequency) and input bit size-32 bit. The proposed FIR filter using new PPA and factorized multiplier has been prototyped on XC3S1600EFG320 in Spartan-3E Platform using Integrated Synthesis Environment (ISE) for 90 nm process. The simulation and synthesis report are presented in (Fig. 6 and 7). The synthesis report shows the longest combinational path between any two registers goes through just one processing element. The synthesis tool report a maximum clock period of 2.256 ns which allows this filter to be run at 300 MHz with the proposed adder.

Table 4 presents the comparison between Filter design with various parallel prefix adders and proposed adder and multiplier.

CONCLUSION

The study presented the implementation of highly efficient FIR Filter design with Delay-Area efficient parallel prefix adder and multiplier circuit using factoring method with minimal depth algorithm and its functional characteristics is compared with the existing architecture in terms of delay and area. The performance evaluation of the proposed PPA and multiplier are examined for the bit sizes of 8, 16, 32 and 64. The proposed FIR filter using new PPA and factorized multiplier has been prototyped on XC3S1600EFG320 in Spartan-3E for 90 nm process. The FIR filter design with the proposed structure produces approximately 14% of the area reduction and 34% of delay reduction when compare to other existing parallel prefix adders.

REFERENCES

Baugh, C. and B.A. Wooly, 1973. A two's complementary parallel array multiplication algorithm. IEEE T. Comput., C-22(12): 1045-1047.

Ching, Y.N., 2005. Low-power high-speed multipliers. IEEE T. Comput., 54(3): 355-361.

Han, T. and D. Carlson, 1987. Fast area efficient VLSI adders. Proceedings of the 8th Symposium on Computer Arithmetic, pp: 49-56.

Jiang, X. and Y. Bao, 2010. FIR filter design based on FPGA. Proceeding of International Conference on Computer Application and System Modeling, (ICCASM 2010).

Knowles, S., 2001. A family of adders. Proceeding of the 15th IEEE Symposium on Computer Arithmetic, pp: 277-281.

Kogge, P.M. and H.S. Stone, 1973. A parallel algorithm for the efficient solution of a general class of recurrence equations. IEEE T. Comput., 22(8): 786-793.

Patel, R.A. and S. Boussakta, 2007. Fast parallel-prefix architectures for modulo 2n-1 addition with a single representation of zero. IEEE T. Comput., 56(11): 1484-1492.

Richard, P.B., 1982. A regular layout for parallel adders. IEEE T. Comput., c-31(3): 260-264.

Richard, E.L. and M.J. Fischer, 1980. Parallel prefix computation. J. Assoc. Comput. Mach., 27(4): 831-838.

Sabyasachi, D. and S.P. Khatri, 2008. A novel hybrid parallel-prefix adder architecture with efficient timing-area characteristic. IEEE T. VLSI Syst., 16(3).

Sheplie, M., 2004. High performance array multiplier. IEEE T. VLSI Syst., 12(3): 320-325.

Sklansky, J., 1960. Conditional-sum addition logic. IRE T. Electron. Comput., EC-9(2): 226-231.

Uma, R., V. Vijayan, M. Mohanapriya and S. Paul, 2012. Area, delay and power comparison of adder topologies. Int. J. VLSI Design Commun. Syst., 3(1): 153.

Wallace, C.S., 1964. A suggestion for a fast multiplier. IEEE T. Comput., 13: 14-17.

Wei, L., R.J. Yang and X.T. Cui, 2008. Design of FIR filter based on distributed arithmetic and its FPGA implementation. Chinese J. Sci. Instrum., 29: 2100-2104.