

## Design and Analysis of Efficient Adder for Reconfigurable Fir Filter

<sup>1</sup>R. Dhayabarani and <sup>2</sup>R.S.D. Wahidabanu

<sup>1</sup>Department of Electronics and Communication Engineering,  
V.S.B. Engineering College, Karur, India

<sup>2</sup>Department of Electronics and Communication Engineering,  
Government College of Engineering, Salem, India

**Abstract:** The Finite Impulse Response (FIR) filter plays an important role in the Digital Signal Processing (DSP). The FIR filter developed based on Binary Signed Sub (BSS) coefficient shows promising improvement in terms of total Logic Elements (LE), Pre-processor Logic Element and Processing Element. The conventional adder unit present in the reconfigurable BSS FIR filter architecture consumes more logic elements; hence this paper presents an analysis with two different adders namely, Carry Look-ahead Adder (CLA) and Carry Save Adder (CSA). The proposed architecture was developed using Verilog HDL and targeted in Alter Stratix FPGA. The result evaluation is made in terms of total Logic Elements (LE), total combinational functions, and dedicated logic register and memory bits. The experimental evaluation reveals that the Carry Save Adder (CSA) is very much opted for the BSS FIR filter.

**Key words:** Reconfigurable • FIR • Adder • DSP

### INTRODUCTION

A filter is a frequency discriminating system, which is used to change an input signal in order to enable further processing. Essentially, there are two types of filters namely, analog and digital filter. Digital Filters are extensively used in different parts, because digital filters have the possibility to attain much better signal to noise ratio than analog filters. The digital filter executes noiseless measured operations at each transitional step in the change and their precise reproducibility allows design engineers to achieve performance levels that are difficult to obtain with analog filters. Digital filters activate on numbers differing to analog filters, which operates on voltages.

The basic process of digital filter is to proceeds a classification of input numbers and subtracts a different classification of output numbers. There exists a variety of different digital filters. FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters are the two mutual filter forms. The problem with IIR filters is that the closed-form IIR proposals are preliminary limited to low pass, band pass, and high pass filter etc.

A FIR filter is a filter assembly that can be used to instrument almost any sort of frequency response digitally. It is generally executed by using a series of delays, multipliers, and adders to generate the filter's output. The architecture of FIR filter is shown in Fig. 1.

From the above structure the transfer function of canonical form of the filter can be easily described in Z-domain as:

$$H(Z) = X_0 + X_1Z^{-1} + X_2Z^{-2} + \dots + X_{p-1}Z^{1-p} \quad (1)$$

The requirements for a digital filter are normally specified in the frequency domain in terms of the desired magnitude response and/or desired phase response. In the lowpass case, the desired magnitude response is usually given by,

$$D(\omega) = \begin{cases} 1 & \text{for } \omega \in [0, \omega_p] \\ 0 & \text{for } \omega \in [\omega_p, \pi] \end{cases} \quad (2)$$

In many digital signal processing utilizations, FIR filters are favoured over their IIR counter parts. The main advantages of the FIR filter designs over their IIR equivalents,

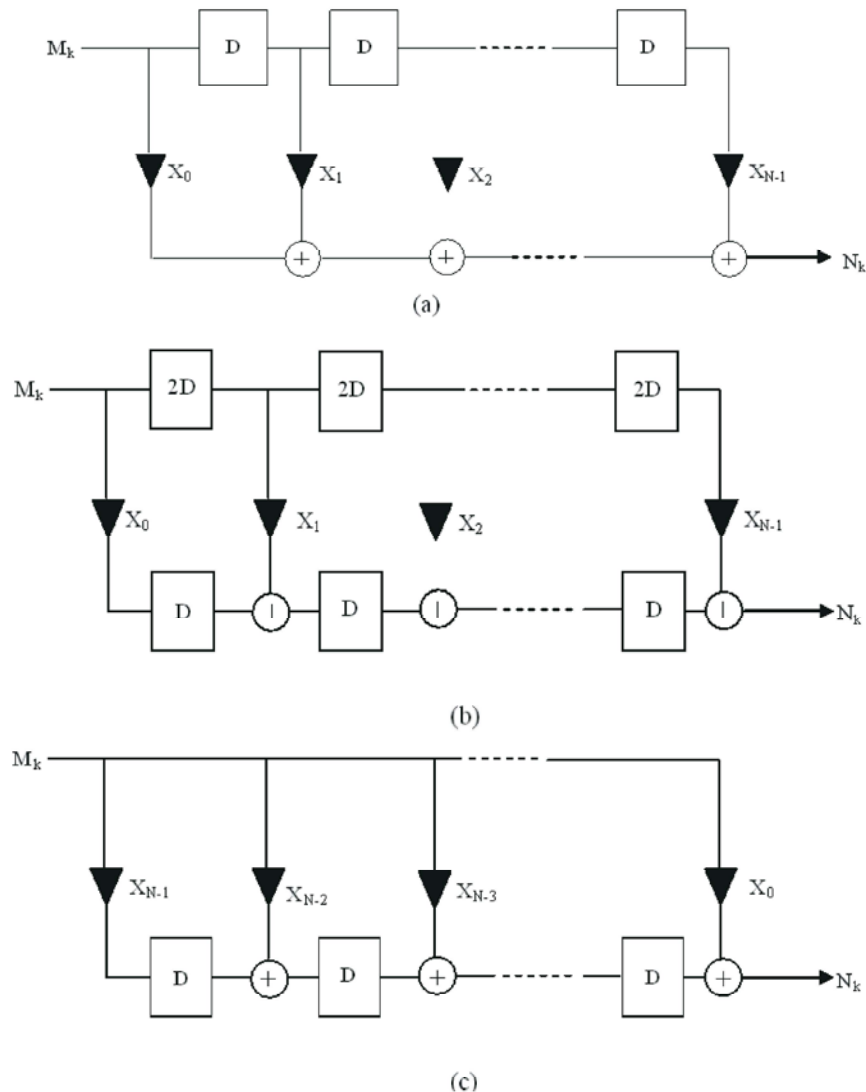


Fig. 1: FIR Filter Architecture, (a) Canonical form, (b) Pipelined, (c) Inverted form.

- FIR filters with exact linear phase can easily be designed
- There exist computationally effective realizations for enforcing FIR Filters. Those filters involve both non-recursive and recursive realizations.
- FIR filters realized non recursively is genetically stable and clear of limit cycle oscillations even though performed on a finite-word length digital system.
- Excellent design procedures are available for different kinds of FIR filter with arbitrary specifications
- The output noise due to multiplication round off errors in an FIR filter is regularly very low and the sensitivity to differences in the filter coefficients is too low.

**FIR Filter Design Techniques:** FIR filters are mainly useful for presentations where exact linear phase response is required. The FIR filter is generally executed in a non-recursive way, which agreements a stable filter [1]. FIR filter design essentially consists of two parts namely approximation and realization problem,

**Approximation Problem:** The approximation moment takes the specification and contributes a transfer function through four steps. They are as follows:

- A desired or ideal response is chosen, usually in the frequency domain.
- An allowed class of filters is chosen (e.g. the length  $N$  for a FIR filters).

- A measure of the quality of the approximation is chosen.
- A method or algorithm is selected to find the best filter transfer function.

**Realization Problem:** The realization method deals with selecting the structure to implement the transfer function whatever may be in the model of circuit diagram or in the model of a program. There are basically three well-known procedures for FIR filter design namely:

- Window technique
- Frequency sampling technique
- Optimal filter design techniques

**Related Works:** In FIR, arithmetic multiplying, illustration of actual numbers in floating point will use an extensive range of values. The floating point unit is generally used in different applications. This makes the developer to work on faster floating point multiplier units. Floatingpoint representation can keep its resolution and precision when estimate to fixedpoint representations.

Multiplication of floating point number can be carried out in 3 parts [2] in the 1st part, the sign product will perform an XOR operation. In the 2nd part, the exponent bits operands are accepted to an adder stage and a bias 127 is subtracted from the output. 8-bit kogge-stone adder is used for device the addition and 2s complement addition for subtraction operations.

**Single Precision Floating Point Numbers:** The single-precision number representations have 32 bits. There are three main fields in a single precision number which are ‘S’, ‘M’ and ‘E’. The 7-digit decimal number represented in 24-bit mantissa, in that while an 8-bit represent to a base 2 which present a scaling factor with a fitting range. Thus, a sum of 32 bit is desired for single-precision number representation. In order to get the stored exponent a bias of  $2n-1, -1$  is added to real exponent.

To increase the performance of multiplier, four stage pipelining is used. In order to increase the performance of multiplier operating frequency of multiplier is increased, pipeline stages are inserted in the critical path. Pipelining stages reduce the latency in the output by four clocks [3-5].

The pipelining stages are followed in following steps:

- Pre-processing the input data.
- Addition of partial products.

- Subtracting the bias and compressing the partial result in Wallace tree.
- Normalization and final carry is adder.

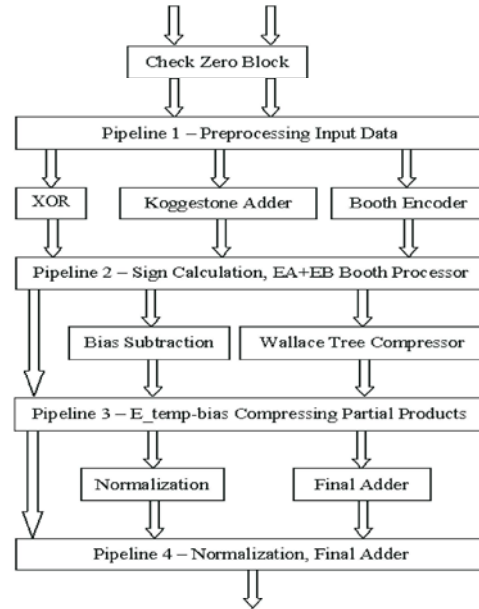


Fig. 2: Pre-processing Multiplier Unit

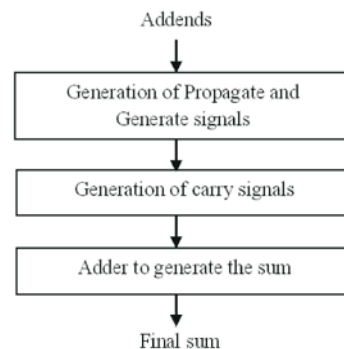


Fig. 3: Three stage structure of a parallel prefix adder

Pipeline registers and stages of pre-processing the input data are shown in Fig. 2. Addition of partial products, subtracting the bias and compressing the partial result in Wallace tree. Normalization and final carry is adder.

A parallel prefix adder is primarily faster than other carry propagate adder. A Parallel prefix adder is the most efficient circuit for the binary addition. A parallel prefix form of carry look ahead adder is a Kogge-stone adder. It creates the carry signals in  $O(\log n)$  time, and is widely considered as the fastest adder design possible. It takes more area to implement as compared to others parallel adders but has a lesser fan-out at every level. A parallel Prefix Addition is generally a three step process.

The first step includes the creation of generate (gi) and propagate (pi) signals for the input operand bits. The second method includes the generation of carry signals and finally a simple adder to generate sum [4]. The three stage structure of carry look ahead adder and parallel prefix adder is shown in Fig. 3.

In common, digital signal processors are used to accomplish the digital signal processing operations like correlation, convolution, filtering and transform. All the above cited digital signal processing processes are in the mode of multiplication and redesigned addition. Unusually Multiply Accumulate Circuit (MAC) is the nature of the digital signal processor. The normal digital Finite Impulse Response (FIR) filter [6] characterized as (1), According to the input and output signal sequences are  $x[n]$  and  $y[n]$ . Here  $N$  is the length of the filter and  $h[n]$  is the filter impulse response. This signal sequence can be characterized as fixed point complex numbers or floating point complex numbers. The complex numbers are playing a crucial role in digital signal processing (DSP) and electronics, as a result, they are an easy way to employ and characterize the most beneficial real world sinusoidal waveforms. This signal features like amplitude and phase, can be disclosed easily by complex numbers than real numbers. By its nature complex numbers are adopted in Fast Fourier Transform (FFT) [2].

$$y[n] = \sum_{k=0}^{N-1} x[n-k]h[k] \quad (3)$$

Fig. 4 shows the fundamental blocks of MAC, whereby the inputs ‘A’ and ‘B’ are multiplied, next the multiplication results are combined with the prior MAC result. In case that, the inputs ‘A’ and ‘B’ are ‘n’ bits immense compare than the multiplication result will have ‘2n’ bits immense. So to bypass the overflow while accumulation, the accumulation directory will have ‘k’ additional bits with its assured length of ‘2n’ bits.

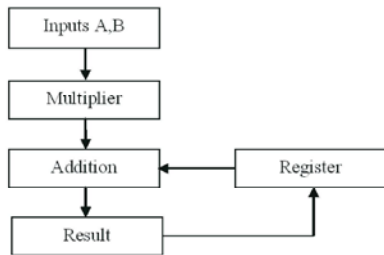


Fig. 4: Architecture of MAC using Amplitude signal

In common, a digital signal can be characterized as amplitude (r) with phase ( $\theta$ ), whichever can be recorded

as a complex number  $z = r \angle \theta = x + jy$ , where  $r = \sqrt{x^2 + y^2}$  and  $\theta = \tan^{-1} \frac{y}{x}$ . The two complex numbers are

characterized as  $(P + jQ)$  and  $(R + jS)$ , where  $P = \{P_s, P_m\}$ ,  $Q = \{Q_s, Q_m\}$ ,  $R = \{R_s, R_m\}$  and  $S = \{S_s, S_m\}$ .

In this direction, the suffix (s) characterizes the sign bit and ‘m’ characterizes a binary number. The complex number multiplication can be finished in several ways. Enable the original part of the multiplication of the two complex numbers is  $X$  and  $Y$ , where  $X = \{X_s, X_m\}$  and  $Y = \{Y_s, Y_m\}$ . As reported by (3), FPCN multiplication needs five fixed point adders and three fixed point multipliers [7].

$$X + jY = (P + jQ)(R + jS) \quad (4)$$

$$X + jY = \{(P - Q)S + (R - S)P\} + j\{(P - Q)S + Q(R + S)\} \quad (5)$$

$$X + jY = \{PR - QS\} + j\{PS + QR\} \quad (6)$$

In this direction, the multiplication will start back of computing  $(P - Q)$ ,  $(R - S)$  and  $(R + S)$ . As reported by (4), FPCN (Fixed Point Complex Number) multiplication needs two fixed point adders and four fixed point multipliers. Hereabouts, the multiplication can start instantly. In consequence of one supplementary multiplier, hardware, area for (4) is greater than (3). However the time complexity for (4) is  $\log_2 n$  depth lower than (3). Remarkably the proposed architecture is followed in (3).

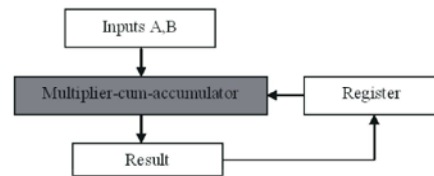


Fig. 5: Architecture of Multiply Accumulator Unit

In common, MAC operation can be achieved with multiplication pursued by accumulation. So the intensity of the MAC circuit is determined by the accumulator and multiplier circuit. In this proposed architecture, the accumulation can be achieved besides multiplication (Multiplication-Cum-Accumulation). Namely, the prior MAC result is joined along with the partial products of the present multiplication. And therefore the separate accumulator circuit is avoided. The proposed architecture of MAC is as shown in Fig 5. In this paper, the various adder unit is evaluated for the proposed reconfigurable FIR filter.

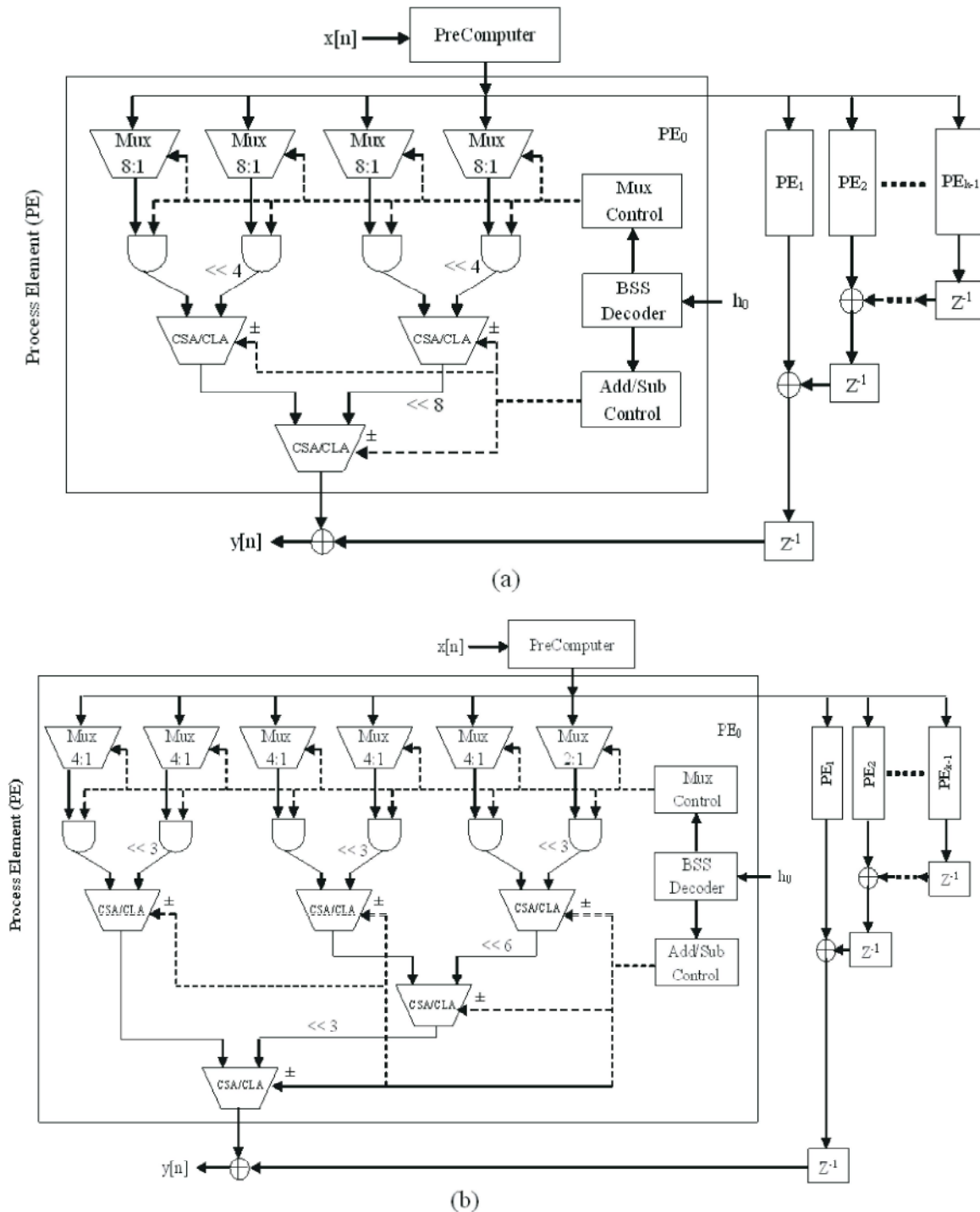


Fig. 6: Proposed Reconfigurable Architecture (a) 4-bit (b) 3-bit

**Proposed Method**

**Reconfigurable FIR Filter Architecture:** The reconfigurable FIR filter architecture with BSS (Binary Signed Sub) coefficients is shown in Fig. 6. The idea of reducing the BSS expression in FIR filter will reduce the

computational complexity, logic elements count and time. In the proposed architecture, adder unit is replaced by two different types of adder namely, Carry Look-ahead Adder (CLA) and Carry Save Adder (CSA) as shown in Fig. 6.

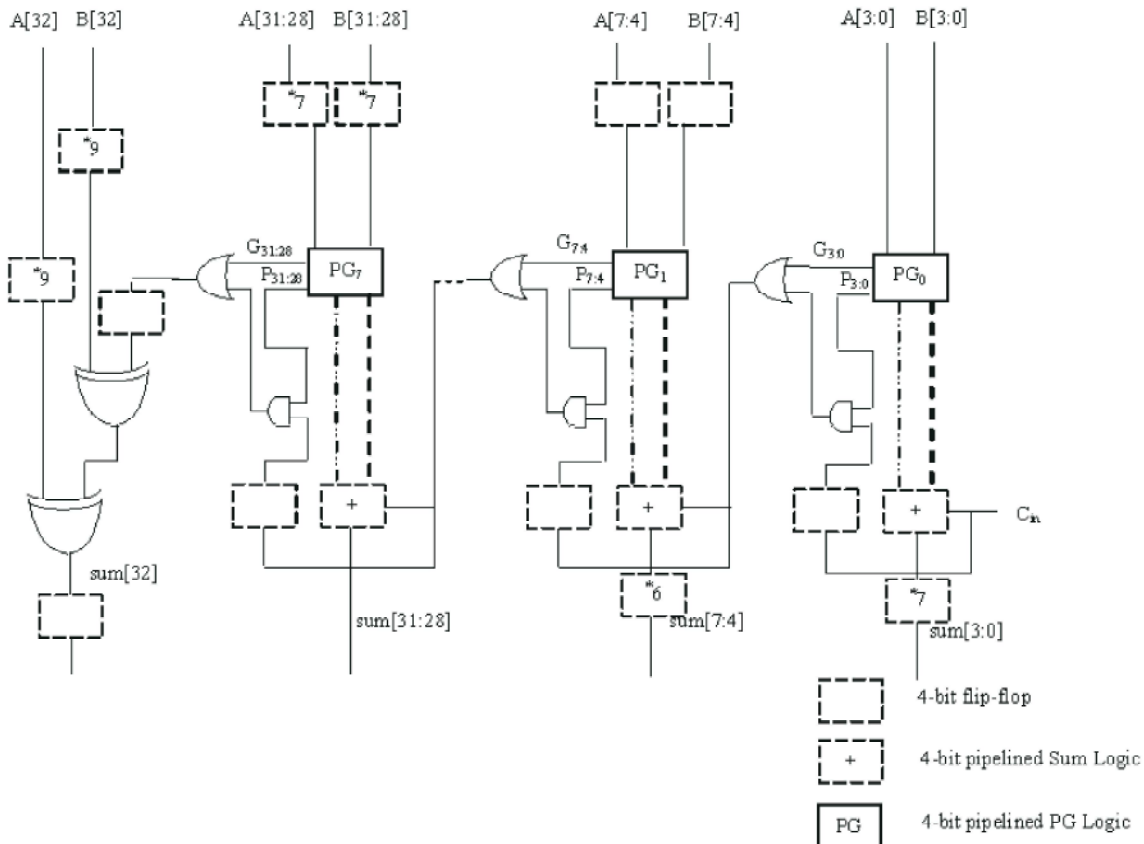


Fig. 7: 33-bit pipelined CLA architecture

**Carry Look-Ahead Adder:** CLA has the advantage of computing carries in parallel with sum calculations using propagate and generate (PG) logic.

It consists of several stages; each stage consists of 4 bits which generate bitwise propagate and generate signals that are used to produce group generate and propagate signals using valence 4 cells PG logic. Then the group generates and propagate signals generate the next group input carry. Bitwise propagate and generate signals also produce summation results through the sum logic unit. As shown in Fig. 7, eight stages are used to implement 33-bit pipelined CLA architecture.

**Carry Saves Adder:** Carry save adder reduces carry propagation delay by using parallelism. It consists of several stages, each stage consists of 4 bits which are added two times using 4-bit pipelined ripple carry adder; once with input carry equals to zero and another time with input carry equals to one. Then, the propagated input carry selects (through a MUX) the correct sum and determines the output carry. In Fig. 8, 33-bit CSA consisting of eight stages is presented.

The multiplier used in this MAC block is a Modified Booth Multiplier which reduces the number of partial products to be generated and is known as the fastest multiplication algorithm. Wallace Tree Carry Save Adder structures have been adopted to sum the sectional products in decreased time. The main objective is to decrease computation time by adopting Booth's algorithm for multiplication and to decrease the chip area by adopting Carry Save Adders organized in a Wallace tree structure.

The proposed design is used for FIR filter application. In filter each tap contains one multiplier, adder and one delay element. The normal multiplier is replaced by the proposed multiplier. The filter process is done stage by stage. Each stage is having separate co-efficient to perform the filter operation. The first input is given to the first tap it multiplies with the co-efficient that presents in the first tap. When the second input is given in the filter, the first input is delayed and moved to the second tap. Now the first input and the second coefficient get multiplied. Similarly the second input and the first co-efficient get multiplied.

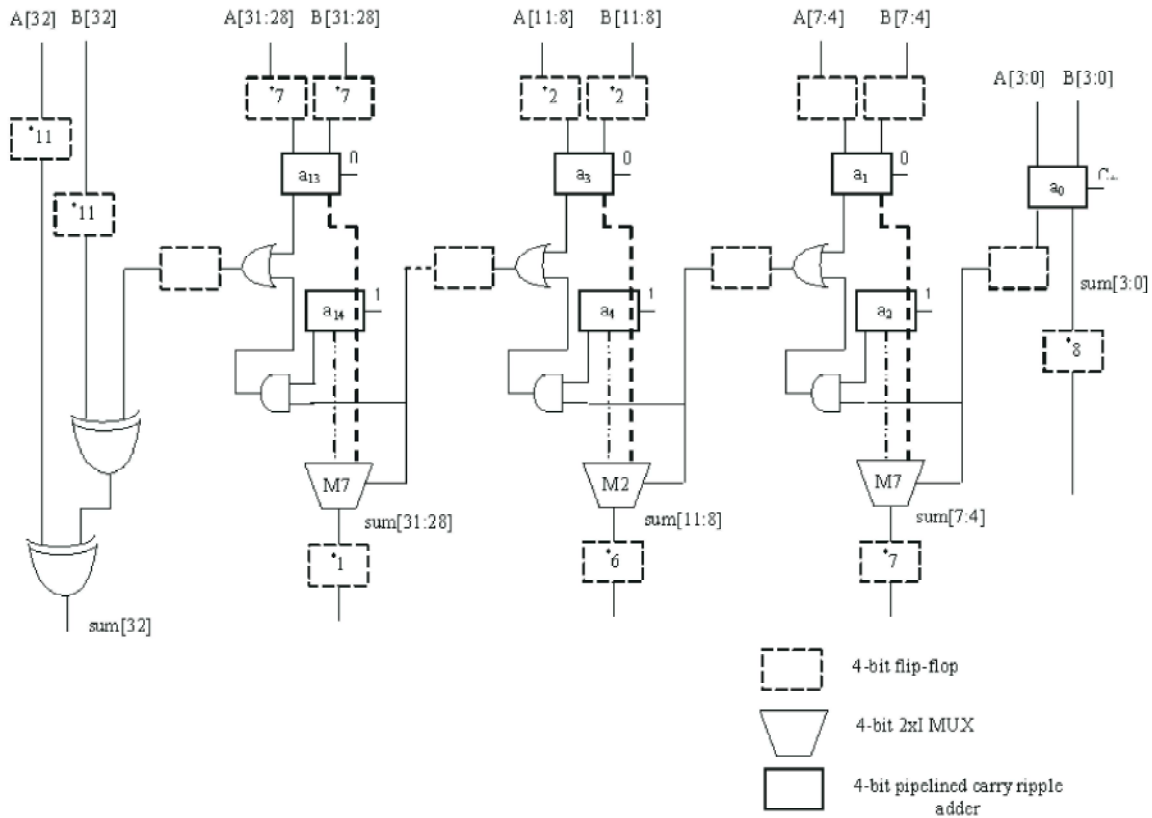


Fig. 8: 33-bit CSLA architecture

Table 1: BSS FIR with CLA

Carry Look-ahead Adder				
Evaluation	Conventional 4-bit BSS FIR Filter	Conventional 3-bit BSS FIR Filter	Proposed 4-bit BSS FIR Filter	Proposed 3-bit BSS FIR Filter
Total Logic Elements	2184	1618	2041	1535
Total Combinational Functions	748	526	655	468
Dedicated Logic Register	650	420	595	380
Memory Bits	9%	7%	5%	4%

Table 2: BSS FIR with CSA

Carry Save Adder				
Evaluation	Conventional 4-bit BSS FIR Filter	Conventional 3-bit BSS FIR Filter	Proposed 4-bit BSS FIR Filter	Proposed 3-bit BSS FIR Filter
Total Logic Elements	2184	1618	1965	1478
Total Combinational Functions	748	526	550	310
Dedicated Logic Register	650	420	537	355
Memory Bits	9%	7%	3%	2%

**Performance Evaluation:** The proposed system is developed using Verilog HDL (Hardware description Language) in Altera Quartus II software and targeted in Altera Stratix FPGA.

**Analysis with Carry Look-ahead Adder (CLA):** The BSS FIR filter with conventional adder unit consumes 2.1k and 1.6k Logic Elements (LE) for 4-bit and 3-bit representation

respectively, whereas, the BSS FIR filter with Carry Look-ahead Adder (CLA) consumes 2.0k and 1.5k Logic Elements (LE) for 4-bit and 3-bit representation respectively. In similar, the total combinational function and dedicated logic register consumed by the conventional BSS FIR filter is 748 & 650 and 526 & 420 for 4-bit and 3-bit represents respectively. Whereas, the BSS FIR filter with Carry Look-ahead Adder (CLA) consumes

655 & 595 and 468 & 380 for 4-bit and 3-bit representation respectively. Finally the memory bit consumed by the conventional BSS FIR filter is 9% and 7% for 4-bit and 3-bit representation respectively. In contrast, the BSS FIR filter with CLA adder consumes 5% and 4% as shown in Table 1.

**Analysis with Carry Save Adder (CSA):** The BSS FIR filter with conventional adder unit consumes 2.1k and 1.6k Logic Elements (LE) for 4-bit and 3-bit representation respectively, whereas, the BSS FIR filter with Carry Save Adder (CSA) consumes 1.9k and 1.4k Logic Elements (LE) for 4-bit and 3-bit representation respectively. In similar, the total combinational function and dedicated logic register consumed by the conventional BSS FIR filter is 748 & 650 and 526 & 420 for 4-bit and 3-bit representation respectively. Whereas, the BSS FIR filter with Carry Save Adder (CSA) consumes 550 & 537 and 310 & 355 for 4-bit and 3-bit representation respectively. Finally the memory bit consumed by the conventional BSS FIR filter is 9% and 7% for 4-bit and 3-bit representation respectively. In contrast, the BSS FIR filter with CSA adder consumes 3% and 2% as shown in Table 2.

### CONCLUSION

The analysis of reconfigurable Finite Impulse Response (FIR) filter in terms of adder unit is studied in this paper. In addition, the two various adders namely, Carry Look-ahead Adder (CLA) and Carry Save Adder (CSA) is evaluated in terms of total Logic Elements (LE), total combinational functions, dedicated logic register and memory bits. The Carry Look-ahead Adder (CLA) shows remarkable improvement when compared to conventional adder, whereas the performance is degraded when compared to Carry Save Adder (CSA). Hence this paper concludes that, the Carry Save Adder (CSA) is best suited for Binary Signed Sub (BSS) coefficient based reconfigurable FIR filter.

### REFERENCES

1. Macpherson, K.N., 2004. Low FPGA area multiplier blocks for full parallel FIR filters, in Proc. IEEE International Conference on Field-Programmable Technology, pp: 247-254.
2. Paschalakis, S. and P. Lee, 2003. Double Precision Floating-Point Arithmetic on FPGAs, In Proc. 2003, 2<sup>nd</sup> IEEE International Conference on Field Programmable Technology (FPT '03), Tokyo, Japan, pp: 352-358.
3. Sunesh, N.V and P. Sathishkumar, 2015. Design and Implementation of Fast Floating Point Multiplier Unit. International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA). IEEE.
4. Bharti Deepshikha and K. Anusudha, 2013. High Speed FIR Filter Based on Truncated Multiplier and Parallel Adder. International Journal of Engineering Trends and Technology (IJETT), 5(5): 243-247.
5. Renxi Gong, Zhang Hainan, Meng Xiaobi and Gong Wenying, 2009. Hardware Implementation of a High Speed Floating Point Multiplier Based on FPGA, 2009 4th International Conference on Computer Science & Education.
6. Kenny Johansson, Oscar Gustafsson, Linda S. De Brunner and Lars Wanhammar, 2011. Minimum Adder Depth Multiple Constant Multiplication Algorithm for Low Power FIR Filter, IEEE International Symposium on Circuits and Systems, pp: 1439-1442.
7. Mohamed, Asan Basiri M. and S.K. Noor Mahammad, 2014. An Efficient Hardware Based MAC Design in Digital Filters with Complex Numbers. International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp: 475-480.