

# VLSI Implementation of FIR Filter Using Computational Sharing Multiplier Based on High Speed Carry Select Adder

<sup>1</sup>S. Karunakaran and <sup>2</sup>N. Kasthuri

<sup>1</sup>Department of Electrical and Electronics Engineering,  
R.M.D Engineering College, Kavarapettai, Tamilnadu, India

<sup>2</sup>Department of Electronics and Communication Engineering,  
Kongu Engineering College, Perundurai, Tamilnadu, India

Received 2012-05-12, Revised 2012-06-03; Accepted 2012-06-03

## ABSTRACT

Recent advances in mobile computing and multimedia applications demand high-performance and low-power VLSI Digital Signal Processing (DSP) systems. One of the most widely used operations in DSP is Finite-Impulse Response (FIR) filtering. In the existing method FIR filter is designed using array multiplier, which is having higher delay and power dissipation. The proposed method presents a programmable digital Finite Impulse Response (FIR) filter for high-performance applications. The architecture is based on a computational sharing multiplier which specifically doing add and shift operation and also targets computation re-use in vector-scalar products. CSHM multiplier can be implemented by Carry Select Adder which is a high speed adder. A Carry-Select Adder (CSA) can be implemented by using single ripple carry adder and add-one circuits using the fast all-one finding circuit and low-delay multiplexers to reduce the area and accelerate the speed of CSA. An 8-tap programmable FIR filter was implemented in tanner EDA tool using CMOS 180nm technology based on the proposed CSHM technique. In which the number of transistor, power (mW) and clock cycle (ns) of the filter using array multiplier are 6000, 3.732 and 9 respectively. The FIR filter using CSHM in which the number of transistor, power (mW) and clock cycle (ns) are 23500, 2.627 and 4.5 respectively. By adopting the proposed method for the design of FIR filter, the delay is reduced to about 43.2% in comparison with the existing method. The CSHM scheme and circuit-level techniques helped to achieve high-performance FIR filtering operation.

**Keywords:** FIR Filter, Carry Select Adder, Computational Sharing Multiplier, VLSI Signal Processing

## 1. INTRODUCTION

The three most widely accepted metrics for measuring the performance of a circuit are power, delay and area. Minimizing area and delay has always been considered important, but reducing power consumption has been gaining prominence recently. With the increasing level of device integration and the growth in complexity of micro-electronic circuits, reduction of power efficiency has come to fore as a primary design

goal while power efficiency has always been desirable in electronic circuits.

Recent advances in mobile computing and multimedia applications demand high performance and low-power VLSI Digital Signal Processing (DSP) systems. One of the most widely used operations in DSP is Finite-Impulse Response (FIR) filtering. In the existing method FIR filter is designed using array multiplier, which is having higher delay and power dissipation. The proposed method presents a

**Corresponding Author:** S. Karunakaran, Department of Electrical and Electronics Engineering, R.M.D Engineering College, Kavarapettai, Tamilnadu, India

programmable digital Finite Impulse Response (FIR) filter for high performance applications. Recently, due to the high-performance requirement and increasing complexity of DSP and multimedia communication applications, FIR filters with large filter taps are required to operate with high sampling rate, which makes the filtering operation very computationally intensive.

In the proposed FIR filter architecture, the Computation Sharing Multiplier (CSHM) is efficiently used for the low-complexity design of the FIR filter. The main idea of CSHM is to represent the multiplications in the FIR Filtering operations as a combination of add and shift operations over the common computation results. The common computations are identified and those are shared without additional memory area. This sharing property enables the computation sharing multiplier approach that achieves high performance and low power in FIR filter implementation.

Due to the rapidly growing mobile industry, not only faster units but also smaller area and become major concerns for designing digital circuits. Adders are critical components of the Arithmetic Logic Unit (ALU's) or Digital Signal Processing (DSP) chips. Therefore, high performance adders with low power consumption are essential for the design of high performance processing units. Several different types of high performance adder algorithms are available in literature. Among them, Carry Look-Ahead Adder (CLA) and Carry Select Adder (CSA) are widely used for high speed operations. The proposed FIR filter was implemented by using modified Carry Select Adder (CSA) architecture to reduce area with minimum speed penalty. In the conventional CSA there are two portions of Ripple Carry Adder (RCA) which occupy large silicon area. The proposed architecture use single RCA and Add one logic circuit, this will reduce the overall area of CSA.

Park *et al.* (2000) proposed FIR filtering can be expressed as multiplication of vector by scalars. We present high-speed implementations for adaptive and nonadaptive filters based on a computation sharing multiplier which specifically targets computation re-use in vector-scalar products. The performance of the proposed implementation is compared with implementations based on carry save and Wallace tree multipliers in 0.6  $\mu\text{m}$  technology. We show that sharing multiplier scheme improves speed by approximately 30 and 21% with respect to the Wallace tree multiplier based implementation for non-adaptive and adaptive filters, respectively.

Muhammad and Roy (2002) proposed computation reduction techniques which can either be used to obtain multiplierless implementation of Finite Impulse Response (FIR) digital filters or to further improve multiplierless implementation obtained by currently used techniques. Although presented in the FIR filtering framework, these ideas are also directly applicable to any task/application which can be expressed as multiplication of vectors by scalars. The presented approach is to remove computational redundancy by reordering computation. The reordering problem is formulated using a graph in which vertices represent coefficients and edges represent resources required in a computation using the differential coefficient defined by the difference of the vertices joined by the edge. This interpretation leads to various methods for computation reduction for which simple polynomial run time algorithms are presented. It is shown that about 20% reductions in the number of add operations per coefficient can be obtained over the conventional multiplierless implementations. It is also shown that implementations requiring less than one adder per coefficient can be obtained using the presented approaches when using nonuniformly scaled coefficients quantized from infinite precision representation by simple rounding.

Park *et al.* (2004) proposed a high performance and low power fir filter design, which is based on Computation Sharing Multiplier (CSHM). CSHM specifically targets computation re-use in vector-scalar products and is effectively used in our FIR filter design efficient circuit level techniques: new carry select adder and Conditional Capture Flip-Flop (CCFF), are also used to further improve power performance.

Chang *et al.* (2005) proposed add one circuit method instead of dual carry-ripple adders. A carry select adder scheme using an add one circuit to replace one carry-ripple adder requires 29.2% fewer transistor with a speed penalty of 5.9% for bit length  $n = 64$ .

Rawat *et al.* (2002) proposed a carry select adder block and add-one circuit instead of using dual adder blocks. The add one circuit is based on "First" zero detection logic and few multiplexers. In modified CSA, one of the  $n$ -bit adder blocks is replaced by an add one circuit consisting of fewer transistors. This scheme considerably reduces the power and area with negligible speed penalty. For 8-bit length  $n = 8$ , this modified CSA requires 38% fewer transistors and consumes only 73% of power compared to conventional design using a 0.5 micron CMOS technology.

Jeong *et al.* (2004) developed a Dual Transition Skewed Logic (DTSL) based Carry Select Adder (CSA) suitable for processing units requiring low power and high performance with high noise immunity. We implemented 31-bit carry select adders in three different logic styles: Dual Transition Skewed Logic (DTSL), Domino and conventional static CMOS in TSMC 0.25 $\mu$ m technology and compared them in terms of performance, power consumption and layout area. CSA using DTSL shows 36.7 and 17.7% improvements in power dissipation and performance, respectively, over domino and 40.4% improvement in performance compared to a static CMOS CSA.

Yiran *et al.* (2005) developed a novel low-power Carry-Select Adder (CSA) design called Cascaded CSA ( $C^2$  SA). Based on the prediction of the critical path delay of current operation;  $C^2$  SA can automatically work with one or two clock-cycle latency and a scaled supply voltage to achieve power improvement. Post-layout simulations of a 64-bit  $C^2$  CSA in 180 nm Technology show that SA. Can operate at a lower supply voltage, attaining 40.7% energy saving, while maintaining a similar (average) Latency Per Operation (LPO) compared to standard CSA.

Amelifard *et al.* (2005) are proposed a method of sharing two adders used in the Carry Select Adder (CSA), a new design of a low-power high performance adder is presented. The new adder is faster than a Ripple Carry Adder (RCA), but slower than a CSA. On the other hand, its area and power dissipation are smaller than those of a CSA.

Jeong *et al.* (2001) developed a Carry-select method has deemed to be a good compromise between cost and performance in carry propagation adder design. However, conventional Carry-Select adder (CSL) is still area-consuming due to the dual ripple carry adder structure. The excessive area overhead makes CSL relatively unattractive but this has been circumvented by the use of add-one circuit introduced recently. The proposed CSL witnesses a notable power-delay and area-delay performance improvement by virtue of proper exploitation of logic structure and circuit technique.

Gerosa (1994) proposed an 80 MHz structured-custom RISC microprocessor design. This 32-bit implementation of the PowerPC architecture is fabricated in a 3.3 V, 0.5  $\mu$ m, 4-level metal CMOS technology, resulting in 1.6 million transistors in a 7.4 mm by 11.5 mm chip size. Low-power design techniques are used throughout the entire design, including dynamically powered down execution units. Typical power dissipation is kept under 2.2 W at 80 MHz. Three distinct levels of

software-programmable, static, low-power operation-for system power management are offered, resulting in standby power dissipation from 2 mW to 350 mW.

Hartley (1996) suggested a common way of implementing constant multiplication is by a series of shift and add operations. As is well known, if the multiplier is represented in Canonical Signed Digit (CSD) form, then the number of additions (or subtractions) used will be a minimum. This study examines methods for optimizing the design of CSD multipliers and in particular the gains that can be made by sharing sub expressions.

Klass (1998) suggested a family of semi-dynamic and dynamic edge-triggered flip-flops to be used with static and dynamic circuits, respectively. The flip-flops provide both short latency and the capability of incorporating logic functions with minimum delay penalty, properties which make them very attractive for high-performance microprocessor design.

Kong *et al.* (2000) proposed a family of novel low-power flip-flops, collectively called Conditional-Capture Flip-Flops (CCFFs). They achieve statistical power reduction by eliminating redundant transitions of internal nodes. These flip-flops also have negative setup time and thus provide small data-to-output latency and attribute of soft-clock edge for overcoming clock skew-related cycle time loss. The simulation comparison indicates that the proposed differential flip-flop achieves power savings of up to 61% with no impact on latency while the single-ended structure provides the maximum power savings of around 67%, as compared to conventional flip-flops.

Lim and Parker (1983) suggested FIR digital filters with discrete coefficient values selected from the powers-of-two coefficient space are designed using the methods of integer programming. The frequency responses obtained are shown to be superior to those obtained by simply rounding the coefficients. Both the weighted minimax and the weighted least square error criteria are considered. Using a weighted least square error criterion, it is shown that it is possible to predict the improvement that can be expected when integer quadratic programming is used instead of simple coefficient rounding.

Neve *et al.* (2004) suggested methods to minimize the power-delay product of 64-bit carry-select adders intended for high-performance and low-power applications. A first realization in 0.18  $\mu$ m Partially Depleted (PD) Silicon-On-Insulator (SOI), using complex Branch-Based Logic (BBL) cells, results in a delay of 720 ps and a power dissipation of 96mW at 1.5 V.

Nikolic *et al.* (1999) suggested Timing elements, latches and flip-flops, are critical to performance of

digital systems, due to tighter timing constraints and low power requirements. Short setup and hold times are essential, but often overlooked. Recently reported flip-flop structures achieved small delay between the latest point of data arrival and output transition. Typical representatives of these structures are Sense Amplifier-Based Flip-Flop (SAFF), Hybrid Latch-Flip-Flop (HLFF) and Semi-Dynamic Flip Flop (SDFF). Hybrid flip-flops outperform reported sense amplifier-based designs, because the latter are limited by the output latch implementation. SAFF consists of the sense amplifier in the first stage and the RS latch in the second stage.

Partovi *et al.* (2001) suggested that in low-energy, constant throughput system, the supply voltage is often scaled down to minimize the energy consumption. The design of the clocking subsystem-register elements and clock distribution network-has to be resistant to noise and timing failures for robust

circuit operation. Noise robust designs are usually fully static or pseudo-static.

## 2. MATERIALS AND METHODS

### 2.1. Architecture of a proposed FIR filter

The input-output relationship of the Linear Time Invariant (LTI) FIR filter can be described as:

$$Y(n) = \sum_{k=0}^{M-1} C_k \cdot x(n-k)$$

Where M represents the length of the FIR filter, they are the filter coefficients and  $x(n-k)$  denotes the data sample at time constant(n-k). **Figure 1** shows a Transposed Direct Form (TDF) implementation of the FIR filter.

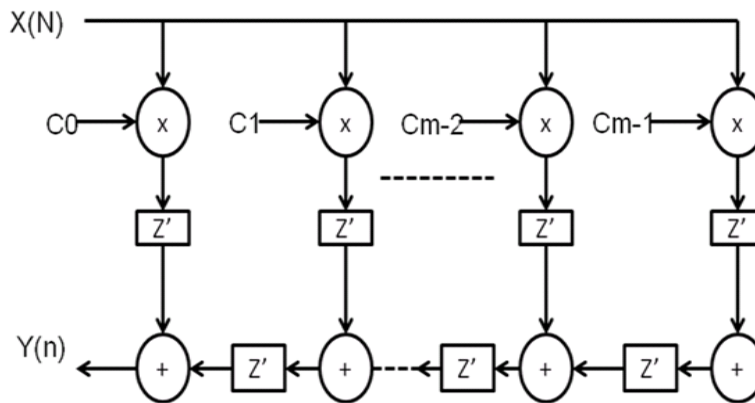


Fig. 1. Transposed Direct (TDF) FIR filter

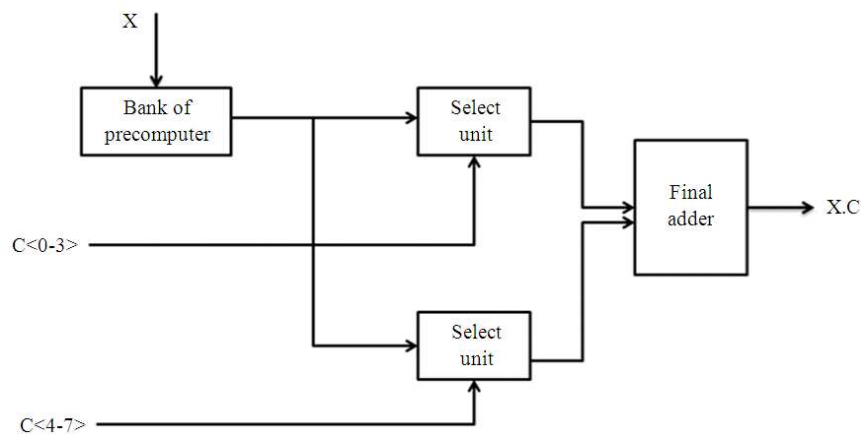
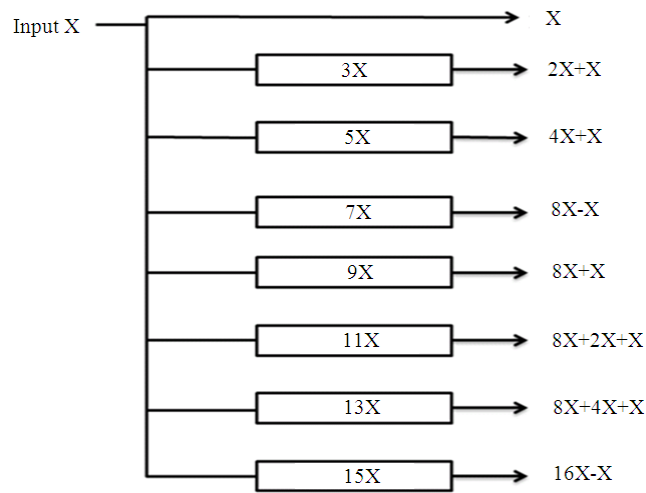
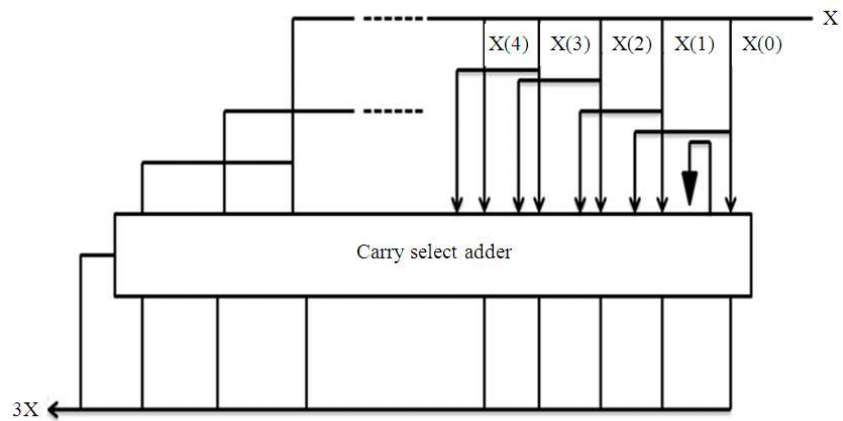


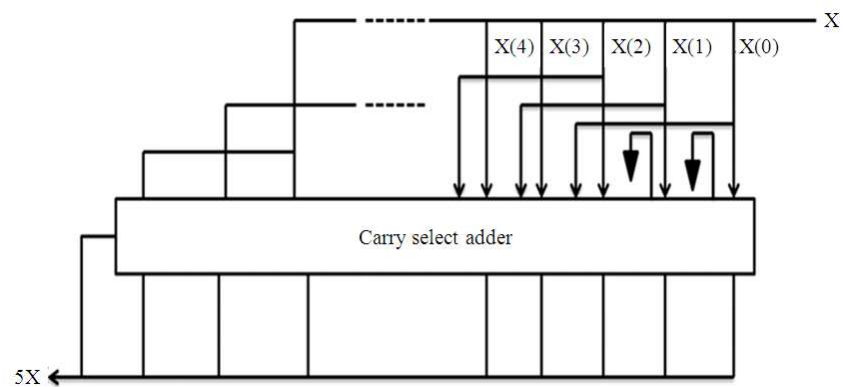
Fig. 2. CSHM architecture



**Fig. 3.** The precomputer unit



**Fig. 4.** Block diagram of 3x precomputer unit



**Fig. 5.** Block diagram of 5x precomputer unit

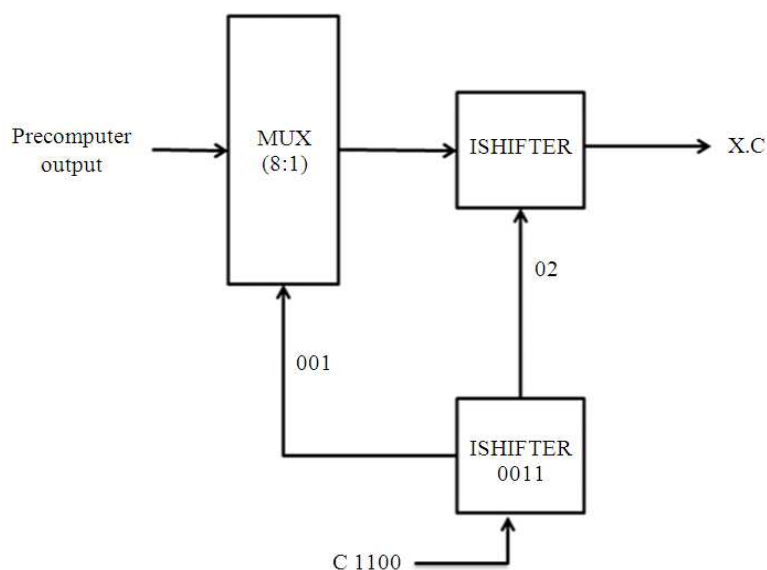


Fig. 6. Block diagram of selector unit

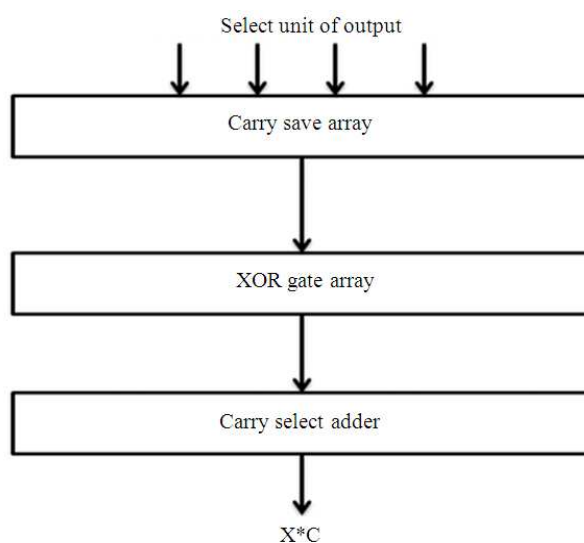


Fig. 7. Block diagram of final adder unit

We notice that the TDF implements a product of the coefficient vector with the scalar by all the coefficients simultaneously. In the sequence, such products will be referred to as a vector scaling operation.

In the vector scaling operations we can carefully select a set of small bit sequences so that the same multiplication result can be obtained by only add and shift Operations.

The precomputer performs the multiplication of alphabets with input. Since alphabets are small bit

sequences, the multiplication with input and alphabets can be done without seriously compromising the performance. The architecture of CSHM is shown in Fig. 2. The advantage of CSHM is that once the multiplications of alphabets with input are calculated by the precomputer, the outputs are shared by the entire S&As. In order to cover every possible coefficient and perform general multiplication operation, we used eight alphabets in the precomputer.

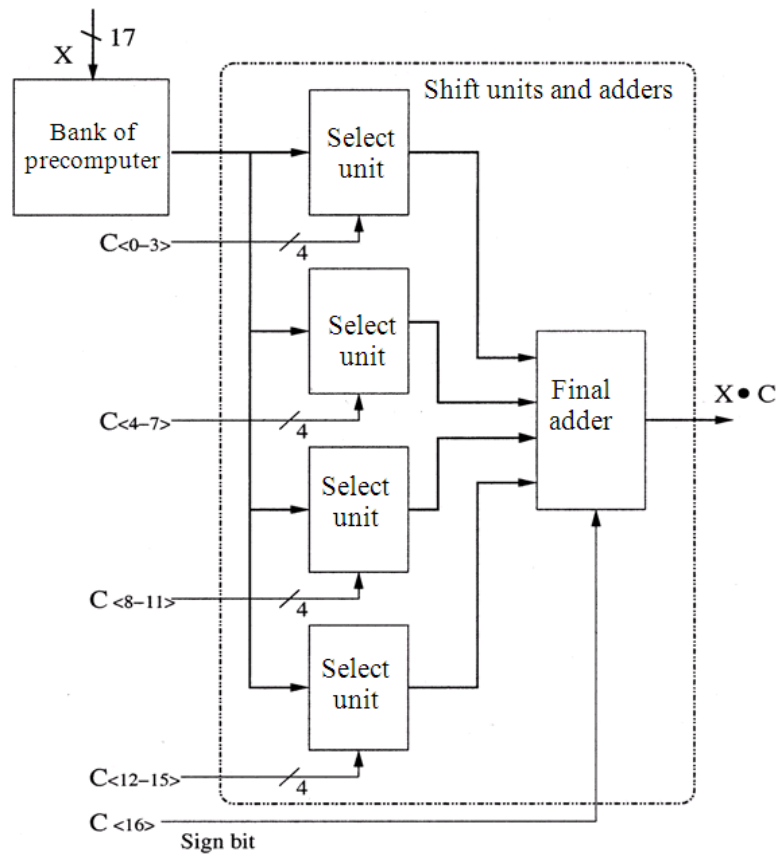


Fig. 8. Block diagram of 17×17 CSHM unit

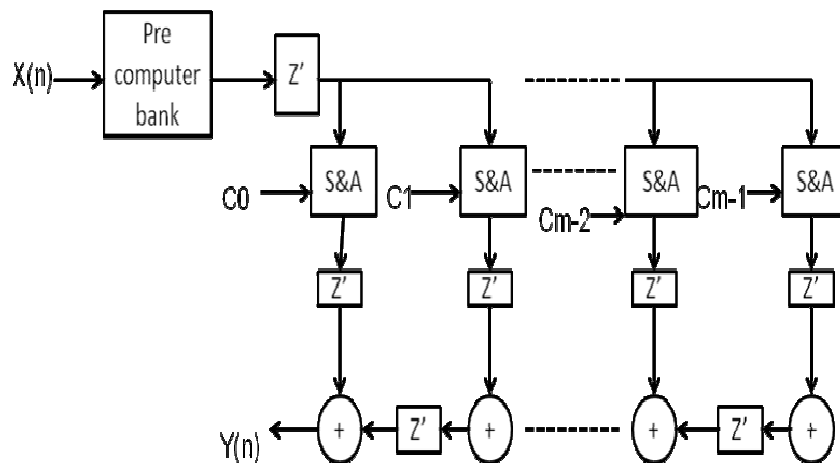


Fig. 9. Architecture of FIR filter based on CSHM



## 2.2. Precomputer

The multiplications performed by the precomputer are simply implemented using the new carry-Select adder, which is proposed. **Figure 3** shows the basic structures of and the precomputer structure. **Figure 4 and 5** are block diagrams of 3x and 5x precomputer.

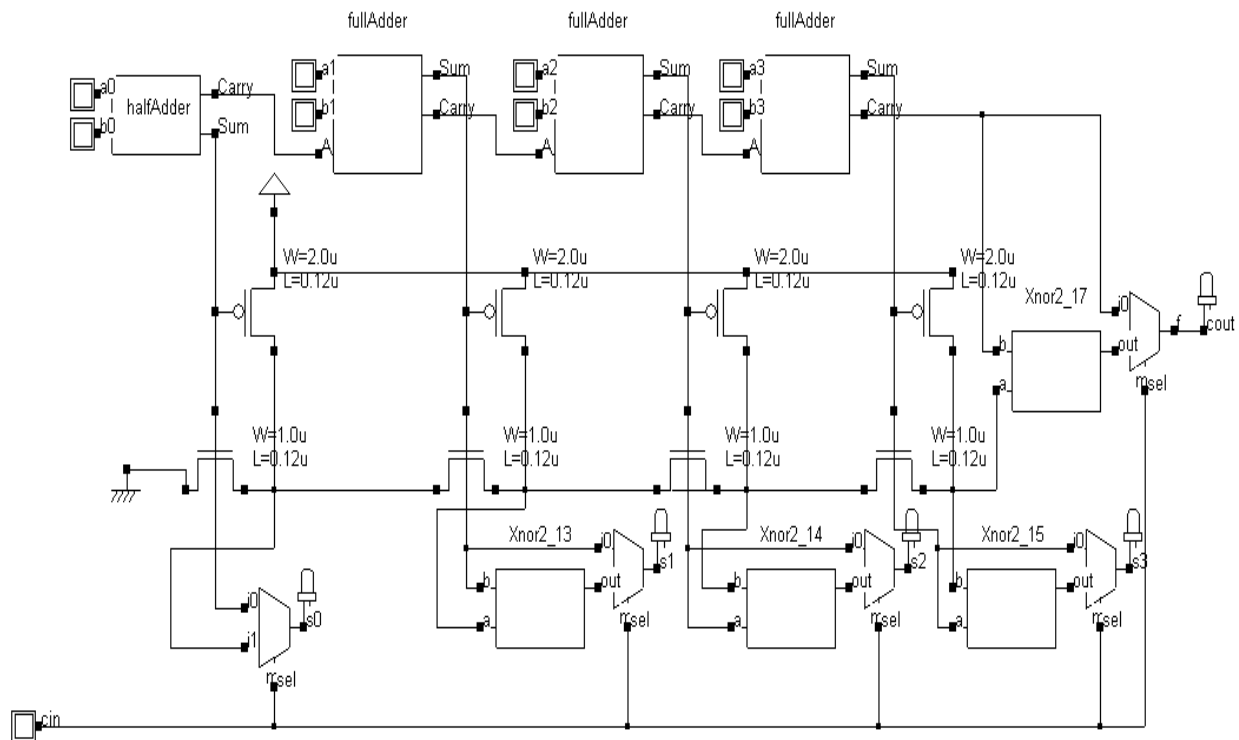
S&As perform appropriate select/shift and add operations required to obtain the multiplication output. The select unit is composed of SHIFTER, MUX (8:1), ISHIFTER and AND gate. To find the correct alphabet, SHIFTERS perform the right shift operation until they encounter 1 and send an appropriate select signal to MUXes (8:1). SHIFTERS also send the exact shifted values (shift signal) to ISHIFTERS. The MUXes (8:1) select the correct answer among the eight precomputer outputs, ISHIFTERS simply inverse the operation performed by SHIFTERS (barrel shifter). When the coefficient input is 0000, we cannot obtain a zero output with shifted value of the precomputer outputs. Simple AND gates are used to deal with the zero (0000) coefficient input. **Figure 7** shows the final adder unit. The final adder adds the outputs of the select units to obtain the final multiplication output.

## 2.3. Select Unit

As shown in **Fig. 6**, the select unit is composed of SHIFTER, MUX, ISHIFTER and AND gates. Since SHIFTER is directly connected to the coefficients, it does not lie on the critical path. Static CMOS design with minimum size is used for SHIFTER implementation. ISHIFTER lies on the critical path and the maximal shift width is 3 bits. A barrel shifter is used since the signal has to pass through at most one transmission gate in the barrel shifter. The MUX using pass-transistor logic was implemented to achieve a compact and high-speed design.

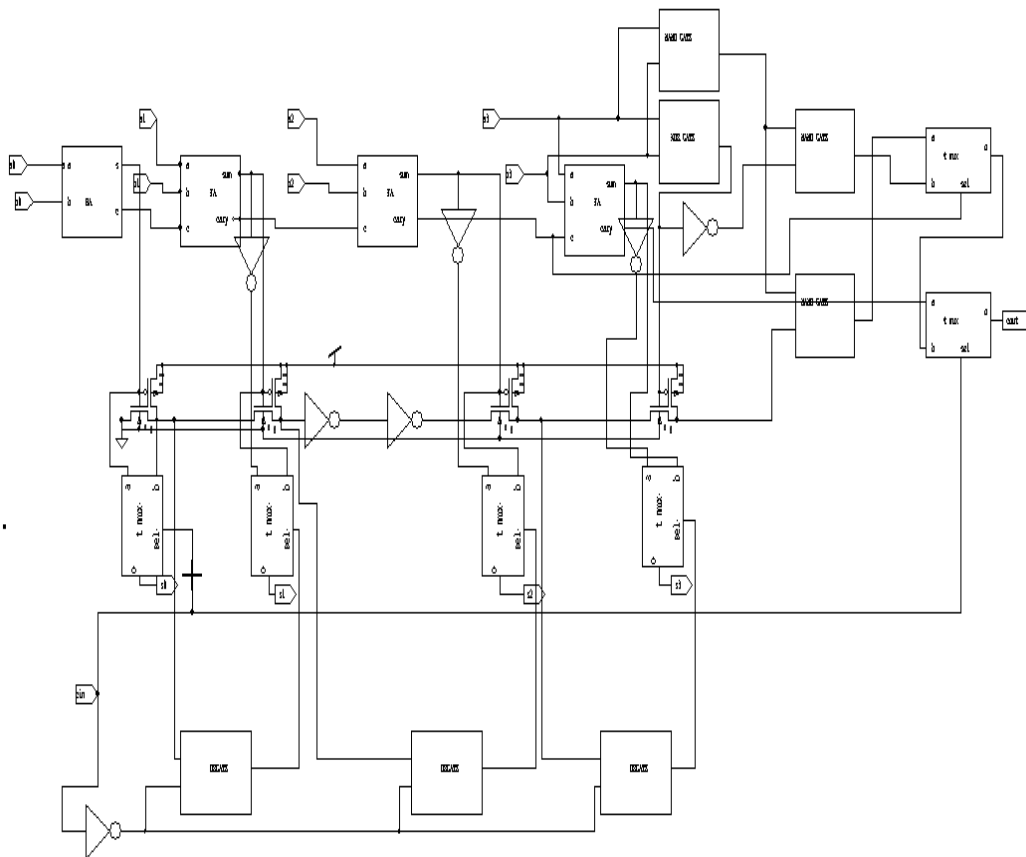
## 2.4. Final Adder

The final adder is the largest component in the S&A, which sums the outputs of four select units. The carry-save array and the new carry-select adder presented are used for high performance. As mentioned before, the input data is in two's complement format, the coefficient in sign and magnitude and the final adder output in two's complement.



**Fig. 10.** Carry select adder using add one circuit





**Fig. 11.** Proposed CSA with fast all one finding circuit

The  $17 \times 17$  CSHM, shown in **Fig. 8**, is implemented using 180-nm TSMC standard cell library. In our CSHM implementation, the input is represented in two's complement format and coefficient is in sign and magnitude format. The output of the CSHM is also in two's complement format.

In our CSHM design shown in **Fig. 8**. The sign bit of coefficient is not used and is considered as a positive number in the select unit. The XOR gate array is efficiently used for controlling the sign of the final adder output. When the coefficient is a positive number (when the sign bit is '0'), since the output of the final adder has the same sign as input data, the inputs of final adder can be added without sign conversion. When the coefficient is a negative number (when the sign bit is '1'), since the output of the final adder has a different sign than the input data, the inputs of the final adder should be converted to numbers with the opposite sign. The architecture is easily realized using the XOR gate array. The addition of the coefficient sign bit and input Least Significant Bit (LSB) can be merged into the carry-select adder.

## 2.5. FIR Filter Based on CSHM

Using the  $17 \times 17$  CSHM, a 10-tap FIR filter with programmable coefficients has been implemented for fabrication. FIR filter can be implemented in Direct Form (DF) or Transposed Direct Form (TDF) architecture (**Fig. 1**). In the DF FIR filter, a large adder in the final stage lies on the critical path and it slows down the FIR filter. For high-performance filter structure, TDF is used in our implementation. In the TDF of the FIR filter shown in **Fig. 1**, multipliers are replaced by S&As and a precomputer is connected to the input. Therefore, as shown in **Fig. 9**, the FIR filter using CSHM consists of one precomputer and ten S&As. We can easily see from the figure that the precomputer outputs are shared by all the S&As. In other words, the computations are performed only once for all's and these values are shared by the entire S&As for generating. The CSHM scheme efficiently removes the redundant computations in the FIR filtering operation, which leads to low-power and high-performance design.

## 2.6. High-Performance Carry-Select Adder Using Fast all one Finding Logic

The blocks in the conventional carry-select adder consists of two ripple carry adders, one for  $C_{in} = 0$  and the other for  $C_{in} = 1$ . If the results for  $= 0$  is known as  $S^0$  the result for  $C_{in} = 1$  ( $S^1$ ) can be obtained by adding one to  $S^0$ . Thus, an add-one circuit can replace the ripple-carry adder for  $C_{in} = 1$  to reduce the area in a block. To design an efficient add-one circuit, the first zero finding circuit is showed in the Fig. 10. In his figure shows Adding one to the result for  $C_{in} = 0(S^0)$ , if the is the  $S_k^0$  first zero count from the least significant bit, the  $S^1$  is just inverting each bit of  $S^0$  starting from the least significant bit until the  $S_k^0$  bit (included) and other bit(s) remain the same.

In other words, the carry-out signal for the add-one circuit is one if and only if all the sum outputs from the n bit block are one. As all sums equal one, the first zero detection circuit generates one at the final node. For all the other cases, it generates a zero carry-out.

## 2.7. An Add One Circuit to Replace One RCA

The 4-bit add-one circuit architecture used by Chang *et al.* (2005) is showed in the Fig. 11. The Full Adder (FA) cell consists of a two-level NAND gate for carry output and two-level two-input exclusive-or gates with the critical delays. And the delay in the unit of the

two input NAND gate was illustrated in Fig. 11. The carry-chain is the critical path in the CSA, so the critical path increase 1.5 units in every block compared with the original RCA structure.

## 3. RESULTS AND DISCUSSION

### 3.1. Tanner EDA

Tanner EDA is a leading provider of Electronic Design Automation (EDA) software solutions for the design, layout and verification of Analog/Mixed signal ICs and MEMS. This tool helps to automate and simplify the design process, enabling engineers to cost-effectively bring out commercially successful electronic products to the market ahead of competition.

### 3.2. Results of Carry Select Adder

#### 3.2.1. Conventional CSA using tanner tool

The Fig. 12 shows the output waveform of conventional CSA, in which “s0, s1, s2, s3, cout” are outputs. The input parameters are namely “A0, A1, A2, A3, B0, B1, B2, B3, Cin”. When the all the inputs are high means all outputs are equal to high.

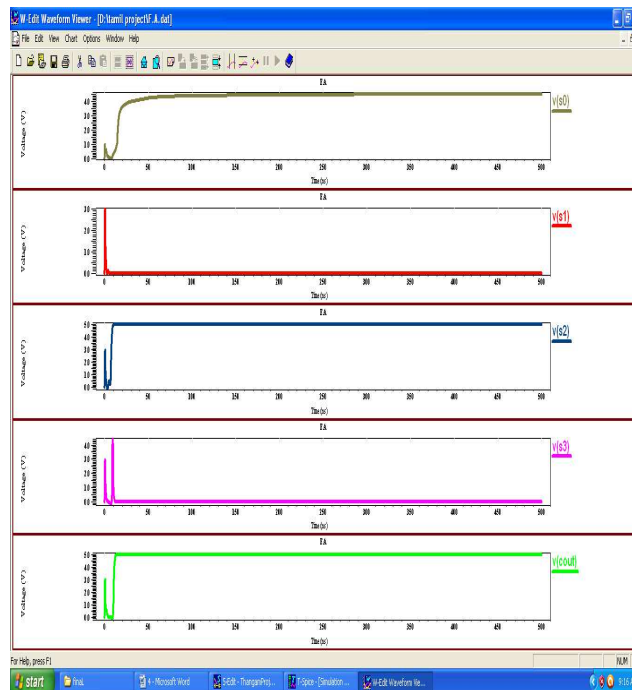


Fig. 12. Output of conventional CSA

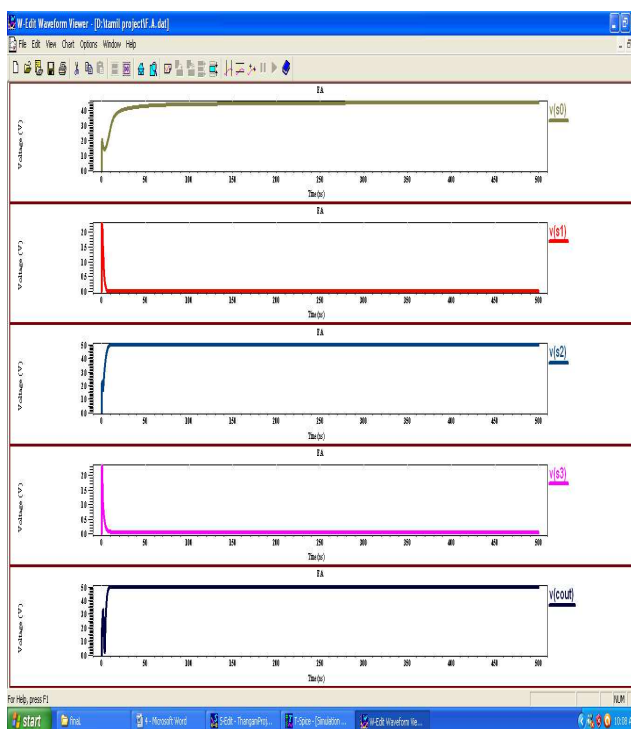


Fig. 13. Output of proposed CSA using tanner tool

### 3.3. Proposed CSA using tanner tool

The Fig. 15 shows the output waveform of DTSL CSA, in which “out0, out1, out2, out3, cout” are outputs. The input parameters namely “A0, A1, A2, A3, B0, B1, B2, B3, Cin”. When the all the inputs are high means all outputs are equal to high. When ‘cin’ zero means all the outputs are equal to high except ‘out0’. When ‘a’ (a0-a3) and cin are all zero means all outputs are equal to one except ‘cout’.

The Fig. 13 and 14 are shows simulation of CSA in different logic styles namely conventional, proposed (fast all one finding logic) respectively. The all styles are working functionality is same but proposed CSA of propagation delay is reduced and the number of transistors are reduced. The Table 1 shows the comparisons of carry select adder.

### 3.4. Multiplier Using Conventional CSA

The conventional carry select adder simulation waveforms are shown in Fig. 15, in which “out0, out1, out2, out3, out4, out5, out6, out7, out8, out9, out10, out11 and out12” are outputs. When any one of the input all the bits are zero means the outputs are zero. when the input ‘x’ is “0100” and ‘y’ is “11100100” means the outputs are “1110010000”.

Table 1. Comparisons of carry select adder

Types of Adder	Delay (ns)	Number of Transistors	Power (mw)
4-Bit Conventional carry select adder	12.36	294	7.34
4-Bit Proposed carry select adder	8.25	190	4.46

### 3.5. Multiplier Using Proposed CSA

Figure 16 shows simulation of multiplier output waveform using proposed carry select adder, in which out0, out1, out2, out3, out4, out5, out6, out7, out8, out9, out10 and out11 are the outputs of the multiplier.. This functionality is same as multiplier using conventional CSA but the propagation delay reduced.

### 3.6. Results of fir filter

Results of 8-tap FIR filter using conventional CSA

The Fig. 17 shows 8 tap fir filter using conventional carry select adder. In that table one input parameter is x with 4bits namely x0, x1, x2 and x3 and another one input is coefficient c0, c1, c2 and c3 are all 8bits size. The output parameters are out0, out1, out2, out3, out4, out5, out6, out7, out8, out9, out10, out11 and out12.

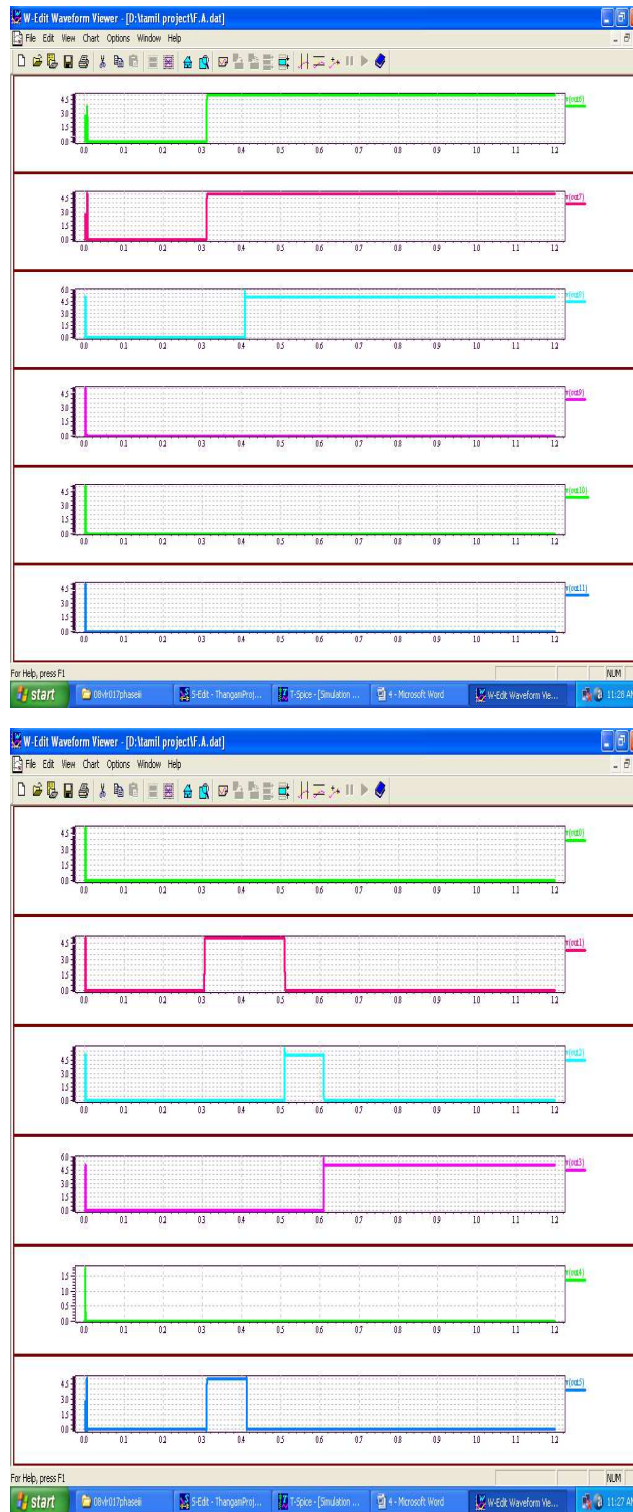


Fig. 14. Output waveform of multiplier using conventional CSA

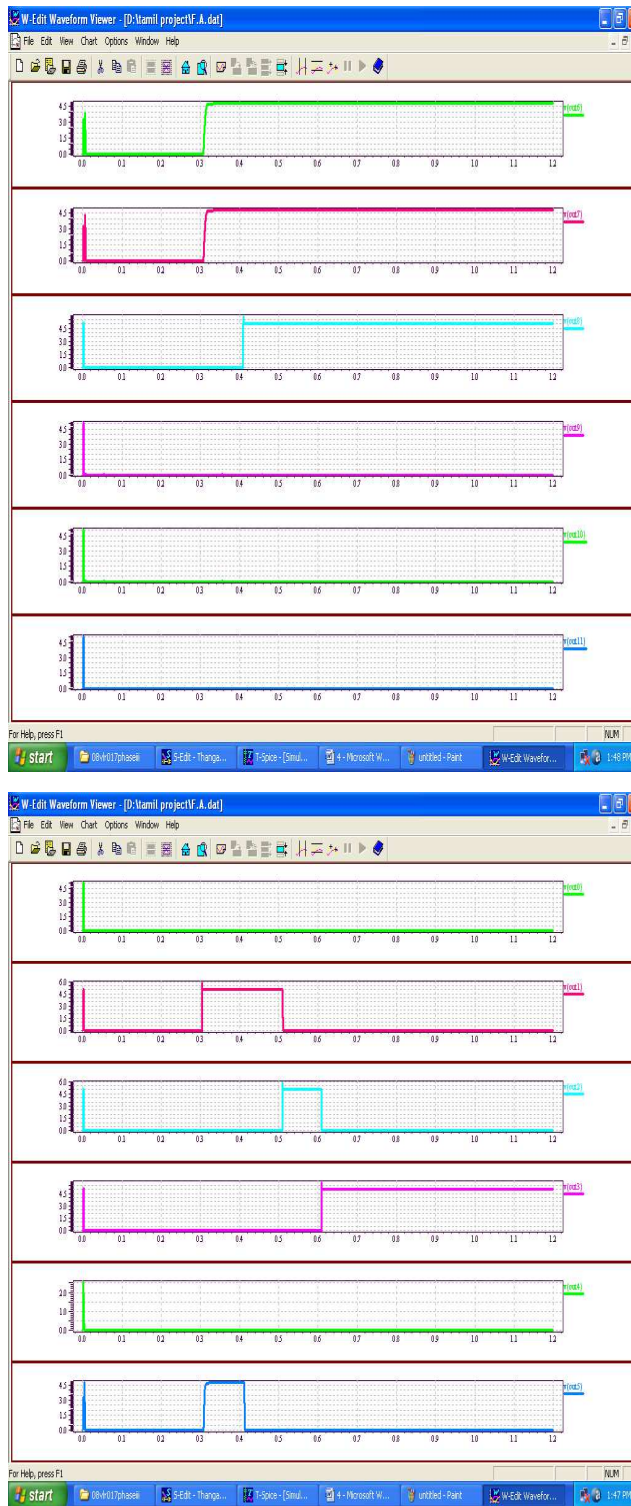


Fig. 15. Output waveform of multiplier using proposed CSA



Fig. 16. Output waveform of 8-tap FIR using conventional CSA





Fig. 17. Output waveform of 4-tap FIR using proposed CSA



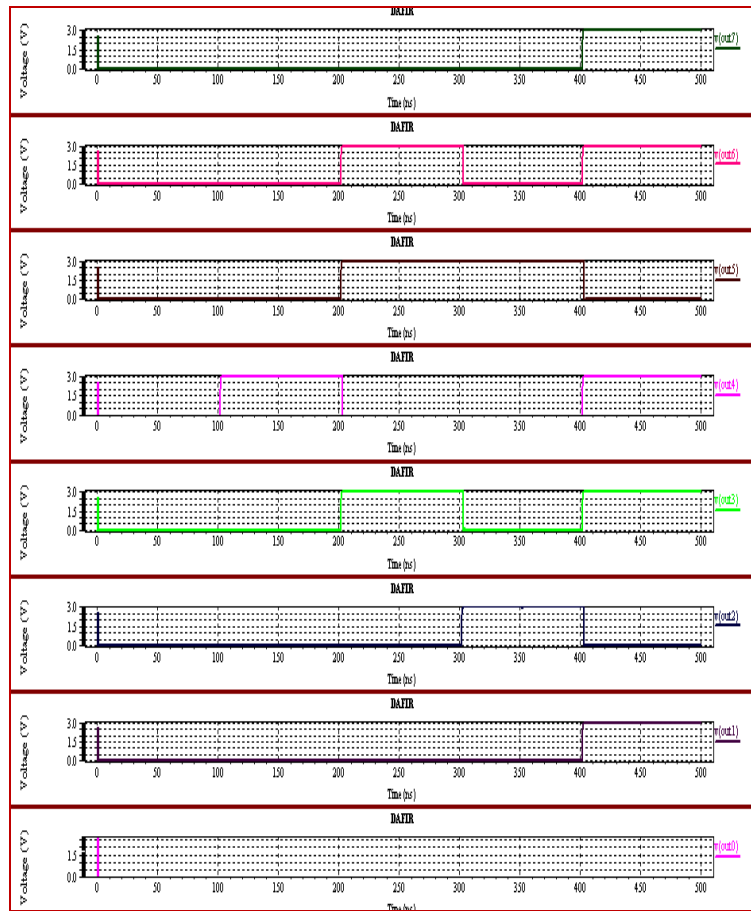


Fig. 18. Output waveform of 8-tap FIR using proposed CSA

During the first clock pulse to simulate the one fir filter (one multiplication) and then second pulse to simulate second tap (second multiplication) and this result add with first multiplication output and so on so.

### 3.7. Results of 8-Tap FIR Filter Using Proposed CSA

The Fig. 18 shows 8 tap fir filter using proposed carry select adder. In that table one input parameter is  $x$  with 4bits namely  $x_0, x_1, x_2$  and  $x_3$ . And another one input is coefficient  $c_0, c_1, c_2$  and  $c_3$  are all 8bits size. The output parameters are  $out_0, out_1, out_2, out_3, out_4, out_5, out_6, out_7, out_8, out_9, out_{10}, out_{11}$  and  $out_{12}$ .

All the three techniques namely conventional, proposed carry select adder are used in filter. All working functionality is same but proposed adder technique of propagation delay is reduced. And also area will be 10% are reduced.

Table 2. Comparisons of FIR filter results

Parameter	Filter using array multiplier	Filter using CSHM
Area (no.of transistor)	6000.000	23500.00000
Power (mw)	3.732	2.62732
Clock cycle (ns)	9.000	4.50000

The Table 2 shows comparisons of FIR filter results, in which the number of transistor, power (mW) and clock cycle (ns) of the filter using array multiplier are 6000, 3.732 and 9 respectively. The FIR filter using CSHM in which the number of transistor, power (mW) and clock cycle (ns) are 23500, 2.627 and 4.5 respectively.

## 4. CONCLUSION

The proposed method presents a programmable digital Finite Impulse Response (FIR) filter for high-

performance applications. The architecture is based on a Computation Sharing Multiplier (CSHM) which specifically doing add and shift operation and also targets computation re-use in vector-scalar products. CSHM multiplier can be implemented by Carry Select Adder which is a high speed adder. A Carry-Select Adder (CSA) can be implemented by using single ripple carry adder and add-one circuits using the fast all-one finding circuit and low-delay multiplexers to reduce the area and accelerate the speed of CSA.

A 4-tap programmable FIR filter was implemented in tanner EDA tool using CMOS 180nm technology based on the proposed CSHM technique. In which the number of transistor, power (mW) and clock cycle (ns) of the filter using array multiplier are 6000, 3.732 and 9 respectively. The FIR filter using CSHM in which the number of transistor, power (mW) and clock cycle (ns) are 23500, 2.627 and 4.5 respectively. By adopting the proposed method for the design of FIR filter, the delay is reduced to about 43.2% in comparison with the existing method. The CSHM scheme and circuit-level Techniques helped to achieve high-performance FIR filtering operation. The proposed CSHM architecture is also applicable to adaptive filter and matrix multiplication implementation.

## 5. REFERENCES

- Amelifard, B., F. Fallah and M. Pedram, 2005. Closing the gap between carry select adder and ripple carry adder: A new class of low-power high-performance adders. Proceedings of the 6th International Symposium on Quality of Electronic Design, Mar. 21-23, IEEE Xplore Press, pp: 148-152. DOI: 10.1109/ISQED.2005.131
- Chang, C.H., Y. He and J. Gu, 2005. An area efficient 64-bit square root carry-select adder for low power applications. Proceedings of the IEEE International Symposium on Circuits and Systems, May 23-26, IEEE Xplore Press, pp: 4082-4085. DOI: 10.1109/ISCAS.2005.1465528
- Gerosa, G., 1994. A 2.2-W 80-MHz superscalar RISC microprocessor. IEEE J. Solid-State Circ., 29: 1440-1452.
- Hartley, R.I., 1996. Subexpression sharing in filters using canonic signed digit multipliers. IEEE Trans. Circ. Syst., 43: 677-688. DOI: 10.1109/82.539000
- Jeong, W., H. Mahmoodi-Meimand, Y. Wang, H. Choo and K. Roy, 2004. Computation sharing programmable FIR filter for low-power and high-performance applications. IEEE J. Solid-State Circuits, 39: 348-357. DOI: 10.1109/JSSC.2003.821785
- Jeong, W., K. Roy and C. Koh, 2001. High-performance low-power carry-select adder using dual transition skewed logic. Proceedings of the 27th European Solid-State Circuits Conference, Sept. 18-20, IEEE Xplore Press, Villach, Austria, pp: 145-148.
- Klass, F., 1998. Semi-dynamic and dynamic flip-flops with embedded logic. Proceedings of the Symposium on VLSI Circuits, Jun. 11-13, IEEE Xplore Press, Honolulu, HI., pp: 108-109. DOI: 10.1109/VLSIC.1998.688018
- Kong, B.S., S.S. Kim and Y.H. Jun, 2000. Conditional capture flip-flop technique for statistical power reduction. Proceedings of the IEEE International Solid-State Circuits Conference, Feb. 07-09, IEEE Xplore Press, San Francisco, CA., pp: 290-291. DOI: 10.1109/ISSCC.2000.839785
- Lim, Y.C. and S.R. Parker, 1983. FIR filter design over a discrete powers-of-two coefficient space. IEEE Trans. Acoust., Speech Signal Proces., 31: 583-591. DOI: 10.1109/TASSP.1983.1164085
- Muhammad, K. and K. Roy, 2002. A graph theoretic approach for synthesizing very low-complexity high-speed digital filters. IEEE Trans. Comp.-Aided Des. Integr. Circ. Syst., 21: 204-216. DOI: 10.1109/43.980259
- Neve, A., H. Schettler and T. Ludwig, 2004. Power-delay product minimization in high performance 64-bit carry-select adders. IEEE Trans. Very Large Scale Integr. Syst., 12: 235-244. DOI: 10.1109/TVLSI.2004.824305
- Nikolic, B., V. Stojanovic, V.G. Oklobdzija, J. Wenyan and J. Chiu *et al.*, 1999. Sense amplifier-based flip-flop. Proceedings of the IEEE International Solid-State Circuits Conference, Feb. 15-17, IEEE Xplore Press, San Francisco, CA., pp: 282-283. DOI: 10.1109/ISSCC.1999.759251
- Park, J., H. Choo, K. Muhammad and K. Roy, 2000. Non-adaptive and adaptive filter implementation based on sharing multiplication. Proceeding of the IEEE International Conference Acoustics, Speech and Signal Processing, Jun. 05-09, IEEE Xplore Press, Istanbul, pp: 460-463. DOI: 10.1109/ICASSP.2000.862012
- Park, J., W. Jeong, H. Mahmoodi-Meimand, Y. Wang and H. Choo *et al.*, 2004. Computation sharing programmable FIR filter for low-power and high-performance applications. IEEE J. Solid-State Circ., 39: 348-357. DOI: 10.1109/JSSC.2003.821785

- Partovi, H., A. Chandrakasan, W.J. Bowhill and F. Fox, 2001. Clocked Storage Elements. In: Design of High-Performance Microprocessor Circuits, Chandrakasan, A.P., F. Fox and W.J. Bowhill (Eds.), 1st Edn., IEEE Press, New York, ISBN-10: 078036001X, pp: 207-234.
- Rawat, K., T. Darwish and M. Bayoumi, 2002. A low power and reduced area carry select adder. Proceedings of the 45th Midwest Symposium on Circuits and Systems, Aug. 4-7, IEEE Xplore Press, pp: 467-470. DOI: 10.1109/MWSCAS.2002.1187259
- Yiran, C., H. Li, K. Roy and K.C. Kok, 2005. Cascaded carry-select adder ( $C^2SA$ ): A new structure for low-power CSA design. Proceedings of the International Symposium on Low Power Electronics and Design, Aug. 8-10, IEEE Xplore Press, pp: 115-118. DOI: 10.1109/LPE.2005.195498