An Efficient VLSI Architecture for FIR Filter using Computation Sharing Multiplier

Thamizharasan .V
Assistant Professor
Department of Electronics and Communication Engineering
Erode Sengunthar Engineering College,
Perundurai,Erode.

Parthipan.V
Assistant Professor
Department of Electronics and Communication Engineering
Erode Sengunthar Engineering College,
Perundurai, Erode.

.

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 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. An 4-tap programmable FIR filter was implemented in tanner EDA tool using CMOS 180nm technology based on the proposed CSHM technique. 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.

***Index Terms*—**Computation sharing, dual transition skewed logic, programmable finite impulse response (FIR) filter.

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-elctronic 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 programmable digital finite impulse response (FIR) filter for high performance applications. The FIR filter performs the weighted summations of input sequences and is widely used in video convolution functions, signal preconditioning, and various communication 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. Canonical-signed-digit and signed-power-of-two coefficient representations are widely used in the parallel implementation of FIR filters. Using those Techniques, the FIR filtering operation can be simplified to add and shift operations. Common sub expressions elimination and differential coefficients method also explore low-complexity design of FIR filters by minimizing the number of additions in filtering operations.

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.

On the other hand, Carry Look-ahead Adders (CLAs) are the fastest adders (O (log (n) time), but they are the worst from the area point of view (O (log (n)) area). Carry Select Adders (CSAs) have been considered as a compromise solution between RCAs and CLAs (O (n) time and O (2n) area) because they offer a good tradeoff between the compact area of RCAs and the short delay of CLAs. As a result, some effort has been done to improve the efficiency of this kind of adder. Due to the rapidly growing mobile industry, not only faster units but also smaller area and become major concerns for designing digitial circuits. Adders are critical components of the ALU's (Arithmetic Logic Unit) or DSP (Digital Signal Processing) 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 circuit style is as important as the adder algorithm and architecture. Traditionally, static CMOS circuits have been mostly used in adder designs. However, as the demand for high performance is increasing, several new circuit techniques such as Domino have been used. However, noise immunity of Domino circuits is worse than static CMOS circuits (especially for scaled technologies) and they consume larger power than static CMOS. Precharge/Evaluate logic using static CMOS technology (e.g. skewed CMOS circuits) is one solution to achieve high performance with low power consumption and good noise immunity.

This paper was modified carry select adder (CSA) architecture to reduce area with minimum speed penalty.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, this reduce overall area of CSA.

2. FIR FILTER ARCHITECTURE

2.1 CSHM Algorithm and architecture

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, the c_k are the filter coefficients and x(n-k) denotes the data sample at time constant(n-k). Fig.1 shows a transposed direct form (TDF) implementation of the FIR filter. We notice that the TDF implements a product of the coefficient vector with the scalar by all the coefficients simultaneously. In the sequel, 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. For instance, a simple vector scaling operation can be decomposed as. If, and are available, the entire multiplication process is significantly simplified to a few add and shift operations. We refer to these chosen basic bit sequences as alphabets. Also, an alphabet set is a set of alphabets that spans all the coefficients in vector. In the above example, the alphabet set is. In this example, is computed once and the result is shared to calculate both and, this shows the concept of computation sharing in the vector Scaling operation.

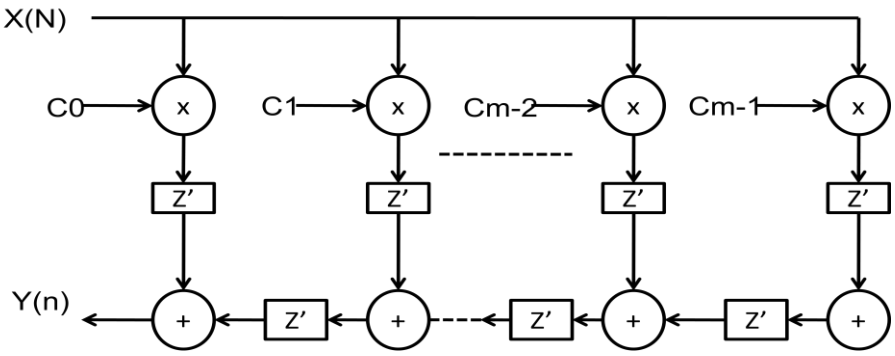

Fig.1.Transposed direct (TDF) FIR filter

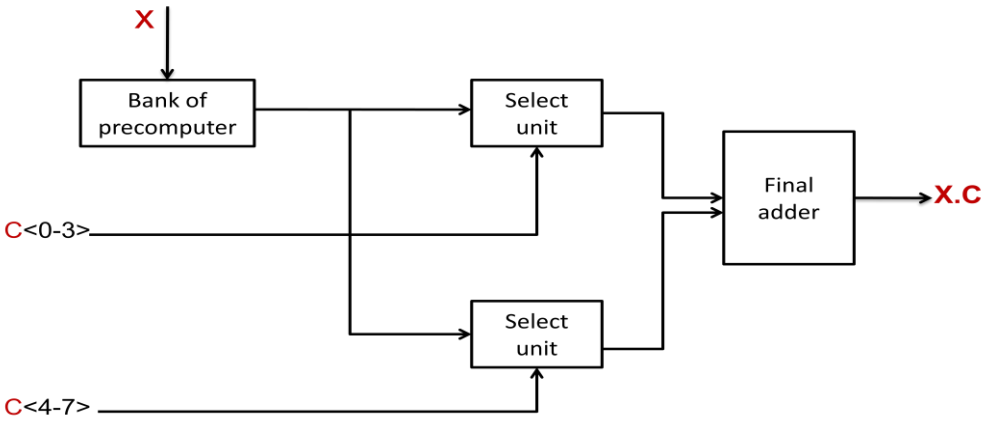

Fig.2. Computation sharing multiplier (CSHM) architecture

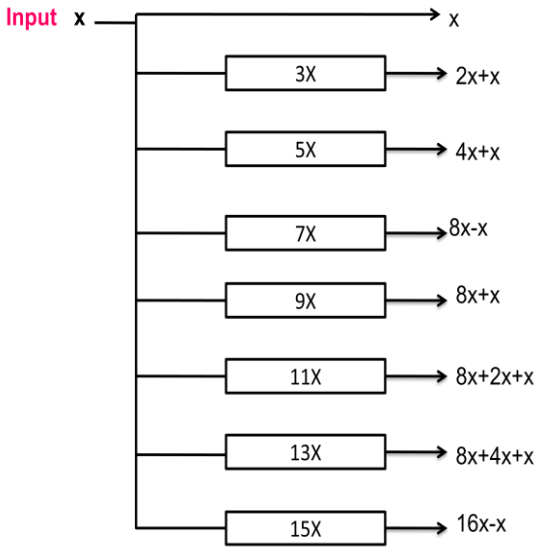

Fig.3.Block diagram of precomputer unit

CSHM architecture is based on the algorithm explained above. Fig.2 shows the CSHM architecture. CSHM is composed of a precomputer, select units and final adders (S&As). Fig.3 shows the precomputer unit.

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. Once the multiplications of alphabets with input are calculated by the precomputer, the outputs are shared by the entire S&As, which is the main advantage of CSHM. In order to cover every possible coefficient and perform general multiplication operation, we used eight alphabets in the precomputer.

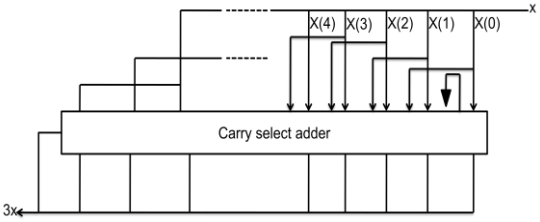

Fig.4.Block diagram of 3x precomputer unit

Precomputer: The multiplications performed by the precomputer are simply implemented using the new carry-Select adder, which is proposed. Fig.3 shows the basic structures of and the precomputer structure. The figure 4 and 5 are example 3x and 5x precomputer respectively.

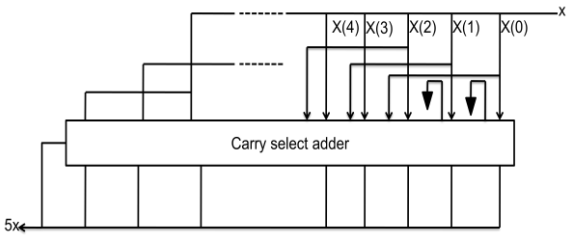

Fig.5.Block diagram of 5x precomputer unit

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. Fig.7 shows the final adder unit. The final adder adds the outputs of the select units to obtain the final multiplication output.

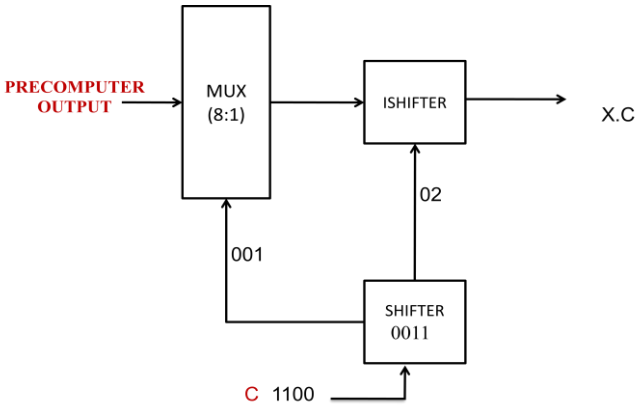

Fig.6. Block diagram of selector unit

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.

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.

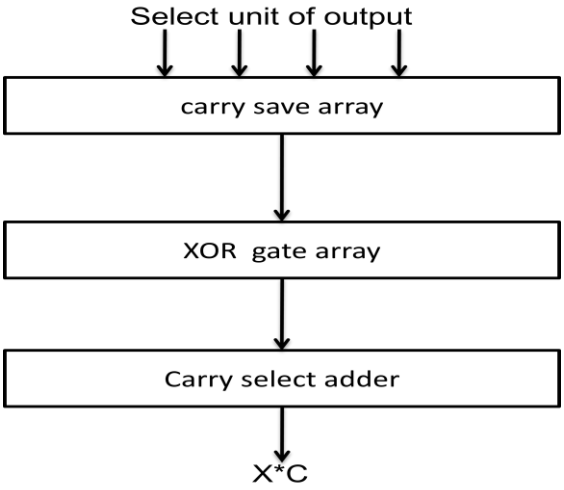

Fig.7.Block diagram of final adder unit

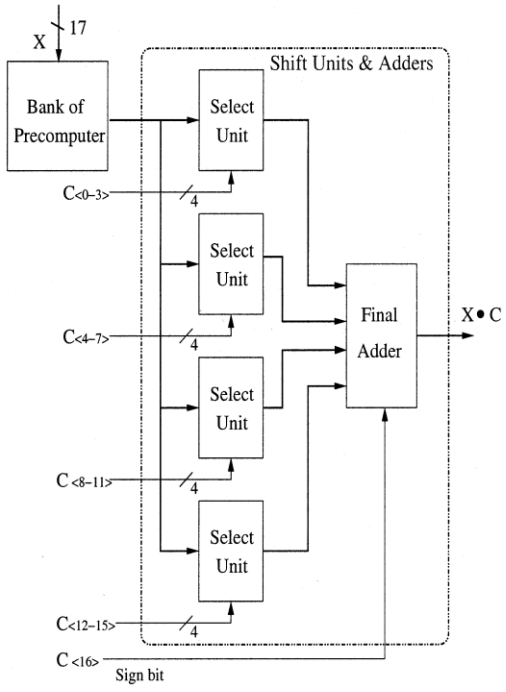

Fig.8. Block diagram of 17x17 CSHM unit

The 17 X 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.

3. FIR Filter Based on CSHM

Using the 17x17 CSHM presented in the previous section, 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.

4. HIGH-PERFORMANCE CARRY-SELECT ADDER USING FAST ALL ONE FINDING LOGIC

4.1 *ADD ONE CIRCUIT*

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}=. If the results for c_{in} = 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 $s_k^0$is the 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.

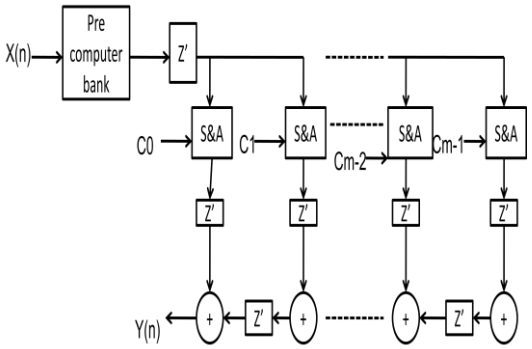

Fig.9. Architecture of FIR filter based on CSHM

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.

4.2 *Examples of first zero detection logic*

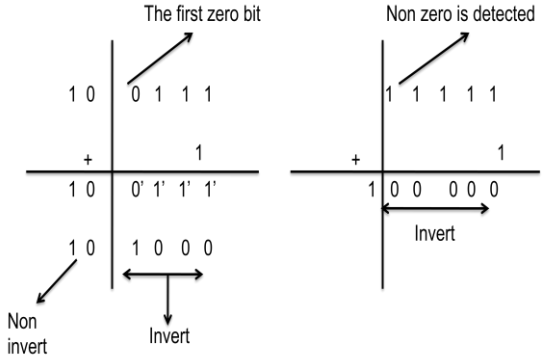

Figure.10:Example of add one logic

4.*3 AN ADD ONE CIRCUIT TO REPLACE ONE RCA*

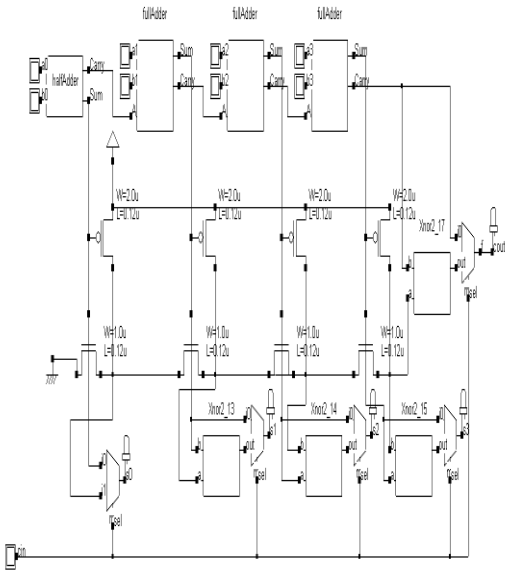

Figure 11.Carry select adder using add one circuit

The 4-bit add-one circuit architecture used by Chang 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.

5. RESULTS AND COMPARISON

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.

Table1.Comparisons of result of CSA

Types Of Adder	Delay (ns)	Number of Transistors	Power (mw)
4-Bit Conventional carry select adder	12.36	294	7.34
4-Bit DTSL carry select adder	15.36	284	1.76
4-Bit Proposed carry select adder	8.25	190	4.46

Table2.Comparisons of FIR filter results

parameter	Filter using Array Multiplier	Filter using CSHM
Area(no.of transistor)	6000	23500
Power(w)	3.732	2.62732
Clock cycle (ns)	9	4.5

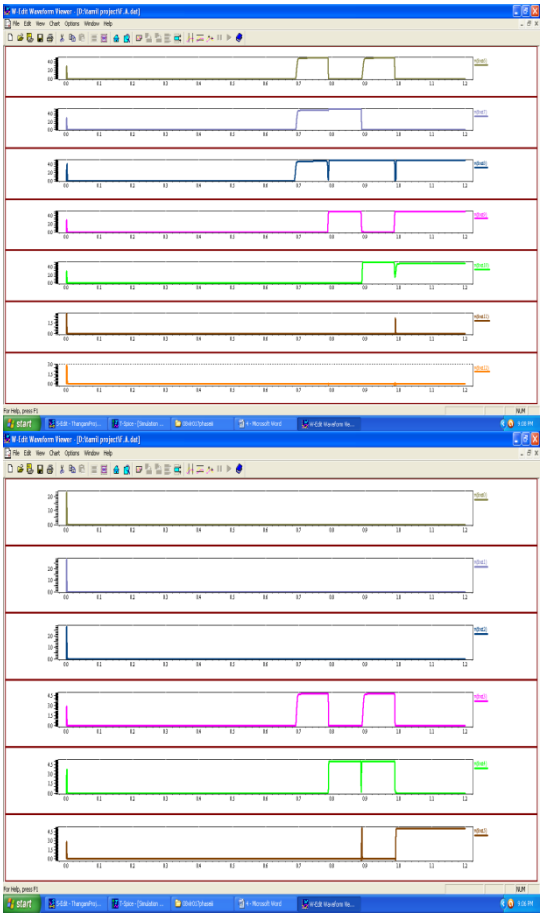

Fig.7. output waveform of 4-tap FIR using proposed CSA

6. 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. 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.

REFERENCES

[1] Jongsun Park, Woopyo Jeong, Hamid Mahmoodi-Meimand, *Student Member, IEEE*, Yongtao Wang, Hunsoo Choo,and Kaushik Roy, *Fellow, IEEE* "Computation Sharing Programmable FIR Filter forLow-Power and High-Performance Applications"

[2] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filter with powers-of-two coefficients," *IEEE Trans. CircuitsSyst.*, vol. 36, pp. 1044–1047, July 1989.

[3] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-31, pp. 583–591, June 1983.

[4] R. I. Hartley, "Subexpression sharing in filters using canonic signed digitmultipliers," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 677–688, Oct.1996

[5] M. Potkonjak, M. Srivastava, and A. P. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 151–165, Feb. 1996.

[6] N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for lowpowerhigh speed FIR filter realization using differential coefficients," *IEEETrans. Circuits Syst. II*, vol. 44, pp. 488–497, June 1997

[7] K. Muhammad and K. Roy, "A graph theoretic approach for synthesizing very low-complexity high-speed digital filters," *IEEE Trans. Computer-Aided Design*, vol. 21, pp. 204–216, Feb. 2002.

[8] W. Jeong, K. Roy, and C. Koh, "High-performance low-power carry-select adder using dual transition skewed logic," in *Proc. ESSCIRC*, 2001,pp. 172–175.

[9] Yan Sun., xin zhang. And xi jin. 'High performance carry select adder using all one finding logic', Second Asia International Conference on Modelling and Simulation. June 2008**.**

[10] J. Woopyo, and R. Kaushik. "RobustHigh-Performance Low-Power Carry Select Adder",*Design Automation Conference*, pp. 503 - 506, Jan.2003

[11] K. Rawat, T. Darwish, and M. Bayoumi, "A low power and reduced area carry select adder", The 45th Midwest Symposium on *Circuits and Systems*., vol.1, pp. 467-470, 2002.

[12] N. Weste and K. Eshragian, "Principle of CMOS VLSI designs: a system perspective", *2nd ed., Addison-Wesley*, 1993.

[13] T. Y. Chang and M. J. Hsiao, "Carry-select adder using ripple-carry adder", *Electron. Lett*. vol. 34, pp. 2101-2103, 1998.

[14] B. S. Kong *et al.*, "Conditional capture flip-flop for statistical power reduction," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2000, pp. 290–291.

[15] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996