Journal of Engineering and Applied Sciences 14 (4): 1070-1077, 2019
ISSN: 1816-949X
© Medwell Journals, 2019

VLSI Implementation of Folded FIR Filter Structures Using High Speed Multipliers

[1]S. Karunakaran and [2]Naveen Kishore Gattim
[1]Department of ECE, Vardhaman College of Engineering, 501218 Shamshabad, Hyderabad, Telangana, India
[2]Department of ECE, KL University, Vadeswaram, Guntur, Andhra Pradesh, India

Abstract: Modern DSP systems are often well suited to VLSI implementation. Indeed, they are often technically feasible or economically viable only if implemented using VLSI technologies. The purpose of designing special purpose DSP systems is an interesting research topic but more important, it has significant industrial and commercial relevance. Many DSP systems are produced in very large numbers and require high performance circuits with respect to throughput and power consumption. A digital Finite Impulse Response (FIR) filter performs the frequency shaping or the linear prediction on a discrete-time input sequence $\{x_0, x_1, -x_2, ...\}$. The FIR filter is commonly used in many applications such as communication or multimedia signal processing. The study is focused on the design of an efficient VLSI architecture for FIR filters which aims at reducing the power consumption and also to reduce the hardware complexity. In the existing method, design of folded FIR filter based on carry-save multiplier is presented. It does not allow the internal pipelining delays to be exploited. It leads to significant increase in hardware as well as considerable increase in power consumption. It leads to less throughput and increases hardware complexity. In the proposed method, design of folded Finite Impulse Response (FIR) filters based on pipelined multiplier arrays is presented. The design is considered at the bit-level and the internal delays of the pipelined multiplier array are fully exploited in order to reduce hardware complexity. Partially folded architectures are also proposed which are implemented by cascading a number of folded FIR filters. The proposed schemes are compared as to the aspect of hardware complexity with a straightforward implementation of a folded FIR filter based on the pipelined carry-save multiplier. The comparison reveals that the proposed schemes may require 20-30% less hardware. Due to the lesser carry propagation, the proposed method can achieve low power consumption and higher computational speedstrategy.

Key words: Finite Impulse Response (FIR), VLSI, design, complexity, linear prediction, bit-level

INTRODUCTION

Now a days, Digital Signal Processing (DSP) is used in a wide variety of real-time applications and is playing an important role in the digital revolution. Finite-Impulse Response (FIR) digital filters are the most fundamental DSP components. FIR filters have the advantage of stability and easy implementation but the large number of filter taps leads to excessive hardware complexity. Power consumption is another important factor in DSP circuits, especially when used in mobile communication systems.

Folding techniques have been proposed as a means of reducing hardware when the processing throughput required by the application is less than the throughput at which the circuit can operate. FIR filters are ideal candidates for folding, since, they are essentially a repetition of multiplications. A reconfigurable folded transposed FIR filter architecture has been presented in but not detailed implementation is given, although, the bit-level design can lead to hardware efficient circuits. An important issue in synchronous designs is the avoidance of problems related to clock-skew. A number of asynchronous architectures have been presented in the bibliography as a solution to this problem at the expense of extra hardware. A significant advantage of the folded FIR architectures is that they lead to reduced hardware in comparison with the corresponding unfolded schemes and the clock-skew problem does not exist. Thus, there is no need to resort to hardware expensive asynchronous solutions.

Corresponding Author: S. Karunakaran, Department of ECE, Vardhaman College of Engineering, 501218 Shamshabad, Hyderabad, Telangana, India

Due to the lesser carry propagation, the proposed method can achieve low power consumption and higher computational speed. The circuit is simulated in T-Spice in 180 nm technology. Power estimation can be obtained through transistor level design simulation.

MATERIALS AND METHODS

Overview of existing method multipliers: Multiplication is one of the most frequently encountered arithmetic operations in floating point processors and digital signal processing applications such as digital filtering, DCT, FFT, CDMA and many others. To increase the speed of computation, digital multipliers are usually implemented in hardware that generates all bit products and sums in parallel using an array or tree of adding elements. These parallel multipliers are fast but need a large number of transistors, resulting in significant power consumption.

Conventional multiplier array: In parallel multiplier, the carry-save array has the highest switching activity, largest capacitance and so, dissipates most of the power (Bougas *et al*., 2015; Mudassir and Abid, 2005; Baugh and Wooley, 1973; Brent, 1982). A major goal of the partial product bypassing techniques is to avoid redundant switching transitions in the array. The main idea is based on observation that conventional array multipliers generate a large number of signal transitions while adding zero products. To eliminate redundant signal transitions, proposed Row-bypassing technique that uses multiplexers in order to bypass those rows which correspond to zero partial products. For example, if the 3rd bit of B is zero, the 3rd row of adders does not need to be activated, since, the corresponding product is 0. Triangles in the figure denote the 3-state buffers. Contrary to row bypassing, proposed to bypass columns when the corresponding bit of the multiplicand (Y) is 0. As one can see, these techniques require an extra circuitry overhead for each carry-save adder. Also they increase the delay of multiplier because an extra logic is inserted into its critical path.

Carry-save multiplier: In carry-ripple multipliers (Parhi *et al*., 1992; Wang *et al*., 2001; Yu *et al*., 2002) carries are propagated during each additions. This carry-propagation limits the speed of multiplication. In a case involving several additions such as partial product accumulation in multiplication. It is not strictly necessary to propagate these carries during each cycle. Instead, the carries generated during the addition of pair of operands can be saved and added with proper alignment to the next operands. This leads to a concept of carry-save addition. In the carry-save array multiplier, the carry outputs are saved and used in the adder in the next row. In this case, the partial product is replaced by a partial sum and partial carry which are saved and passed onto the next row. The advantage is that the addition at different bit positions in the same row are now independent of each other and can be carried out in parallel.

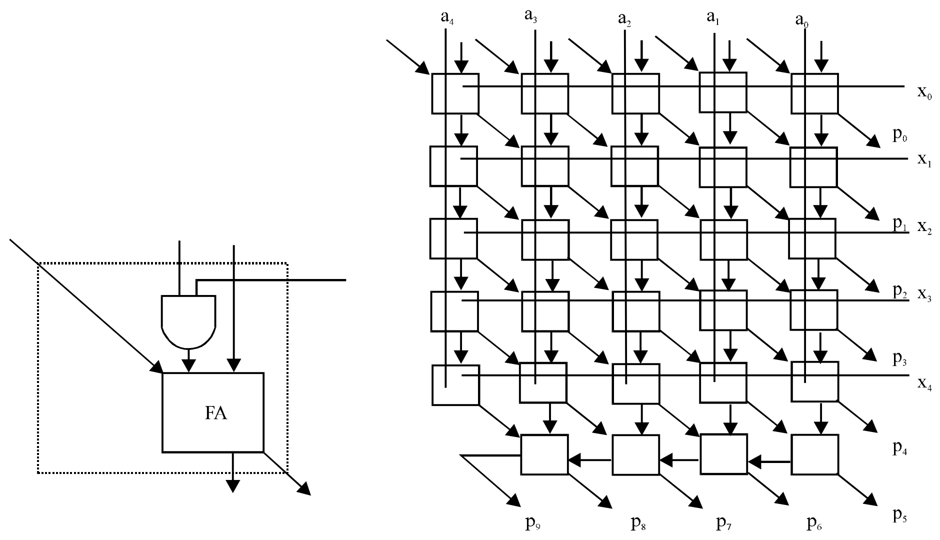
This carry-save addition can be applied to all but the last step where there is no more multiplicand-multiple to be added but the partial sum and partial carry. The addition of partial sum and partial carry is performed by a Vector Merging Adder (VMA). The VMA can be implemented either as a carry save array which contains only half adders and a more regular and easier to pipelined structure. When bot level pipelining is not required, the carry-ripple and carry-save principles can also be combined into a single structure where the carry output can ripple through at most 1 bit in the same row and the resulting carry from each ripple portion is passed on to the next row like the carry-save architecture.

Problem statement: The elaborate design of folded FIR filter is considered. In the existing method, it does not allow the internal pipelining delays to be exploited. It leads to significant increase in hardware as well as considerable increase in power consumption. It leads to less throughput and increases hardware complexity.

Proposed system: In Fig. 1-3, a folded implementation of this filter is given. It consists of an M-A unit that performs one multiplication and addition at each clock cycle. Such operations are required to produce one result at the output. The filter coefficients are stored in a cyclic shift register Coefficient Register (C-R) in descending order (at the right end of the register). The samples are stored in the cyclic shift.

Architecture of carry-save multiplier

Architecture of folded FIR filter structure: Figure 3 folded architecture of FIR filter in direct form register Data Register (D-R) in ascending order (the oldest at the left end of the register). The timing diagram in Fig. 3 clarifies the operation of the circuit. The computation cycle lasts clock cycles. During those cycles the most recent samples as well as the coefficients are cyclically shifted in order to compute the convolution. At the last clock of the operation cycle when the control signal SYN is high, a



FA
a_4 a_3 a_2 a_1 a_0
x_0 x_1 x_2 x_3 x_4
p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9

Fig. 1: Conventional multiplier array

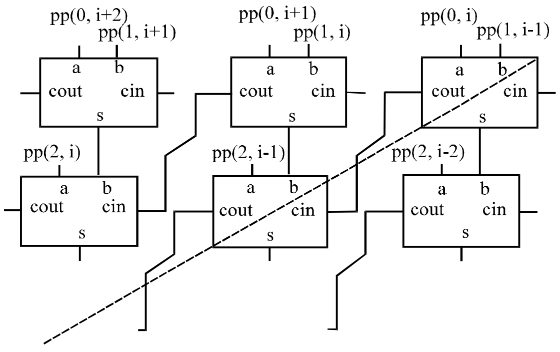


Fig. 2: Architecture of carry-save multiplier

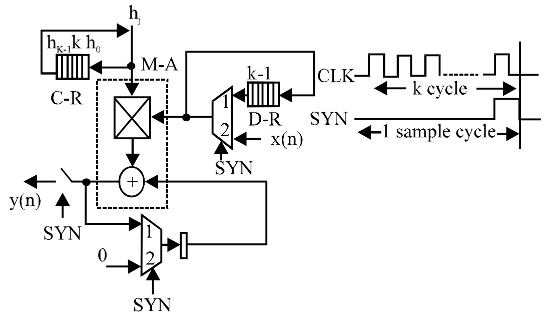


Fig. 3: Folder architecture of FIR filter in direct form

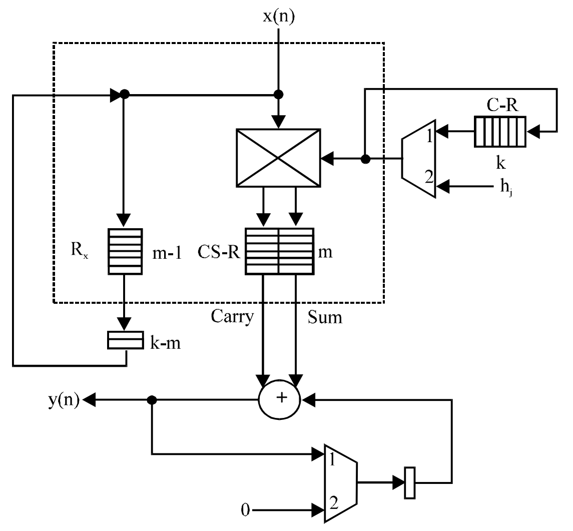


Fig. 4: Folder FIR scheme based on both multiplier

new input sample enters the circuit and replaces the oldest stored input sample in D-R. At the same clock, the delay element of the accumulator is cleared while the switch at the output of the circuit closes and the final result is produced. Thus, the circuit is of immediate response. The circuit operates at clock frequency. An input sample is processed every clock cycles and therefore, the filter operation frequency is f_s. T_e frequency of the control signal SYN is. The D-R at the data input corresponds to the delay elements in the data line. The C-R corresponds to the data latches where the filter coefficients are stored when the filter is programmed.

Folded FIR filter scheme based on booth multiplier: The circuit, drawn in detail in Fig. 4, implements the above filter scheme for filter taps. The i-order bit of the input sample x(n), the coefficient h_j and the output y(n) is represented the dashed lines represent the accumulation feedback loop. The feedback lines of x(n) and y(n) are not shown in order the complexity of the schematic to be kept low. In this example the length of Rx is equal to k-2, so, one external delay for is required. The length of Rh is b = 4 (<k), so, 5 additional delays in each bit line of have been added on the right side of the multiplier. The external delays are represented with empty boxes and the internal delays with filled boxes. The diagonal row of cells at the right edge of the array is used for the accumulation of the least significant part of the result. For the accumulation of the most significant part an additional row of full-adders has been added at the bottom of the array. This row is extended leftwards by full-adders to avoid accumulation overflows. The structure of the M-A unit using a booth multiplier is shown. The booth multiplier has internal delays, denoted within the figure, only in the bit-lines. These delays can be used for the implementation of D-R. The register for the implementation of C-R must be added externally as shown where the implementation of a folded FIR filter in direct form is given based on the M-A unit of Fig. 4-13.

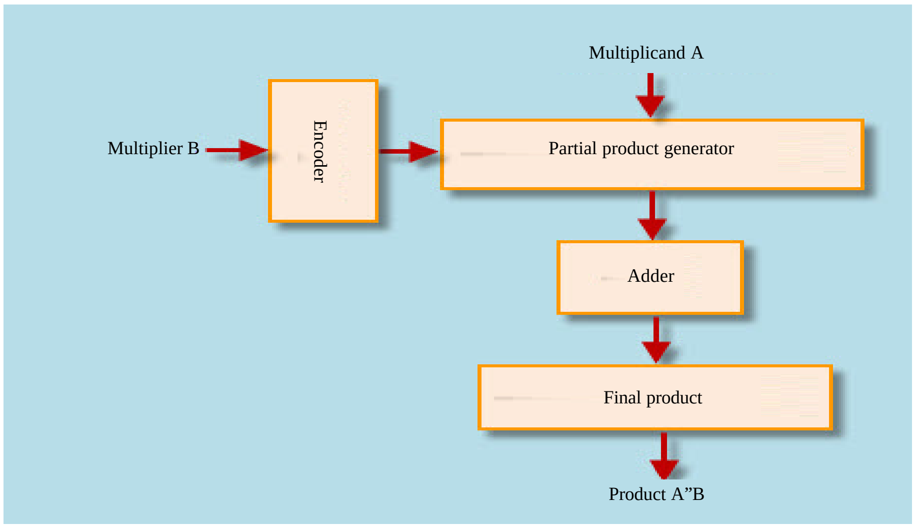


Fig. 5: Architecture of booth multiplier

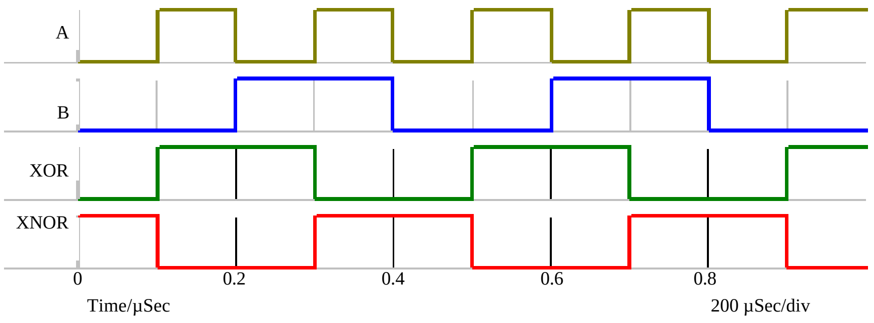


Fig. 6: Simulation result of XOR gate in T-SPICE

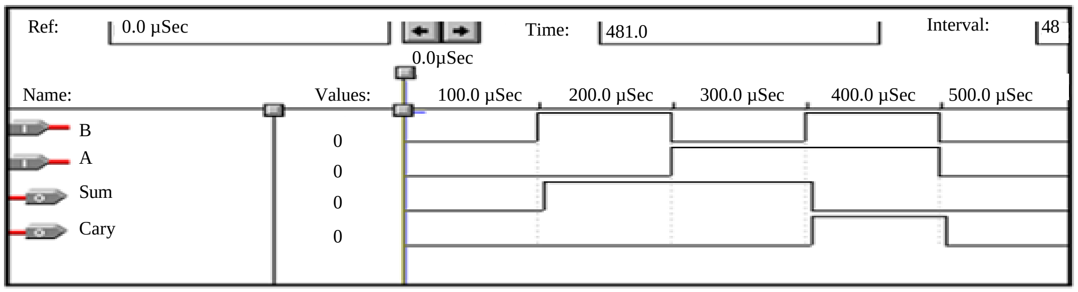


Fig. 7: Simulation result of half Adder in T-SPICE

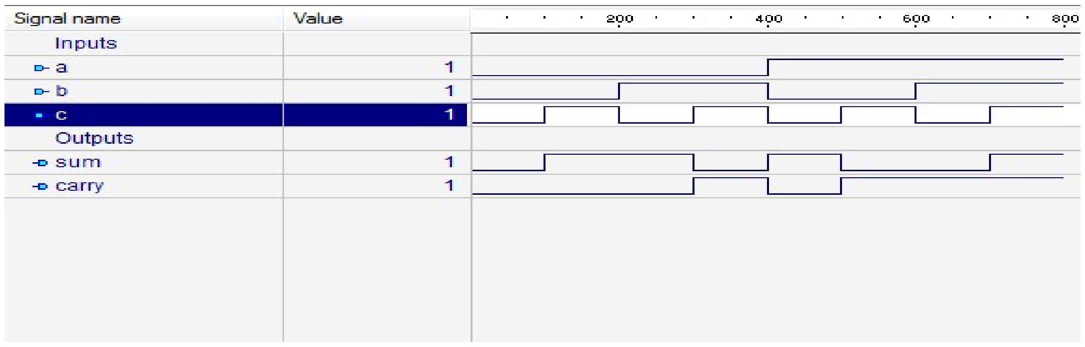


Fig. 8: Simulation result of full adder in T-Spice

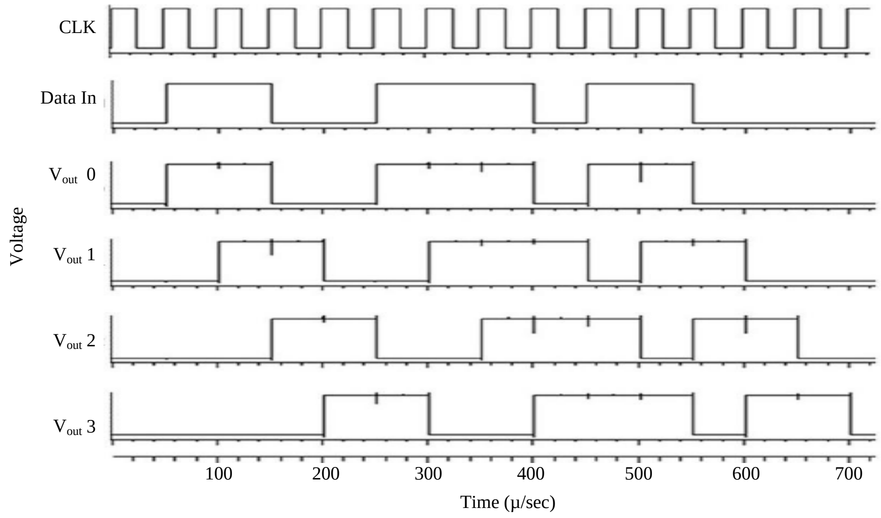


Fig. 9: Simulation result of shift register (C-R and D-R) in T-Spice

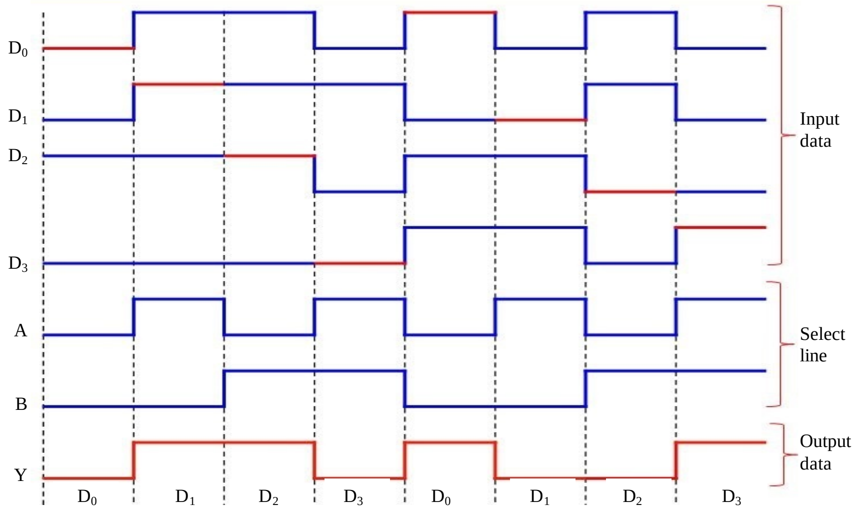


Fig. 10: Simulation result of multiplier in T-Spice

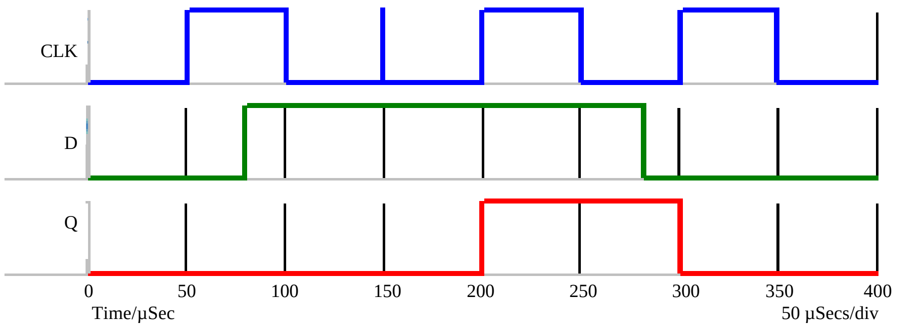


Fig. 11: Simulation result of D-flipflop in T-Spice

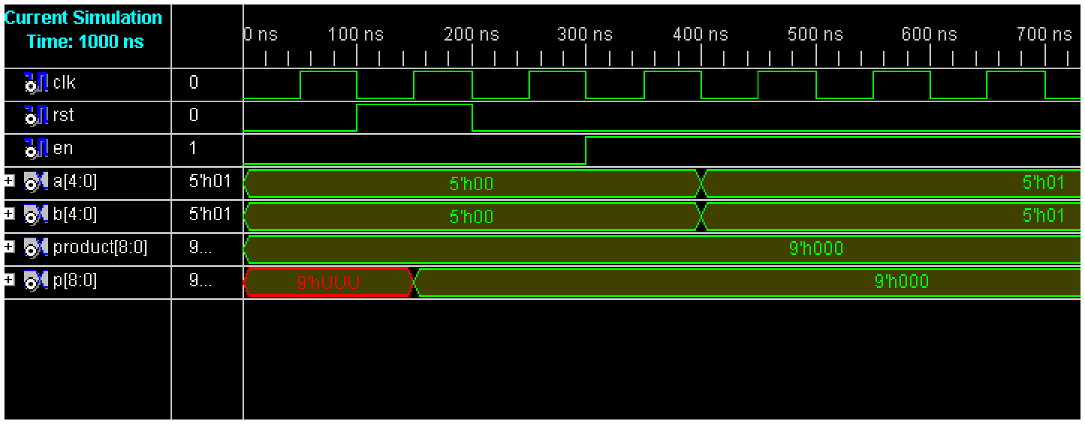


Fig. 12: Simulation result of carry-save multiplier in T-Spice

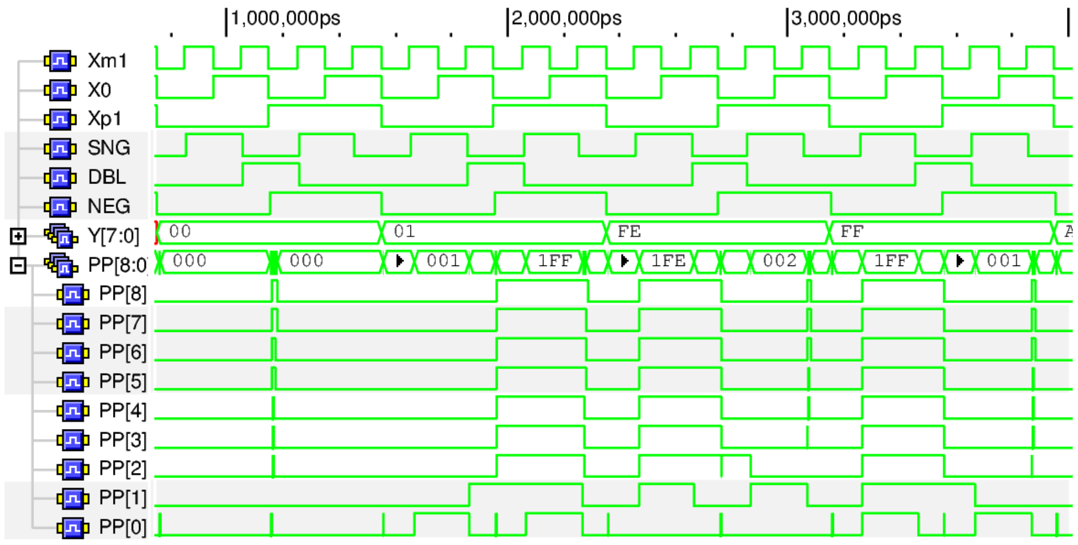


Fig. 13: Simulation result of both multiplier in T-Spice

External delays must also be added to in order to become equal to the required length for the D-R. The circuit implements the above filter scheme. The accumulation is performed by a bit-skew adder which is implemented by the diagonal line of full-adders at the right edge of the array. It is extended leftwards by full-adders, in order to avoid the accumulation overflow. The dashed lines represent the accumulation feedback loop. The external delays are represented with empty boxes. The feedback lines of and have been omitted in order to keep the complexity of the schematic low. The most significant part of the result is in carry-save form. Therefore, a triangle of delays and a row of full-adders is attached to the bottom of the multiplier in order to convert the result in bit-skew binary form.

Booth multiplier: A low-complexity and high-speed transposed direct-form Finite-Impulse-Response (FIR) architecture is developed based on the radix-4 Booth algorithm. It includes a pre-processing unit. Input sub-data latches, a control unit, booth decoders. filter coefficient registers an accumulation path and a post-processing unit. To decrease the hardware complexity the pre-processing unit, input sub-data latches and booth decoders are explored by using the 2 bit word length of sub-data latches instead of the conventional 3 bit one. In addition, the accumulation path using the carry save adders is designed by using the addition delay scheme to minimize the numbers of half adders and latches. As compared to the pipelined carry-save multiplier architecture, the proposed FIR architecture can reduce more hardware complexities as the word length of input data and the tap number of FIR increase. For example, when the % tap FIR with % bit input data and the 256-tap FIR with 16 bit input data are designed. the proposed FIR architecture would save about 11 and 25% of hardware complexity, respectively.

To effectively reduce computational time and hardware complexities researchers have developed high-speed FIR tilters using the Booth algorithm for multiplication. In the Booth-algorithm FIR design, it decomposes a multiplication into addition subtraction and shifting operations that are determined by the encoded input data and filter coefficients generally. The complexity of a multiplication using the Radix-2 booth algorithm is smaller than that of a straightforward multiplication algorithm. In order to improve computational efficiency.

Architecture of booth multiplier: Figure 5 shows the architecture of booth multiplier.

RESULTS AND DISCUSSION

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 commercially successful electronic products to market ahead of the competition. Tanner's fully-integrated solutions consists of tools for schematic entry, circuit simulation, waveform probing, full custom layout editing, placement and routing, net list extraction, LVS and DRC verification. Tanner EDA's innovative solutions are used in a range of application in next-generation wireless, consumer electronics; imaging, power management, biomedical, automotive and RF market segments.

Table 1: Result comparison

Variables	Carry-save multiplier	Booth multiplier
Number of MOSFETs	544	425
Power consumed at 10 MHz frequency	8.974168e-004 (W)	5.9017e-004 (W)

Comparison of results: From the simulation result presented in Table 1, it is found that the power consumption for carry-save multiplier and booth multiplier are found to be 8.974168e-004 W and 5.9017e-004 W. From the results, it is inferred that the booth multiplier has reduced hardware complexity and lesser power consumption when compared with carry-save multiplier.

CONCLUSION

The possibility of incorporating a whole signal processing system into a chip has a multitude of effects. It will dramatically increase the processing capacity and simultaneously reduce the size of the system.

The study is focused on the design of an efficient VLSI architecture for folded FIR filter which aims at reducing the hardware complexity and also to reduce the power consumption. In the proposed method, booth multiplier is employed instead of carry-save multiplier to get the better efficiency.

From the simulation, it is found that the number of MOSFET's for carry-save multiplier and booth multiplier are 544 and 425, respectively. The power consumption for carry- save multiplier and booth multiplier are found to be 8.974168e-004 watts and 5.9017e-004 watts From the results, it is inferred that the booth multiplier has reduced hardware complexity and lesser power consumption when compared with carry-save multiplier.

Due to the lesser carry propagation, the proposed method can achieve low power consumption and higher less hardware complexity. The circuit is simulated in T-Spice in 180 nm technology. Power estimation can be obtained through transistor level design simulation.

RECOMMENDATION

The future research is to incorporate the modules to form a FIR structure and to find out the results.

REFERENCES

Baugh, C.R. and B.A. Wooley, 1973. A two's complement parallel array multiplication algorithm. IEEE. Trans. Comput., 100: 1045-1047.

Bougas, P., P. Kalivas, A. Tsirikos and K.Z. Pekmestzi, 2015. Pipelined array-based FIR filter folding. IEEE. Trans. Circuits Syst. I Regul. Pap., 52: 108-118.

Brent, R.P., 1982. A regular layout for parallel adders. IEEE Trans. Comput., c-31: 260-264.

Mudassir, R. and Z. Abid, 2005. New parallel multipliers based on low power adders. Proceedings of the Canadian Conference on Electrical and Computer Engineering, May 1-4, 2005, Canada, pp: 694-697.

Parhi, K.K., C.Y. Wang and A.P. Brown, 1992. Synthesis of control circuits in folded pipelined DSP architectures. IEEE. J. Solid State Circuits, 27: 29-343.

Wang, Y., Y. Jiang and E. Sha, 2001. On area-efficient low power array multipliers. Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS'01) Vol. 3, September 2-5, 2001, IEEE, Malta, Europe, pp: 1429-1432.

Yu, Z., M.L. Yu, K. Azadet and A.N. Willson, 2002. A low power adaptive filter using dynamic reduced 2's-complement representation. Proceedings of the 2002 IEEE Conference on Custom Integrated Circuits, May 15, 2002, IEEE, Orlando, Florida, pp: 141-144.