ISSN:2348-2079

Volume-6 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

Design of high speed constant multiplier based on VHBCSE algorithm with BRENT kung and ling adders

Mahalakshmi Marimuthu[1], Dr.Brindha Palanisamy[2], Parimaladevi Muthusamy[3]

[1]II Year M.E. VLSI Design, [2,3]Assistant Professor, Velalar College of Engineering and Technology, Erode-638 012, Tamilnadu, India.

***Abstract*—Inmany DSP applications, the design of Multiply and accumulate (MAC) unit defines the efficiency of the overall system. Similarly Multiple Constant Multiplication (MCM) is the multiplication operation between the particular input variable and the variable coefficients constants, which is used in FIR filters. This MCM is performed by using constant multiplier, which is implemented using some prior algorithms. Variable-bit BCSE algorithms is the one which belongs to BCSE algorithm. In this methodparallel prefix computation logic is used for the addition process in Vertical-Horizontal Binary Common Sub-expression Elimination (VHBCSE) based constant multiplier design. In this paper ripple carry adder (RCA) is replaced by the parallel prefix adders (PPA) like a Brent Kung adder (BKA) and Ling adder in the PPG and control addition layer. The VHDL language is used to design the constant multiplier, using ripple carry adder, Brent Kung adder and Ling adder. Thus the comparison between this different design in terms of performance parameters like power, delay and area is achieved by the FPGA implementation using Xilinx ISE 9.2i synthesis tool.**

***Index Terms*— VHBCSE algorithm, BCSE technique, Brent Kung Adder (BKA), Ling Adder.**

I. INTRODUCTION

The complexity of the multiplication operation in the DSP applications, mainly depends on the number of adders, delay due to adders and computation cost. Thus the repetitive application of this multiplication and addition operation accomplish the use of nonlinear function in DSP processing systems. FIR filter design is implemented using an efficient algorithm which can be categorized into two groups based on this MCM: 1) Graph based algorithms and 2) Common sub-expression elimination (CSE) algorithms [3], [7] and [16]. Graph based algorithms didn't consider about the delay which is the important parameter in high performance systems. BCSE algorithm mainly focused on the delay reduction process by reducing the operand length using elimination technique.

The performance of any processor mainly depends on the speed and power consumption of adders and multipliers. Multiplication is nothing but the repeated addition process. So it is important to design the adders and multipliers with low power, high speed and minimized area. According to the concept of the multiplier, constant multiplier plays an important role in the design the polynomial basis (PB) finite field multiplier. In this polynomial basis finite field multiplier design constant multiplier module realizes multiplication between a field element and the constant x^k [13]. One of the technique for the low power, constant multiplier design is binary common sub-expression elimination (BCSE) algorithm which eliminating the common sub-expression in binary form. The CSE algorithm uses both canonical signed digits (CSD) as well as binary representation. In CSD based CSE algorithm requires more memory to store the signed bits.

In BCSE algorithm, $2^n - (n+1)$ number of BCSs can be used for n-bit binary number and to generate the partial products for n-bit BCS requires a total of $2^{n-1} - 1$ number of adders [2]. Normally BCSE technique is applied across the adjacent coefficients and also within the coefficient. The choice of BCS defines the type of BCSE algorithm as a fixed bit BCSE (FBCSE) algorithm [12] and [6] or vertical-horizontal BCSE (VHBCSE) algorithm. According to FBCSE algorithm the choice of the BCS is fixed in length. In terms of speed and area 2-bit BCSE algorithm [6] is more efficient than 3-bit BCSE algorithm [12]. But both the two methods having some of the problems, like only it supports signed magnitude number format, BCSE algorithm only used in the first layer, etc.

The VHBCSE algorithm which can solve the problems in the FBCSE algorithm and also can work for

signed decimal number of both the input and the coefficients. VHBCSE algorithm is the combination of vertical-BCSE and horizontal-BCSE technique which is used for designing an efficient reconfigurable FIR filter. Horizontal BCSE algorithm utilizes CSs occurring within each coefficient to get rid of redundant computations, while vertical BCSE uses CSs found across adjacent coefficients to eliminate redundant computations. In VHBCSE algorithm, a 2-bit vertical BCSE has been applied to the adjacent coefficients, followed by 4-bit and 8-bit horizontal BCSEs. Vertical BCSE produces more effective BCS elimination than the horizontal BCSE [15].

In constant multiplier design with VHBCSE algorithm achieves the reduced probability of the use of the adders to sum up the partial product generator (PPG) by extending the BCSE at the lower level. In this constant multiplier, ripple carry adder (RCA) is used for the addition process in the partial product generation unit and control addition layer. In our proposed technique the VHBCSE based constant multiplier design is achieved by the parallel prefix addition (PPA) technique. To speed up the addition, carry look ahead adder is introduced which is the basis for parallel prefix adder design to achieve the efficient speed rather than the simple parallel adders like carry skip adder, carry select adder and ripple carry adder, etc. Carry Look-Ahead adders terminology is equivalent to Parallel-prefix adders [4]. The parallel prefix adder logic is mainly used in microprocessors, DSPs, mobile devices and other high speed applications like VLSI chip implementation. Thus the VLSI chips rely heavily on fast and reliable arithmetic computation. These contributions can be provided by PPA [8].

The above parallel prefix adder technology reduces logic complexity and delay thereby enhancing the overall performance of the design of constant multiplier. The rest of this paper is organized as follows. In section II, the basic operation of the parallel prefix adder (PPA) is explained. In section III, detailed analysis of VHBCSE based constant multiplier with parallel prefix adders are presented. In section IV, result and discussion about the adder replacement in the design is explained. Concluding remarks are given in section V.

II. PARALLEL PREFIX ADDER (PPA)

The parallel prefix adders (PPA) are high speed adders rather than the parallel adders. PPA technique is a parallel form of obtaining the carry bit that makes it performs addition arithmetic faster. The operation of a prefix computation in a PPA consists of three stages, namely pre-processing, carry computation and post-processing which is shown in fig. 1. On pre-processing stage the propagate (p) and generate (g) bits are generated. Prefix carry tree is constructed for the parallel carry generation in the carry computation stage. The post-processing stage calculates the final sum and carry signals for the given input. For the overall addition process PPA uses the black cell, grey cell and buffer.

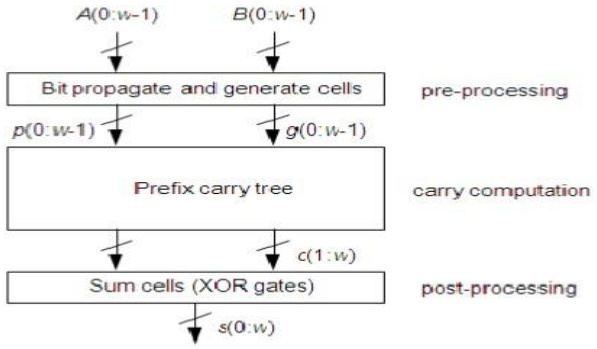

Fig. 1: Structure of Parallel Prefix Adder.

Parallel Prefix Adder (PPA) consist of different types, namely Kogge-Stone Adder (KSA) [10], Brent-Kung Adder (BKA), Han-Carlson Adder (HCA) [14], S. Knowles (SKA), Sklansky Adder, Ladner-Fischer Adder (LFA) [11] whose area, fan-out, wire track and logic levels are explained in [4]. Ling adder is also operated using the parallel prefix technique for carry computation which is used in the proposed multiplier design due to its high speed design [5]. Beaumont adder is also based on the parallel prefix computation [9], [1].

III. VERTICAL HORIZONTAL BCSE (VHBCSE) ALGORITHM BASED CONSTANT MULTIPLIER WITH PARALLEL PREFIX ADDERS

In this method a 2-bit vertical BCSE has been applied first on the adjacent coefficient, followed by 4-bit and 8-bit horizontal BCSEs to detect and eliminate as many BCSs as possible which are present within each of the coefficient. Instead of using ripple carry adders in the partial product generator and also the control addition layers, parallel prefix adders are used.

The data flow diagram of the proposed vertical-horizontal BCSE algorithm based constant multiplier design is shown in fig. 2.

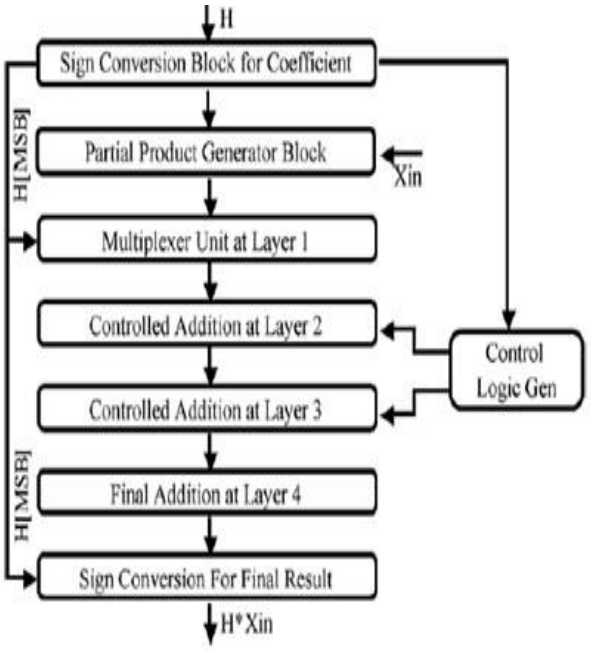

Fig. 2: Data flow diagram of VHBCSE based constant multiplier.

The designed multiplier considers the length of the input (Xin) and coefficient (H) as the 16 - bit and 17-bit respectively, while the output is assumed to be 16-bit length. Functionality of different blocks of the VHBCSE based constant multiplier:

*a) Sign Conversion Block***:** This block is needed to support the signed decimal format data representation for both the input and the coefficient. The architecture of the sign conversion block consists of one 1's complement circuit which generates the inverted version of the 16-bit coefficient (excluding MSB) and one 16-bit 2:1 multiplexer which produces the multiplexed coefficients depending on the value of the most significant bit (MSB) of the coefficient. The architecture is shown in fig. 3. If the MSB is 1 (for negative coefficient), the multiplexed coefficient will be inverted; otherwise it will remain the same.

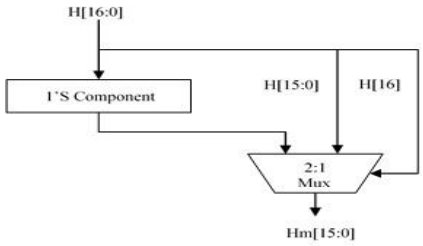

Fig. 3: Sign conversion block.

b) Partial Product Generator: Partial product has been generated by shift and add based technique in BCSE method which will be summed by the following addition layer. For layer-1, 2-bit BCSs are considered where extraadder is required for the pattern '11' whereas the rest are generated by hardwired shifting. For 16-bit coefficient, eight (P8-P1) partial products is obtained by right shifting the first partial products P8. This reduces the size of the multiplexer. The architecture of partial product generator is shown in fig. 4.

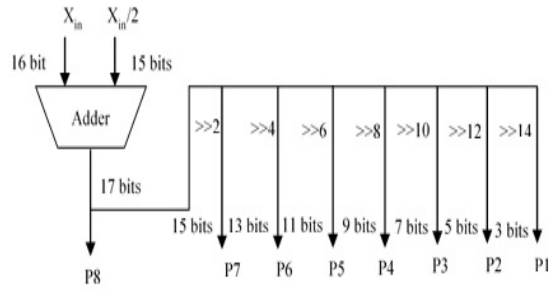

Fig. 4: Block diagram of Partial Product Generation Unit.

c) Control Logic (CL) Generator: The multiplexed coefficient (Hm[15:0]) is grouped into four bit each as (Hm[15:12], Hm[11:8], Hm[7:4] and Hm[3:0]) which is given as the input of the control logic generator. Another grouping produce 8-bit of each (Hm [15:8] and Hm[7:0]). This control logic generator produces seven control signals from seven equality checks for 7 different cases. The architecture of the block is shown in fig. 4. The control signal for 8-bit equality check (C7) is generated by using the control signal generated from the 4-bit equality check (C2, C5).

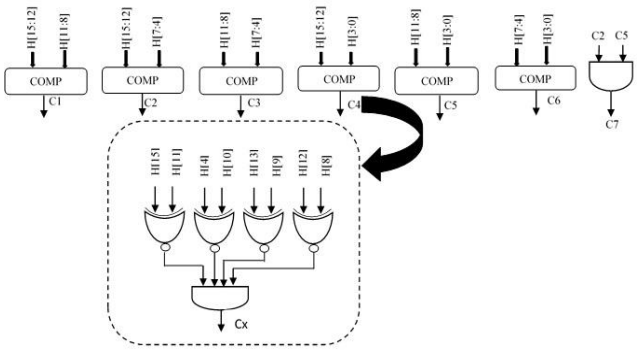

Fig. 5: Block diagram of Control Logic Generator unit.

*d) Multiplexer Unit at layer-1:*The multiplexer unit is used to select the appropriate partial product from the result of the PPG unit by the coefficient's binary value. It consists of eight 4:1 multiplexer with different width depends on the width of partial products. This reduces the hardware complexity and power consumption.

*e) Controlled addition at layer-2:*This layer consists of four adders (A1-A4) for addition operations of the eight PPs under the control signals (C1-C6) generated from the control addition layer. Instead of normal addition of these PPs, the controlled addition is takes place according to the VHBCSE algorithm and generate AS1 (16-bit), AS2 (12-bit), AS3 (8-

bit), AS4 (4-bit). Adder A1 is designed as a parallel prefix adder. The architecture of this block is shown in fig. 6.

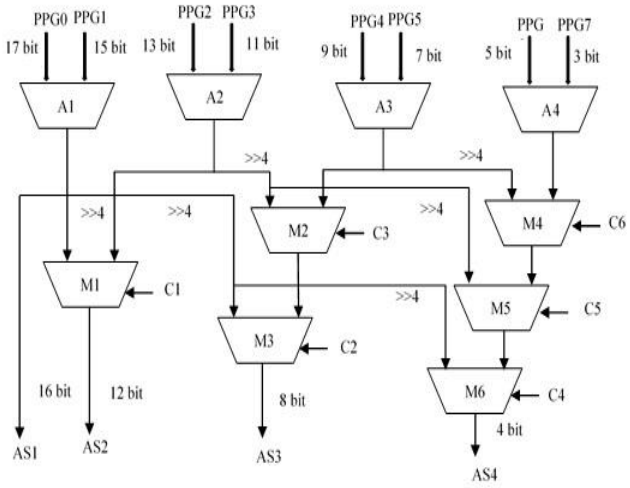

Fig. 6: Architecture of Control Addition Layer-2.

g) Controlled addition at layer-3: Thefour multiplexed sums generated from layer-2 are summed up in layer-3. The controlled addition is performed on the four multiplexed results (AS1, AS2, AS3, AS4) under the control signal C7, which is generated based on 8-bit BCSE from the CL generator unit to produce AS5 (16-bit) and AS6 (8-bit) using parallel prefix adder. The architecture of this block is shown in fig. 7.

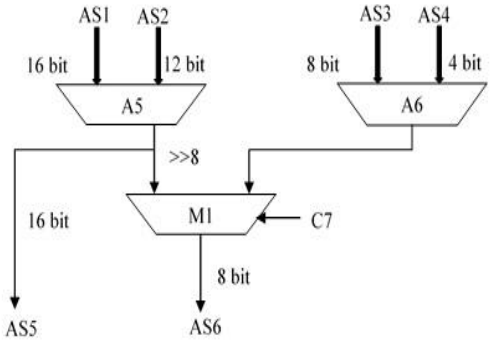

Fig. 7: Architecture of Control Addition Layer-3.

*h) The final addition at layer-4:*In this layer the direct addition is performed instead of control addition by using the parallel prefix adder to sums AS5 (16-bit) and AS6 (8-bit) to produce the final multiplication results between input and coefficient. The architecture of the constant multiplier is shown in fig. 8.

The constant multiplier design based on VHBCSE algorithm has been coded using VHDL hardware description language for simulation. Xilinx ISE 9.2i synthesis tool is used to implement the design in the targeted FPGA device XC3S250E (Spartan 3E Xilinx Kit). In this method adders used in the PPG unit and the control addition layer-3 (17-bit adder) and the final addition have been replaced by the adders with parallel prefix computation logic, such as Brent-Kung adder and Ling adder, which having the high operation speed than the normal parallel adders like carry skip, carry save, carry propagate adder.

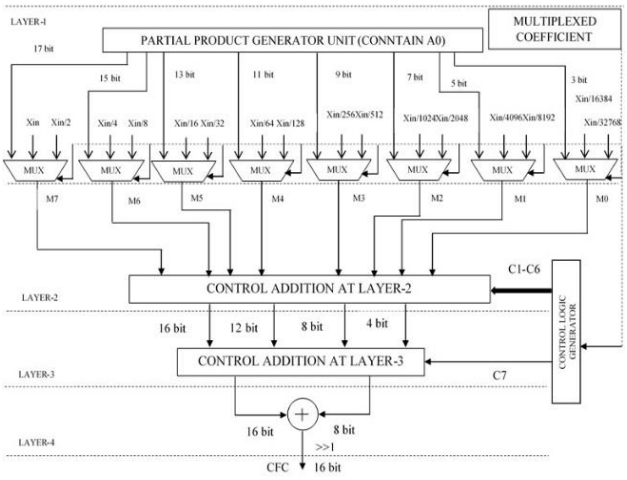

Fig. 8: Architecture of Constant Multiplier based on VHBCSE Algorithm.

IV. RESULT AND DISCUSSION

The partial product generation (PPG) unit and control addition layer in the constant multiplier design using the adders to produce the final result. In proposing method this adders are replaced by the parallel prefix adders (PPA) such as Brent-Kung and Ling Adder. The implementation is done using Xilinx ISE 9.2i synthesis tool. VHDL hardware description language is used to code the design to synthesize the design in the targeted FPGA device. The frequency of 100MHz is maintained at the time of analysis of the VLSI parameters such as area, delay and power. Parameter comparison is shown in TABLE I.

TABLE I

COMPARISON OF THE VLSI DESIGN PARAMETERS OF PPG LAYER WITH DIFFERENT ADDERS

Type of adder used for PPG layer	Ripple Carry Adder	Brent-Kung Adder	Ling Adder
Total Power(mw)	57.40	57.40	57.40
No. of slices	11	12	15
No. of 4 input LUTs	20	22	28
Delay (ns)	11.577	11.635	11.316
Logic Delay (ns)	8.010	8.045	8.010
Route Delay (ns)	3.567	3.590	3.30

Total Memory Usage (kilo bytes)	144744	144744	145768

Similarly, parameter comparison of the control addition layer-3 and the final addition layer with Ripple Carry Adder (RCA) and Ling Adder is shown in TABLE II. From the tabulation total power, number of slices used and the delay exposes the analysis report, which results that the Ling adder consumes less time rather than the Ripple Carry Adder (RCA) and the Brent-Kung Adder.

TABLE II

COMPARISON OF THE VLSI DESIGN PARAMETERS OF CONTROL ADDITION LAYER - 3 AND FINAL ADDITION LAYER WITH DFFERENT ADDERS

Type of adder used for Control Addition Layer	Ripple Carry Adder	Ling Adder
Total Power (mw)	53.36	53.41
No. of slices	36	64
No. of 4 input LUTs	63	114
Delay (ns)	29.222	27.997
Logic Delay (ns)	17.866	16.493
Route Delay (ns)	11.356	11.504
Total Memory Usage (Kilobytes)	147816	149864

V. CONCLUSION

Thus the result shows that memory usage for the PPG with RCA and BK adder consumes same size, but the PPG with Ling adder consumes 0.707% of the more memory size than the other two designs. Power utilization is equal for all the three designs. Compared to PPG with RCA, PPG unit with BK adder delay is 0.5% higher, and the PPG unit with Ling adder delay is 2.254% less. Compared to the PPG unit with RCA, PPG unit with BK adder and the Ling adder consume more area of about 9.09% and 36.36% respectively. Similarly the control addition layer with Ling adder consume more memory space of about 1.38% than the control addition layer with RCA. Power consumption of control addition layer with Ling adder is also more of about 0.093% than the control addition layer with RCA. The trade-off is achieved by the high speed. Thus the required time for control addition layer with Ling adder is 4.192% less than the control addition layer with RCA and area utilization is 77.77% high for the design of control addition layer with Ling adder. Thus the overall analysis shows that the Ling adder is more useful for higher speed applications. The complete replacement of Ling adder instead of RCA will results the most efficient design.

REFERENCES

[1] AdilakshmiSiliveru and M. Bharathi, "Design of Ladner-Fischer and Beaumont-Smith Adders Using Degenerate Pass Transistor Logic", *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 1, pp. 177-181, March 2013.

[2] A. P. Vinod and E. M. K. Lai, "An efficient coefficient partitioning algorithm for realizing low-complexity digital filters," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1936–1946, Dec. 2005.

[3] C. H. Vun, A. B. Premkumar, and W. Zhang, "A new RNS based DA approach for inner productcomputation," *IEEE Trans. Circuits SystemsI, Reg. Papers*, vol. 60, no. 8, pp. 2139–2152, Aug. 2013.

[4] Geeta Rani, Sachin Kumar, "Delay analysis of Parallel Prefix Adder", *International Journal of Science and Research (IJSR)*, vol. 3, no. 6, pp. 2339-2342, June 2012.

[5] H. Ling, "High Speed-Binary Adders", *IBM J. of R&D*, vol. 25, no. 2-3, pp. 156-166, May-June 1981.

[6] I. Hatai, I. Chakrabarti, and S. Banerjee, "An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation filter for multi-standard DUC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, May 2014 [Online]. Available: http://ieeexplore.ieee.org

[7] I. Kouretas, C. Basetas, and V. Paliouras, "Low-power logarithmic number system addition/subtraction and their impact on digital filters," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2196–2209, Nov. 2013.

[8] NurdianiZamhari Peter Voon, KuryatiKipli, Kho Lee Chin and MaimunHujaHusin, "Comparison of Parallel Prefix Adder (PPA)", *Proceedings of the World Congress on Engineering*, vol. 11, July 2012.

[9] Pakkiraiahchakali and Madhu Kumar Patnala, "Design of High Speed Beaumont-Smith Based Carry Select Adder", ", International Journal of Research in Engineering and Applied Sciences, vol. 4, no. 4,pp. 892-895, April 2013.

[10] P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," *IEEE Trans. on Computers*, vol. C-22, no. 8, Aug 1973.

[11] R. E. Ladner and M. J. Fischer, "Parallel Prefix Computation", *Journal of the ACM*, vol.27, no. 4, pp. 831-838, Oct. 1980.

[12] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 275–288, Feb. 2010.

[13] ShoalehHashemiNamin, Huapeng Wu, "Low-Power Design for a Digit-Serial Polynomial Basis Finite Field Multiplier Using Factoring Technique", *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 2, Feb 2017.

[14] Swapna K. Gedam and Pravin p. Zode, "Parallel Prefix Han-Carlson Adder", *International Journal of Research in Engineering and Applied Sciences*, vol.2, no.2, July 2014.

[15] Y. Jang and S. Yang, "Low-power CSD linear phase FIR filter structure using vertical common sub-expression," *Electron. Lett.*, vol. 38, no. 15, pp. 777–779, Jul. 2002.

[16] Y. Pan and P. K. Meher, "Bit-level optimization of adder-trees for multiple constant multiplications for efficient FIR filter implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 455–462, Feb. 2014.