African Journal of Basic & Applied Sciences 9 (1): 53-58, 2017
ISSN 2079-2034

DOI: 10.5829/idosi.ajbas.2017.53.58

Design of a Multiplier Architecture Based on LUT and VHBCSE Algorithm For FIR Filter

M. Jenifer Shopana and S. Poorna Lekha

Department of ECE, V V College of Engineering, Tirunelveli, Tamilnadu, India

Abstract: This paper presentsan efficient multiplier architecture based Upon Vertical Horizontal Binary Common Subexpression Elimination (VHBCSE) algorithm and memory based structure for FIR filter synthesis is proposed. The key requirement of FIR filter is low complexity. In a Finite Impulse Response (FIR) filter, the multiplier plays a vital role which defines the performance of the filter. Binary Common Subexpression Elimination (BCSE) algorithm is introduced to eliminate the Common Subexpression (CS) in binary form for designing an efficient multiplier. First, horizontal BCSE algorithm is introduced to find the CSs occurring within each coefficient (multiplier) followed by vertical BCSE which uses CSs found across adjacent coefficients inorder to eliminate redundant computations. Also Look Up Table (LUT) based multiplication are suggested to reduce the size over the conventional design. As a result, the proposed multiplier offers good reduction in the number of slices and LUT usage in realizing FIR filters. The proposed scheme is coded in Verilog language and simulation is performed using Xilinx ISE 13.4.

Key words: Binary Common Subexpression Elimination (BCSE) algorithm · Common Subexpression Elimination (CSE) · Finite Impulse Response (FIR) filter · Look Up Table (LUT)

INTRODUCTION

The ever increasing growth in laptop and portable systems in cellular networks has intensified the research efforts in low power microelectronics. Today, there are numerous portable applications requiring low power and high throughput than ever before. For example, notebook and laptop computers, representing the fastest growing segment of the computer industry, are demanding the same computation capabilities as found in desktop machines. The developments in personal communication services (PCS's), employ complex speech compression algorithms and sophisticated radio modems in a pocket sized device. Thus, low power system design has become a significant performance goal. So the designers faced more constraints: high speed, high throughput, less silicon area and at the same time, consumes as minimal power as possible.

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip [14]. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and area remain to be the two major design tools, power consumption has become a critical concern in today's VLSI system design. The FPGA is one of the best devices for hardware implementation of the DSP applications. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices.

The Finite Impulse Response (FIR) Filter is the important component for designing an efficient digital signal processing system [10]. When considered the elementary structure of an FIR filter, it is found that it is a combination of multipliers and delays, which in turn are the combination of adders. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in

Corresponding Author: M. Jenifer Shopana, Department of ECE, V V College of Engineering, Tirunelveli, Tamilnadu, India.

the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. Thus, the goal of the research is to design an efficient multiplier which could be used to build FIR filter.

Subexpression Elimination: Constant multiplication can be implemented efficiently by using dedicated shift-and-add multipliers. Subexpression elimination is a numerical transformation of the constant multiplication that can lead to efficient hardware implementations in terms of area, power and speed. Subexpression elimination is applied to a set of constant multiplications with the same constant multiplicand. Subexpression elimination can be performed on constant multiplications that operate on a common variable. Essentially, subexpression elimination is the process of examining the shift-and-add implementations of the constant multiplications and finding redundant operations. Once the redundancies are found, these operations can be performed once and shared among the constant multiplications [9].

Multiple Constant Multiplication(MCM): Subexpression elimination can be applied to a set of constant multipliers that multiply a common variable. The multiple constant multiplication problem determines how subexpression elimination can be applied to the set of constant multipliers so that the number of shifts and additions required for implementation is minimized. One of the attractive is its versatility and adaptability to various forms of problem.

Linear Transformation: Subexpression elimination can also be applied to linear transformations. When subexpression elimination is applied to constant multiplication, the constants. The nonzero bits in the binary representation of a constant determine the number of partial products that need to be generated and summed in order to realize the constant multiplier [9]. Consider the general form of the linear transformation: y=T*x, where T is m by n matrix, y is a length-m vector and x is a length-n vector. Writing this in an equivalent form, we have:

$$y_i = \sum_{j=1}^{n} t_{ij} x_j, i = 1,2 \quad (1)$$

When considering linear transformations, the subexpression elimination problem consists of 3 basic steps. The 1st step is to minimize the number of shifts and additions required to compute the products $t_{ij}x_j$, i=1,2.

Polynomial Evaluation: Subexpression elimination can be applied to polynomial evaluation to reduce the computational complexity. Suppose we are to evaluate the polynomial:

☻ + ☻ + ☻ + ☻ + ☻ (2)

If the redundancies in this calculation are not considered, the evaluation of this polynomial would require 22 multiplications. However, if the exponents are examined, the number of multiplications required to evaluate this polynomial can be reduced.

MATERIALS AND METHODS

VHBCSE Algorithm: VHBCSE algorithm is a combination of vertical and horizontal BCSE which is used for designing a FIR filter. Vertical and horizontal BCSEs are the two types of BCSE which eliminates the Common Subexpressions (CSs) present across the adjacent coefficients (multipliers) and within the coefficients (multipliers) respectively in aVHBCSE algorithm. Vertical BCSE is more effective in CS elimination compared to the horizontal BCSE. The proposed Vertical Horizontal Binary Common Subexpression Elimination (VHBCSE) algorithm is made of the following steps:

- Get the input of 16-bits (x[15:0]).
- Store coefficients of 16-bits (h[15:0]) in LUT.
- Partition coefficient (h[15:0]) into fixed groups of 2 bits each and use these groups as the select lines to the corresponding multiplexer (M7-M0) at layer1 as shown in Fig. 1.
- Partition the coefficient into groups of 4 bits each (h[15:12], h[11:8], h[7:4], h[3:0]).
- Compare h[15:12] with h[11:8] in layer 2.
- If match is found then skip the output of the adder A2.
- Otherwise, take the output of the adder A2.
- Compare h[15:12] with h[7:4] in the layer-2.
- If match is found then skip the output of the adder A3.
- Otherwise, compare h[11:8] with h[7:4] in the layer 2.
- If match is found then skip the output of the adder (A3).
- Otherwise, use the output of the adder (A3).
- Compare h[15:12] with h[3:0] in the layer-2.
- If match is found then skip the output of the adder A4.

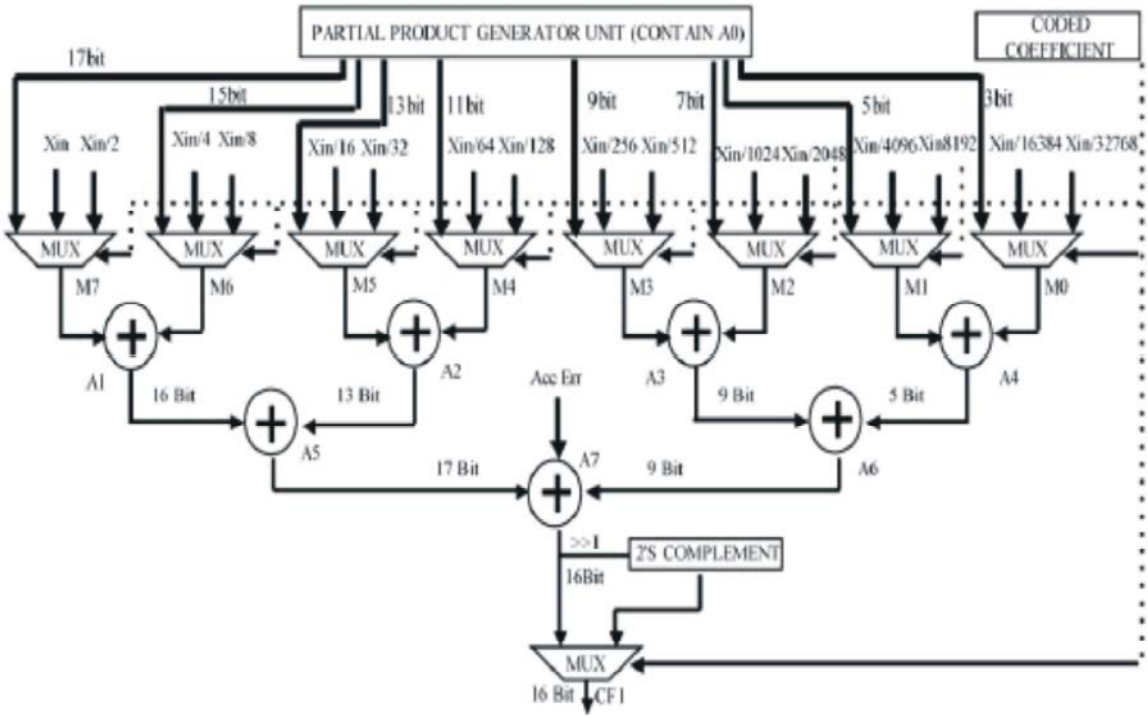

Fig. 1: Block Diagram of Multiplier and Final Addition Unit

- Otherwise compare h[11:8] with h[3:0] in the layer-2.
- If match is found then skip the output of the adder (A4).
- Otherwise, compare h[7:4] with h[3:0] in the layer-2.
- If match is found then skip the output of the adder (A4).
- Else use the output of the adder (A4).
- Partition the coefficient into fixed group of 8-bit (h[15:8], h[7:0]).
- Compare h[15:8] with h[7:0] in the layer 3.
- If match is found then skip the output of the adder A6.
- Else take the output of the adder A6.
- Obtain the final addition result by performing 1-bit right shift on the output of the adder A7.
- Multiplication is completed. Store this result, h*x, in the register.

Hardware Architecture of VHBCSE Algorithm: Consider the length of the input as Xin and coefficient as H with 16-bit and 16-bit respectively while the output is assumed to be 32-bit long. First the input (Xin) is stored in the register and the coefficients are stored in the LUTs. The data flow diagram of the proposed vertical-horizontal algorithm based on constant multiplier (CM) design consists of partial product generator block, multiplexer unit at layer-1, controlled addition at layer-2, controlled addition at layer-3, final addition at layer-4 and control logic generator. The data flow diagram of the proposed vertical-horizontal algorithm based Constant Multiplier (CM) design is shown in Fig. 2.

Functionality along with hardware architecture of different blocks of the designed VHBCSE based multiplier are explained in details below.

Partial Product Generator (PPG): In this algorithm, shift and add based technique is used to generate the partial product and summed up for producing the final multiplication result. The number of partial products is defined by the Choice of the size of BCS. Partial Product Generator block helps in reducing the size of multiplexer.

Control Logic (CL) Generator: Control logic generator block takes the coefficient (H[15:0]) as its input and groups it into one of 4-bit each (H[15:12], H[11:8], H[7:4] and H[3:0]). Another of 8-bit each (H[15:8], H[7:0]).

Multiplexer Unit: The multiplexer unit is used to select the data generated from the PPG unit depending on the coefficients binary value.This is used to reduce the hardware and power consumption.

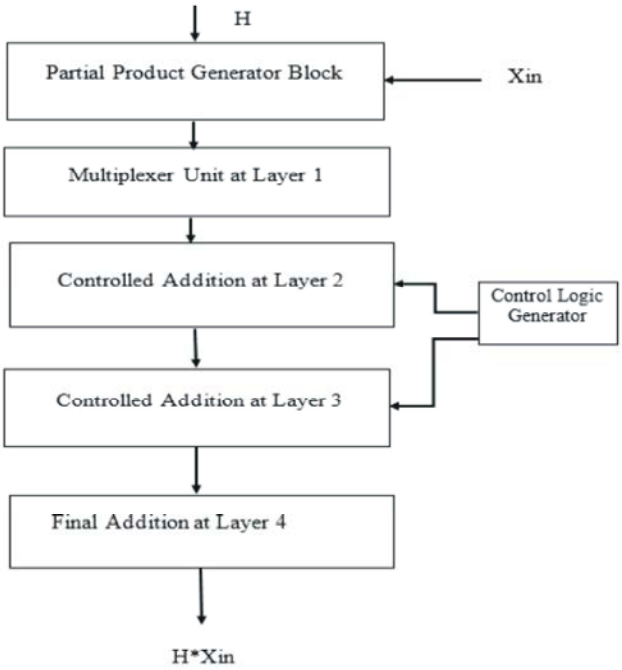

Fig. 2: Flow Diagram of VHBCSE algorithm

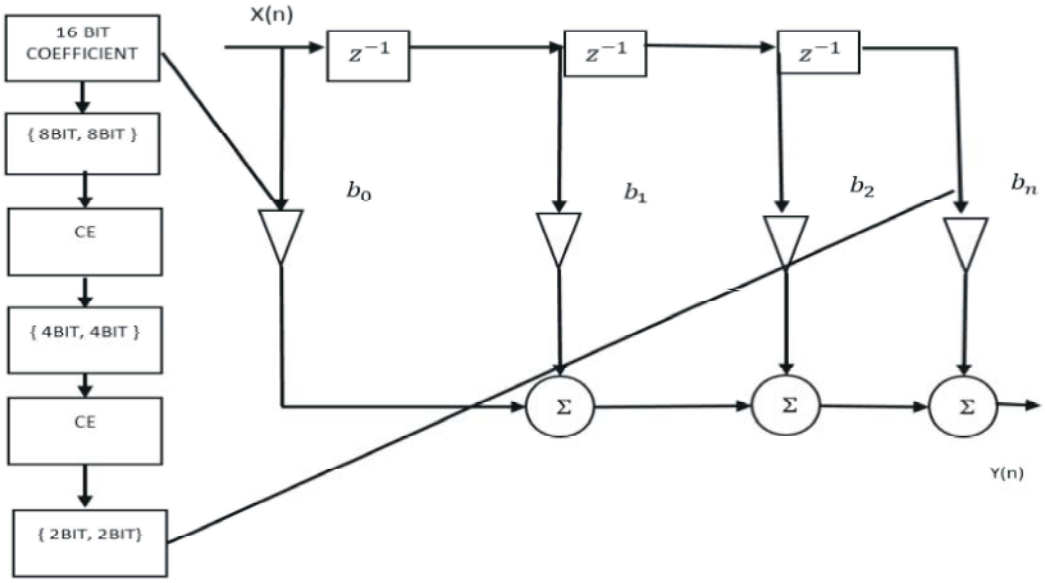

Fig. 3: LUT Multiplier Based FIR Filter

Controlled Addition at Layer-2: The Partial Products (PP) generated from eight groups of two-bit BCSs are added to produce the final multiplication results which have been performed in three layers.

Controlled Addition at Layer-3: The four multiplexed sums (AS1, AS2, AS3 and AS4) generated from layer-2 are summed up in layer-3. In our algorithm, controlled additions are performed, instead of direct addition of these four sums as shown in Figure 3. This addition (A6) is controlled by the control signal (C7) which is generated based on 8-bit BCSE from the CS generator block.

Final Addition at Layer-4: This block performs the addition operation between the two sums (AS5-AS6) produced by layer-3 to finally produce the multiplication result between the input and the coefficient.

Look Up Table (LUT)-Multiplier Based Filter: The "memory-based structures" or "memory-based systems" the memory elements like RAM or ROM is employed either as a section or whole of an arithmetic unit. Memory-based structures are more regular compared with the multiply-accumulate structures and have several alternative benefits.

Example, larger potential for high-throughput and reduced-latency implementation, (since the memory-access-time is much shorter than the usual multiplication-time) and are expected to own less dynamic power consumption due to less switching activities for memory-read operations compared to the standard multipliers. Memory-based structures are well suited for several digital signal processing (DSP) algorithms, thatinvolve multiplication with a set of coefficients. Figure 3 shows Look Up Table (LUT) multiplier based FIR filter which uses the VHBCSE algorithm in combination with the LUT approach.

RESULTS AND DISCUSSION

The proposed VHBCSE-LUT based multiplier is coded in Verilog language. The simulation results are obtained using Xilinx ISE 13.4. Using VHBCSE algorithm simulation is performed for 16 bit multiplication, x as its multiplicand and h as its multiplier which eliminates the common subexpression and produce the result. The Figure 4 shows the simulation result of vertical horizontal binary common subexpression algorithm based multiplier for FIR filter.

The Figure 5 shows the common subexpression found during multiplication operation.

From the Figure 6, the input Xin is given as 1010101010101010, initially the reset is given 1 and then reset is given 0. When the reset is given 0 the results of h0, h1, h2, h3, h4 are loaded for each clock cycle.

The Figure 7, shows that there is a considerable reduction in area based on the based on the usage of slices and LUT. The device utilization of VHBCSE algorithm as shown in Figure 7, the number of slices utilized is 469, number of 4 input LUT used is 838. The device utilization of VHBCSE and LUT based multiplier for FIR filter is shown in Figure 7, the number of slices utilized is 438, number of 4 input LUT used is 786. Based on the number of slices utilized and LUTs the result can be compared. The proposed VHBCSE-LUT algorithm significantly reduces area consumption when compared to the VHBCSE algorithm.

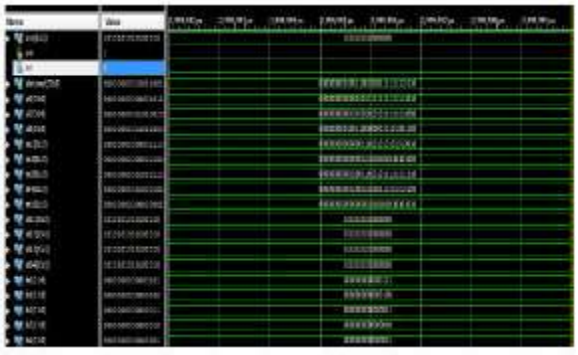

Fig. 4: Waveform of VHBCSE algorithm based multiplier

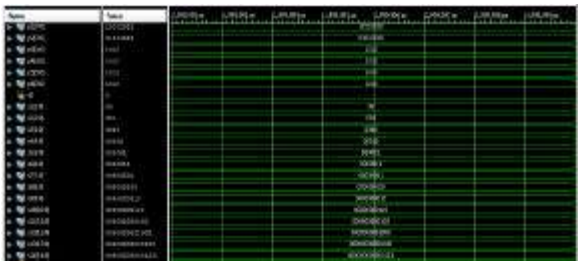

Fig. 5: Waveform of common subexpression

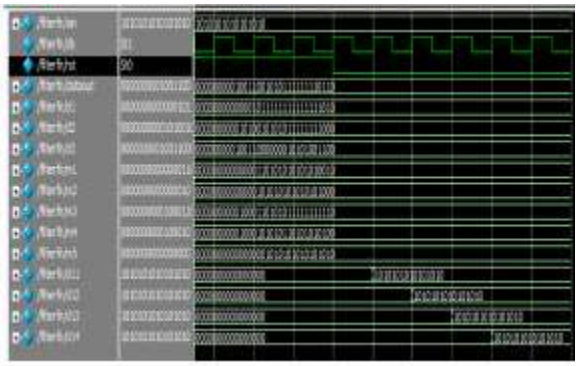

Fig. 6: Waveform of VHBCSE and LUT based multiplier for FIR filter

Fig. 7: Performance analysis interms of slice and LUT utilization

CONCLUSION

The proposed VHBCSE and LUT-multiplier-based filter presents one new Vertical-Horizontal BCSE algorithm which removes the initial common sub-expressions (CSs) by applying 8 bit and 4 bit common expressions combined with LUT method. New approaches to LUT-based-with multiplication are suggested to reduce the LUT-size over that of conventional design. The proposed method has nearly the same memory requirementand less number of slices and LUT usage than the existing methods.

REFERENCES

1. IndranilHatai, Indrajit Chakrabarti and S. Banarjee, 2015. 'An efficient VLSI architecture of a reconfigurable pulse-shaping FIR interpolation filter for multi-standard DUC', IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 23(6):1150-1154.
2. Baugh, C.R. and B.A Wooley, 1973. 'A two's complement parallel array multiplication algorithm', IEEE Transactions on Computer, c-22: 1045-1047.
3. Meher, P.K., 2010. 'New approach to look-up-table design and memory-based realization of FIR digital filter', IEEE Transactions on Circuits Systems-I, Regular Papers, 57(3): 592-603.
4. Wallace, C., 1965. 'A suggestion for a fast multiplier', IEEE Transactions on Electronics Computer, EC-13(1): 14-17.
5. Chia-Yu Yao, Hsin-Horng Chen, Tsuan-Fan Lin, Chiang-JuChien and Chun-Te Hsu, 2004. 'A Novel Common-Subexpression-Elimination Method for Synthesizing Fixed-Point FIR Filters', IEEE Transactions on Circuits and Systems-I: Regular Papers, 51(11): 2215-2221.
6. Ko, H.J. and S.F. Hsiao, 2011. 'Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation and rounding', IEEE Transactions on Circuits Systems-II, Analog Digital Signal Processing, 58(5): 304-308.
7. Indranil Hatai, Indrajit Chakrabarti and Swapna Banerjee, 2015. 'An efficient constant multiplier architecture based on vertical horizontal common subexpression algorithm for reconfigurable FIR filter synthesis', IEEE Transactions on Circuits and Systems-I: Regular Papers, 62(4): 1071-1080.
8. Dandapat, A., S. Ghosal, P. Sarkar and P. Mukhopadhyay, 2009. 'A 1.2 ns 16×16-Bit Binary Multiplier Using High Speed Compressors', International Journal of Electrical, Computer and Systems Engineering, pp: 234-239.
9. Keshab K. Parhi, 2008. 'VLSI Signal Processing Systems', John Wiley & Sons.
10. Morris Mano M., 2006. 'Digital Design', vol.3rd Edition, Prentice Hall of India Private Limited.
11. Nazieh Botros, 2000. 'HDL Programming fundamentals VHDL & Verilog', Da Vinci Engineering Press.
12. Wayne Wolf, 2008. 'Modern VLSI Design System-on-Chip Design', vol. 3rd Edition, prentice Hall.