Optimized Design of FIR Filter using Vedic Multiplier for Reconfigurable Applications

S.Keerthana and J.Julie Antony Roselin

Abstract—The likelihood of realization of the block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters is performed for both fixed and reconfigurable applications. FIR filter is the fundamental building block of Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), signal processing and Image processing applications. The transposed form block FIR filter contains adder, inner product unit (multiplier), and Coefficient Storage Unit (CSU) and register units. The CSU stores coefficients of the considerable number of channels to be utilized for the reconfigurable application. The reconfigurable FIR filter is designed with an array multiplier. So it provides high die size (area) and high power utilization. To overcome this problem, the reduced Vedic multiplier is composed by urdhva tiryagbhyam technique. Vertical and horizontal multiplication is carried out to reduce the partial product generation stages. Finally, this reduced Vedic multiplier unit is applied into the transposed form block FIR filter to achieve the low area, delay and low power. This structure involves significantly less Area Delay Product (ADP) and less Energy Per Sample (EPS) than the transposed form block FIR filter using an array multiplier.

Index Terms— Array multiplier, FIR filter, Multiple Constant Multiplication, Vedic multiplier.

1 INTRODUCTION

FINITE impulse response(FIR) filter is a filter whose impulse response is of finite duration, it settles to zero in finite time. FIR filter plays an important role in designing an efficient digital signal processing system. So, in this project a FIR channel is developed, which is effective as far as power and range, as well as far as delay. Uses of FIR filters are speech processing, loud speaker equalization, echo cancellation, adaptive noise cancellation [8].

Software Defined Radio (SDR) [4] is a radio communication system in which components are implemented by programming on PC or on embedded system rather than typically being implemented in hardware. An efficient coefficient coding scheme using Pseudo Floating Point [9] representation for implementing FIR filters in SDR receivers is proposed.

Multiple Constant Multiplication (MCM) dealt with efficient implementation of the filter using addition, subtraction and left shift of the input signal. Multiple Constant Multiplication (MCM) [2] is an arithmetic operation where fixed point variable is products with a set of fixed point constants. For MCM based implementation filter uses vertical and horizontal common sub expression elimination [3] method. This reduces the number of shift and add operation. Avoiding costly multipliers is important for hardware implementation. MCM scheme is more effective when number of constant multiplies with common operand. In this way MCM scheme is more successful for large order filters for fixed applications. Computation sharing multiplier [10] targets computation reuse in vector scalar product computation sharing multiplier for low power and high performance applications.

[1] Second year PG Student, VVCOE, Tisaiyanvilai-627657.
Email.id: c.josesherin10@gmail.com
[2] Assistant Professor, Dept of ECE, VVCOE, Tisaiyanvilai-627657.

The rest of the paper is organized as follows: Section 2 portrays the system model and Vedic multiplier operations.In section 3, we presented synthesis results with discussion. Finally, we conclude the paper in section 4.

2 SYSTEM DESCRIPTION

2.1 SYSTEM MODEL

The block FIR filter with block size L = 8 is considered. Fig. 1 demonstrates the structure of the block FIR filter. It comprises of one coefficient storage unit (CSU), one register unit (RU), M number of inner product units (IPUs), and one pipeline adder unit (PAU). The Coefficient Storage Unit stores coefficients of the considerable number of channels to be utilized for the reconfigurable application. Using N ROM LUTs, CSU can be implemented such that the filter coefficients of the particular channel filter are acquired in one clock cycle, where N is the filter length. During the Kth cycle, the register unit get input X_k and generates output $S^0{}_K$ in parallel.

Each IPU gets S0K parallel inputs from the register unit and short-weight vectors from CSU, such that during the Kth cycle, the (m + 1) th IPU receives L rows of S0K from the Register Unit and the weight vector CM-m-1 from the CSU. Each Inner Product Unit computes the product of S0K with the short-weight vector Cm and produces partial filter outputs $r^m{}_k$.

The pipeline adder unit adds the partial inner products and results in a block of L filter outputs Yk. In each cycle, the transpose form block FIR filter structure receives L rows of inputs Xk and produces L rows of filter outputs Yk. The time duration of each cycle is given by (1):

$$T = TM + TA + TFA \log 2L \quad (1)$$

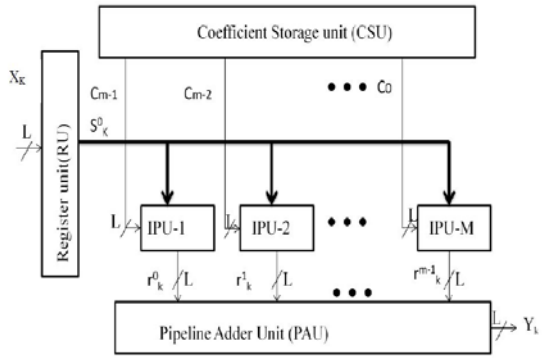

Fig. 1. Transpose form block FIR filter

Where, TM is one multiplier delay, TA is one adder delay, and TFA is one full-adder delay.

In previous system models, 8x8 Array multiplier [1] was used to perform the product operations in the Inner product unit. Adding and shift algorithm are used in multiplier circuit where the multiplicand is multiplied with the multiplier bit and the partial product is generated. Shifting of the partial products are done according to their bit orders and then added using carry propagate adder. Carry Save Adders are used in which every carry and sum signal are passed to the adders of the next stage. The Final product is obtained in a final adder by any fast adder.

2.2 Vedic Multiplier

Inner product unit are built using Vedic multiplier to perform any N x N bit multiplication. Here, 16 bit multiplication operations are performed using and 4 x 4 bit Vedic multiplier. Fig. 2 depicts the architecture of 4×4 bit Vedic multiplier. The 4-bit input sequence is divided into two 2-bit numbers (a[3:2] & b[3:2], a[1:0] & b[1:0]) and given as inputs to the 2-bit multiplier blocks.

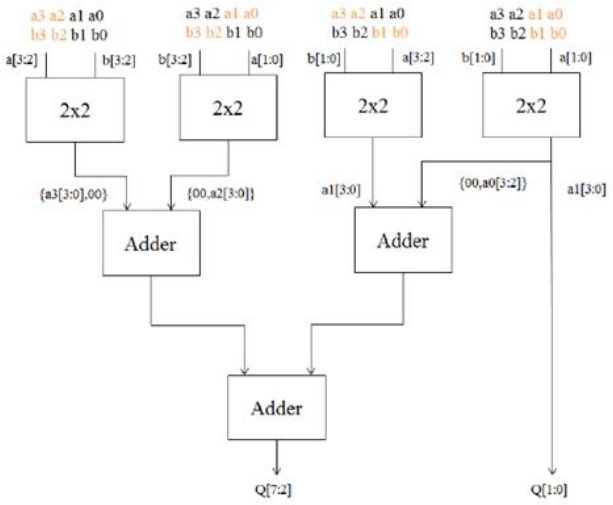

Fig. 2 Structure of 4×4 bit Vedic multiplier.

The 4x4 bit Vedic multiplier multiplies the 2 bit input sequence and generates an intermediate product of 4-bits which are added using overlapping logic in the three modified. The LSB product output Q[1:0] are directly obtained from the first multipliers. ADDER-1 adds the outputs from the second and third multiplier and ADDER-2 adds the outputs from the third and fourth multiplier and both the outputs are added by ADDER 3, which gives the MSB product output Q[7:2].

As the partial product and their sum are calculated in parallel in Vedic multiplier, the multiplier is independent of its clock frequency and so it does not require high clock frequencies for multiplication and results in less switching delay with minimal power.

3 SYNTHESIS RESULTS AND DISCUSSIONS

The transposed form block FIR filter using Array multiplier and Vedic multiplier are coded in Verilog language. The simulation results are obtained in ModelSim- Altera 6.6c and synthesis results are obtained in Xilinx ISE 13.2. Synthesis results using array multiplier are obtained from Xilinx ISE 13.2 and are depicted in Fig. 3.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	3,080	21,504	14%
Number of 4 input LUTs	6,514	21,504	30%
Number of occupied Slices	3,571	10,752	33%
Number of Slices containing only related logic	3,571	3,571	100%
Number of Slices containing unrelated logic	0	3,571	0%
Total Number of 4 input LUTs	6,700	21,504	31%
Number used as logic	5,835		
Number used as a route-thru	186		
Number used as Shift registers	679		
Number of bonded IOBs	43	448	9%

Fig. 3 Device utilization report of FIR filter using Array multiplier

The number of slice Flip Flops utilized by the array multiplier are 3,080 and the number of 4 input LUTs are 6,514. Number of occupied slices are 3,571. Number of slices containing only related logic are 3,571.

The timing summary using Array multiplier are shown in Fig. 4. The minimum input arrival time before clock is 12.810ns and the maximum output time required after clock is 6.878ns. Fig. 5 shows the power analysis using Array multiplier.

```
Timing Summary:
---------------
Speed Grade: -12

   Minimum period: 5.894ns (Maximum Frequency: 169.661MHz)
   Minimum input arrival time before clock: 12.810ns
   Maximum output required time after clock: 6.878ns
   Maximum combinational path delay: No path found
```

Fig. 4 Synthesis summary

Fig. 5 Power analysis using Array multiplier by Xpower Analysis

The Array multiplier consumes more power and it requires more number of components and gates for its implementation which requires more area. Synthesis results using vedic multiplier are obtained from Xilinx ISE 13.2 and are depicted in Fig. 6.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	1,828	21,504	8%
Number of 4 input LUTs	5,311	21,504	24%
Number of occupied Slices	3,049	10,752	28%
Number of Slices containing only related logic	3,049	3,049	100%
Number of Slices containing unrelated logic	0	3,049	0%
Total Number of 4 input LUTs	5,507	21,504	25%
Number used as logic	5,311		
Number used as a route-thru	196		
Number of bonded IOBs	43	448	9%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig. 6 Device utilization report of FIR filter using Vedic multiplier

The number of slice Flip Flops utilized by the Vedic multiplier are 1,828 and the number of 4 input LUTs are 5,311. The number of occupied slices are 3,049. The timing summary using Vedic multiplier are shown in Fig. 7.

```
Timing Summary:
---------------
Speed Grade: -12

   Minimum period: 7.662ns (Maximum Frequency: 130.519MHz)
   Minimum input arrival time before clock: 12.774ns
   Maximum output required time after clock: 6.878ns
   Maximum combinational path delay: No path found
```

Fig. 7. Synthesis summary

The minimum input arrival time before clock is 12.774ns and the maximum output required time required after clock is 6.878ns. Fig. 8 shows the power analysis using Vedic multiplier.

Fig. 8 Power analysis using Vedic multiplier by Xpower Analysis

The proposed Vedic multiplier of the transposed form block FIR filter reduces power consumption compared with the array multiplier. Fig. 9 shows the performance analysis of transpose form block FIR filter using both multipliers. The graph are plotted based on the slice and LUT utilization of both array and Vedic multiplier blocks in IPU.

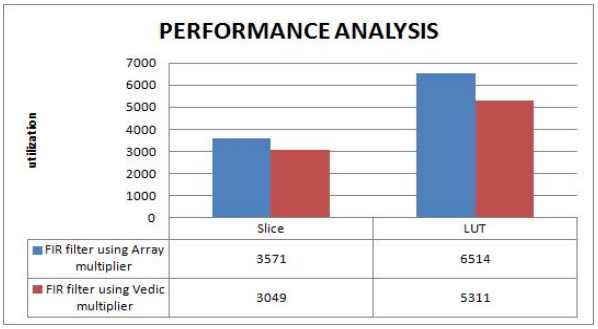

Fig. 9 Performance Analysis

By using Vedic multiplier, the number of occupied slices have been reduced to 5% when comparing it with the array multiplier. Similarly, the number of input LUTs have been reduced to 6% when comparing it with the array multiplier. Table 1 compares the Slices, delay and the power utilization of FIR filter by using both array multiplier and Vedic multiplier.

TABLE 1
COMPARISION OF FIR FILTER USING ARRAY MULTIPLIER AND VEDIC MULTIPLIER

Methods	Slices(area)	Delay	Power
FIR filter using Array multiplier	3571	12.810ns	5.751w
FIR filter using Vedic multiplier	3049	12.774ns	4.267w

The Vedic multiplier occupies 522 slices less than that of Array multiplier with delay reduction of about 0.036 ns and consumes 1.484w less than that of array multiplier. Hence, Vedic multiplier utilizes less area, delay, and power consumption compared with Array multiplier.

4 CONCLUSION

In this paper, Vedic multiplier is used to perform the multiplication operations in the IPU of the transpose form FIR filter as this multiplier is suitable for multiplying large number of bits in parallel. By comparing Vedic multiplier over array multiplier, Vedic multiplier is more convenient than an array multiplier. The simulation results demonstrate that the Vedic multiplier delivers low area, delay and lower power utilization as compared to the array multiplier. The hardware complexity of Vedic algorithm is less than the array multiplier. From the obtained results, the FIR filter with Vedic multiplier requires low area and low power than FIR filter using Array multiplier and as a result an efficient processor with low delay and minimum power consumption is achieved.

In future work, the filter design are modified for fixed applications by using parallel prefix adder and wave pipeline adder unit in Multiple Constant Multiplication (MCM) structure and its synthesis results are compared with the existing works.

REFERENCES

[1] Basant Kumar Mohanty, and Pramod Kumar Meher, 'A High- Performance FIR Filter Architecture for fixed and Reconfigurable Applications',*IEEE Transaction.*VLSI, Jan 2016.

[2] Birdawade Kishori , 'FPGA Realization of FIR Filter by Efficient Multiple Constant Multiplication for Fixed Application', *IJIRCCE* Vol. 4, Issue 6, June 2016.

[3] Mahesh R and Vinod A.P, 'A new common sub expression elimination algorithm for realizing low-complexity higher order digital filters', *IEEE Trans. Compute.-Aided Design Integer. Circuits Syst*., vol. 27, no. 2,pp. 217–219, Feb 2008.

[4] Mahesh R and Vinod A.P, 'New reconfigurable architectures for implementing FIR filters with low complexity', *IEEE Transaction Computer.-Aided Design Integer. Circuits Syst.*, vol. 29, no. 2, pp. 275–288, Feb 2010.

[5] Meher P.K , "Hardware-efficient systolization of DA-based calculation of finite digital convolution," *IEEE Trans. Circuits Syst. II, Exp. Briefs*,vol. 53, no. 8, pp. 707–711, May 2005.

[6] Meher P.K,Chandrasekaran S, and Amira A (2008), 'FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic', *IEEE Transaction. Signal Process*, vol. 56, no. 7, pp.3009–3017, July 2008.

[7] Meher P.K and Mohanty B.K, 'Memory Footprint Reduction for Power-Efficient Realization of 2-D Finite Impulse Response Filters', *IEEE Transactions on Circuits and Systems*, Jan 2014.

[8] Mirchandani E, Zinser R.L, and Evans J.B, 'A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals]', *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 10,pp. 681–694, Oct 1995.

[9] Vinod A.P and Lai E.M, 'Low power and high-speed implementation of FIR filters for software defined radio receivers', *IEEE Transaction. Wireless Communication*., vol. 7, no. 5, pp. 1669–1675, July 2006.

[10] Park J, Jeong W, Mahmoodi-Meimand H,Wang Y,Choo H, 'Computation sharing programmable FIR filter for low-power and high performance applications', *IEEE J. Solid State Circuits*,vol. 39, no. 2, pp. 348–357, Feb 2004.