Design and implementation of DDA architecture for FIR Filters

T. Ranjith Kumar

M.Tech (DSCE) & Department of ECE & JNTU-Hyderabad
H.no:15-21-31, Balaji nagar, Kukatpally, Hyderabad

Abstract –**Traditionally, direct implementation of a K-tap FIR filter requires K multiply-and-accumulate (MAC) blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. To resolve this issue, we first present DA, which is a architecture without multiplier. This paper implements the DA architecture. This architecture is applicable to only one type of filter Coefficients i.e., fixed filter coefficient. In case if we want to operate on variable filter coefficients we have been using Dynamic Distributed Arithmetic (DDA) Architecture. In this we are providing the flexibility to operate on variable filter coefficients. Here also compare DA(Distributed Arithmetic),D-DA(Decomposed-Distributed Arithmetic),DDA(Dynamic Distributed Arithmetic) by using of XILINX ISE 9.1.i tool, for simulation and synthesis, dumping on sparton-3E FPGA.**

Keywords– Distributed Arithmetic; FIR; Decomposed DA; dynamic DA.

I. INTRODUCTION

In the recent years, there has been a growing trend to implement digital signal processing functions in Field Programmable Gate Array (FPGA). In this sense, we need to put great effort in designing efficient architectures for digital signal processing functions such as FIR filters, which are widely used in video and audio signal processing, telecommunications and etc. Traditionally, the implementation of a K-tap FIR filter requires K multiply-and-accumulate (MAC) blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. To resolve this issue, we first present DA, which is a multiplier-less architecture.

Implementing multipliers using the logic fabric of the FPGA is costly due to logic complexity and area usage, especially when the filter size is large. Modern FPGAs have dedicated DSP blocks that alleviate this problem, however for very large filter sizes the challenge of reducing area and complexity still remains. An alternative to computing the multiplication is to decompose the MAC operations into a series of lookup table (LUT) accesses and summations. This approach is termed distributed arithmetic (DA), a bit serial method of computing the inner product of two vectors with a fixed number of cycles.

The original DA architecture stores all the possible binary combinations of the coefficients w[k] of equation (1) in a memory or lookup table. It is evident that for large values of L, the size of the memory containing the pre computed terms grows exponentially too large to be practical. The memory size can be reduced by dividing the single large memory (2Lwords) into m multiple smaller sized memories each of size 2k where L = m × k. The memory size can be further reduced to 2L−1 and 2L−2 by applying offset binary coding and exploiting resultant symmetries found in the contents of the memories.

This technique is based on using 2's complement binary representation of data, and the data can be pre-computed and stored in LUT. As DA is a very efficient solution especially suited for LUT-based FPGA architectures, many researchers put great effort in using DA to implement FIR filters in FPGA. Patrick Longa introduced the structure of the FIR filter using DA algorithm and the functions of each part. Sangyun Hwang analyzed the power consumption of the filter using DA algorithm. Heejong Yoo proposed a modified DA architecture that gradually replaces LUT requirements with multiplexer/adder pairs. But the main problem of DA is that the requirement of LUT capacity increases exponentially with the order of the filter, given that DA implementations need 2Kwords (K is the number of taps of the filter). And if K is a prime, the hardware resource consumption will cost even higher. To overcome these problems, this paper presents a hardware-efficient DA architecture.

This method not only reduces the LUT size, but also modifies the structure of the filter to achieve high speed performance. The proposed filter has been designed and synthesized with ISE 9.11i, and implemented with a 4VLX40FF668 FPGA device. Our results show that the proposed DA architecture can implement FIR filters with high speed and smaller resource usage in comparison to the previous DA architecture.

II. DISTRIBUTED ARITHMETIC FIR FILTER ARCHITECTURE

Distributed Arithmetic is one of the most well-known methods of implementing FIR filters. The DA solves the computation of the inner product equation when the coefficients are pre knowledge, as happens in FIR filters.
An FIR filter of length K is described as:

$$y[n]=\sum_{k=0}^{K-1} h[k]x[n-k].......(1)$$

Where h[k] is the filter coefficient and x[k] is the input data. For the convenience of analysis, x'[k] =x [n - k] is used for modifying the equation (1) and we have:

$$y[n]=\sum_{k=0}^{K-1} h[k].x'[k].........(2)$$

Then we use B-bit two's complement binary numbers to represent the input data:

$$x'[k]=-2^B.x_B[k]+\sum_{b=0}^{B-1} x_b[k].2^b..........(3)$$

Where $x_b[k]$ denotes the b’th bit of x[k], $x_b[k] \in$ {0, 1}.
Substitution of (3) into (2) yields:

$$y=\sum_{k=0}^{K-1} h[k]\cdot(-2^B\cdot x_B[k]+\sum_{b=0}^{B-1} x_b[k]\cdot 2^b)$$

$$=-2^B\cdot\sum_{k=0}^{K-1} h[k]\cdot x_B[k]+\sum_{b=0}^{B-1} 2^b\cdot\sum_{k=0}^{K-1} h[k]\cdot x_b[k]$$

$$=-2^B\cdot f(h[k],x_B[k])+\sum_{b=0}^{B-1} 2^b\cdot f(h[k],x_b[k])\;............\;(4)$$

We have

$$f(h[k],\,x_b[k])=\sum_{k=0}^{K-1} h[k]\cdot x_b[k]\quad ....(5)$$

In equation (4), we observe that the filter coefficients can be pre-stored in LUT and addressed by X_b = [$x_b[0], x_b[1],... x_b[K-1]$].This way, the MAC blocks of FIR filters are reduced to access and summation with LUT.

The implementation of digital filters using this arithmetic is done by using registers, memory resources and a scaling accumulator.

Original LUT-based DA implementation of a 4-tap (K=4) FIR filter is shown in Figure 4.1. The DA architecture includes three units: the shift register unit, the DA-LUT unit, and the adder/shifter unit.

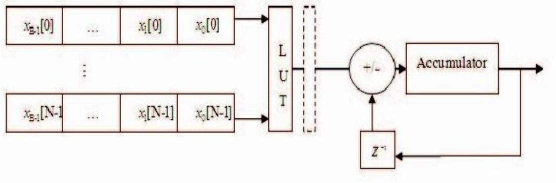
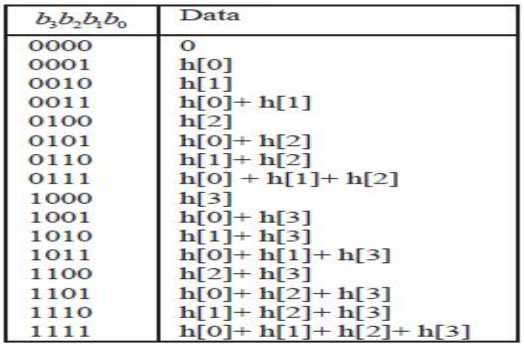


Fig.1: Basic DA MAC architecture.



$b_3b_2b_1b_0$	Data
0000	0
0001	h[0]
0010	h[1]
0011	h[0]+ h[1]
0100	h[2]
0101	h[0]+ h[2]
0110	h[1]+ h[2]
0111	h[0] + h[1]+ h[2]
1000	h[3]
1001	h[0]+ h[3]
1010	h[1]+ h[3]
1011	h[0]+ h[1]+ h[3]
1100	h[2]+ h[3]
1101	h[0]+ h[2]+ h[3]
1110	h[1]+ h[2]+ h[3]
1111	h[0]+ h[1]+ h[2]+ h[3]

Fig. 2: Co-efficient values of LUT for a 4-tap filter

III. DECOMPOSED DA-LUT UNIT

In Fig. 2, we can see that the lower half of LUT (locations where b_3 =1) is the same with the sum of the upper half of LUT (locations where b_3 =0) and h [3]. Hence, LUT size can be reduced 1/2 with an additional 2x1 multiplexer and a full adder, as shown in Fig. 3. On other side, for the use of combination logic circuit, the filter performance will be affected. But when the taps of the filter is a prime, we can use 4-input LUT units with additional multiplexers and full adders to get the tradeoff between filter performance and small resource usage.

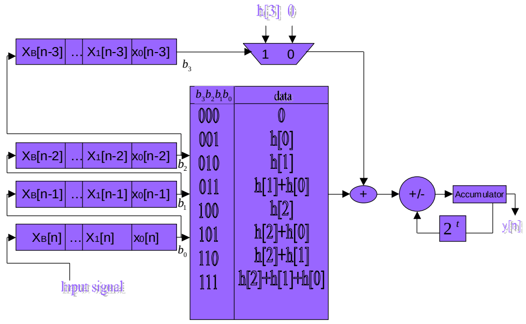


Fig. 3: Decomposed DA architecture for a 4-tap filter

IV. DYNAMIC DISTRIBUTED ARITHMETIC

Dynamic Distributed Arithmetic (DDA) Algorithm is suitable for designing digital fir filter with varying co-efficient as compared to the conventional Distributed Arithmetic (DA) Algorithm based fir filter design where the filter co-efficient are constant. Whenever there is a change in filter co-efficient, the co-efficient buffer is updated dynamically and then performs the defined operations.

Working of the DDA

The co-efficient are stored in the Co-efficient buffer which outputs the value based on the C[n] value. The input buffer generates the address from the given inputs and are applied to the LUT. The LUT get updated with filter co-efficient available in the co-efficient buffer corresponding to the value of C[n]. The shift register will shift the value in the accumulator. At last, the outputs of registered LUTs are added and loaded to the scaling accumulator from LSB to MSB and the result which is the filter output will be accumulated on to the output register

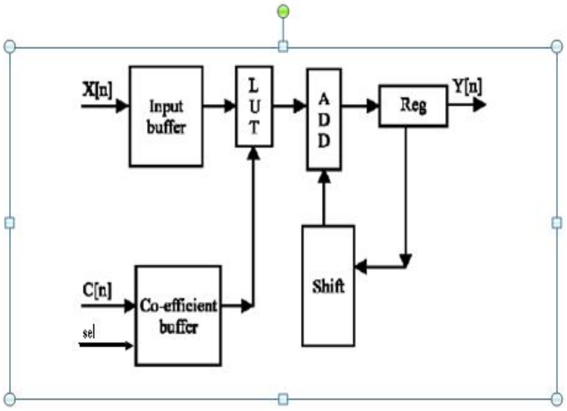
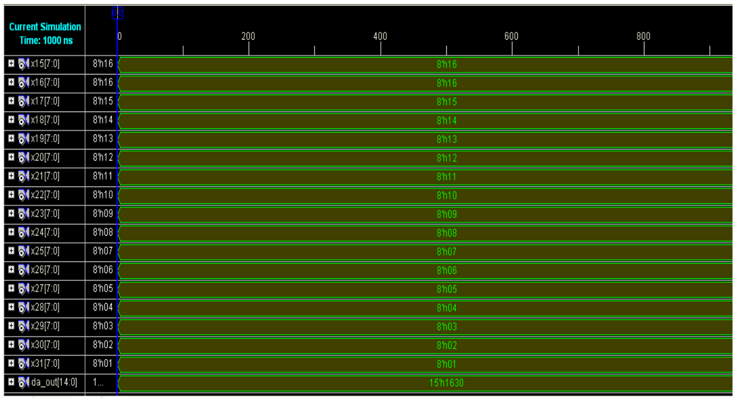


Fig. 4: Dynamic DA architecture

A Fast LUT based Dynamic DA FIR configuration with multiple coefficient banks which is able to change the coefficients of the filter taps according to signal by using table partitioning algorithm in designing the FIR structure.

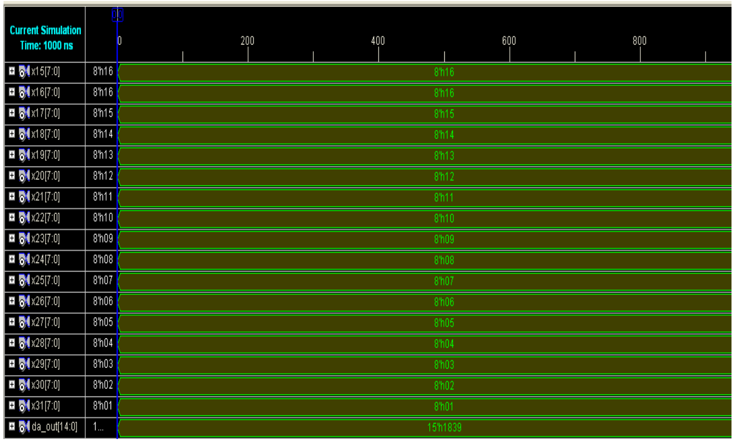
V. RESULTS

Here, the simulation results for Dynamic DA as well as the comparison results for the three architectures by using Xilinx ISE tool are shown below:



```
x15[7:0]      8'h16   8'h16
x16[7:0]      8'h16   8'h16
x17[7:0]      8'h15   8'h15
x18[7:0]      8'h14   8'h14
x19[7:0]      8'h13   8'h13
x20[7:0]      8'h12   8'h12
x21[7:0]      8'h11   8'h11
x22[7:0]      8'h10   8'h10
x23[7:0]      8'h09   8'h09
x24[7:0]      8'h08   8'h08
x25[7:0]      8'h07   8'h07
x26[7:0]      8'h06   8'h06
x27[7:0]      8'h05   8'h05
x28[7:0]      8'h04   8'h04
x29[7:0]      8'h03   8'h03
x30[7:0]      8'h02   8'h02
x31[7:0]      8'h01   8'h01
da_out[14:0]  1...    15'h1630
```

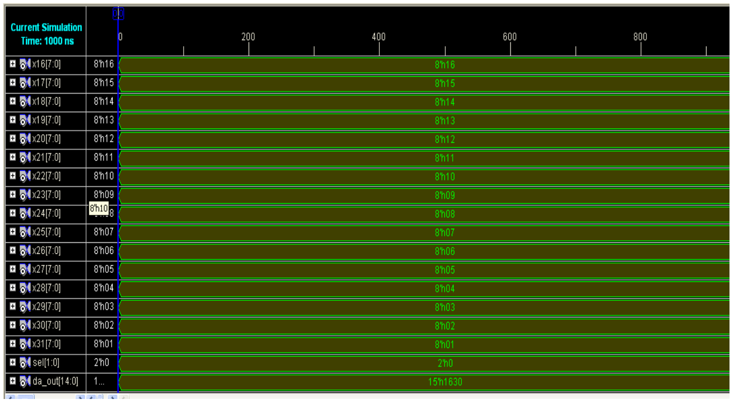
(a)



Current Simulation Time: 1000 ns

x15[7:0]	8'h16	8'h16
x16[7:0]	8'h16	8'h16
x17[7:0]	8'h15	8'h15
x18[7:0]	8'h14	8'h14
x19[7:0]	8'h13	8'h13
x20[7:0]	8'h12	8'h12
x21[7:0]	8'h11	8'h11
x22[7:0]	8'h10	8'h10
x23[7:0]	8'h09	8'h09
x24[7:0]	8'h08	8'h08
x25[7:0]	8'h07	8'h07
x26[7:0]	8'h06	8'h06
x27[7:0]	8'h05	8'h05
x28[7:0]	8'h04	8'h04
x29[7:0]	8'h03	8'h03
x30[7:0]	8'h02	8'h02
x31[7:0]	8'h01	8'h01
da_out[14:0]	1...	15'h1839

(b)



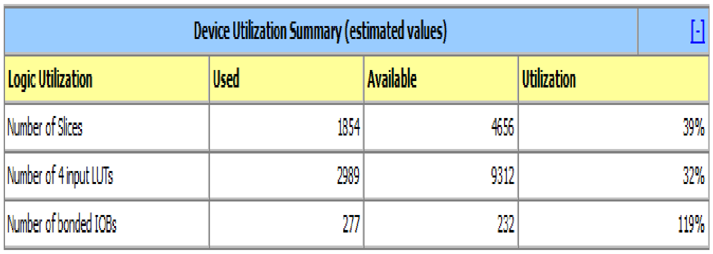
Current Simulation Time: 1000 ns

x16[7:0]	8'h16	8'h16
x17[7:0]	8'h15	8'h15
x18[7:0]	8'h14	8'h14
x19[7:0]	8'h13	8'h13
x20[7:0]	8'h12	8'h12
x21[7:0]	8'h11	8'h11
x22[7:0]	8'h10	8'h10
x23[7:0]	8'h09	8'h09
x24[7:0]	8'h10	8'h08
x25[7:0]	8'h07	8'h07
x26[7:0]	8'h06	8'h06
x27[7:0]	8'h05	8'h05
x28[7:0]	8'h04	8'h04
x29[7:0]	8'h03	8'h03
x30[7:0]	8'h02	8'h02
x31[7:0]	8'h01	8'h01
sel[1:0]	2'h0	2'h0
da_out[14:0]	1...	15'h1630

(c)

Fig. 5: Simulation results for (a) DA, (b) Decomposed DA, (c) Dynamic DA.

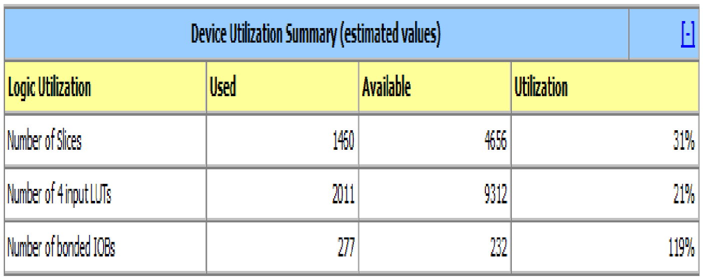
The design utilization values for the DA and Decomposed Da are given in below table:



Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slices	1854	4656	39%
Number of 4 input LUTs	2989	9312	32%
Number of bonded IOBs	277	232	119%

(a).



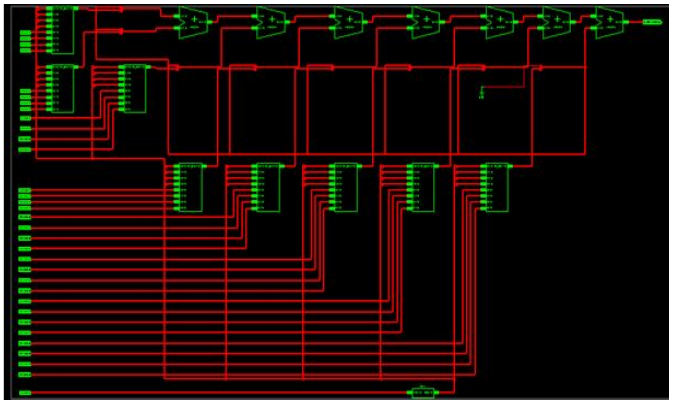
Device Utilization Summary (estimated values) [-]

Logic Utilization	Used	Available	Utilization
Number of Slices	1460	4656	31%
Number of 4 input LUTs	2011	9312	21%
Number of bonded IOBs	277	232	119%

(b)

Table 1: Design utilization of (a) DA, (b) Decomposed DA

The RTL schematic for Dynamic DA was as follows



V. CONCLUSION

This paper concludes that the design and implementation of DDA (dynamic DA) and comparison results between the basic DA, Decomposed DA and DDA. This architecture provides the flexibility to operate on variable filter coefficients as per our requirement. The test results indicate that the designed filter using Distributed Arithmetic can work stable with high speed and can save hardware resources. Meanwhile, it is very easy to transplant the filter to other applications through modifying the order parameter or bit width and other parameters and therefore have great practical applications in digital signal processing.

REFERENCES

[1] Vhdl programming by J.Baskar.

[2] Distributed Arithmetic for FIR Filter implementation on FPGA by Yajun Zhou**,** Pingzheng Shi.

[3] Uwe Meyer-Baese.Digital signal processing with FPGA[M]. Beijing:Tsinghua University Press,2006:50~51

[4] Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010,PP(99):1~5.

[5] Chao Cheng and Keshab K Parhi. Low-Cost Parallel FIR Filter Structures With 2-Stage Parallelism[J].IEEE Transactions on Circuits and Systems I: Regular 2007,54(2):280~290.

[6] Tearney G J and Bouma B E. Real-Time FPGA Processing for High-Speed Optical Frequency Domain Imaging [J]. IEEE Transactions on Medical Imaging, 2009,28(9):1468~1472.

-

[7] Hu Guang-shu. Digital signal processing-theory,algorithm and realizes[M]. 2nd ed.Beijing: Tsinghua University Press,2003:296~307.

[8] Chun Hok Ho,Chi Wai Yu and Leong P. Floating-Point FPGA: Architecture and Modeling [J]. IEEE Transactions on Very Large Scale Integration Systems, 2008,17(12): 1709~1718.

[9] Evans J B. Efficient FIR filter architectures suitable for FPGA implementation[J].IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 2002,41(7):490~493.

[10] Meher P K , Chandrasekaran S and Amira A. FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic [J].
IEEE Transactions on Signal Processing, 2008,56(7): 3009~3017.

[11] Xia Yu-wen. Digital system design with Verilog[M]. 2nd ed.Beijing:Higher Education Press,2008:102~103.

[12] Sungwook Yu and Swartziander E E. DCT implementation with distributed arithmetic[J]. IEEE Transactions on Computers, 2001,50(9):985~991.