

## FPGA Hardware Resource Specific Optimal Design for FIR Filters

Hira Ilyas<sup>1</sup> and Shoab Khan<sup>2</sup>

<sup>1,2</sup> Computer Engineering, Center for Advance Studies in Engineering, Islamabad, Pakistan.

<sup>1</sup>hirailyas786@yahoo.com, <sup>2</sup>kshoab@yahoo.com

### ABSTRACT

This paper presents a strategy to use a particular implementation that only uses a set of available resources and minimize the use of other. As an FPGA has many resources like multipliers, adders, distributed RAM, look up table and equivalent millions of gates and it completely maps application that utilizes different hardware resources. Some algorithms can be mapped to use particular resources or the other algorithms for the same application utilize other resources. In this paper FIR filters with different techniques are implemented and the resource utilization of these different algorithms are compared using Virtex 6 FPGA. The time efficient technique is also presented in this paper. At the end the tool is designed, which takes resources and coefficients from user and generates RTL verilog code according to them.

**Keywords:** *Canonic Sign Digit (CSD), Distributed Arithmetic (DA), Field-Programmable Gate Arrays (FPGA), Global Correction Vector (GCV).*

### 1. INTRODUCTION

Filters are used to remove unwanted component of signals. They extract the required signal from the noisy signal which contains unwanted disturbances. In recent years filters has been widely applied to voice, image and communication. Filtering is a fundamental DSP technique having many applications. There are two type of digital filters Finite Impulse Response and Infinite Impulse Response. FIR filter implementations with different algorithms are discussed in this paper. In FIR algorithms a large proportion of multiplications are by a constant number. These algorithms can be specified in many programming languages and can be executed on FPGA because of its reconfigure-ability and reprogram-ability [3]-[4].

The FIR filter is mathematically expressed in Eq. 1 where each output  $y(n)$  is equal to a weighted sum of a finite number of past and present input samples. Figure 1 shows  $n$ -tap FIR filter [2].

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad \text{Eq.(1)}$$

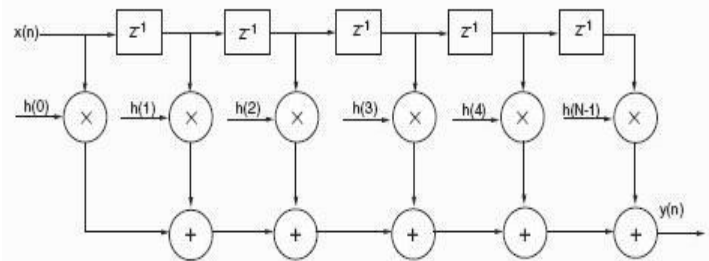


Fig. 1.  $n$ -tap FIR Filter

Four algorithms are used to implement FIR filter with 10 coefficients. The first approach used multiplier and adders to implement the FIR filter whereas the second approach converts the filter coefficients in CSD format while considering maximum of four non-zero CSD digits for each coefficients. There are a number of partial products that are generated the RTL code simply adds the PP, another approach to implement the filter by computing GCV. The vector is added in place of all the sign bits and 1s that are there to cater for two's complement in the PPs. Finally the Distributed Arithmetic architecture can be effectively used for implementing FIR filter. This design eliminates the need to use hardware multiplier and uses only look up table to provide high throughput execution and yields faster output irrespective of the filter length and width of the coefficients [1].

This paper is organized in Six sections. Section 2 presents the resource utilization of these techniques. Section 3 describes a comparison of these techniques with respect to the usage of hardware resources. Section 4 presents time efficient technique. In Section 5 the application is designed which generates FIR filter RTL using most suitable resources. Section 6 concludes the paper.

### 2. HARDWARE RESOURCE UTILIZATION

In this section the resource utilization of FPGA virtex 6 is presented with different techniques and number of coefficients.

H. Ilyas and S. Khan

## 2.1 MULIPLIERS

This is the simplest method to implement FIR filter. Verilog RTL code was written using adders and multipliers for 2 to 10 number of coefficients on Xilinx ISE 12.1 with FPGA virtex 6 and found the following utilization of resources. Fig 2 shows the resource utilization where a to i are the number of coefficients from 2 to 10. The number of look up table and flip- flops for coefficient two is zero.

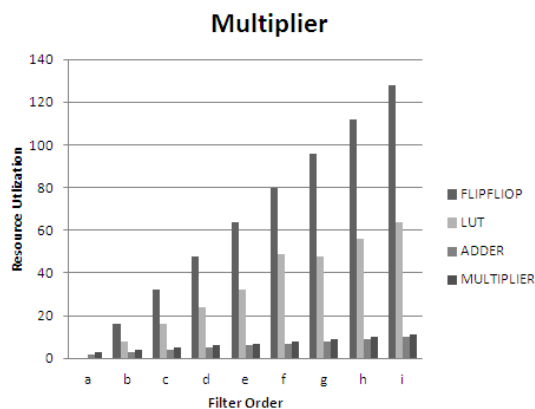


Fig. 2. utilization of resources in conventional FIR filter

The look up table here represents the number used exclusively as route-thru and number used as logic is 0 so this design is not using look up tables. Number of flip-flops, adders and multipliers increases as we increase the number of coefficient.

Multipliers play an important role to increases the hardware complexity of filters on FPGA. For a real time application such as filtering multipliers are used because of their high speed. The multiplier-based design of FIR filters are highly expensive in term of area the complexity grows as the number of coefficients increases [9]. As the number of coefficients increases multipliers increases, the high order demand more hardware requirement, more arithmetic operations, more area and power consumption. [8] Therefore the most important task is to reduce these parameters. It is done in the next techniques which are multiplier less designs.

## 2.2 CANONIC SIGN DIGIT

In many digital systems the signal is multiplied with a constant number so half of the information is given

already. In FIR filter all coefficients are constants. For a fully parallel implementation general purpose multipliers are not required and coefficients are converted in canonic sign digit form [6]. The CSD number system minimizes the number of non-zero digits so therefore number of partial products additions in hardware multiplier reduced in this system. More than two consecutive non-zero bits are not allowed. This form contains minimum possible number of non-zero bits [5]-[8]. Whenever find more than two 1, convert the first 1 to negative 1 then put all 0 and at the end put positive 1, generate partial products only for 1s in the constant multiplier. Instead of multipliers, adders and subtractors are utilized the resulting hardware complexity is very less then the previous design with multipliers and thus a larger number of taps can be integrated in to a one single chip.

Fig 3 shows the FPGA resources utilized by CSD. No multipliers are used, number of look up tables and flip-flops increases with the increase in the number of coefficients. This technique utilized highest number adders LUT's and flip-flops.

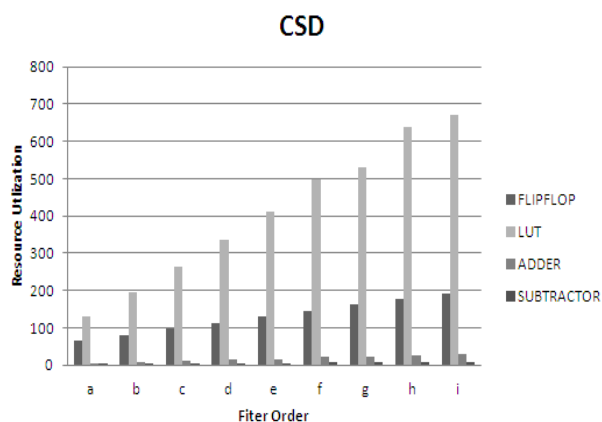


Fig. 3. utilization of resources in FIR filter with CSD

## 2.3 GLOBAL CORRECTION VECTOR

Correction vector enable us to remove the sign extension logic. For sign extension elimination CV for each coefficient is computed and added to form GCV. The resource utilization is shown in Fig 4 [1].

GCV and CSD used almost same number of adders and flip- flops but subtractors are not utilized in GCV. Look up tables, adders and flip-flops increases with the increase in number of coefficient. This design is also not using the multipliers so the hardware complexity is reduced.

H. Ilyas and S. Khan

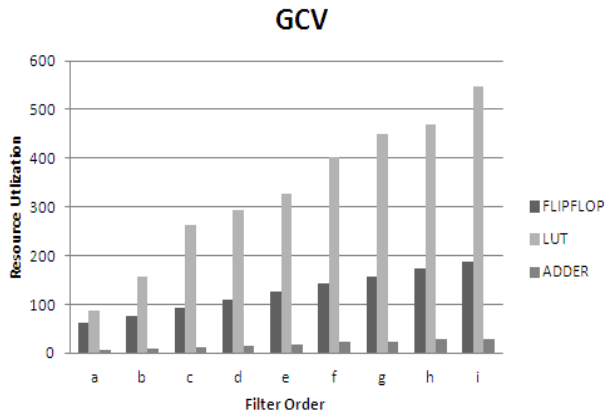


Fig. 4. utilization of resources in FIR filter with GCV

### 2.4 DISTRIBUTED ARITHMETIC

Distributed arithmetic is another multiplier less technique for implementing digital filters. It gained popularity due to their high through put, processing capability which results in cost effective and area-time efficient computing structure. Distributed Arithmetic is a memory based design, the all possible combinations of filter coefficients are pre-computed and stores in the LUT [9] follow by shift-accumulation operation [7]. The memory elements of the LUT increase exponentially and the memory (shift registers) are increases linearly as the filter order grows.

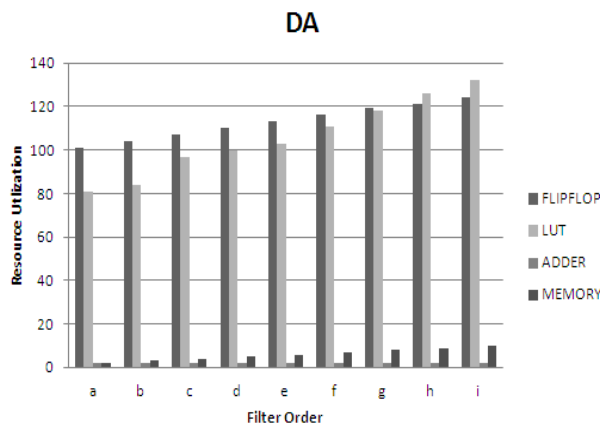


Fig. 5. utilization of resources in FIR filter with DA

DA based design is well suited for FPGA, because the LUT and the shift-add operation can be mapped to the LUT based FPGA logic structure. This technique yield faster output then multiplier based design because the partial results are pre-computed on the paper and stores in LUT. This design is well suited for lower order filters because as the number of coefficients increases LUT size increases, for  $2^{16}$  there are 65536 possible combinations

to avoid this problem the LUT partitioning is used discussed in [10]. Fig 5 shows the resource utilization, this design utilized a constant same number of adders for every coefficient. Numbers of LUT's and flip-flops are increases with the filter order [1]-[7].

### 3. COMPARISONS AND ANALYSIS

In this section resource utilization of FIR filter with respect to different techniques are compared for different filter order. When FIR filter was implemented with multipliers hardware complexity increased, but this technique utilize less number of LUT's and flip-flops. On the other hand multiplier less techniques used more number of LUT's, flip-flops plus other resources like adders, subtractors and memory. In conventional FIR filter technique very less number of look up tables and flip-flops are utilized. Even the number of look up tables design summary shows are route thru, so number used as logic is 0. While the DA technique utilized more number of LUT's and flip-flops. In this technique RAM is utilized so memory is increase with filter order. This technique did not utilize adders. GCV used more number of adders, LUT's and flip-flops then DA based design. Finally the CSD technique utilized the highest number of LUT's, flip-flops. It also utilized the subtractors which are also an adder with 2's complement.

In the following figures this comparison is shown for 5, 6, 7 and 8 number of coefficients. Other number of coefficients follows the same behavior. In the figure it is shown the resource utilization is increases with the filter order and CSD is using most resources among all.

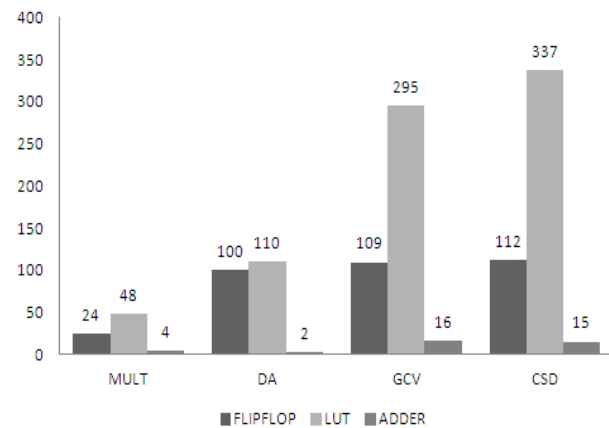


Fig. 6. comparison for filter order 5

H. Ilyas and S. Khan

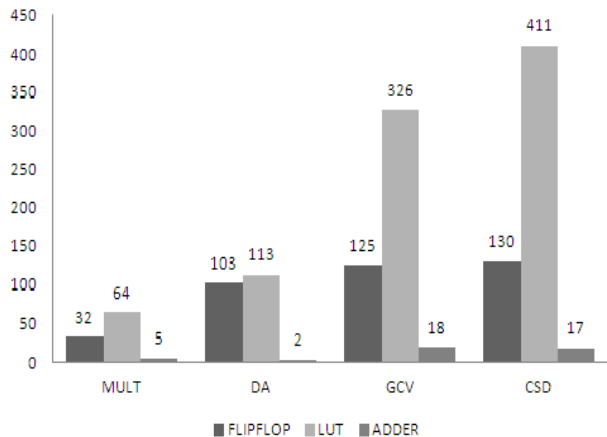


Fig. 7. comparison for filter order 6

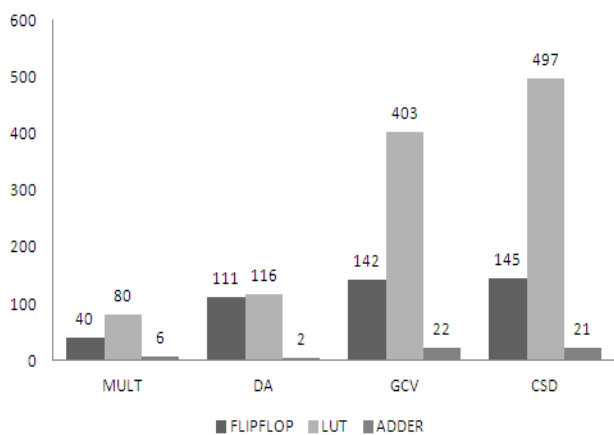


Fig. 8. comparison for filter order 7

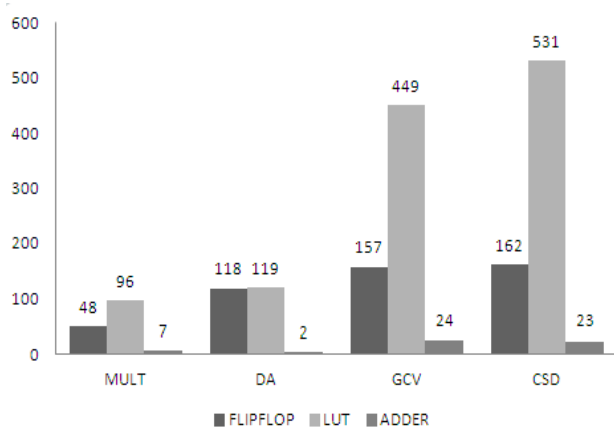


Fig. 9. comparison for filter order 8

### 4. TIME EFFICIENT TECHNIQUE

If the performance metric is time then the time efficient technique is Distributed Arithmetic based Digital filter. In table 4.10 the performance metric is time.

Table 1: Time Efficient Technique for Digital Filters

Technique	Timing
DA	2.498 ns
GCV	4.370 ns
CSD	5.670 ns
MULT	6.278 ns

### 5. RESOURCE IMPLEMENTATION

In this section we have developed a Graphical User Interface which takes the number of resources and coefficients from the user and generates RTL verilog code of the particular algorithm according to the analysis shown in the previous section. When user gives resources and filter coefficients compare it with the section II and III and use that technique which suits the best. In Fig. 10 the example of this Application is shown in which the resources have multipliers, Flip-Flops and adders. The number of coefficients is 4 so after analysis the RTL verilog code is generated for the 1<sup>st</sup> algorithm where the number of coefficients is 4.

H. Ilyas and S. Khan



Fig. 10. Resource Specific Implementation

## 6. CONCLUSION

In this paper we have presented the resource utilization of FIR filter on Virtex 6 FPGA. This paper shows that when FIR filter is implemented with conventional multiplier based design hardware complexity is increased as it takes more area to store the partial results. On the other hand multiplier less technique reduced hardware complexity by not using hardware multipliers but other resources are utilized heavily. The multiplier based design used least number of resources. In this paper the most time-efficient technique is also presented. And at the end we have designed the tool which suggests the best technique according to the available resources.

## REFERENCES

- [1] Shoab Ahmed Khan "Digital design of signal processing Systems" John Wiley & Sons, Ltd, 2011, pp.261-289
- [2] Shoab Ahmed Khan, Sheikh M. Farhan, and Muhammad Sohail Sadiq, "Optimal Time-Shared Design of Digital Signal Processing Architectures", 4th IEEE national multi-topic conference, september2010, pp.1-5
- [3] A. Antoniou, Digital filters: analysis, design and applications New York: McGraw-Hill, 1993
- [4] Suvarna Joshi and Bharati Ainapore, "FPGA based FIR filter", international journal of Engineering Science and Technology, Vol. 2(12), 2010, pp.7320-7323
- [5] Reid M. Hewlitt, and Earl S. Swartzlander, "Canonical Signed Digit Representation for Fir digital Filters," IEEE Workshop on signal Processing Systems, 2000, pp.416-426
- [6] M. Yamda and A. Nishihara, " High Speed FIR digital filter with CSD coefficients implemented on FPGA", in proc. IEEE Design Automation Conference(ASP-DAC 2001), pp.7-8
- [7] Pramod Kumar Meher, Shrutisagar Chandrasekaran, and Abbes Amira, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic", IEEE Transactions On Signal Processing , Vol. 56, No. 7, July 2008, pp.3009-3017
- [8] Vijender Saini, Balwinder singh and Rekha Devi, "Area Optimization of FIR Filters and its Implementation on FPGA", International Journal of Recent Trends in Engineering, Vol. 1, No. 4, May 2008, pp.55-58,
- [9] Narendra Singh Pal, Harjit Pal Singh, R. K. Sarin and Sarabjeet Singh, "Implementation of High Speed Serial and Parallel Distributed Arithmetic Algorithm", international Journal Of computer Applications, Vol. 25, July 2011, pp. 26-32
- [10] Ramesh .R andNathiya .R, "Realization of FIR filter Using Modified Distributed Arithmetic Architecture", Signal & Image Processing : An International Journal (SIPIJ), Vol.3, February 2012, pp. 83-94.