Area Optimization of FIR Filter and its Implementation on FPGA

Vijender Saini[1], Balwinder Singh [2], and Rekha Devi[3]
[1]Dept ECE,Thapar UniversityPatiala,India Email: Vijendra.kumar.saini@gmail.com
[2]VLSI-ES Dept, Centre for development of Advanced computing (CDAC),Mohali,India
Email: balwinder_cdacmohali @yahoo.com
[3] Research Scholar,Punjab University, Chandigarh Email: Er.rekha@yahoo.co.in

Abstract— **Digital Signal Processing is omnipresent in the modern world. Filtering is most important operation of Digital Signal Processing. FIR digital filters are widely used in DSP by the virtue of its stability, linear phase, fewer finite precision error and efficient implementation. First Low pass FIR Filter is designed by choosing CSD algorithms and MATLAB FDA Tool is used Coefficients calculation. The CSD numbers has the minimum number of non-zero digits and no consecutive nonzero digits. Now the multipliers in the digital filters are realized with shifters, adders and subtractors. The use of CSD expression can reduce the number of adders and subtractors. To multiply with CSD number only shift and add operations are required. In this paper multiplication with CSD and binary number is simulated and implemented on the Spartan device, the results shows that the area in the terms of number of slices optimized by 80% in CSD algorithm.**

Index Terms—**CSD, FIR filter, FPGA**

I. INTRODUCTION

Conventional FIR filters consist of cells equal in number to the length of the filter i.e. the number of data taps. Each cell consists of a storage register, a second register and a multiplier. The storage register stores the data tap values, which are digital samples of the signal being processed by the filter. The second register stores the filter coefficients for a particular tap and the multiplier generates the product of the two register contents.

The latter product serves as the output of the cell, and the weighted sum that constitutes the FIR filter output is generated by adding the outputs of all of the cells. [1] .FIR filters are said to be finite because they do not have any feedback. FIR digital filters are widely used in DSP by the virtue of its stability, linear phase, fewer finite precision error and efficient implementation. The FIR Filter consists of Shifters, adders, and multipliers. These constitutes can be chosen on the demand of the designer. An adder is chosen depending upon the performance constraints, different adder implementations are preferred. It depends upon the designer which adder to use, when there is tradeoff between size and speed. Ripple carry adder is used if the designer constraints only on area but if there are requirement of faster adder than Carry look ahead adder is used. As multiplier plays an important role in the hardware complexity of filter on FPGA. Two different multipliers that can be implemented in hardware are combinational multiplier and sequential multiplier. The area of Combinational multiplier increases with increase exponentially with increasing bit widths. Instead, sequential multipliers are often implemented because of substantial saving in the chip area. Though sequential implementations take a finite number of clock cycles in which to perform the operation, unless the design is for real time critical system where speed is essence, a sequential implementation is the better compromise. So for a real time applications such as filtering, combinational multipliers are used because of their high speed. Most of the hardware complexity is due to multipliers, as filters require large number of multiplication, leading to excessive area, delay and power consumption even if implemented in a full-custom integrated circuits. Now the problem faced is that how to reduce the hardware complexity of a combinational multiplier. The main stress is on the reduction of multipliers in FIR filter, the major disadvantages of high order need. The high order demand imposes more hardware requirements, arithmetic operations, area usage, and power consumption when designing and fabricating the filter. Therefore, minimizing or reducing these parameters, is a major goal or target in digital filter design task. It is desired to find efficient algorithms (low-complexity algorithms) that require as few arithmetic operations as possible, as this in the end minimizes the device size and energy consumption, provided that algorithm dedicated ICs (as opposed to general-purpose DSP processors) are used which is the underlying assumption here. A key is to identify and remove redundant computations which leads to more efficient algorithms.The technique chosen is CSD,which is used to optimize the area of Low pass FIR Filter.

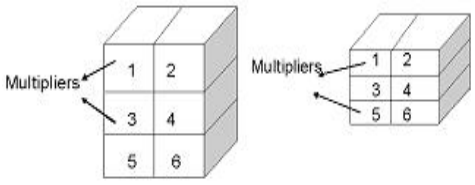

Figure1 (a) : FIR Filter (b) FIR Filter using CSD

Figure 1(a) shows that area occupied by the simple multiplier. The number written on the each box represents the filter coefficient. This type of multiplier used in general-purpose DSP processors. Where the value of filter coefficient can change. In CSD form your filter coefficients are fixed. In CSD form multiplier area get reduced as shown in fig1b. The box represents one multiplier and area of the box in figure 1(b) small as compare to box shown in figure 1(a).

II. CANONICAL SIGN DIGIT ALGORITHM (CSD)

An Encoding a binary number such that it contains the fewest number of non-zero bits is called Canonical Sign Digit .A CSD representation is a kind of sum of signed power of two representations. Unlike binary numbers, that is expressed using only 0 and1, but the CSD representation and the SPT representation use 0, 1 and -1. In CSD, an integer X can be represented by $X=\sum_{i=0}^{n} C_i 2^i$ with $C_i \in \{-, 1,0\}$, where (- denotes -1). It is signed digit number system that minimizes the number of non-zero digits .It can reduce the number of partial product additions in a hardware multiplier. They are successful in implementing multipliers with less complexity. Since the complexity of the multipliers is typically estimated through the number of non-zero elements, which can be reduced by using signed digit numbers. Adjacent CSD Digits are never both non-zero. This property implies that for an n-bit number, there are at most [n/2] non-zero digits. The value [x] denotes the largest integer less then x. On the other hand, for a 2's complement number there can be n non-zero digits for n-bit number. The probability of a digit being zero is roughly 2/3 for CSD and exactly 1/2 for 2's complement. For negative numbers, the numbers of non-zero digits is less for the CSD Representation than the 2's Complement representation. The CSD numbers has the minimum number of non-zero digits and no consecutive nonzero digits .The CSD Representation have fewer nonzero digits than the normal binary expression. Now the multipliers in the digital filters are realized with shifters, adders and subtractors. The use of CSD expression can reduce the number of adders and subtractors For example, the normal binary representation would need 3 adders, as 15 is represented as 1111_2. The number of adders and subtractors is less than the number of non-zero digits by 1.The CSD Multiplier is based on shifts and adds (or subtracts) instead of conventional multipliers. This results in the area reduction of multiplier of the digital filters. The Complexity of a digital filter design is a function of the number of non-zero design in the filter Coefficients.
Encoding the filter Coefficient using the CSD representation reduces the number of partial products as well as the area and the power consumption. Hence, it is useful technique for implementing of FIR Filters with fixed coefficients. Digital FIR filters can take advantage of the CSD representation. [4] [2]

The CSD representation is one of the techniques for implementing FIR filters. Instead, the CSD representation permits the minimization of the total number of non-zero bits in all of the filter coefficients that nevertheless maintains acceptable filter performance.

For each tap in a CSD representation filter, the data tap value is shifted by the number of bit positions corresponding to the position of each non-zero bit in the coefficient for that tap. The resulting shifted data tap values are then added. This is done for every coefficient in the filter. A primary consequence of this is that a larger adder i e. one with more data inputs is required than would be needed for a conventional FIR filter implementation. However, this is still desirable since no multipliers are used. The high hardware cost of multipliers makes the CSD implementation especially attractive for very long digital filters. Such filters are often needed, for example, in communications systems, particularly for the demodulation of digital data signals. [1]

The Coefficient of Low pass FIR Filter is represented in signed and unsigned numbers. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. In DSP filtering applications, coefficients are made up of both positive and negative numbers. Depending on the applications data is either positive or negative. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. The advantages of two's complement are that we can use the same hardware to add negative numbers and positive numbers and the carry out is discarded. So first the Coefficient firstly convert into two's Complement then CSD Algorithm is applied which convert the representation into maximum number of non-zero terms. [2]. There is a flowchart which will convert the two's Complement number into CSD representation shown in Fig.2. A filter represented by a CSD code is called CSD filter. The multiplication can be easily implemented by using CSD code coefficients. The number of adders/subtractors required to realize a CSD multiplier is one less than the number of nonzero digits in the CSD code [3].

Convention systems use coefficient multipliers in digital filter such as Finite impulse response, as well as within other applications. The multiplier may contain set of coefficients. The Coefficients are multiplied with operands that are supplied to multiplier. Each operand may be multiplied by a coefficient. Conventional multipliers are generally implemented with combinations of shift and add operations. Some of such implementations result in inefficient use of area [3]. The Canonical Signed Digit (CSD) representation of numbers, which further reduces the computational load when used with this Horner's method for multiplication. Here, the multiplication can be performed efficiently with just shift and add operations. The operand is denoted by X, and the

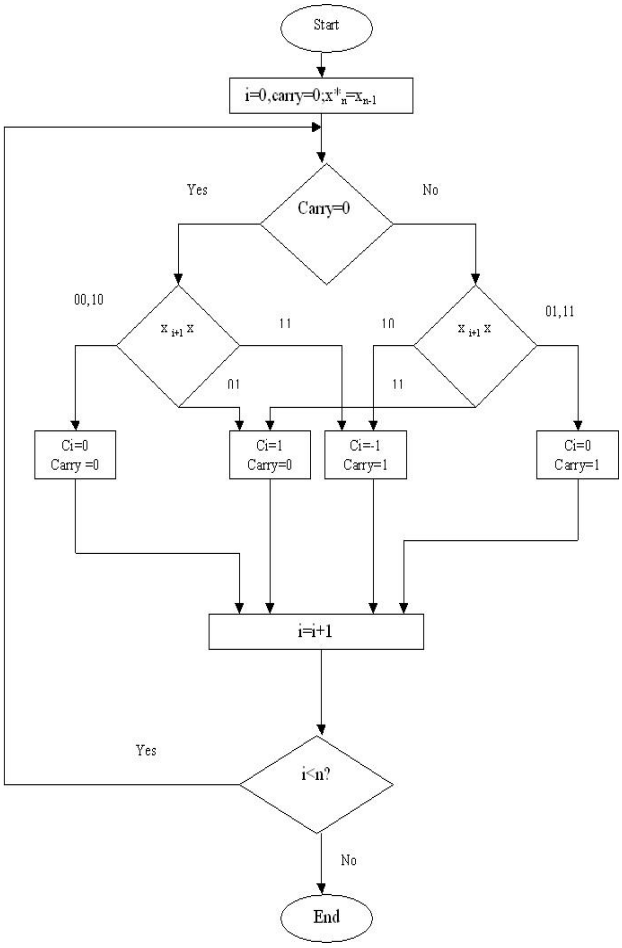

Figure 2: Two's complement to CSD Form

II. CSD MULTIPLIERS

multiplier by M. The efficiency of Horner's method can be further improved by using CSD format to represent the multiplier or divisor. The CSD format aims to reduce the number of add operations during multiplication and division. The CSD format has a ternary set as opposed to a binary set in number representation. The symbols used in this format are {0, 1, 1}, with 1 representing –1.

III. METHODOLOGY USED:

One of the key objectives is to learn how to implement actual design problem in hardware. The following steps will depict the procedure exactly as done by the author by taking a relevant example:

1. Define the filter specifications such as order, window function, cut-off frequency and sampling frequency.
2. Calculate the filter coefficient using MATLAB FDA Tool.
3. Apply CSD Algorithm on filter Coefficients.
4. Configured the target FPGA Spartan 3E kit for real time debugging.
5. Checked the CSD Filter coefficients on target FPGA Chip.
6. Configured the target FPGA Spartan 3E kit for real time debugging.
7. Checked the Multiplication for CSD number 1000000-. on target FPGA kit.
8. Checked the Multiplication for CSD number 01111111.on target FPGA kit.
9. Compare the selected device for both 01111111 and 1000000-.

In this paper the specifications of low pass FIR Filter having filter order 31,Kaiser window used with β=1, cut-off frequency 4kHz, sampling frequency 12kHz. The frequency response of the filter is shown in figure 3.

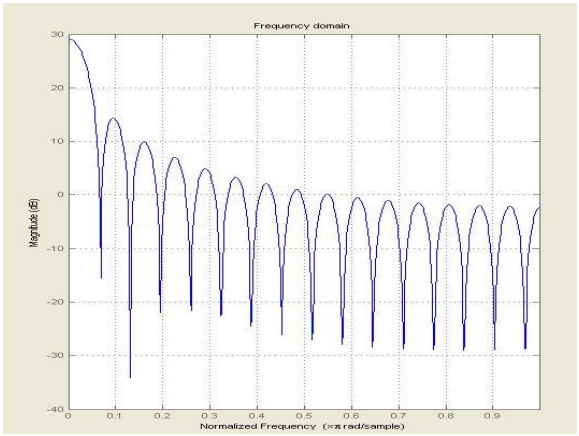

Figure 3: Low Pass Filter Response

In this it is observed that leakage factor is 6.32 %, relative side lobe attenuation is –14.7 dB and the Main lobe width is (-3dB) 0.054688.For implementation and the Horner's method is used for multiplication of CSD number with any integer number.

Unsigned Integer Multiplication With CSD

Consider the example of an integer-integer multiplication using Horner's method using the CSD format, with the operand X being 41 and multiplier 441.

X = 41 = 0101001b

M = 441 = 0110111001b = 100-00-001CSD

The design equations are

$X_1 = X * 2^3 – X = 00101001000b – 00000101001b$

$X_1 \rightarrow \overline{00100011111b}$

$X_2 = X_1 * 2^3 – X = 100011111000b – 000000101001b$

$X_2 \rightarrow \overline{100011001111b}$

$X_3 = X_2 * 2^3 + X =$
$100011001111000b + 000000000101001b$

$X_3 \rightarrow \overline{100011010100001b}$

Final result = X_3 = (100011010100001) b = 18081.

The absolute error obtained for this multiplication is zero. The design equations remain the same if the operand X were a negative number. [5]

IV. SIMULATION RESULTS AND CONCLUSIONS

The Canonical sign digit algorithm, the area in the terms of number of slices optimized by 80%as shown in table 1. CSD representation for FIR filters can reduce the complexity of the hardware implementation by a significant amount for the fixed Coefficients. There will be further reductions in the number of non-zero CSD digits of the filter coefficients reduce the multiplication process to a few additions or subtractions of partial products.

TABLE I. COMPARISON OF BINARY AND CSD MULTIPLIER FOR THE 8-BIT NUMBER

Components	Comparison of Area overhead		
	Total Available	Binary multiplier for 8 bit number	CSD Multiplier for 8 bit number
Slices (CLB)	4656	39	13
4 input LUTS	9312	71	26
IOBS	66	17	18
Bonded IOs	232	17	18

Both the multiplication algorithms are simulated in Model Sim Starter 6 & implemented on FPGA Spartan 3E Kit. The components like CLBs used in selected device(3s500efg320-4) are compared and which shows that area is optimized by 80%..The comparison of these algorithms on the basis of hardware is shown in Figure 4.

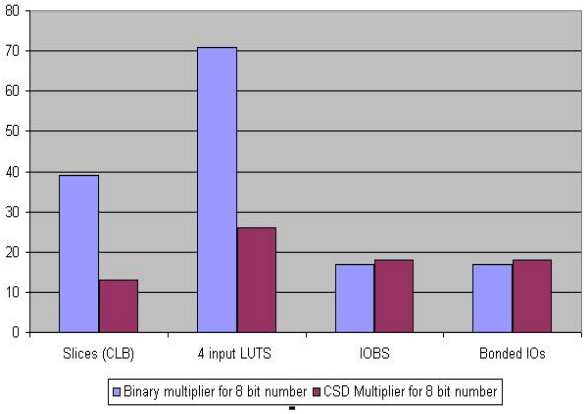

Figure 4 Binary and CSD Multiplier for 8 bit Number

REFERENCES

[1] Karl Wittig, "Reconfigurable FIR Filter using CSD Coefficients", United State, Patent No. US 7,164,712, B2, Date of Patent: Jan. 16, 2007.

[2] Reid M. Hewlitt, "Canonical Signed Digit Representation for Fir Digital Filters", IEEE Workshop on Signal Processing Systems, 2000,pp.416-426.

[3] Murko Kosunen, Kuri Halonen,"Multiplier and shift using signed digit representations" Patent No. US 7,257,609 B1, Date of Patent: Aug 14, 2007

[4] Joseph B. Evans," Efficient FIR Filter Architectures Suitable for FPGA Implementation "VOL.41, No. 7,July1994.

[5] Kripasagar Venkat "Efficient Multiplication and Division Using MSP430" SLAA329–Sep 2006, Texas Instrument, pp .1-14.