ASIC Implementation Of Low Power FIR Filter

Prof. S. J. Dashavant
Dept of Electronics and Telecommunication
BMIT, Solapur

Prof. A. I. Merchant
Dept of Electronics and Telecommunication
BMIT, Solapur

Abstract

The project intended with the ASIC design of Low power FIR Filter. Even though designing a FIR Filter is a traditional trend, achieving a low power FIR Filter using enhanced low power technique is most concerned area of the filter design. It is mandatory for any filter designer to propose a low power multiplier as most of the power consumption of the filter occurs in multiplier unit. Hence, in this paper novel Wallace tree multiplier has been proposed. The proposed multiplier consumes 43% lesser power than the conventional multiplier architecture. The power consumed by the adder structure is also very significant while designing a low power filter. It is found that for a 16 bit input the ripple carry adder consumes more power than carry lookahead adder.

*With proposed multiplier unit and carry lookahead adder, the designed FIR Filter (8*8) consumes 55% lesser power than the conventional filter without significant increase in area. The power area comparison is performed for different orders of the filter and input bit lengths. It is noted that for any order, the power consumed by the proposed filter reduces irrespective of the input length.*

Keywords: *ASIC Implementation, FIR filte , Wallace tree multiplication, FDA tool.*

1. Introduction

In recent times, designing low power filter has emerged as one of the valuable requirements in DSP applications. In this view, a low power Finite Impulse Response (FIR) filter has been designed considering the various aspects such as power consumption in multiplier unit and adder unit. The low power filter is designed modifying the filter equation and employing Decorrelation (DÉCOR) algorithm to feed the input to the multiplier units. Thus, a considerable amount of power has been reduced to achieve a low power FIR filter.

This paper describes the implementation technique of the decorrelating transformation for low power FIR filters. The technique was introduced in the past, but was not fully evaluated for its area and power performance. Early evaluations did not consider the whole implementation and were merely based on either some analytical methods or high level simulation models. This paper presents the complete VLSI implementation of the FIR filter and a study of its area and power performance with different order of the filter and various input bit lengths. Since most of the power consumes in multiplier unit, a novel, modified Wallace tree multiplier is proposed. **.**

2. Multiplier Design

A modified 8x8 multiplier architecture based on Wallace Tree, efficient in terms of power and regularity without significant increase in delay and area has been proposed. Here, the the partial products are generated using AND gates. The modified Wallace tree is used for the addition of these partial products. The parallelism in generating the partial product is realized by ANDing the first bit (LSB) of the multiplier with the multiplicand bits. The second partial product is generated by ANDing the second multiplier bit with the multiplicand bits proceeded by a single zero. The third partial product is obtained by ANDing the third multiplier bit with the multiplicand bits preceded by double zeros, and so on.

Figure 1.1 shows the hierarchical decomposition of a 8x8 proposed Wallace tree logic. For (8X8) bits, 8 partial products are generated, and are added in parallel as shown in Figure 1.1-stage A. The 8 partial products thus generated are divided into two parts, where each part contains four adjacent rows of partial products. The four adjacent partial products in the same parts are subdivided to two columns each of 4 bits as shown by dotted lines. Therefore, we have four parallel blocks, two from each adjacent four rows, working in parallel. The addition operation in the columns of each block can be performed by appropriately choosing the half adders, full adders, 4:2 compressors and 5:2 compressors according to the number of bits to be added. This represents the first level (Stage A) of computation. The partial sums thus generated are again appropriately divided and added again in the same manner, forming the second level (stage B) of computation. The partial sums generated in the second level are added by using a carrylook ahead adder in the third level (stage C) to arrive at the final product.

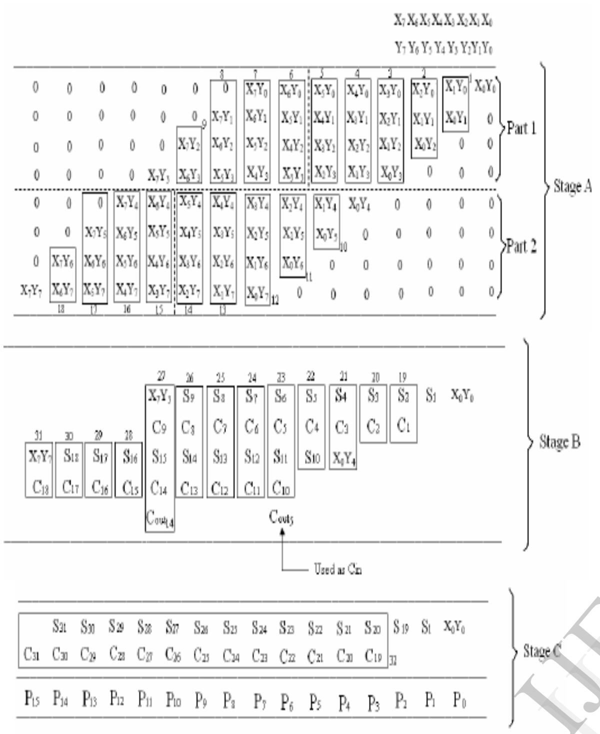


Figure 1.1 Proposed Modified tree logic having hierarchical decomposition

Significant amount of power can be reduced by applying low power concepts to the conventional design. Figure 1.2 shows the block diagram of the proposed modified Wallace tree multiplier. The gray boxes refers to the places where the proposed architecture differs from the conventional Wallace tree logic.

The Modelsim simulated waveforms and synthesis report of 8 bit proposed multiplier is shown 1.3

A comparison of power consumption in proposed and conventional method in each 8bit and 16 bit multiplier is given in table1.1. It is evident from the above table that a significant power is reduced in a proposed design. With a low power technique in 8x8 bit multiplier, 42.5% power can be saved in a proposed design as compared to conventional design without significant increase in area.

It is important to note that in 16x16 bit multiplier, the area occupied by proposed design reduces with considerable reduction in the power consumption. Mathematically, the power and area in a proposed design reduces by 9.8% and 2.5% respectively.

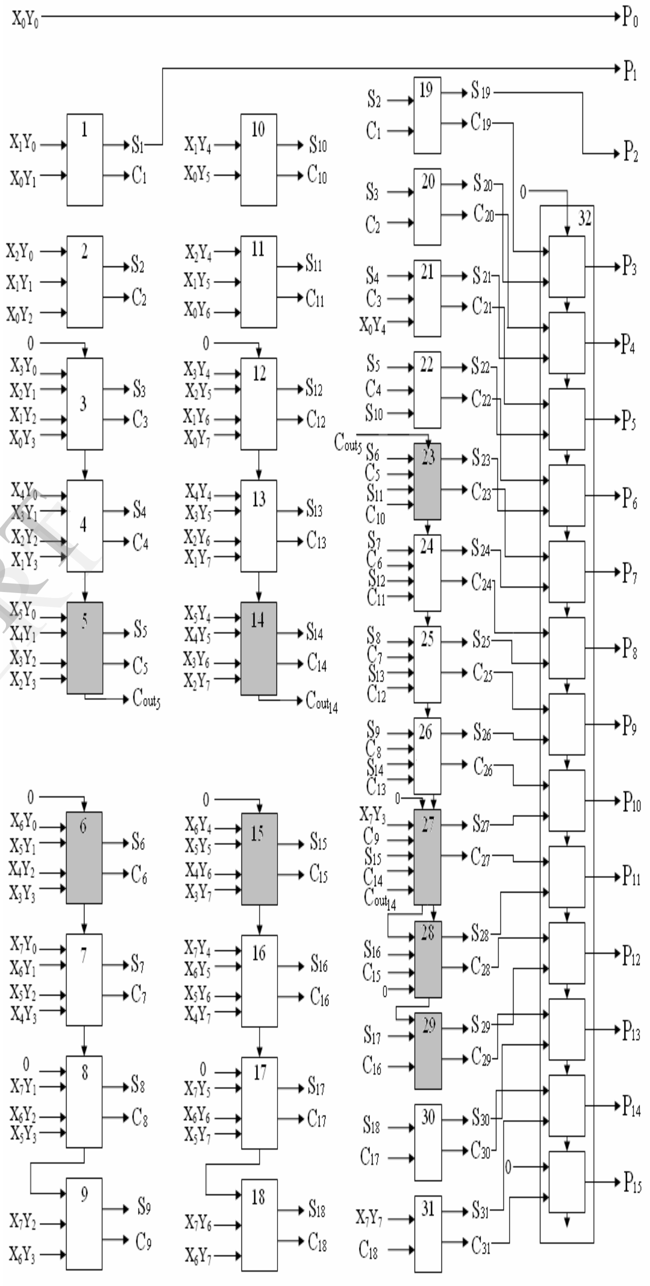
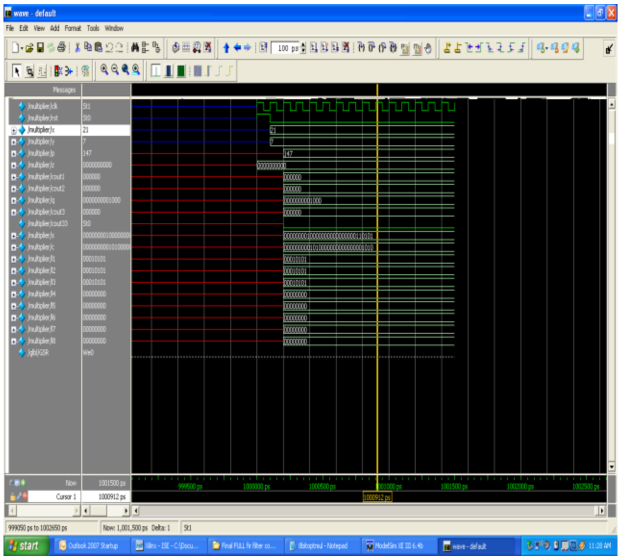


Figure 1.2 Block Diagram of the Proposed Modified Wallace Tree Multiplier



```
/multiplier/clk     St1
/multiplier/rst     St0
/multiplier/x       21
/multiplier/y       7
/multiplier/p       147
/multiplier/z       0000000000
/multiplier/cout1   000000
/multiplier/cout2   000000
/multiplier/q       0000000001000
/multiplier/cout3   000000
/multiplier/cout33  St0
/multiplier/s       000000000100000000
/multiplier/c       000000000010000000
/multiplier/l1      00010101
/multiplier/l2      00010101
/multiplier/l3      00010101
/multiplier/l4      00000000
/multiplier/l5      00000000
/multiplier/l6      00000000
/multiplier/l7      00000000
/multiplier/l8      00000000
/glbl/GSR           We0
```

Figure 1.3 Modelsim simulated waveform for 8bit proposed multiplier

Table 1.1 Power and Area comparison of 8x8 and 16x16 bit multipliers

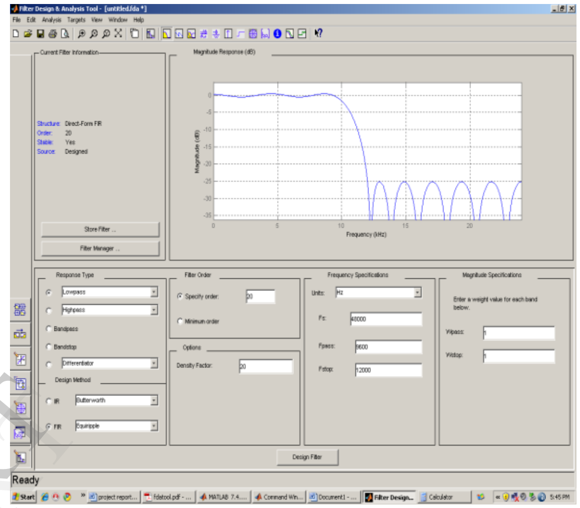
MULTIPLIER	TYPE	POWER (mW)	AREA
8x8	Conventional	0.1172	2063.17
	Proposed	0.0674	2240.99
16x16	Conventional	0.5153	8446.74
	Proposed	0.4680	8237.17

3. Filter coefficient generation using FDA tool

In the design of FIR filter, the number of coefficients generated depends on the order of the filter chosen. If the order of the filter is N, then there will be N+1 coefficient terms in the filter. These coefficient terms represent the impulse response of the filter .In this project, the filter coefficient are determined by a matlab tool known as Filter Design and Analysis (FDA) tool. The upcoming section describes the procedure for generating the filter coefficient using FDA tool in detail.

The Filter Design and Analysis Tool (FDATool) is a powerful user interface for designing and analyzing filters quickly. FDA Tool enables you to design digital FIR or IIR filters by setting filter specifications, by importing filters from your MATLAB workspace, or by adding, moving or deleting poles and zeros. FDA Tool also provides tools for analyzing filters, such as magnitude and phase response and pole-zero plots.

The frequency response of 20 order, FIR equiripple filter with pass band and stop band frequencies of 9600Hz and 12000Hz at a sampling rate 48kHz is shown in figure 1.4.



Structure: Direct-Form FIR
Order: 20
Stable: Yes
Source: Designed

Response Type: ● Lowpass ○ Highpass ○ Bandpass ○ Bandstop ○ Differentiator

Design Method: ○ IIR Butterworth ● FIR Equiripple

Filter Order: ● Specify order: 20 ○ Minimum order

Options: Density Factor: 20

Units: Hz
Fs: 48000
Fpass: 9600
Fstop: 12000

Enter a weight value for each band below.
Wpass: 1
Wstop: 1

Figure 1.4 frequency response of a 20 order FIR filter

4. Mathematical Concepts

An N-tap FIR filter performs the following convolution:

$$Y_j = \sum_{k=0}^{N-1} C_k X_{j-k} \qquad \ldots\ldots\ldots\ldots(1.1)$$

where Ck's are the coefficients of the filter, Xj and Yj are the jth terms of the input and output sequences, respectively.

The z-transfer of (1.1) is given below:

$$Y(z) = H(z)X(z) \qquad \ldots\ldots\ldots..(1.2)$$

where Y(z), H(z), and X(z) are the z-transforms of the output, filter, and input respectively. In decorrelating technique, the transfer function H(z) is multiplied and divided by the polynomial

$$T(z) = (1 + \alpha z^{-\beta})^m \qquad \ldots\ldots\ldots\ldots\ (1.3)$$

where m denotes the order of coefficient difference, α and β are parameters whose value is to be chosen depending on the type of FIR filter. The frequency response is not altered by multiplying and dividing the transfer function H(z) by this polynomial. For example, the z-transfer of the first order lowpass FIR filter is given by (α = -1, β = 1, m = 1) the equation as follows:

$$Y(z) = \frac{[(1-z^{-1})H(z)]}{1-z^{-1}}\, X(z) \qquad \text{………(1.4)}$$

$$Y(z) - Y(z)z^{-1} = H(z)X(z) - H(z)X(z)z^{-1} \qquad \text{…….(1.5)}$$

According to Eq.(1.1) and Eq.(1.5) the transformed filter can be expressed as:

$$Y_j - Y_{j-1} = \sum_{k=0}^{N-1} C_k X_{j-k} - \sum_{k=0}^{N-1} C_k X_{j-k-1} \qquad \text{…….(1.6)}$$

Re-arranging the Eq (1.6) we can obtain the following equation for first order (m=1) differential coefficients:

$$Y_j = C_0 X_j + \sum_{k=1}^{N-1} (C_k - C_{k-1}) X_{j-k} - C_{N-1} X_{j-N} + Y_{j-1} \qquad \text{……(1.7)}$$

The above equation (1.7) clearly shows, for first order differential coefficients, the filter outputs can be obtained using the differences between adjacent coefficients (except for first and last coefficients) and the previous filter output. It is also observed that the transformed filter requires one additional multiplication and subtraction operation in order to realize the term $(-C_{N-1}X_{j-N})$ in Eq (1.7).

5. Structural Implementation of DÉCOR FIR filter

The structural implementation of DÉCOR filter of Nth order as per Eq (1.7) is shown in fig 1.5.

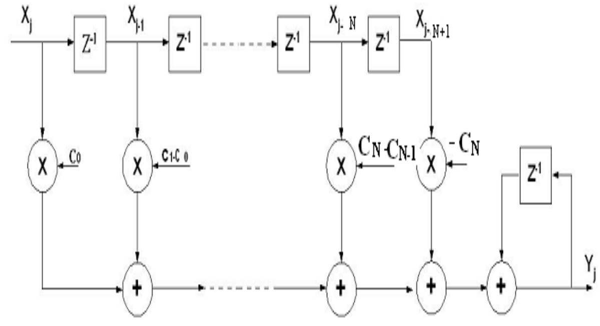


Figure 1.5 DECOR FIR filter structure using first order differential coefficients

6. Block diagram of DÉCOR FIR filter

The block diagram of the DECOR FIR filter is shown in 1.2. There are eight blocks in this DECOR FIR filter core implementation, as shown in Fig 1.6. It contains two memory blocks for storing the input data (X_RAM) and the coefficients (COEFF_ROM), two registers for storing the input data (INPUT_MEM) and coefficient (CODIFF_MEM), the control block (CONTROL), the output register (OUT_STORE) for holding the output data, DÉCOR BLOCK and the main arithmetic block (MAC).

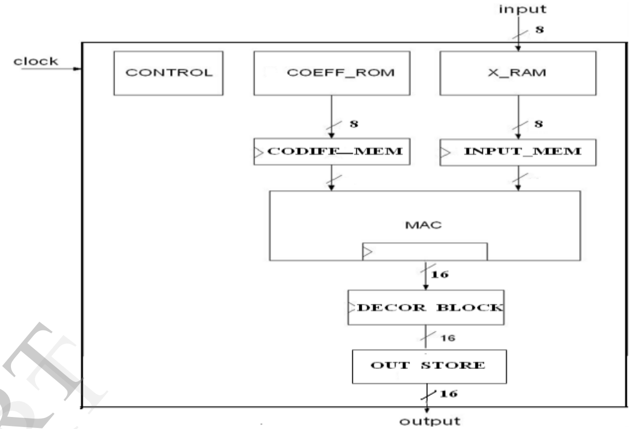


Figure 1.6 Block diagram of DÉCOR FIR filter

A brief description of these blocks is given below:

- CONTROL: The controller is responsible for generating every control signal to synchronize the activity of each block in the filter.
- X_RAM: This is the RAM memory used for the storage of the input data.
- COEFF_ROM: This is the ROM memory used for the storage of the coefficients of the filter.
- INPUT_MEM: This is a 8-bit register implemented in the form of a flip-flop based circuit for synchronization with clock.
- CODIFF_MEM: This is a 8-bit register implemented in the form of a flip-flop based circuit for synchronization with clock. It is responsible for storing the difference between adjacent coefficient values.
- MAC: It consists of a delay register, multiplier, an adder, an accumulator. The block diagram of MAC is shown the figure 1.7. The MAC is used to multiply an input data with a coefficient and adding the previous stored accumulator register value to the product at the same clock period.

- DÉCOR BLOCK: The DÉCOR BLOCK is used for implementing the additional terms in the DECOR FIR filter equations 1.7. The DÉCOR BLOCK consists of a number of registers for storing the previous filter outputs (Yj-1, Yj-2, etc.) and adder/subtractor units for adding a combination of previous filter outputs ([2Yj-1-Yj-2], [3Yj-1-3Yj-2+Yj-3], etc.) to the current filter output, depending on the value of m (order of coefficient difference) used.
- OUT_STORE: This is a 16-bit register implemented in the form of a flip-flop based circuit for synchronization with the clock.

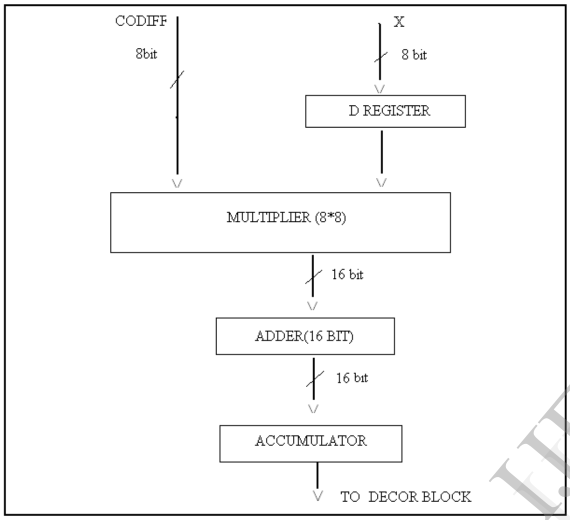


Figure 1.7 Block diagram of MAC unit

7. Simulated Waveforms

The 20 order, 8 bit fir filter has been simulated and waveform is shown in figure 1.4.

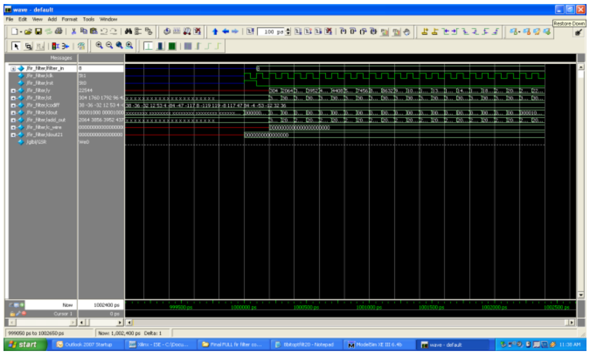


Figure 1.8 Modelsim simulated waveform for 8bit, 20 order FIR filter

8. Power-area comparison

As a summary, the power consumed for various filter order, for different input bit length and area requirement is tabulated in table 1.2. It is evident from the table 1.2 that, the proposed design consumes lesser power in contrast with the conventional filter design.

Table 1.2 power-area requirements for different order filter and different input bit sequence

Filter order	Filter type	i/p bit	Power (mW)	Area
20	Conventional	8x8	2.0876	27452.07
	Proposed		0.9487	29942.84
20	Conventional	16x16	6.7610	113255.86
	Proposed		6.3998	113077.34
35	Conventional	16x16	13.4119	191911.91
	Proposed		12.5849	191616.97

9. Conclusion and future work

The DÉCOR FIR filter has been successfully simulated using Matlab and Cadence tool. The simulated waveforms for different order of the filter are observed using Modelsim tool. The result obtained for different order of the filter as well as for different input bit length are tabulated for the purpose of power area comparison. It has been easily concluded from the tabulation that proposed filter design has consumed the lesser power than the conventional design in all the cases, thereby verifying phrase 'LOW POWER' to the project title. The great care has been taken to simulate the entire FIR design as close the practical systems.

The future works to the project are the detection of continuous input bit stream and reconstruction of original bit stream from the filtered output. The various architecture can be proposed for designing the FIR filter so as to consume low power.

REFERENCES

[1] A text book titled **"FPGA-based Implementation of Signal Processing Systems"** by Roger Woods (Queen's University, Belfast, UK), John McAllister (Queen's University, Belfast, UK), Gaye Lightbodym (University of Ulster, UK),Ying Yi (University of Edinburgh, UK).

[2] A IEEE paper on **"Implementation of the decorrelating transformation for Low power fir filters"**, by A.T. Erdogan , T. Arslan , and R. Lai (School of Engineering and

Electronics, The University of Edinburgh The King's Buildings, Edinburgh EH9 3JL, United Kingdom).

[3] A IEEE paper on **"Low power fir filter realization with Differential coefficients and input"**, by Tian-Sheuan Chang and Chein- Wei Jen (Dept. of Electronics Engineering, National Chiao-Tung University, Taiwan.

[4] A text book titled "VERILOG HDL" by Samir Palnitkar.

[5] A text book titled **"Digital Signal Processing: Principles, Algorithms, and Application"** by J.G.Proakis and D.G. Manolakis, MacMillan Publishing,1992.

[6] A text book titled " A Verilog HDL Primer", by J. Bhasker.

[7] A text book titled "Practical Low power VLSI Design", by Gary Yeap, Kulwer Academic Publishers, 1998.