Design of Linear Phase FIR Filter for Minimum Number of Adders by Using MILP

M.Pratheba and P.SatheesKumar

*Abstract:***Linear phase Finite Impulse Response (FIR) filter are widely used in digital signal applications such as speech coding, image processing, multi rate systems. Although the stability and linear phase is guaranteed, the complexity and power consumption of linear phase FIR filter are usually much higher than that of the Infinite Impulse Response (IIR) filter which meets the same magnitude response specification. Therefore many efforts have been dedicated to the design of low complexity and low power linear phase FIR filter.The proposed work is the design of low complexity linear phase FIR filter with optimum discrete co efficient. The proposed algorithm is based on Mixed Integer Linear Programming (MILP), efficiently traverses the discrete co efficient solutions and searches for the optimum one that results in an implementation using minimum number of adders. During the searching process, discrete co efficient are dynamically synthesized based on a continuously updated sub expression space and most essentially, a monitoring mechanism is introduced to enable the algorithms awareness of optimality.**

*Keywords:***Linear phase FIR filter,MILP,Optimum discrete coefficient,CSD**

1.INTRODUCTION

Digital Signal Processing (DSP) techniques have been increasingly applied in most engineering and science fields due to the explosive development in digital computer technology and software development. Digital filters are basic building blocks for DSP systems. There are two types of filters Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. Since FIR filters possess many desirable features such as exact linear phase property, guaranteed stability, free of limit cycle oscillations, and low coefficient sensitivity, they are preferred in most of the wireless communication systems and biomedical applications. However, the order of an FIR filter is generally higher than that of a corresponding IIR filter meeting the same magnitude response specifications.

MOTIVATION OF THE WORK

FIR filters require considerably more arithmetic operations and hardware components - delay, adder and multiplier. This makes the implementation of FIR filters, especially in applications demanding narrow transition bands, very costly when implemented in VLSI (Very Large Scale Integration) Technology the coefficient multiplier is the most complex and the slowest component. The large number of arithmetic operations in the implementation also increases the power consumption. In the modern applications, such as military devices, wearable devices and portable mobile communication devices, the portability and low power dissipation play a very important role.

M.Pratheba is a Assistant Professor Department of Electrical & Electronics Engineering ,SNS College Of Engineering, Coimbatore Tamilnadu, India and P.SatheesKumar PG Scholar Department Of Electrical and Electronics Engineering,SNS College Of Engineering, Coimbatore-641107, Tamilnadu, India, prathekrishna@gmail.com, sathees61@gmail.com

To address the problem, considerable attention and efforts have been made on reducing the complexities and power consumptions for the DSP systems. The cost of implementation of an FIR filter can be reduced by decreasing the complexity of the coefficients and the complexity reduction includes reducing the coefficient word length and representing coefficients in effective form. One of the most efficient ways is to design filters with coefficients restricted to the sum or difference of signed powers-of-two values. This leads to a so-called multiplication-free implementation, i.e. the filter's coefficient multipliers can be replaced by simple shift-and add circuits. Thus, the implementation complexity can be reduced, resulting in significant increase in the speed and reduction in power dissipation.

PROPOSED WORK

- To design a Finite Impulse Response (FIR) filter with optimum discrete coefficients.
- To design a filter structures with Multiplier less technique using Mixed Integer Linear Programming.
- To extend the above structure to reduce the number of adders and power complexity.

LAYOUT OF FILTER DESIGN

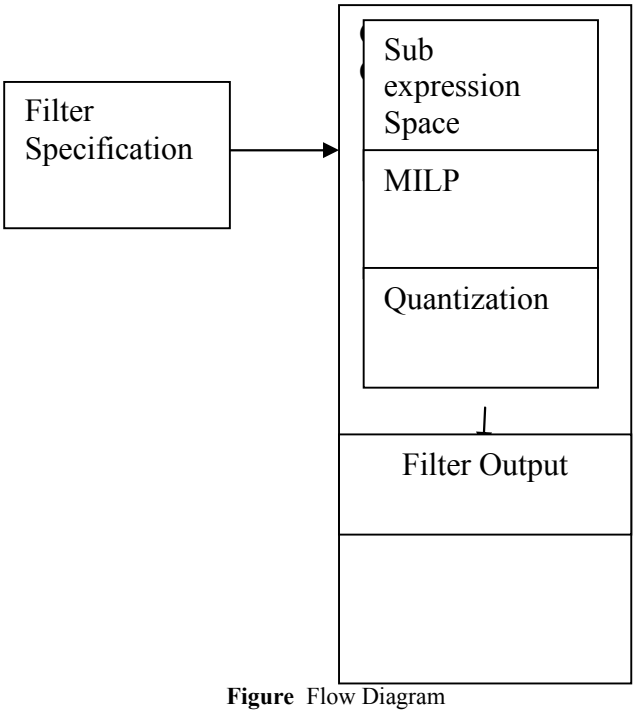


Figure Flow Diagram

COEFFICIENT OPTIMIZATION

FILTER SPECIFICATION

The basic step in the filter design is the filter specification. The filter specifications are

1) The pass band frequency
2) Stop band frequency
3) Pass band ripple

4) Stop band ripple

From the above specifications the filter co efficient are generated.

7.2 COEFFICIENT OPTIMIZATION

The co efficient optimization process consist of three steps .

- Sub expression space.
- Traverse of discrete solution using MILP.
- Detailed quantization process.

Sub Expression Space

Finite-Impulse Response (FIR) filters play a vital role in modern communication system because of its versatility, stability, and simplicity. As multipliers are generally agreed to be a power-hungry device and occupy a large silicon area, the trend toward the design of fixed-point FIR filters is to replace the expensive multiplication operations by simpler additions and hardwired shifters. The basic principle behind the design of FIR filter for multiplier less implementation is to approximate each filter coefficient with a minimal number of signed-power-of-two (SPT) terms.

A common way of implementing constant multiplication is by a series of shift and adds operations. If the multiplier is represented in Canonical Signed Digit (CSD) form, then the number of additions (or subtractions) used wil1 be minimum. In this method, the co efficient optimization is done through the design of CSD multipliers, and in particular the gains that can be made by sharing sub expressions. In the case where several multipliers are present in a network of operators, for instance in an FIR filter, the savings achieved by identifying common sub expressions can be as much as 50% of the total number of operators.

The asymptotic frequency of the most common sub expression is analyzed mathematically, and it is shown that sharing the two most common sub expressions can be expected to lead to a 33% saving of the number of additions.

Signed Power of Two Term

One of the most successful strategies is to optimize the filter coefficients in signed power-of-two (SPT) space, where each coefficient is represented as a sum of a limited number of SPT terms. In this method, the filter complexity is determined by the number of additions/subtractions required to implement the multiplications, which, in turn, is directly related to the number of SPT terms used to synthesize the filter coefficients. Thus, the constrained optimization problem becomes one of finding a set of filter coefficients with a minimal number of SPT terms that satisfy a given magnitude response specification. A minimum representation refers to a representation of a numeral that has the minimum number of SPT terms. Thus the coefficient multiplications can be replaced by shifters and adders, so that the implementation of the filter is essentially multiplier less.

In mathematics, a power of two means a number of the form 2^n where n is an integer, the result of exponentiation with as base the number two and as exponent the integer n. In a context where only integers are considered, n is restricted to non-negative value, so we have 1, 2, and 2 multiplied by itself a certain number of times.

Because two is the base of the binary numeral system, powers of two are common in computer science. Written in binary, a power of two always has the form 100…0 or 0.00…01, just like a power of ten in the decimal system.

The two's complement of a binary number is defined as the value obtained by subtracting the number from a large power of two (specifically, from 2^N for an N-bit two's complement). The two's complement of the number then behaves like the negative of the original number in most arithmetic, and it can coexist with positive numbers in a natural way.A two's-complement system or two's-complement arithmetic is a system in which negative numbers are represented by the two's complement of the absolute value; this system is the most common method of representing signed integers on computers. In such a system, a number is negated (converted from positive to negative or vice versa) by computing its two's complement. An N-bit two's-complement numeral system can represent every integer in the range -2^{N-1} to $+2^{N-1}-1$.The two's-complement system has the advantage of not requiring that the addition and subtraction circuitry examine the signs of the operands to determine whether to add or subtract. This property makes the system both simpler to implement and capable of easily handling higher precision arithmetic. Also, zero has only a single representation, obviating the subtleties associated with negative zero, which exists in ones'-complement systems.

SPT number characteristics and existing optimization techniques for the design of digital filters subject to SPT coefficients is shown below A number, S, is called an SPT number, if it is represented to a precision 2^Q by R - Q ternary digits s (i) according to

$$S = \sum s(i)2^i,\ s(i) \in \{-1,0,1\},\ Q \le i \le R-1, \text{ where } i=Q \text{ to } R-1 \quad (7.1)$$

Where R and Q are integers. Each nonzero digit term, s (i) ≠ 0, is counted as a SPOT term. The word length of S is (R-Q) bits. S is discrete values in increments of 2^Q in the range, in which there are $2^{R-Q+1}-1$ distinct values.

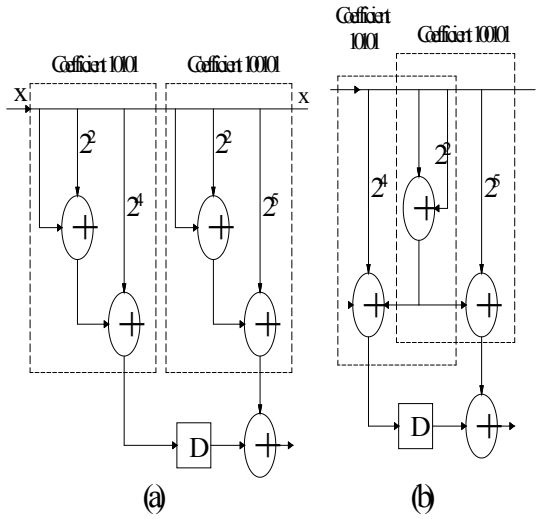
Similarly, a discrete sub expressions space can be constructed by defining its element as,

$$n = \sum y(i)2^{q(i)},\quad y(i) \in S, \text{ where } i=0 \text{ to } k-1 \quad (7.2)$$

Where S is a set of permissible sub expression bases. $y(i)2^{q(i)}$, is a shifted version of a sub expression basis, named a sub expression term, and K is defined as the number of sub expression terms. Usually, a Sub expression basis refers to a trinary string with more than 1 nonzero digit, such as 1011 or 1001.

Canonic Signed Digit Representation

The Canonical Signed Digit (CSD) representation is one of the most commonly used minimum representations in digital filter coefficient synthesis. There exist a number of algorithms for synthesizing CSD coefficients to minimize the number of SPT terms that are required for the implementation of a low-complexity FIR filter.An improved algorithm for the optimization of FIR filter with SPT coefficient value is proposed, which allocates an additional nonzero digit in the Canonic Signed-Digit (CSD) code to the larger coefficients to compensate for the non-uniform nature of CSD coefficient distribution. The two-stage algorithm consists of search for an optimum scale factor and a bivariate local search in the neighborhood of the scaled and rounded CSD coefficients. It is illustrated that a significant improvement in the frequency response can be obtained at the price of minimal increase in filter complexity resulting from the additional CSD Digits.Instead of the entire integer space, the algorithm searches for the feasible solutions within the specified sub expression space. The reduced searching space thus results in significant time saving compared with the algorithm B&B. Instead of being predefined, the sub expression basis set is dynamically expanded and updated during the optimization procedure. Hence, more useful basis sets are generated automatically and better results are obtained.



Implementation of two coefficient multiplication.

(a) Each coefficient is independently implemented by shifters and adders.

(b) Common sub expression of 101 is extracted and implemented to be shared between two coefficients.

The number of adders to implement the FIR filter can be further reduced by extracting common sub expressions from the coefficients. The adders used to realize the common sub expressions can be shared within a coefficient as well as among coefficients. For example, when two coefficients with binary numbers 100101 and 10101 are implemented independently, two adders are required for each coefficient, as shown in Fig. 7.1(a). Thus, in total four adders are used for the coefficients. If the common sub expression of 101 is extracted and implemented first, only an additional adder is required for each coefficient, ending up with three adders in total, as shown in Fig. 7.1(b). Even in such a simple example, an adder is saved.

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. 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/sub tractors required to realize a CSD multiplier is one less than the number of nonzero digits in the CSD code .

The canonical signed digit (CSD) representation is one of the existing signed digit representations. In addition the CSD representation always results in a minimal and unique representation. With the use of CSD the number of nonzero bits is at most N/2, rounded to the nearest integer, compared with N bits for the two's complement representation. In a digital filter with a fixed filter specification all multipliers may have constant coefficients. A constant coefficient in CSD requires only the number of partial products determined by the number of nonzero bits in the coefficient. The asymptotic frequency of the most common sub expression is analyzed mathematically, and it is shown that sharing the two most common sub expressions can be expected to lead to a 33% saving of the number of additions.

The canonical signed digit code (CSD) is a signed digit representation with minimal Hamming weight, i.e., it has a minimum number of ones, and contain no adjacent nonzero digits .The conversion of a two's complement number into CSD code is done according to Table

b_{i+1}	b_i	c_i	a_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	1

Table Two's Complement to CSD Conversion

The b_i' s are the bits of the two's complement number to be converted to CSD code and the a_i's are the CSD code after the conversion. The c_i is the carry generated in step i−1 and c_{i+1} is the carry out at step i, i.e., the carry used in step i+1 together with b_{i+1} and b_{i+2} to generate the CSD output digit a_{i+1}.

Consequently, for multipliers with fixed coefficients the number of partial products can be minimized in each multiplier. For example, the two's complement coefficient {0, 1, 1, 1, 1} (decimal 15) is represented by {1, 0, 0, 0,−1} or {1, 0, 0, 0, 1}in CSD. In this case the multiplication by 15 results in only two partial products compared with four if the original two's complement representation is used. In a multiplier using CSD code for a variable coefficient the number of partial products will be the nearest integer to N/2, where N is the number of bits in the coefficient. A multiplier with a variable coefficient of 5 bits represented in CSD therefore requires three partial products. Consequently, multipliers with constant coefficients may benefit more from signed digit representation than a multiplier with variable coefficients. To encode a two's complement number to CSD representation a carry has to propagate from LSB to MSB. This carry propagation is generally the critical path of the circuit performing the CSD encoding.

MIXED INTEGER LINEAR PROGRAMMING

The Mixed Integer Linear Programming (MILP) is used to optimize the filter coefficient still further. Mixed Integer Linear Programming is the technique employed to optimize the filter coefficients to meet the given specifications. In this technique, the frequency response ripple is minimized subject to a given number of adders. The obtained results may not be the optimum in terms of the number of adders, but the saving in the number of adders can be achieved by using this technique is significant compared with those obtained using other techniques that can be used to design filters with respectable length and bit width. The computation time needed to optimize a filter with order 60 and coefficient bit width 12 is typically within a few minutes to a few hours.

Mixed Integer Linear Programming is capable of traversing the entire discrete solutions efficiently for a given word length, during the traverse the optimality of the synthesis of a set of discrete coefficients is monitored and flagged by a Certainty whenever a new coefficient is discretized. In this manner, when the traverse is completed then the algorithm is able to be aware of whether an optimum solution is obtained. Major aim of this

algorithm is capable of designing FIR filters with minimum number of adders.

Linear programming (LP) is a mathematical method for determining a way to achieve the best outcome (such as maximum profit or lowest cost) in a given mathematical model for some list of requirements represented as linear relationships.

More formally, linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Given a polytope and a real-valued affine function defined on this polytope, a linear programming method will find a point on the polytope where this function has the smallest (or largest) value if such point exists, by searching through the polytope vertices.

The general steps of the filter design problem using MILP, are given as follow**s:**

a. Formulation the filter optimization problem
b. Obtaining the continuous optimum solution
c. Choosing the effective word length (EWL)
d. Traverse of the discrete solutions

OPTIMIZING FILTER COEFFICIENT

For a given set of filter specifications such as pass band edge, stop band edge, pass band and stop band ripple ratio, and filter length N, the optimum continuous coefficients minimizing the magnitude ripple can be found by formulating the problem as a linear programming problem and this is given by

Minimize: δ subject to:

$1-\delta \leq H(\omega) \leq 1+\delta$ for $\omega \varepsilon [0, \omega_p]$

$-\delta_s \delta / \delta_p \leq H(\omega) \leq \delta_s \delta / \delta_p$ for $\omega \varepsilon [\omega_s, \pi]$

To find the coefficient values in a discrete space for the same set of filter specifications, the linear programming problem is replaced by a MILP problem. Branch and bound algorithm is one of the most efficient techniques to solve MILP problem. Branch and bound (BB or B&B) is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization.

CHOOSING EFFECTIVE WORD LENGTH

After the continuous optimum solution for the problem is obtained, all the filter coefficients are scaled up to be within a certain effective wordlength (EWL). These scaled coefficients are then to be fixed to discrete values during the traverse later. Obviously, longer EWL results in less quantization error and hence better frequency response. However, from the perspective of low complexity, low power consumption, and high-speed implementation, large EWL is usually undesirable. Therefore, in this paper, we choose the EWL as small as possible provided that feasible discrete solutions can be found within it.

TRAVERSE OF DISCRETE SOLUTIONS

This explains, roughly the traverse process of the algorithm. Before commencing the traverse, it is beneficial to know the lower bound h^l_k and upper bound h^u_k of coefficient h(k), for k=0,1,2……..N-1. These bounds can be found by solving the following linear programming problem:

Minimize; f = h(k)

Subject to : $b - \delta \leq H(\acute{\omega}) \leq b+ \delta$, for $\acute{\omega} \in [o,\acute{\omega}_p]$

$-\{ \delta s\ \delta)/ \delta p \leq H(\acute{\omega}) \leq \{ \delta s\ \delta)/ \delta p$, for $\acute{\omega} \in [\acute{\omega}_{s,}\pi]$

The above optimization finds the lower bound of coefficient h (k), denoted as h^l_k. To find the upper bound, simply set the objective function to f = -h (n). Therefore, for N coefficients, there are totally 2N runs of linear programming for finding the bounds. The traverse process is then performed using MILP to find the optimum discrete solution. Basically, the traverse is a depth-first width-recursive search. Filter coefficients are quantized to certain integers one by one according to the rules. When a filter coefficient is quantized, the remaining unquantized ones are reoptimized to compensate for the loss in frequency response.

Thus the complexity of an FIR digital filter can be reduced by quantizing its coefficients into SPT (Signed power of two) values. This converts multiplication to simple operations of shift and add. Relatively small chip area is required in VLSI realization, resulting in low cost, high speed, and high yield.

ADVANTAGES OF MILP OVER LINEAR PROGRAMMING

Here we are going for MILP, Since for any given filter specifications, there is no known lower bound for the total number of adders, in order the achieve the optimum synthesis, all the feasible discrete solutions must be taken into account. We propose an algorithm that is capable of traversing the entire discrete solutions efficiently for a given word length. What is of more importance is that during the traverse, the optimality of the synthesis of a set of discrete coefficients is monitored and flagged by a certainty, whenever a new coefficient is discretized. In such a manner, when the traverse is completed, the algorithm is able to be aware of whether an optimum solution is obtained.

While the number of adders used to realize an FIR filter is an important criterion of the implementation complexity, the power consumption and circuit speed, on the other hand are much more related to the Adder Depth (AD) of the filter .

AD is defined as the number of adders that the input signal goes through before reaching the delay element. Obviously, low AD is preferable for the concerns of low power consumption and high throughput if the same numbers of adders are required. Taking this into account, our MILP algorithm is also capable of designing FIR filters using minimum number of adders under a maximum AD constraint.

Mixed Integer Linear Programming is a considerable field of optimization for several reasons. Many practical problems in operations research can be expressed as linear programming problems. Certain special cases of linear programming, such as network flow problems and multicommodity flow problems are considered important enough to have generated much research on specialized algorithms for their solution. A number of algorithms for other types of optimization problems work by solving LP problems as sub-problems. Historically, ideas from linear programming have inspired many of the central concepts of optimization theory, such as duality, decomposition, and the importance of convexity and its generalizations. Likewise, linear programming is heavily used in microeconomics and company management, such as planning, production, transportation, technology and other issues. Although the modern management issues are ever-changing, most companies would like to maximize profits or minimize costs with limited resources.

QUANTIZATIONQuantization is the process of converting a continuous range of values into a finite range of discreet values. This is a function of analog-to-digital converters, which create a series of digital values to represent the original analog signal. The bit depth (number of bits available) determines the accuracy and quality of the quantized value. Quantization has a number of applications in digital image and audio production. Values can be "rounded" to a commonly-agreed standard for simplicity.

FINDING THE APPROXIMATION OF A NUMBER IN SUB EXPRESSION SPACE

Having the sub expression space constructed based on a basis set, we wish to find the best approximation of a real number using a sum of not more than K sub expression terms $y(i)2^{q(i)}$, where y(n) is an element of the set of permissible sub expression bases and q(n) is a non-negative integer. Let [x]k be the K- term sub expression number that is a best approximation to x. Since the number of elements in the basis set is limited, an exhaustive search can be employed to find the combination of K

pairs of y(i) and q(i) which minimize the error between x and the approximation.

When the order of the basis set and/or K are/is increasing, exhaustive search becomes computational impractical. A greedy algorithm for computing is proposed. In this algorithm, initially, the approximation to [x]k, denoted as [x], is set to 0. Sub expression terms are assigned to one at a time. At each time, the sub expression term which can minimize the difference between and is assigned to x.

The algorithm runs as follows

Step 1) Initialize m=1 and Z0=x

Step 2) Find $y(m)2^{q(m)}$ which minimized $|z_{m-1} - y(m)2^{q(m)}|$

Step 3) If either y(m) = 0 or m = k,go to Step 6; other wise go to Step 4

Step 4) Update $Z_m=Z_{m-1}- y(m)2^{q(m)}$

Step 5) Increment m. Go to Step 2

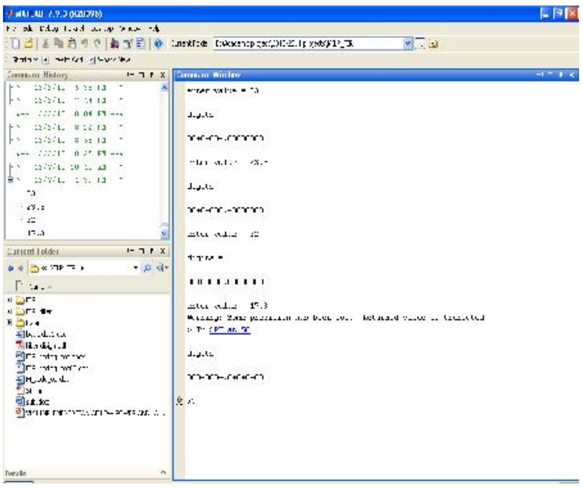
Step 6) $[x]_k = y(i)2^{q(i)}$. Stop

9.2 DYNAMICALLY EXPANDING SUB EXPRESSION BASIS SET

A sub expression basis set is defined as a set that contains zero and odd-valued integers. The order of a sub expression basis set is defined as the number of adders that is required to realize the elements in the set. For instance, the order of the sub expression basis set S2 = {0, ±1, ±5, ±7} is 3. Here, it is assumed that if a positive odd number is realized, its negative value is automatically realized and the elements 0 and ±1are always included. Therefore, for expository convenience, we omit the negative values and 0. Thus, the above sub expressions basis set is simplified as{1,3,5,7} .During the traverse process of the algorithm, each node is associated with a sub expression basis set, and the sub expression basis sets are updated dynamically. When the traverse searches forward along a path with more and more coefficients being quantized to integers, the sub expression basis set expands, while the search is traced back to a coefficient, the original basis set of that coefficient is recovered.

RESULTS AND DISCUSSIONS

SIGNED POWER OF TWO (SPT) COEFFICIENT OUTPUT



NORMAL FILTER DESIGN

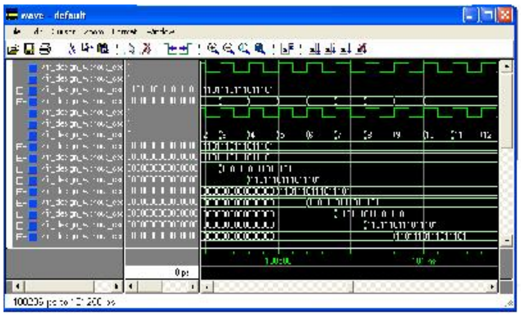
Synthesis Report

Final Report

Device utilization summary:

Selected Device : 3s200ft256-4

Number of Slices:	1019	out of	1920	53%
Number of Slice Flip Flops:	193	out of	3840	5%
Number of 4 input LUTs:	1759	out of	3840	45%
Number of bonded IOBs:	50	out of	173	28%
Number of GCLKs:	2	out of	8	25%



Normal Filter Design

FILTER DESIGN USING CSD

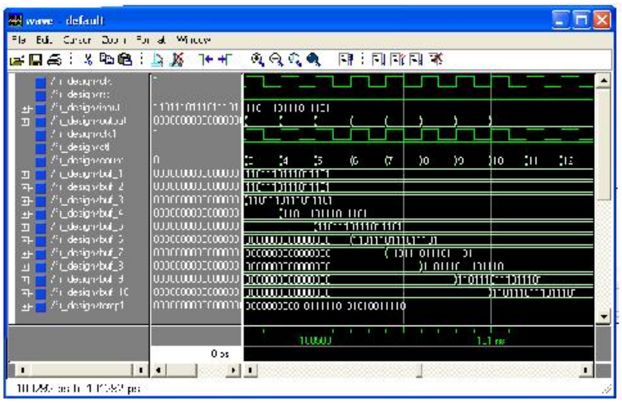
Synthesis Report

Final Report

Device utilization summary:

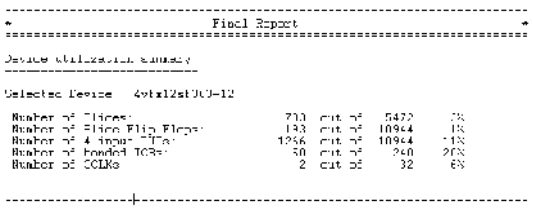
Selected Device 3s200ft256-4

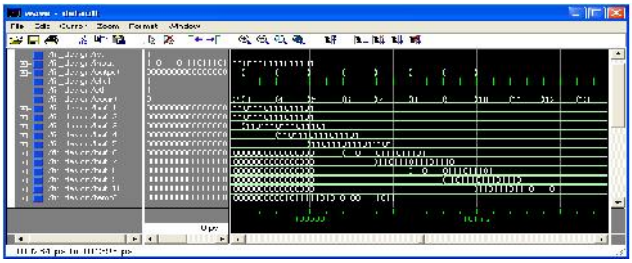
Number of Slices	977	out of	1920	50%
Number of Slice Flip Flops:	1[illegible]	out of	3840	5%
Number of 4 input LUTs:	1[illegible]1	out of	3840	44%
Number of bonded IOBs:	50	out of	173	28%
Number of GCLKs	2	out of	8	25%



Filter Design Using CSD

FILTER DESIGN USING MILP
Synthesis Report





FILTER DESIGN USING MILP

CONCLUSION

The technique of sub expression has the potential of effecting significant savings in the numbers of additions used in the implementation of FIR filters. It has been observed that other techniques utilizing common sub expressions in coefficient representations other than CSD may on occasions lead to fewer adders, and in general can achieve similar savings. Preliminary studies have showed that only a limited number of SPT terms are required to meet a respectable set of specifications if a good optimization technique exists. Hence, to represent the coefficients of a filter in this way, the coefficient multipliers can be replaced by a small number of add/subtract-shift operations. The hardware complexity is thus largely reduced. By using MILP adder usage is further minimized.

REFERENCES

[1] Dong Shi and Ya Jun Yu,(June 2010), " Design of Linear Phase FIR filters with high probability of Achieving Minimum Number of adders", IEEE Trans.Circuits Syst.I.

[2] Chen C.L and Willson A. N,(1999), " A trellis Search algorithm for the design of FIR filters with signed-powers-of-two coefficients", IEEE Trans.Circuits Syst II, Analog Digit. Signal Process., Vol. 46, No.1, pp. 29-39.

[3] Dempster A.G. and Macleod M.D (1995), "Use of minimum- adder multiplier blocks in FIR digital filters", IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.,Vol. 42, No. 9, pp 569-577.

[4] Gustafsson O,(2007) "A difference based adder graph heuristic for multiple constant multiplication problems", in Proc.IEEE ISCAS'07,New Orleans,LA, pp. 1097-1100.

[5] Gustafsson O, (2007) "Lower bound for constant multiplication problems," IEEE Trans. Circuits Syst. II, Exp, Briefs, Vol .54, No.11, pp.974-978.

[6] Gustafsson O and Wanhammar L, "Design of linear-phase FIR filters combining subexpression sharing with MILP", in Proc.MWSCAS' 02, pp.9-12.

[7] Gustafsson O and Wanhammar L, "Improved Multiple Constant multiplication using minimum spanning trees", in proc.Asilomar Conf. Signals,Syst.,Comp., 2004,pp.63-66.

[8] Hartley R.I , (1996) " Subexpression sharing in filters using canonic signed digit multipliers", IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., Vol.43, No. 10, pp.677-688.

[9] Samueli. H,(1989) "An improved search algorithm for the design of multipliers FIR filters with powers-of-two coefficients ", IEEE Trans.Circuits Syst.,Vol.36, no 7, pp.1044-1047.

[10] Yao. C.Y and Chien C.J, (2002) " A Partial MILP algorithm for the design of linear phase FIR filters with SPT Coefficients", IEICE Trans. Fundam., Vol. E85-A,pp. 2302-2310.

[11] Yao C.J.(2004) , " A novel common - subexpression-elimination method for synthesizing fixed-point FIR filters", IEEE Trans. Circuits Syst.I, Vol. 51, no 11, pp 2215-2221.

[12] Yu Y.J , Shi D and Lim Y.C, "Subexpression encoded extrapolated impulse response FIR filter with perfect residual compensation", in Proc.IEEE ISCAS'08, pp. 2446-2449.

[13] Yu Y.J , Shi D and Lim Y.C, (2009) "Design of extrapolated impulse response FIR filters with residual compensation in subexpression space", IEEE Trans. Circuits Syst. I, Reg.Papers, Vol.56, No.12 , pp 2621-2633.

[14] Yu Y.J and Lim Y.C,(2010) "Optimization of linear phase FIR filters in dynamically expanding subexpression space", Circuit Syst. Signal Process., Vol. 29,pp.65-80.