ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 7, Issue, 04, pp.14980-14985, April, 2015

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

REVIEW ARTICLE

DESIGN OF DIGITAL SERIAL FIR FILTER

***,[1]Kalbande, A.G. and [2]Choudhari, P.S.**

[1]M.E Student of PRMCEAM, Badnera-Amravati, (M.S) India.
[2]Department of EXTC, PRMCEAM, Badnera-Amravati,(M.S)India.

ARTICLE INFO

Article History:
Received 23rd January, 2015
Received in revised form
09th February, 2015
Accepted 09th March, 2015
Published online 30th April, 2015

Key words:

Integer linear programming (ILP),
Digit serial arithmetic,
Finite impulse response ((FIR) filters,
Gate level area optimization,
Multiple constant multiplication.

ABSTRACT

In the last two decades, many efficient algorithms and architectures have been introduced for the design of low complexity bit-parallel multiple constant multiplications (MCM) operation which dominates the complexity of many digital signal processing systems. On the other hand, little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations albeit at the cost of an increased delay. In this paper the design of a digital-serial N tap FIR filter with programmable coefficients is presented. The design considers the general case of W-bit sample word and M-bit coefficient word. The processing of the data within the filter takes place with full precision. The output data is truncated to W bits. also in this paper we address the problem of optimizing the gate-level area in digit-serial MCM designs and introduce high level synthesis algorithms, design architectures, and a computer aided design tool.

INTRODUCTION

Finite Impulse Response (FIR) filters are of great importance in digital signal processing (DSP) systems since their characteristics in linear-phase and feed-forward implementations make them very useful for building stable high-performance filters (Leven Aksoy and Cristiano Lazzari, 2013). The N-tap FIR digital filter is normally described by the equation (Wanhammar, 1999).

$$y(n) = \sum_{i=0}^{N-1} H_i X_{n-i} \quad \text{----------} \quad \text{(Equ.1)}$$

In case of DSP, the output sample is the sum of a number of terms while the term itself represents a multiplication of M-bit sample word and W-bit coefficient word. Accordingly, the processing of the data in case of DSP has two levels; the first is on the level of the calculation of the term (multiplication process) and the second level is on the system level to get the output by accumulating the outputs of the first level. The fully serial implementation uses one serial-bit multiplier to calculate the terms in serial form and one accumulator to accumulate the results. Such implementation guarantees minimum hardware but at the same time gives a system that cannot be used with any real-time application (Suzuki *et al.,* 1998). On the other hand, using parallel multiplier for each term in the expression of the filter together with parallel adding network to get the output, results in a tremendous amount of hardware with speed so high such that it is not needed in many applications (Pasko et al., 1999). Many of the researches in the field of DSP concentrate on finding algorithms and implementation techniques that results in real-time system with reasonable hardware complexity and that can be implemented using FPGA devices (Daurland, 2004; Javier Valls, 1998). Bit parallel designs process all of the bits of an input simultaneously at a significant hardware cost. In contrast, a bit serial structure processes the input one bit at a time, generally using the results of the operations on the first bits to influence the processing of subsequent bits. The advantage enjoyed by the bit serial design is that all of the bits pass through the same logic, resulting in a huge reduction in the required hardware. Typically the bit serial approach requires $1/M_{th}$ of the hardware required for the equivalent M-bit parallel design. The organization of this paper is as follows. In the next section the FIR filter architecture, the different canonical and inverted form topologies to design FIR filters are summarized. The expression and the topology that we are going to consider in implementation are explained. Section 3 covers all the design and implementation aspects. The digit-serial architectures are briefly exposed, the basic digit-serial adder cell is explained, the proposed digit- serial multiplier is also given and finally the input output control block is given. Section 4 deals with the FPGA implementation while in section S some conclusions and results are given. The FIR filter is implemented in two forms

Corresponding author: Kalbande, A.G.
M.E Student of PRMCEAM, Badnera-Amravati, (M.S) India

that is Direct Form and Transpose Form. The direct and transposed-form FIR filter implementations are illustrated in Fig.1 and 2 respectively. Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency (Wanhammar, 1999; Wallace, 1964). The multiplier block of the digital FIR filter in its transposed form [Fig. 2], where the multiplication of filter coefficients with the filter input is realized, has significant impact on the complexity and performance of the design because a large number of constant multiplications are required. This is generally known as the multiple constant multiplications (MCM) operation and is also a central operation and performance bottleneck in many other DSP systems such as fast Fourier transforms, discrete cosine transforms (DCTs), and error-correcting codes (Dempster and Macleod, 1995).

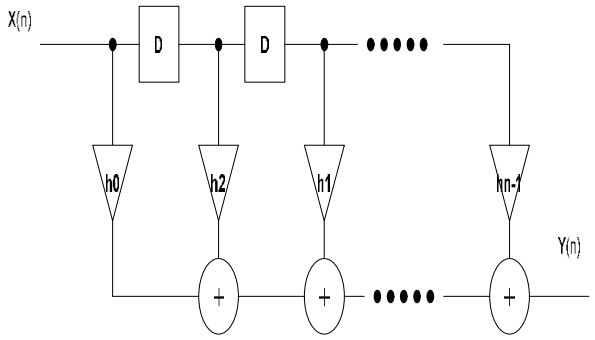

Fig 1. FIR filter representation (direct form)

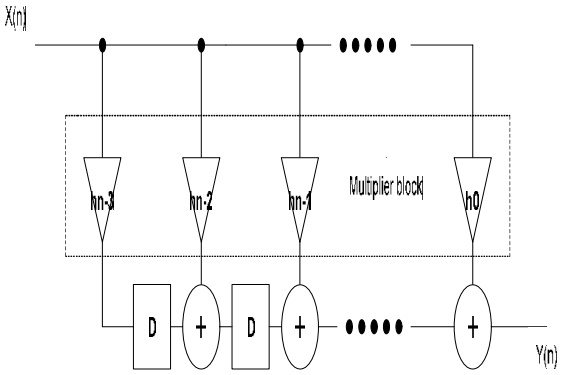

Fig 2.FIR filter representation (transpose form)

Although area-, delay-, and power-efficient multiplier architectures, such as Wallace (Wanhammar, 1999) and modified Booth (Wallace, 1964) multipliers, have been proposed, the full flexibility of a multiplier is not necessary for the constant multiplications, since filter coefficients are fixed and determined beforehand by the DSP algorithms (McClellan *et al.,* 1973). Hence, the multiplication of filter coefficients with the input data is generally implemented under a shift adds architecture (Nguyen and Chatterjee, 2000), where each constant multiplication is realized using addition/subtraction and shift operations in an MCM operation [Fig. 1(c)]. For the shift-adds implementation of constant multiplications, a straightforward method, generally known as digit based recoding [5], initially defines the constants in binary.

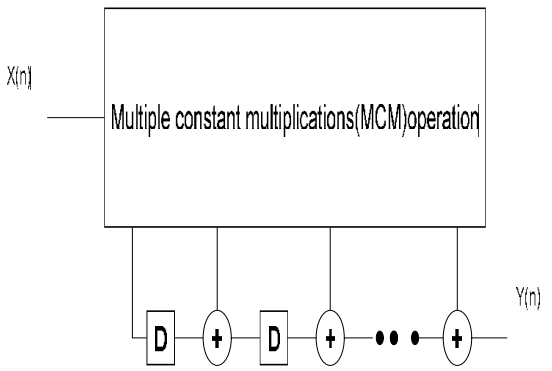

Fig 3. FIR filter representation (direct form with MCM block)

Then, for each "1" in the binary representation of the constant, according to its bit position, it shifts the variable and adds up the shifted variables to obtain the result. As a simple example, consider the constant multiplications 29*x* and 43*x*. Their decompositions in binary are listed as follows:

29x = (11101) bin x=x<<4x+ <<3x+ <<2x+x;
43x = (101011) bin x=x<<5x+ <<3x+ <<1x+x;

This requires six addition operations as illustrated in Fig. 4, In Fig. 4, the exact CSE algorithm of [8] gives a solution with four operations by finding the most common partial products 3x = (11) bin x and 5x = (101) bin x when constants are defined under binary, as illustrated in Fig. 4(b). On the other hand, the exact GB algorithm (Aksoy, 2008; Cappello and Steiglitz, 1984; Flores, 2005; Gustafsson and Wanhammar, 2002) finds a solution with the minimum number of operations by sharing the common partial product 7x in both multiplications, as shown in Fig. 4(c).

Fig 4. Shift-adds implementations of 29*x* and 43*x*. (a) Without partial product sharing [6] and with partial product sharing. (b) Exact CSE algorithm [9]. (c) Exact GB algorithm [12].

Note that the partial product 7x = (111) bin x cannot be extracted from the binary representation of 43x in the exact CSE algorithm (Aksoy *et al.,,* 2007; Suzuki *et al.*, 1998). However, all these algorithms assume that the input data x is processed in parallel. On the other hand, in digit-serial arithmetic, the data words are divided into digit sets, consisting of d bits, which are processed one at a time. The algorithms designed for the MCM problem can be categorized in two classes (Hartley and Corbett, 1990): common Sub expression elimination (CSE) algorithms (Park and Kang, 2001; Aksoy, 2008) and graph-based (GB) techniques (Aksoy *et al.*, 2007; Aksoy, 2011; Hartley and Corbett, 1990; Cappello and Steiglitz, 1984; Flores, 2005; Gustafsson and Wanhammar, 2002; Potkonjak, 1996; Suzuki, 1998; Pasko, 1999; Magnus Karlson, 2005; Daurland, 2004; Javier Valls, 1998; Rakesh and Patney, 1979; Chearl Park and Hyeong Jukung, 2002). The CSE algorithms initially extract all possible Sub expression from the representations of the constants when they are defined under binary, canonical signed digit (CSD) , or minimal signed digit (MSD) (Hartley, 1996). Then, they find the "best" Sub expression, generally the most common, to be shared among the constant multiplications. The high-level algorithms should take into account the sharing of shift operations as well as the sharing of addition/subtraction operations in digit-serial MCM design. Furthermore, finding the minimum number of operations realizing an MCM operation does not always yield an MCM design with optimal area at the gate level (Cappello and Steiglitz, 1984). Hence, the high-level algorithms should consider the implementation cost of each digit-serial operation at the gate level. we introduce the exact CSE algorithm (Flores *et al.,* 2005) that formalizes the gate-level area optimization problem as a 0–1 integer programming (ILP) problem when constants are defined under a particular number representation (Aksoy *et al.,* 2007). In a digit-serial arithmetic implementation, the W bits of a data word are processed in units of the digit size N in W/N clock cycles. This leads to arithmetic operators that have smaller area than equivalent to parallel arithmetic and have a larger throughput than equivalent bit serial arithmetic design. By considering a range a of value for the digit size, one can search the design space to find optimum implementation for a given application. The basic functions required for most any digital signal processor include addition, negation, and storage registers (delay). These basic functions can then be used to construct more complex structures such as accumulators and multipliers. With a digit-serial architecture, these hardware units are parameterized by the digit-size.

Digit-serial adder

A basic element in a digit-serial arithmetic implementation is the digit-serial adder, as shown in Fig. 1 (a). A digit-serial-adder is a circuit that adds two digits along with a previous carry bit and produces the sum digit and a new carry bit. The two operands, A and B, are fed one digit at a time into the digit-serial adder. The addition is done N bits at a time, with the carry rippling from one full adder to the next. The carry-out from the digit-serial adder is fed back into the first full adder during the next clock cycle, when the next digits of the inputs have arrived. The latency of digit-size inputs for each pipelined digit serial adder is one clock cycle time. In a Xilinx XC4010 FPGA implementation (Daurland, 2004; Javier Valls , 1998), a digit-serial adder with N=2 uses all 2 of the lookup tables (LUTs) and 1 of the CLB flip-flops within a configurable logic block (CLB).

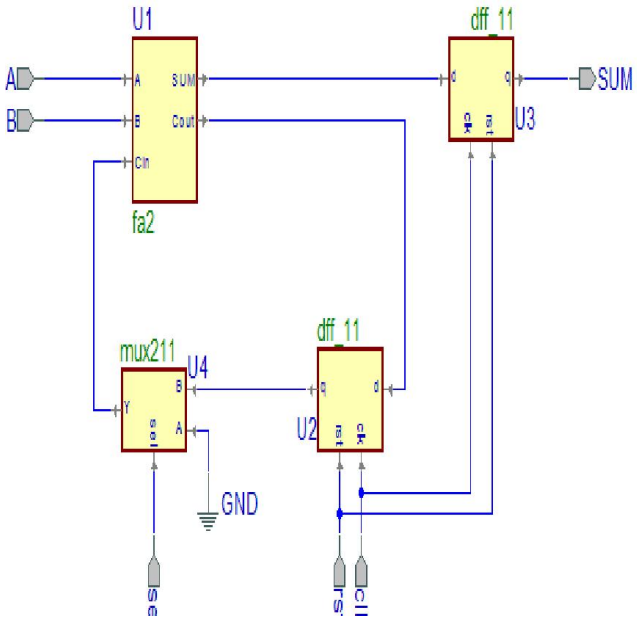

Fig 5. Digit serial Adder

Digit-serial multiplier

Multipliers are essential in most DSP applications. A digit-serial multiplier with a fixed coefficient can be implemented by generalizing the structure of a bit-serial multiplier. The digit-serial multiplier basic module and multiplier structure with N=2 are shown in Fig. 1 (b) and (c), respectively. A 2-input AND gate generates each partial product. Each digit-serial multiplier modules can be connected in a systolic array fashion to implement a very fine grained pipeline.

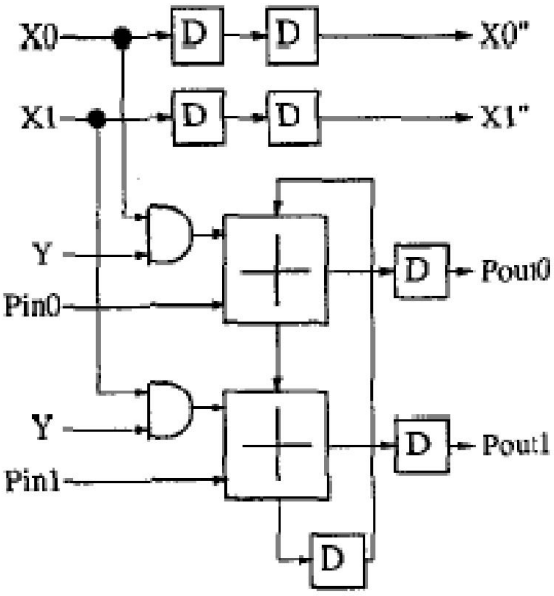

Fig 6.Digit Serial Multiplier

This network of digit-serial multiplier modules can easily be mapped into FPGAs because of their sparse and regular interconnections. The bits of the multiplier are supplied one digit at a time, starting with the least significant digit, while the bits of the multiplicand are supplied as a parallel word. Each partial product is shifted and then added to the previous partial products.

In order to increase the throughput of the digit-serial multiplier, the architecture is highly pipelined. Pipelining is done in order to limit the critical path propagation delays between registers. In the example shown, the pipelining limits the propagation to a 2-bit adder in the digit-serial multiplier with N=2. Due to the prevalence of local routing in this architecture, both high performance and high density can be attained in a straightforward way using the FPGA building blocks and routing resources.

Delay

The delay element 2-l in an FIR filter signifies a full word delay. The delay is used to align data words and to propagate control signals that must also be properly synchronized. Word delays can be implemented as shift-registers of the proper length. Assume that each data word consists of L digits. The unit of delay is the delay in processing a single digit. Thus, the delay in the transfer of one word is equal to L times this amount.

Literature review

Leven Aksoy, Cristiano Lazzari. "Design Of Digit Serial FIR Filter"

Theory detail: In this paper author describe all the detail information regarding to design Digit serial FIR filter and also he describe the problem associate during the design.

Our finding: From this paper we got information regarding to the problem occur during the design of digit serial FIR filter. Also we get the algorithm which is to used for optimization.

L. Wanhammar. The book" DSP Integrated Circuits"

Theory detail: In this book detail of DSP and VLSI combine application is given with different feature.

Our finding: From this book we got basic information related to FIR filter.

C. Wallace, "A suggestion for a fast multiplier".

Theory detail: In this paper author give the suggestion regarding to the design of combinational multiplier and division which perform the operation more fast as compared to conventional multiplier.

Our finding: From this paper we observe that the designing method of multiplier is more efficient than conventional multiplier. But it offers more complexity.

W. Gallagher and E. Swartzlander, "High radix booth multipliers using reduced area adder trees"

Theory detail: In this paper author introduces a booth multiplier, which offer more simplicity and flexibility as compared to Wallace multiplier .it also introduce booth multiplication algorithm for to design it. The use of a fast adder tree, such as that found in a Reduced Area multiplier, permits straightforward design of very high radix Booth multipliers.

Our finding: In this paper observe the technique of to reduce the adder tree. And also we get design techniques of high operands multiplication technique.

J. McClellan, T. Parks, and L. Rabiner "A computer program for designing optimum FIR linear phase digital filters"

Theory detail: This paper contain a general-purpose computer program which is capable of designing a large class of optimum FIR linear phase digital filters. The program has options for designing such standard filters. The program can also be used to design filters which approximate arbitrary frequency specifications which are provided by the user.

Our finding: From this we observe that the speed of the algorithm, as well as its simplicity make this program an attractive one for a wide variety of design applications.

H. Nguyen and A. Chatterjee, "Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis,"

Theory detail: In this Paper author propose an optimization that combines a numerical transformation called number-splitting with a shift-and-add decomposition scheme.

Our finding: From this paper, we get Optimization framework based on numerical techniques. These optimization methods are based on real-number splitting and shift-and-add decomposition for Sub expression sharing. For two's complement encoding of coefficients, the number of operations can be reduced as much as possible.

R. Hartley, "Sub expression sharing in filters using canonic signed digit multipliers,"

Theory detail: In this paper author examines methods for optimizing the design of CSD multipliers, and in particular the gains that can be made by sharing sub expressions.

Our finding: From this paper we observe that the technique of Sub expression sharing has the potential of effecting significant savings in the numbers of additions used in the implementation of FIR filters.

I.C. Park and H.J. Kang, "Digital filter synthesis based on minimal signed digit representation,"

Theory detail: In this paper author introduce digital filter synthesis algorithm that is based on the minimal signed digit

(MSD) representation. The MSD representation is attractive because it provides a number of forms that have the minimal number of non-zero digits for a constant.

Our finding: In this paper, we observed new digital filter synthesis algorithm that is based on the MSD representation The proposed filter synthesis algorithm is to select one coefficient at a time and synthesize it using previously synthesized patterns, which is different from the conventional method.

L. Aksoy, E. Costa, P. Flores, and J. Monteiro. "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications".

Theory detail: In this paper author introduce common Sub expression elimination algorithm for the optimum sharing of partial terms in multiple constant multiplications (MCMs). We model this problem as a Boolean network that covers all possible partial terms that may be used to generate the set coefficients in the MCM instance.

Our finding: From this paper we got different algorithm by which we can minimize the area and delay during multiplication.

A. Dempster and M. Macleod. "Use of minimum-adder multiplier blocks in FIR digital filters".

Theory detail: In this paper author gives CSD algorithm by which the minimization of area and time will take place.
Our finding: From this paper we get idea regarding to design optimize circuit using different algorithm.

L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of area in digital FIR filters using gate-level metrics,"

Theory detail: In the paper ,author gives new metric for the minimization of area in the generic problem of multiple constant multiplications, and demonstrate its effectiveness for digital FIR filters.

Our finding: From this paper we observe that exact algorithm that minimizes the area of the multiplier block of a digital filter is proposed.

Proposed work

Basically here in this paper our main aim is to design serial digit FIR filter and generally there are two type of filter that is finite impulse response and infinite impulse response but in that we are not able to design infinite impulse response filter practically, so here we are interested in to design of finite impulse response filter. Then according to the way of applied input we can apply input in two ways that is serial and parallel, but in parallel design we required more area and power so we are avoid that disadvantages by designing the serial FIR filter. Again on the basis of applied input we can apply the input in two way i; e bit and digit. It means we can apply input as bit by bit or by using digit. But by referring the reference paper we got that while applying input in bit wise has certain disadvantages, for to avoid it we are applying the input digit wise that is the grouping of number of bit. So overall name of our topic is digit serial fir filter .so for to design the digit serial FIR filter we are distribute overall process in certain step which are as given below.

- At first we design simple fir filter in direct form without any modification and check the result in term of area, power dissipation and delay.
- Then we design simple fir filter in direct form without any modification and check the result in term of area, power dissipation and delay.
- Then perform certain modification in term of adder and multiplier circuit required for the design of digit serial fir filter.
- Then check the result of serial digit fir filter in term of area, power and delay.
- Compare all the result of different filter with each other and make a comment that why our design serial digit filter is better as compared to other.

Conclusion

In this paper we are explain different type of filter, on the basis of different specification we are compare them such as an according to area, power dissipation , delay and by comparing all of them we got certain conclusion which is that digit serial FIR filter which is design in our paper is excellent among all the filter.

REFERENCES

Aksoy, L., Costa, E., Flores, P. and Monteiro, J. 2007. "Optimization of area in digital FIR filters using gate-level metrics," in Proc. DAC, 2007, pp. 420–423.

Aksoy, L., Costa, E., Flores, P. and Monteiro, J. 2008. "Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications," IEEE Trans. Computer.-Aided Design Integr. Circuits Syst., vol. 27, no. 6, pp. 1013–1026,.

Aksoy, L., Lazzari, C., Costa, E., Flores, P. and Monteiro, J. 2011. "Optimization of area in digit-serial multiple constant multiplications at gate-level," in Proc. ISCAS, pp. 2737–2740.

Cappello, P. and Steiglitz, K. 1984. "Some complexity issues in digital signal processing," IEEE Trans. Acoust., Speech, Signal Process., vol. 32, no. 5, pp. 1037–1041.

Chearl Park and Hyeong Jukung. 2002. "Digit Filter Synthesis Based On Algorithm To Generate All Minimal Signed Digit Parameter "In IEEE Transaction.

D.S.Daurland "Design and FPGA Implementation Of Digit Serial FIR Filter "In IEEE Transaction African 2004.

Dempster, A. and Macleod, M. 1995. "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 42, no. 9, pp. 569–577,.

Flores, P., Monteiro, J. and Costa, E. 2005. "An exact algorithm for the maximal sharing of partial terms in multiple constant multiplications," in Proc. Int. Conf. Comput.-Aided Design, Nov. 2005, pp. 13–16.

Gallagher, W. and Swartzlander, E. 1994. "High radix booth multipliers using reduced area adder trees," in Proc. Asilomar Conf. Signals, Syst. Computer., vol. 1. Pacific Grove, CA, pp. 545–549.

Gustafsson, O. and Wanhammar, L. 2002. "ILP modelling of the common subexpression sharing problem," in Proc. ICECS, pp. 1171–1174.

Hartley, R. and Corbett, P. 1990. "Digit-serial processing techniques," IEEE Trans. Circuits Syst., vol. 37, no. 6, pp. 707–719.

Hartley, R. 1996. "Subexpression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 43, no. 10, pp. 677–688,

Javier Valls, 1998. "Design and FPGA Implementation Of Digit Serial Fir Filter "In IEEE Transaction 1998.

Leven Aksoy and Cristiano Lazzari. 2013. "Design Of Digit SerialFIR Filter" Transaction on VLSI Systems, Vol.21.No.3.

Magnus Karlson. 2005. "Algorithm Transformation In A Design Of Digit Serial Fir Filter "in IEEE Transaction.

McClellan, J., Parks, T. and Rabiner, L. 1973. "A computer program for designing optimum FIR linear phase digital filters," IEEE Trans. Audio Electroacoust. vol. 21, no. 6, pp. 506–526,.

Nguyen, H. and Chatterjee, A. 2000. "Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 8, no. 4, pp. 419–424.

Park, I.C. and Kang, H.J. 2001. "Digital filter synthesis based on minimal signed digit representation," in Proc. DAC, pp. 468–473.

Pasko, R., Schaumont, P., Derudder, V. and Vernalde S. 1999. "A New Algorithm for Elimination Of Common Subexpressions" IEEE Transaction on Computer Aided Design Of Integrated Circuit And System, Vol. 18, No. 1,

Potkonjak, M., Srivastava, M. and Chandrakasan, A. 1996. "Multipl constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," IEEE Trans. Comput- Aided Design Integr. Circuits Syst., vol. 15, no. 2, pp. 151–165, .

Rakesh, N. and Patney. 1979. " Design Of Linear Phase Fir Filter Using Pseudo- Boolean Methods" in IEEE Transaction On Circuit and System , Vol.CAS-26 No.4.

Suzuki, H., Chang, Y.N. and Parhi, K. 1998. "Performance tradeoffs in digitserial DSP systems," in Proc. Asilomar Conf. Signals, Syst. Comput., 1998, pp. 1225–1229.

Wallace, C. 1964. "A suggestion for a fast multiplier," IEEE Trans. Electron. Computer. vol. 13, no. 1, pp. 14–17,.

Wanhammar, L. 1999. DSP Integrated Circuits. New York: Academic.

****