Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VLSI ARCHITECTURE FOR LOW POWER MINIMUM SIGNED DIGIT MULTIPLIER FOR FIR FILTER AND ITS SIGNAL PROCESSING APPLICATIONS

[1] K.N.VIJEYAKUMAR, [2] Dr.V.SUMATHY, [3] E.J.AISHWARYA, [4]S.SARAVANAKUMAR, [5]M.GAYATHRI DEVI

[1]Asst Prof, Department of ECE, Anna university of Technology, Coimbatore-India.

[2]Assoc. Prof., Department of ECE, Government College of Technology , Coimbatore-India.

[3]Asst Prof, Department of ECE, Sri Shakthi Institute of Engineering and Technology, Coimbatore-India.

[4,5]PG scholar, Department of ECE, Anna university of Technology, Coimbatore-India.

E-mail: [1]vijey.tn@gmail.com, [2]sumi2001_gct@yahoo.co.in,[3]aishu.ej@gmail.com, [4]saravanakumar.vlsi@gmail.com, [5]gdevi1988@gmail.com

ABSTRACT

Most of the Digital Signal Processing applications (DSP) involve multiplication and accumulation operation. As multiplier decide the critical computation time and power dissipation of a signal processing architecture, design of high throughput and power efficient multiplier is a major challenge. In this paper, we proposed a new architecture for high speed multiplication based on shifting and adding inputs based on bit values of filter coefficients. Minimum Signed Digit (MSD) representation of constants with Common Sub expression (CS) algorithms has been used to reduce the number of addition operations in filters for reducing dynamic power consumption. The proposed architecture has been designed using VHDL and simulated in ALTERA Quartus II 8.1. Experimental results demonstrates that the dynamic power dissipation of our MSD-CS multiplier is better compared to the previous designs and shows a delay reduction up to 17.2% compared to standard design. Design implementation of proposed multiplier in an18 tap FIR filter is done to verify its functionality.

Keywords: *Common Sub-Expression Elimination (CSE), Shift And Add, Canonic Signed Digit(CSD) Representation, Finite Impulse Response (FIR),Infinite Impulse Response(IIR).*

1. INTRODUCTION

A Fixed transforms (e.g., FIR/IIR filters with fixed coefficients like Discrete Cosine Transform (DCT), Discrete Fourier Transform(DFT),etc., in DSP algorithms, do not require the flexibility of a general purpose multiplier as the multiplicand has a limited number of values. Hence multiplication in filters can be easily realized using shifts and addition. The shifts can be realized using hard-wired shifters and addition using conventional adders. Furthermore, we can reduce the number of adders by using CSE techniques. The CSE tackles the Multiple Constant Multiplication (MCM) problem [6], [8] by minimizing the number of additions through extracting common parts among the constants represented in CSD [1],[3],[7], [9],[11],[13].

MCM[4] is a transformation used to reduce the number of Logic Operations(LOs), closely related to the widely used substitution of multiplications with constants by shifts and additions[15] .While the latter considers multiplication of only one constant at a time, the MCM considers simultaneous multiplication of one variable with multiple constants at same instance[8]. However the uniqueness of constant representation is an important criteria for MCM. In [13] A.P.Vinod *et al,* have proposed a multiplier based on CSE algorithm for FIR filter implementation. The major advantage of CSE algorithm is that it eliminates redundant computations in Multiplier Blocks (MBs) with minimum number of additions. However the logic depth is same as that of binary sub expression multiplier. A reconfigurable architecture for FIR filter using coefficient representation in CSD was proposed in [16] by Chen and Chiueh. Though the architecture offers high flexibility it suffers from high area overhead. In another pioneer work on design of reconfigurable architectures for FIR filters [17] Mahesh and Vinod used CSE algorithm for eliminating redundant architectures. Though the

Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

architecture offers better performance in terms of area and power dissipation the design suffers from high hardware complexity for large *n* in case of Programmable Shift Multiplier (PSM). In this paper, we proposed a new architecture for high speed multiplication based on shifting and adding of inputs by constants represented in MSD. CSE algorithms have been used to reduce the number of addition operations, saving dynamic power dissipation.

The rest of the paper is organized as follows. In section 2, the CSE method is reviewed. Section 3 discusses about features of MSD representation and conversion of binary co-efficient to MSD system. Also the architecture of proposed MSD multiplier and its working are discussed. In section 4 the performance of the proposed MSD based shift multiplier is analyzed and compared with conventional binary multiplier .An implementation of the proposed multiplier in an FIR filter is also verified in this section. The functionality of FIR filter designed using proposed multiplier for a signal processing applications is reported in Section 5. Section 6 gives a brief conclusion of the work.

2. REVIEW OF CSE METHOD

The proposed architecture considers coefficients as constants stored in Look Up Table (LUT) and input bit pattern as a variable. Coefficient multiplication is done using MCM approach i.e., multiplication of one variable (input signal) with multiple constants (filter coefficients) .CSE algorithm is used along with MCM approach for minimizing the number of additions by extracting the common parts among the constructs represented in binary form.

The basic idea of CSE method can be illustrated using an example 16 bit coefficient h_k =0.001011000111101. Using direct method i.e., implementation without CSE algorithm requires seven adders to compute the output y_k corresponding to h_k (number of adders needed is one less than the number of nonzero bits in h_k). In h_k, the bit pattern [1 1] is present thrice, which can be expressed as 2 bit sub-expression $x_1 = x + 2^{-1}x$, where x is the input signal. Using the CS, the output of the filter can be expressed as

$$y_k = 2^{-3}x + 2^{-5}x_1 + 2^{-10}x_1 + 2^{-12}x_1 + 2^{-15}x \quad (1)$$

Note that only five adders are required to obtain y_k using CSE (one adder for obtaining the sub-expression x_1, and four adders for (1)). Hence the coefficient multiplier optimized using the CSE requires two adders less than the structure obtained using direct method. Thus using CSE, the number of adders required to implement the coefficient multiplications of the filter structure can be minimized [4]. The implementation of multiplier in FIR filter structure is shown in figure 1.

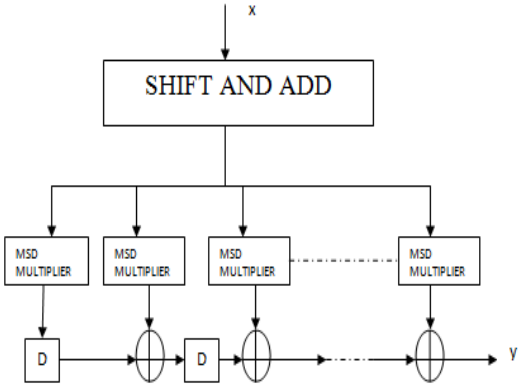


Figure 1.Transposed direct form of an FIR filter realized using shift-and add constant multiplier.

3. PROPOSED MSD SHIFT- AND- ADD MULTIPLIER

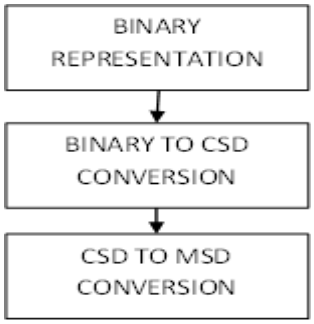
3.1 MSD representation

As MSD representation is more appropriate in finding CS for multiple constants, we have designed a shift-and add multiplier using MSD representation for each constant to be synthesized. The CSD representation is a radix-2 signed digit system with the digit set $\{1,0,\bar{1}\}$. The CSD representation for a given number is unique and has two properties; the first is that the number of non-zero digits is minimal and the second is that the product of adjacent two digits is zero. The CSD number system is an efficient way of representing the coefficients, as it reduces 33% of non-zero digits compared with the equivalent binary representation [2]. The CSD representation is widely used in implementing MCMs since it reduces the number of additions for a given constant multiplication. However, with CSD representation only limited sub-expressions can be realized for multiple constants. Another form of binary representation called as MSD representation can be realized if the second property of CSD representation is relaxed. The transformations needed to convert the CSD representation to MSD representation [5] are as follows: First the binary number is converted into equivalent CSD form. In the next step, a pattern of either $10\bar{1}$ or $\bar{1}01$ is

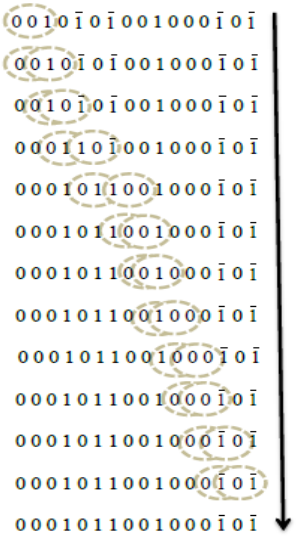
Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

searched, starting from the leftmost bit position i.e., Most Significant Bit (MSB) and transformed into 011 or $0\bar{1}\bar{1}$ respectively. A new MSD representation is generated for each transformation. The iterative procedure in previous step is continued until there is no such pattern in the transformation. The pattern is searched in MSD representation from the next position of the digit where a transformation is applied to generate the MSD representation [5], [2]. The basic block for conversion of binary to MSD with an example is shown in figure 2.



(a)



(b)

Figure 2. CSD to MSD conversion (a) Block diagram (b)Example

3.2. Architecture of proposed MSD- CS shift-and-add multiplier

In this section the architecture of proposed MSD shift-and-add CS multiplier is presented. Our architecture works on the concept of shifting and adding of partial products to realize the multiplied result.

The functions of different blocks are explained below.

a) MSD Converter: It is used to convert the binary coefficient into its equivalent MSD representation. Two steps involved in MSD conversion are, conversion of binary coefficient to equivalent CSD using CSD algorithm[4] and then a pattern of either $10\bar{1}$ or $\bar{1}01$ is searched next, starting from the MSB and transformed into 011 or $0\bar{1}\bar{1}$ respectively [2] which is the equivalent MSD.

i) CSD Algorithm

$\bar{a}_{-1}=0$
$\gamma_{-1}=0$
$\bar{a}_W=\bar{a}_{W-1}$
For (i=0 to W-1)
{
$\theta_i=\bar{a}_i \oplus \bar{a}_{i-1}$
$\gamma_i=\bar{\gamma}_{i-1}\ \theta_i$
$a_i=(1-2\bar{a}_{i+1})\ \gamma_i$
}
$\bar{a}_{w-1.}\ \bar{a}_{w-2.....}\ \bar{a}_0$---Binary value
$a_{w-1.}\ a_{w-2.....}a_0$-----CSD value
w-----Word length

ii) CSD to MSD Conversion

c(i) = digit position where the transformation is applied to generate the i^{th} MSD representation.
cx = digit position where the search is being done.

Step 1: Convert N into the CSD representation and store in Z. Initialize $m_{new}=1$, $m_{total}=1$, M $\{m_{total}\}$ =Z, m_{set} =Z and c (1) =1.
Step 2: Initialize cx=c $\{m_{new}\}$.
Step 3: If cx > n, increment m_{new} =1. If not, go to Step 5.
Step 4: If $m_{new} = m_{total}$, end.
Step 5: If the digits from positions cx to cx+2 in m_{set} are $10\bar{1}$ or $\bar{1}01$, make a new MSD by transforming $10\bar{1}$ to 011 and $\bar{1}01$ to $0\bar{1}\bar{1}$ respectively.
Step 6:Z gets the new value of the MSD representation. Store this new value in the array M.
Step 7: Increment m_{total} by 1 and cx by 2.
Step 8: Loop Steps 5, 6 and 7 as long as the value of cx is less than the total number of bits.

Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

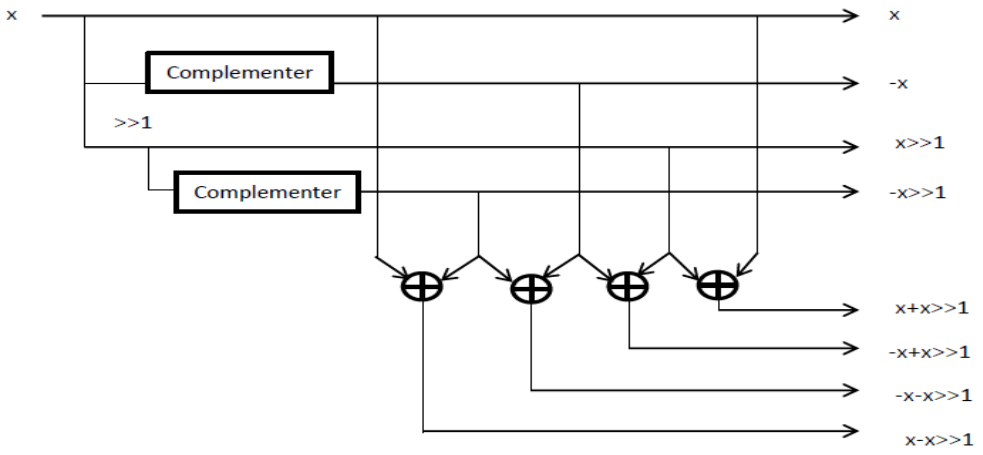


Figure.3 Architecture of shift and add unit

Glossary of terms

M =array of MSD representations.
N = n bit number to find all MSD representations.
Z = stores the current value of MSD representation.
m_{set}= set of MSD representations.
m_{total}= number of MSD representations in m_{set}.
m_{new}=MSD representation where the new one is being searched.

b) Shift and Add unit: One of the efficient ways to reduce the complexity of multiplication operation is to realize it using shift and add operations. Here CSE logic is followed for MCM. Two bits are combined to realize CS in the shift and add unit to reduce the complexity in MCM. The architecture of shift and add unit is shown in figure 3.

The shift and add unit is used to realize all the 2-bit sub expressions for the constant coefficients.8 combinations of inputs are possible for 2 bit sub expressions as MSD stores two bit value along with sign. The input combinations of sign bits (represented as *abcd*) shown in table 1 are encoded to produce (*xyz*) and used as the select signals for multiplexers.

All the possible two bit sub expressions can be realized using two adders. All these sub expressions are then fed to the multiplexer units.

3) *8:1 Multiplexer Unit:* The multiplexer units are used to select the appropriate output from the shift and add unit. The inputs to the multiplexer are 8 outputs from the shift and add unit and 8:1 multiplexer units are employed in architecture. The select signals of the multiplexers are the filter coefficients which are stored in LUT in MSD form. The value bits and their sign bits are fed to the encoder shown in figure 4 and outputs of the encoder are used as the select signal for multiplexer.

Table 1.Encoder output for the value and sign bit combination of MSD represented coefficient.

Two bit sub expression(abcd)				**Encoder output (*xyz*)**
sign bit(*a*)	**value bit(*b*)**	**sign bit(*c*)**	**value bit(*d*)**	
X	0	0	1	000
X	0	1	1	001
0	1	X	0	010
1	1	X	0	011
0	1	0	1	100
0	1	1	1	101
1	1	0	0	110

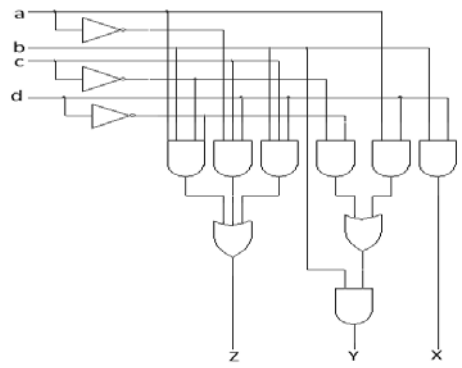


Figure 4. Encoder Circuit

Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

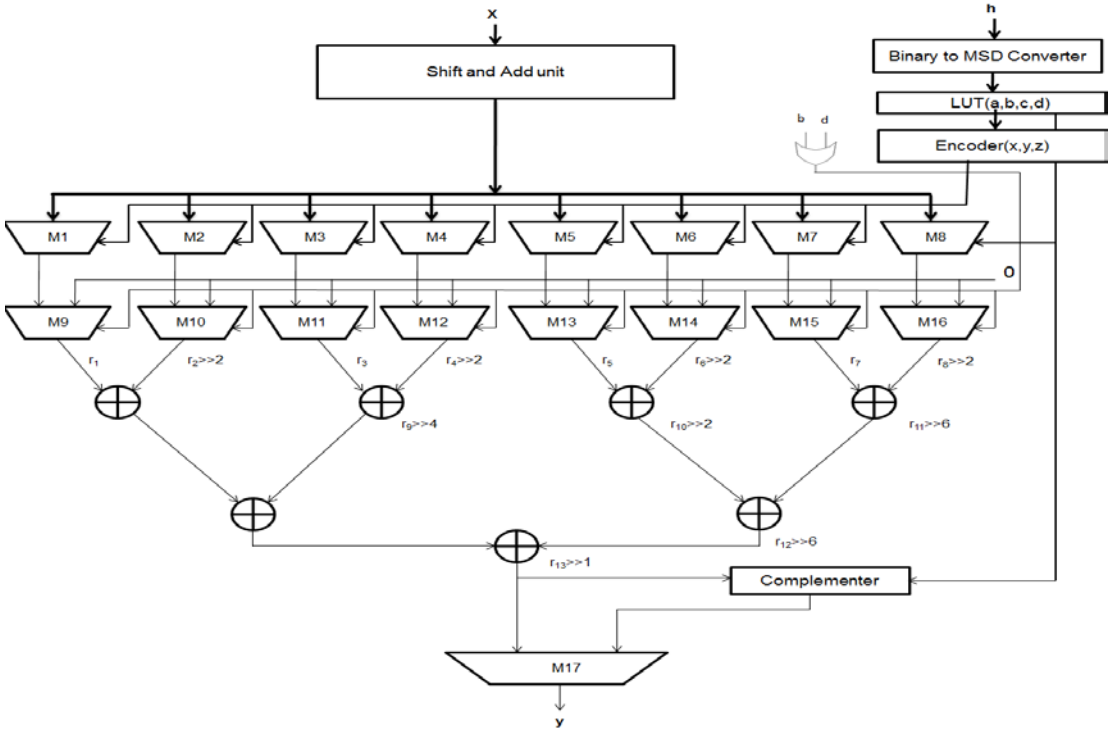


Figure 5.Architecture of MSD- CS shift-and-add multiplier

d) 2:1 multiplexer Unit: The 2:1 multiplexer is used to select either the output of 8:1 multiplexer or zero based on the value bit(*bd*) .The *bd* is ORed and is used as the select signal for this multiplexer. If *bd* = "00" the 2:1 mux selects zero and 8:1 multiplexer output vice versa.

e) Final shifter unit: The final shifter unit will perform the shifting operation after all the intermediate additions are done. This can be illustrated with an example.

Consider the output expression

$$y = 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-6}x - 2^{-7}x + 2^{-14}x + 2^{-15}x \quad --(1)$$

By partitioning into group of two bits from MSB we have

$$y = 2^{-1}(x + 2^{-1}x) + 2^{-3}(x + 2^{-1}x) + 2^{-6}(x - 2^{-1}x) + 2^{-14}(x + 2^{-1}x) \quad --(2)$$

The terms $x + 2^{-1}x$, $x - 2^{-1}x$ can be obtained from the shift and add unit. These terms can be steered inside the architecture by the multiplexer units. The final shifter unit will then perform the shift operations 2^{-1}, 2^{-3}, $2^{-6}$and 2^{-14} of multiplexer outputs.

f) Final selector unit: The final selector unit will select the direct or complemented output of final adder based on the value of the sign bit of constant coefficient. In case of negative number i.e. sign bit is 1, the value after final addition will be complemented, but in case of positive number i.e. sign bit 0, the direct value of the final adder will be passed out of multiplexer.

3.2.1 Architecture design

The architecture is explained with help of an 16-bit co-efficient. Consider 16-bit binary coefficient 0.001011000111101, the equivalent CSD and MSD representation are $0.010\bar{1}\ 0\bar{1}\ 00100\bar{1}\ 0\bar{1}$ and $0.0010110010000\bar{1}\ 0\bar{1}$.

The output based on binary representation is

$$y= 2^{-3}x+2^{-5}x+2^{-6}x+2^{-8}x+2^{-10}x+2^{-11}x+2^{-12}x+2^{-13}x + 2^{-15}x \quad --(3)$$

$$y= 2^{-3}x + 2^{-5}x_1 + 2^{-10}x+2^{-11}x_1+2^{-13}x +2^{-15}x \quad --(4)$$

The output based on MSD representation is

$$y= 2^{-3}x+2^{-5}x+2^{-6}x+2^{-9}x-2^{-13}x -2^{-15}x \quad --(5)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2.Powe,Delay and Area Estimates of proposed MSD-CSE shift-and add multiplier and previous approaches

Shift-and add multiplier	Binary based [15]	CSD based [4]	MSD based [5]	Binary CSE [17]	ours
Switching power (mw)	7.83	7.46	7.31	7.45	6.77
Delay (ns)	63.59	68.18	80.41	61.89	66.55
Gate count	542	1063	1056	462	1098

$$y= 2^{-3}x + 2^{-5}x_1 + 2^{-9}x-2^{-13}x-2^{-15}x \quad --(6)$$

Comparing expressions (4) and (6), it can be seen that the number of addition operations is less for MSD representation than the equivalent binary representation of constants. Hence more number of adders will be idle in case of MSD multiplication, saving significant dynamic power dissipation

Note that the terms $2^{-1}x$, $-2^{-1}x$ can be obtained from the shift and add unit. Then by using 8:1 multiplexer the intermediate sums shown in the bracket can be obtained. The final shifter will perform the shift operations 2^{-6}, 2^{-4}, 2^{-2}. The shifters are hardwired as these shifts are always constant.

The architecture of MSD-CS shift-and-add multiplier is shown in figure 5. The filter coefficient word length is considered as 16-bits and is stored in the 32 bit LUTs. The coefficient is partitioned into groups of 2 bits. As we use MSD representation for coefficients, binary to MSD conversion is done before storing the coefficient in 32 bit LUT. As MSD representation has both positive and negative bits, sign of the respective bits and bit values are stored in separate registers. The values of coefficient bits and sign bits (one for negative bit and zero for positive bit) are fed to an encoder shown in figure 4 and the outputs of encoder are used as select signals for 8:1 multiplexers (M_1-M_7). The OR of value bit *bd* is used as the select signal for 2:1 multiplexer whose inputs are 00 and the 2-bit sub expression from 8:1 multiplexer. The output of 2:1 multiplexers is passed through adder units. The output of final adder and its complement are passed to a 2:1 multiplexer M_{17} whose select signal is the sign bit of constant coefficient. For zero value of sign bit, direct output of final adder and for sign bit is 1, the complemented output of final adder will be routed through the multiplexer M_{17}.

4. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

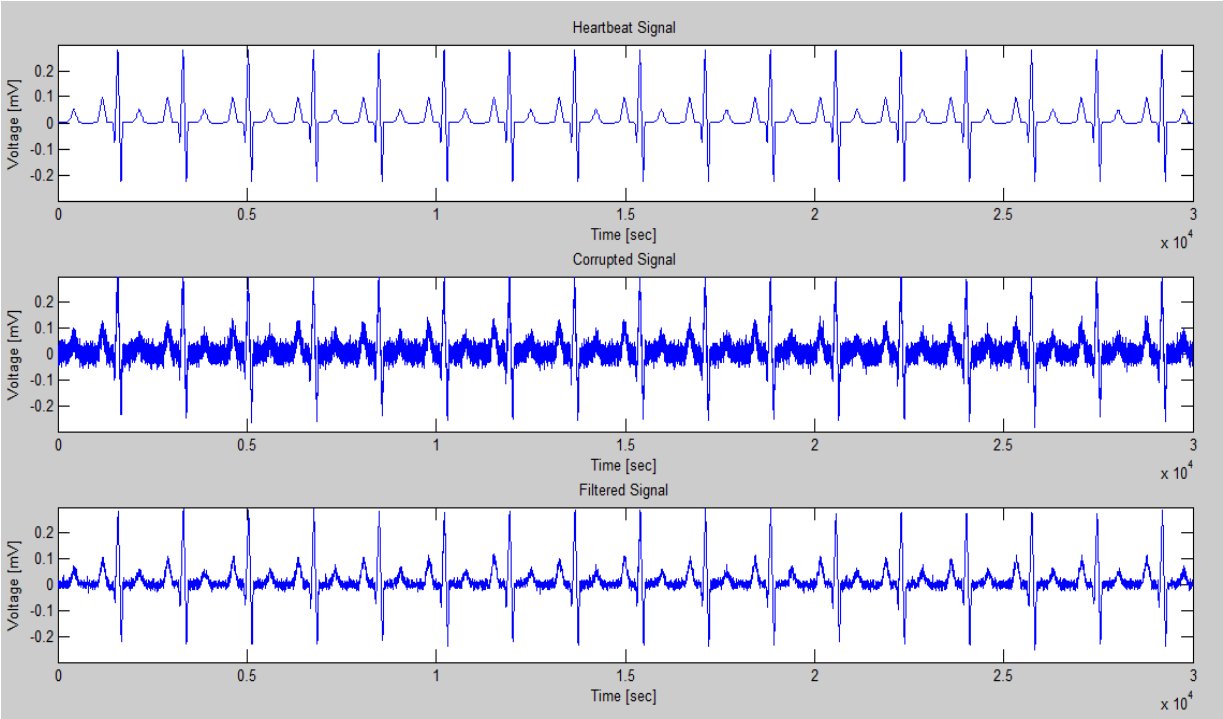
The proposed shift and add based on MSD- CS multiplier is designed using VHDL ,a hardware description language and implemented in ALTERA Quartus II. Shift and add unit is designed as a separate block with multiplexers and adders forming another block. The word length is set to 16 bits. Table 2 shows the power, delay and area estimates of our proposed MSD-CS shift-and add multiplier and previous shift-and add multipliers. It is seen that the proposed MSD-CS shift-and add multiplier demonstrate less power dissipation when compared with binary based, CSD based and standard MSD based shift-and add multipliers. The delay of our proposed multiplier design is 2.4% and 17.2% better when compared to CSD based and standard MSD based multipliers respectively. However our proposed multiplier demonstrates poor delay when compared with binary and binary-CSE based multipliers. This is due to the fewer gates in the critical path of binary based multipliers. Area (represented in terms of gate count) of our proposed MSD-CSE multiplier is high compared to all other approaches used for comparison. This is due to the MSD converter circuit and shift-and add unit used in our proposed multiplier design which contribute for significant gate count. Table 3 in addition shows the power, delay and area estimates of 18 tap FIR filter (both direct –D and Transpose-T form) implemented with proposed multiplier. The results of synthesis are compared with FIR filter systems implemented with conventional, CSD based, MSD based , binary-CSE shift-and- add multipliers and designs in [16]and [17] .It is seen from table 3, that the FIR filter designed using our MSD-CS shift-and-add multiplier demonstrates

Journal of Theoretical and Applied Information Technology
15 August 2012. Vol. 42 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3.Synthesis Results for 18 tap FIR Filter Implementation

Parameters / FIR Architectures		Dynamic power(mw)	T_{co} (critical path delay)(ns)	Number of logic gates
FIR using Binary multiplier	D	152.9	9.90	10299
	T	171.6	9.67	10299
FIR using CSD Multiplier	D	149.1	9.06	19705
	T	164.9	8.64	19674
FIR using MSD multiplier	D	146.6	8.66	19255
	T	159.0	8.88	19112
FIR using [16]	D	160.7	9.93	14485
	T	178.6	9.06	14467
FIR using [17]	D	126.3	9.34	8443
	T	126.3	8.81	7738
FIR using our multiplier	D	113.4	8.59	20206
	T	99.4	8.66	17720



Fiureg. 6. (a) sample of heart beat signal(b)corrupted signal (c)filter output.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

power reduction of more than 5.7% and 12.3% in direct and transpose form respectively when compared with filters implemented using conventional counterparts . The critical delay of FIR filter implemented with our multiplier is better both in direct and transpose form compared to all other approaches used for comparison. The gate count of FIR filter implemented with our multiplier is better in transpose form implementation compared to the filter systems implemented with CSD based and standard MSD based shift-and add multipliers. This is due to the CSE used in our proposed multiplier which generates sub expressions through a single shift and add unit .However the gate count of FIR system designed with our multiplier is high compared to the binary based and binary-CSE based filter systems. This is due to the encoder and converter circuitry used in the proposed multiplier. Also the storage of value bit along with sign bit for
coefficients represented in MSD form necessitates more number of registers thus increasing the gate count in the proposed filter system.

5. IMPLEMENTATION OF THE REALIZED FIR FILTER IN A SIGNAL PROCESSING APPLICATION

To verify the functionality of the proposed MSD-CS multiplier, we have subjected the FIR filter implemented with our multiplier to a sample heart beat signal taken from MATLAB. The sampled signal is corrupted by a noisy signal and the corrupted signal is shown in figure 6(b). The noisy signal is sampled and represented using 16 bits. The filter coefficients are generated using MATLAB and represented using 16 bits. The sampled signal is fed into the our FIR filter system having 18 taps whose output is shown in figure 6(c) . It is seen that the output waveform is symmetrical with the input thus verifying the functionality of our multiplier design.

6. CONCLUSION

A simple approach for multiplication based on MSD representation of coefficients with CSE algorithm is proposed for FIR filters. The multiplier works on the principle of shift of inputs based on the bit values of coefficients and addition of the shifted values. The MSD based shift-and-add multiplier with CSE results in high speed and low power filters since MSD representation of coefficients reduces the number of nonzero bits. Experimental evaluation demonstrated significant improvement in performance in terms of power and delay reduction of filter systems implemented with proposed MSD-CS based shift-and-add multiplier compared with systems implemented with conventional binary and CSD based shift-and-add multipliers. Thus, our proposed multiplier design methodology is a general approach for high speed and low power reconfigurable channel filters.

REFRENCES:

[1] Hartley.R.I," Optimization of canonic signed digit multipliers for filter design", *Proc. IEEE Int. Symp. On Circuits and Systems (ISCAS1991), Singapore*, June11-14, 1991,pp.1992-1995.

[2] In-Cheol Park, Hyeong-Ju Kang, "Digital filter synthesis based on Minimal Signed Digit representation", *Proceedings of the 38th conference on Design automation*, 2001,pp. 468-473.

[3] Jang.YandYang.S,"Low-power CSD linear phase FIR filter structure using vertical common sub-expression",. *Electron. Lett.*, vol. 38, no. 15,2002,pp. 777-779.

[4] Keshab K .Parhi,chapter 13 of "VLSI digital signal processing systems", *Wiley, Inter-Science*,1999,pp.505.

[5] Khalid,H. Abed, Vivek Venugopal and Shailesh,B.Nerurkar, "High Speed Digital Filter Design using Minimal Signed Digit Representation", *IEEE* ,2005.

[6] Mehendale,M, Sherlekar,S.D, and Venkatesh,G. "Synthesis of multiplier-less FIR filters with minimum number of additions",. *In Proc. 1995 IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD 1995), San Jose, CA*, Nov. 5.9, 1999,pp. 668-671.

[7] Pasko,R.,Schaumout,P.,Derudder,V., Vernalde, S., and Durackova,D. "A new algorithm for elimination of common sub expressions",. IEEE Trans. Computer-Aided Design, vol. 18, no. 1,1999, pp. 58-68.

[8] Potkonjak,M., Srivastava,M.B., and Chandrakasan,A., "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common sub expression elimination",. *IEEE Trans. Computer-Aided Design*, vol. 15, no. 2, 1996,pp. 151.165.

[9] Takahashi,Y., Sekine,T., and Yokoyama.M.,"70 MHz multiplier less FIR Hilbert transformer in 0.35 _m standard CMOS library",. *IEICE Trans. Fundamentals*, vol. E90-A, no. 7,2007, pp. 1376-1383.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] Takahashi,Y and Yokoyama,M,"New cost-effective VLSI implementation of multiplierless FIR filter using common sub express ion elimination",. In Proc. ISCAS 2005, Kobe, Japan, pp.845-848.

[11] Vinod,A.P., M-K. Lai.E, Premkumar, A.BandLau,C.T. ,"FIR filter implementation by efficient sharing of horizontal and vertical common sub expressions",. *Electron. Lett.*, vol. 39, no. 2,2003, pp. 251.253.

[12] Vinod,A.P and M-K. Lai.E.,"Comparison of the horizontal and the vertical common sub expression elimination methods for realizing digital filters",. In *Proc. ISCAS 2005*, pp. 496-498.

[13] Vinod.A.P and Vijay.S."A Greedy Common Sub-expression Elimination Algorithm for Implementing FIR Filters with Minimum Adders", Submitted to *Integration the VLSI Journal, Elsevier*.

[14] Yao.C.Y.,Chen.H.H.,Lin.T.F., Chien.C.J and Hsu.C.T, .A novel common-sub expression-elimination method for synthesizing fixed-point FIR filters" ,.*IEEE Trans. Circuits Syst. I*, vol. 51, no. 11, 2004,pp. 2215-2221.

[15] Charles H.Roth Jr "Fundamentals of Logic Design" *Thomson Learning* 2004.

[16] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 617–621, Dec. 2006.

[17] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing filters with low complexity," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 2, 2010pp. 275–288.