ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Design of Memory-Based Realization of FIR Digital Filter

Ms.K.Neeraja
M.Tech, VLSI Design,
Department of ECE.,
Sri Venkateswara College of Engineering and Technology, Chittoor.

Mr. Dr.C.Chandrasekhar, M.Tech, Ph.D
Professsor & HOD,
Department of ECE.,
Sri Venkateswara College of Engineering and Technology, Chittoor.

Abstract:

In the last two decades, many efficient algorithmsand architectures have been introduced for the design of lowcomplexity bit-parallel multiple constant multiplications (MCM)operation which dominates the complexity of many digital signalprocessing systems. On the other hand, little attention has beengiven to the digit-serial MCM design that offers alternative lowcomplexityMCM operations albeit at the cost of an increased-delay. In this paper mainly addressing the problem of optimizing the gate-level area in digit-serial MCM designs and introduce highlevelsynthesis algorithms, design architectures, and a computeraideddesign tool. Experimental results show the efficiency of theproposed optimization algorithms and of the digit-serial MCMarchitectures in the design of digit-serial MCM operations andfinite impulse response filters.

Keywords:

Adders, digit-serial arithmetic, finite impulse response (FIR) filters, gate-level area optimization, multiple constant multiplications (MCM).

I. INTRODUCTION:

FINITE impulse response (FIR) filters are of greatimportance in digital signal processing (DSP) systemssince their characteristics in linear-phase and feed-forwardimplementations make them very useful for building stablehigh- performance filters. The direct and transposed-form FIRfilter implementations are illustrated in Fig. 1(a) and (b),respectively. Although both architectures have similar complexityin hardware, the transposed form is generally preferredbecause of its higher performance and power efficiency [1].The multiplier block of the digital FIR filter in its transposedform [Fig. 1(b)], where the multiplication of filter coefficientswith the filter input is realized, has significant impact onthe complexity and performance of the design because alarge number of constant multiplications are required. Thisis generally known as the multiple constant multiplications(MCM) operation and is also a central operation and performancebottleneck in many other DSP systems such as fastFourier transforms, discrete cosine transforms (DCTs), anderror-correcting codes.

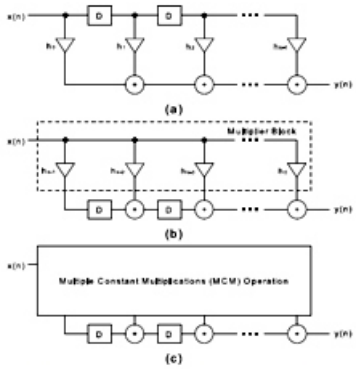


Fig. 1. FIR filter implementations. (a) Direct form. (b) Transposed form with generic multipliers. (c) Transposed form with an MCM block.

Although area, delay, and power efficient multiplier architectures,such as Wallace [2] and modified Booth [3] multipliers,have been proposed, the full flexibility of a multiplieris not necessary for the constant multiplications, since filtercoefficients are fixed and determined beforehand by the DSPalgorithms [4]. Hence, the multiplication of filter coefficientswith the input data is generally implemented under a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

shiftaddsarchitecture [5], where each constant multiplication isrealized using addition/subtraction and shift operations in anMCM operation [Fig. 1(c)].For the shift-adds implementation of constant multiplications,a straightforward method, generally known as digit-basedrecoding [6], initially defines the constants in binary.Then, for each "1" in the binary representation of the constant,according to its bit position, it shifts the variable and addsup the shifted variables to obtain the result. As a simple example, consider the constant multiplications 29x and 43x.Their decompositions in binary are listed as follows:

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

Which requires six addition operations as illustratedin Fig. 2(a).

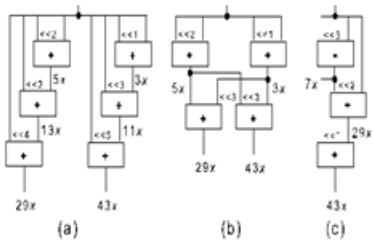


Fig. 2. Shift-adds implementations of 29x and 43x. (a) Without partial product sharing [6] and with partial product sharing. (b) Exact CSE algorithm [9]. (c) Exact GB algorithm [12].

However, the digit-based recoding technique does notexploit the sharing of common partial products, which allows great reductions in the number of operations consequently,in area and power dissipation of the MCM design at the gate level. Hence, the fundamental optimization problem,called the MCM problem, is defined as finding the minimumnumber of addition and subtraction operations that implement the constant multiplications. Note that, in bit-parallel designof constant multiplications, shifts can be realized using only-wires in hardware without representing any area cost.

The algorithms designed for the MCM problem can becategorized in two classes: common sub-expression elimination(CSE) algorithms [7]–[9] and graph-based (GB) techniques[10]–[12]. The CSE algorithms initially extract all possiblesubexpressions from the representations of the constantswhen they are defined under binary, canonical signed digit(CSD) [7], or minimal signed digit (MSD) [8]. Then, they findthe "best" subexpression, generally the most common, to beshared among the constant multiplications.

The GB methodsare notlimited to any particular number representation andconsider a larger number of alternative implementations of a constant, yielding better solutions than the CSE algorithms, asshown in [11] and [12]. Returning to example in Fig. 2, the exact CSE algorithmof [9] gives a solution with four operations by finding the mostcommon partial products 3x = (11)bin x and 5x = (101)binx when constants are defined under binary, as illustrated inFig. 2(b). On the other hand, the exact GB algorithm [12]finds a solution with the minimum number of operations bysharing the common partial product 7x in both multiplications,as shown in Fig. 2(c). Note that the partial product 7x =(111)binx cannot be extracted from the binary representationof 43x in the exact CSE algorithm [9].

However, all these algorithms assume that the input datax is processed in parallel. On the other hand, in digit-serial arithmetic, the data words are divided into digit sets, consisting of d bits that are processed one at a time [13]. Since digitserialoperators occupy less area and are independent of thedata wordlength, digit-serial architectures offer alternative lowcomplexitydesigns when compared to bit-parallel architectures.However, the shifts require the use of D flip-flops, as opposed to the bit-parallel MCM design where they are freein terms of hardware. Hence, the high-level algorithms should take into account the sharing of shift operations as well as thesharing of addition/subtraction operations in digit-serial MCMdesign. Furthermore, finding the minimum number of operationsrealizing an MCM operation does not always yield anMCM design with optimal area at the gate level [14]. Hence,the high-level algorithms should consider the implementationcost of each digit-serial operation at the gate level.

II. LUT DESIGN FOR MEMORY-BASED MULTIPLICATION:

The basic principle of memory-based multiplication is depictedin Fig. 2. Let be a fixed coefficient and be an inputword to be multiplied with. If we assume to be an unsignedbinary number of word-length, there can be possiblevalues of, and accordingly, there can be possiblevalues of product. Therefore, for the conventional implementationof memory-based multiplication, a memoryunit of words is required to be used as look-up-table consistingof pre-computed product values corresponding to all possiblevalues of.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The product-word, for, is stored at the memory location whose address is thesame as the binary value of, such that if bit binary valueof is used as address for the memory-unit, then the corresponding product value is read-out from the memory.Although possible values of correspond to possiblevalues of, recently we have shown that onlywords corresponding to the odd multiples of may only bestored in the LUT. One of the possible product words iszero, while all the rest are even multiples of A whichcould be derived by left-shift operations of one of the odd multiplesof A.

Address $d_2d_1d_0$	Word symbol	Stored value	Input $x_3x_2x_1x_0$	Product value	# of shifts	Control s_1s_0
000	P0	A	0001	A	0	00
			0010	2^1×A	1	01
			0100	2^2×A	2	10
			1000	2^3×A	3	11
001	P1	3A	0011	3A	0	00
			0110	2^1×3A	1	01
			1100	2^2×3A	2	10
010	P2	5A	0101	5A	0	00
			1010	2^1×5A	1	01
011	P3	7A	0111	7A	0	00
			1110	2^1×7A	1	10
100	P4	9A	1001	9A	0	00
101	P5	11A	1011	11A	0	00
110	P6	13A	1101	13A	0	00
111	P7	15A	1111	15A	0	00

Table 1: Look up table

We illustrate this in Table I for at eightmemory locations, eight odd multiples are storedare derived by left-shift operations of. Similarly, and are derived by left-shifting, while and are derived by left-shifting and, respectively. The address corresponds to, which can be obtained by resetting the LUT output. For an input multiplicandof word-size similarly, only odd multiple values needto be stored in the memory-core of the LUT, while the othernon-zero values could be derived by left-shift operationsof the stored values. Based on the above, an LUT forthe multiplication of a bit input with bit coefficient is designedby the following strategy:

- A memory-unit of words of bit width is used to store all the odd multiples of A.
- A barrel-shifter for producing a maximum of left shifts is used to derive all the even multiples of A.
- The bit inputword is mapped to bit LUT address by an encoder.

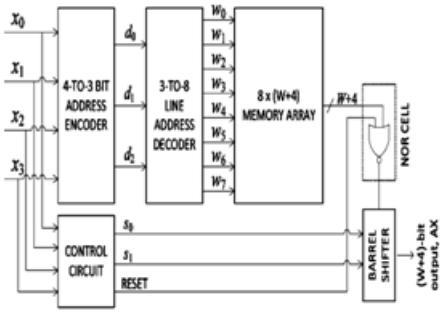


Fig (a):LUT multiplier

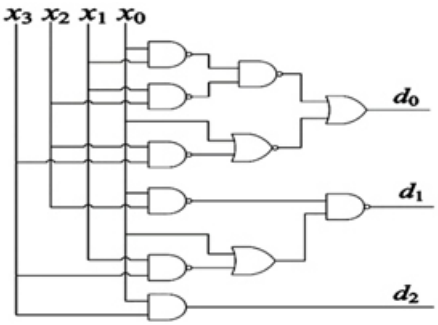


Fig (b):4 to 3 bit encoder

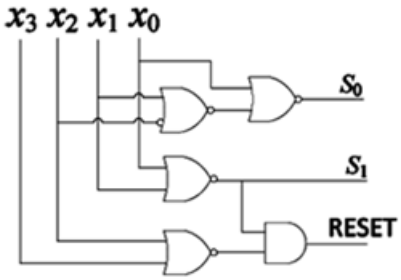


Fig (c): Control circuit

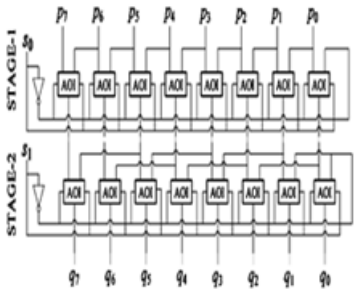
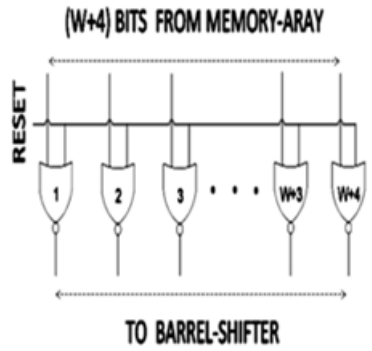


Fig (d): Two stage logarithmic barrel-shifter

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



Fig(e): NOR cell

• The control-bits for the barrel-shifter are derived by a control-circuit to perform the necessary shifts of the LUT output.Besides, a RESET signal is generated by the samecontrol circuit to reset the LUT output.

III. SIMULATION:

System-level testing may be performed with the ModelSim logic simulator.Simulation is done through the ModelSim 6.3g_p1 for the FIR filter implantation as shown below.

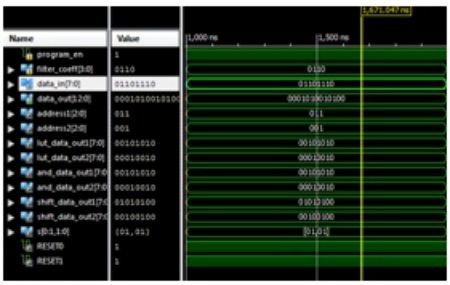


Fig: Simulation results

IV.SYNTHESIS REPORT:

Xilinx's patented algorithms for synthesis allow designs to run upto 30% faster than competing programs, and allows greater logic density which reduces project costs.The synthesis report can be done through the Xilinx 9.2 to know the area delay.

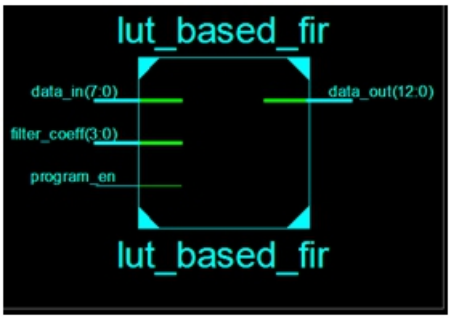


Fig:RTL schematic

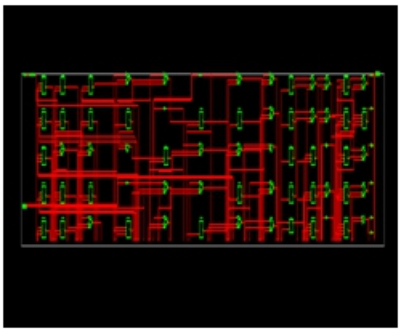


Fig:Technology schematic

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	31	4656	1%
Number of Slice Flip Flops	45	9312	0%
Number of 4 input LUTs	155	9312	1%
Number of bonded IOBs	25	232	10%
Number of GCLKs	1	24	4%

Fig: Design summary

Total delay
15.662ns(10.839nslogic,4.823ns route)
(69.2% logic, 30.8% route)

Fig:Delay calculation

V.CONCLUSION:

New approaches to LUT-based-multiplication are suggestedto reduce the LUT-size over that of conventional design.By odd-multiple-storage scheme, for address-length 4, theLUT size is reduced to half by using a two-stage logarithmic barrel-shifter and number of NOR gates, whereis the word-length of the fixed multiplying coefficients.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Three memory-based structures having unit throughput rate aredesigned further for the implementation of FIR filter. One ofthe structures is based on DA principle, and the other two arebased on LUT-based multiplier using the conventional and the proposed LUT designs. All the structures are found to havethe same or nearly the same cycle periods, which depend onthe implementation of adders, the word-length and the filterorder. The conventional LUT-multiplier-based filter has nearlythe same memory requirement and the same number of adders,and less number of input registers than the DA-based design at the cost of higher adder-widths.

REFERENCES:

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles,Algorithms and Applications. Upper Saddle River, NJ: Prentice-Hall, 1996.

[2] G. Mirchandani, R. L. Zinser Jr., and J. B. Evans, "A new adaptivenoise cancellation scheme in the presence of crosstalk [speech signals],"IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.,vol. 39, no. 10, pp. 681–694, Oct. 1995.

[3] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering andvariable gain ranging seismic data acquisition system," in Proc. IEEESoutheastcon'93, Apr. 1993,p.6.

[4] K. K. Parhi, VLSI Digital Signal Procesing Systems: Design and Implementation.New York: Wiley, 1999.

[5] H. H. Kha, H. D. Tuan, B.-N. Vo, and T. Q. Nguyen, "Symmetric orthogonalcomplex-valued filter bank design by semi-definite programming,"IEEE Trans. Signal Process., vol. 55, no. 9, pp. 4405–4414, Sep.2007.

[6] H. H. Dam, A. Cantoni, K. L. Teo, and S. Nordholm, "FIR variable digitalfilter with signed power-of-two coefficients," IEEE Trans. CircuitsSyst. I, Reg. Papers, vol. 54, no. 6, pp. 1348–1357, Jun. 2007.

[7] R. Mahesh and A. P. Vinod, "A new common sub-expression elimination algorithm for realizing low-complexity higher order digital filters,"IEEE Trans. Computer-Aided Ded. Integr. Circuits Syst., vol. 27, no.2, pp. 217–229, Feb. 2008.

[8] International Technology Roadmap for Semiconductors, [Online].Available: http://public.itrs.net/

[9] B. Prince, "Trends in scaled and nanotechnology memories," in Proc.IEEE Conf. Custom Integr. Circuits, Nov. 2005, pp. 7.

[10] K. Itoh, S. Kimura, and T. Sakata, "VLSI memory technology: Currentstatus and future trends," in Proc. 25th Eur. Solid-State Circuits Conference,{ESSCIRC'99, Sep. 1999, pp. 3–10.

[11] T. Furuyama, "Trends and challenges of large scale embedded memories,"in Proc. IEEE Conf. Custom Integrated Circuits, Oct. 2004, pp.449–456.

[12] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R.Mckenzie, "Computational RAM: Implementing processors inmemory," IEEE Trans. Design Test Comput., vol. 16, no. 1, pp. 32–41,Jan. 1999.

[13] H.-R. Lee, C.-W. Jen, and C.-M. Liu, "On the design automation ofthe memory-based VLSI architectures for FIR filters," IEEE Trans.Consum. Electron., vol. 39, no. 3, pp. 619–629, Aug. 1993.

[14] S. A. White, "Applications of the distributed arithmetic to digital signalprocessing:Atutorial review," IEEE ASSP Mag., vol. 6, no. 3, pp. 5–19,Jul. 1989.

[15] M. Mehendale, S. D. Sherlekar, and G.Venkatesh, "Area-delay tradeoffin distributed arithmetic based implementation of FIR filters," in Proc.10th Int. Conf. VLSI Design, Jan. 1997, pp. 124–129.

[16] H. Yoo and D. V. Anderson, "Hardware-efficient distributed arithmeticarchitecture for high-order digital filters," in Proc. IEEE Int. Conf.Acoustics, Speech, Signal Processing, (ICASSP'05), Mar. 2005, vol.5, pp. v/125–v/128.

[17] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson,"LMS adaptive filters using distributed arithmetic for highthroughput," IEEE Trans. Circuits Systems I, Reg. Papers, vol. 52, no 7, pp. 1327–1337, Jul.2009