Vol 06, Article 01532; February 2015
http://ijves.com

International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

EFFICIENT POWER CONSUMPTION AND COMPLEXITY EFFECTIVE DESIGN OF NARROWBAND USING FRM

N.SUBBULAKSHMI[#1]
[#1]ECE, [#1]AssistantProfessor ECE, Sri Ramakrishna Engineering College, Coimbatore-22, India
[1]lakshu.125@gmail.com

ABSTRACT

To reduce the complexity of narrow transition bandwidth of FIR filter through Frequency Response Masking Technique (FRM) and to achieve low power with the help of time multiplexed FRM filters. This technique uses a cascade structure of periodic model filter and masking filter. The complexity is a reduced both by the number of multipliers and also the logic elements used which results in low power consumption. This design is implemented on FPGAs.

Keywords— FRM, Time-multiplexing, FPGAs

I. INTRODUCTION

Finite-length Impulse Response (FIR) filters are digital filters whose impulse response is of finite length [1].The difference equation that defines the output of an FIR filter of length L is:

$$p(n) = \sum_{z=0}^{L-1} q(z) r(n-z) \tag{1}$$

Where $p(n)$ is output sequence, $q(z)$ are the coefficients, $r(n)$ is the input sequence

The FIR filter complexity depends on number of multiplications, based on equation (1) it is proportional to filter length. For linear phase Singe Stage Filters (SSF) filter length is proportional to the inverse of the width of the transition band. The equation (2) shows that narrow transition bands would results in large filter length [1]

$$L \approx -\frac{2}{3}\log(10\gamma_c\gamma_s)\frac{2\pi}{\rho_s T - \rho_c T} + 1 \tag{2}$$

Where $\gamma_c, \gamma_s, \rho_s T, \rho_c T$ indicates pass band ripple, stop band ripple, pass band edge and stop band edge respectively. Complexity of FIR filters can be reduced by frequency response masking techniques. This technique involves two filters in cascade: periodic model filter and masking filter. The impulse response of the periodic model filter is interpolated with the factor K and its transfer function is written as $H(Z^K)$[1]-[4]

Field Programmable Gate Arrays (FPGAs) is a platform where signal processing algorithm can be implemented efficiently.

In the case where sample rate is lower than maximum obtainable clock frequency time-multiplexing is the efficient way to use the FPGA resources which results in reduction of number of multipliers needed. This in turn reduces the complexity when combined with sparseness

The FRM filter designs and the related structures have received considerable attention [1]-[4].reference [5] studies multiplier less narrow-band frequency response masking filters with fixed tap count, in [6] it works on reducing memory fetches between the FPGA and external memory, while in [7] the authors compare fully parallel FRM filters with conventional, sharp FIR filters developed using Xilinx core generator tool. The authors in [8] compare the impact of different sparsity factors and placement of zeros on FPGA utilization while implementing a 200-order fully parallel FIR filter. Furthermore, the effect of time multiplexing and Sparseness of a periodic model filter was studied in [9]

II. FREQUENCY RESPONSE MASKING

Frequency Response Masking is used for realizing filter with very narrow transition bands. The required structure has a cascade of two filters such as narrow band filters and wide band filters. Narrow band filters is shown in fig. Wide band filters can be done by computing the complementary output of a narrow band filter. The periodic model filters possess a frequency response with many pass bands in which only one required. The masking filter masks the unwanted passbands.This structure is known as interpolated FIR (IFIR) filter [4].

In order to attain a narrow transition band FIR filter with stop band and pass band edges at ρ_ST andρ_CT ,the model filter should have pass band and stop band edges at S ρ_ST and Sρ_CT .It is sampled by a factor S this will provide periodic images, which is masked by a masking filter with pass band and stop band edges at ρ_CT and $^{2\pi}/_{F} - \rho_s T$.The required magnetic responses of the model, periodic model, masking and overall filter is shown

Vol 06, Article 01532; February 2015
http://ijves.com

International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

in fig for a narrow band filter. The cascaded structure of these two filters decreases the multiplier count with the cost of an increased number of delay elements .By inserting the zeros in the model filter, the filter order will get increases but the arithmetic complexity of the model filter decreases as L is increased but that of the masking filter increases.

III. IMPLEMENTING FILTERS ON FPGAs

Generally FPGA which is programmable hardware are programmed using hardware description language (HDLs) and can be able to implement any given logic function. Look up table is nothing but a basic building block of an FPGA, which can be used to perform complex combinational functions or simple logic gates. The combination of look up table is to form a larger block which might consists of a multiplexer, flip flop and even a carry chain. Current state of the art FPGAs with general purpose LUTs and registers, a number of dedicated blocks is used for different specialized functions. FPGAs such as Xilinx virtex series and Altera stratix series have dedicated blocks known as DSP blocks, for implementing multipliers, multiply accumulates(MACs) and multiply adds(MADs).These DSP blocks are very efficient in doing the convolution operation which is at the center of filter operation .FPGAs have resources is mainly used for implementing memories .there are two types such as dedicated memory blocks(known as block RAMs) and memories provided by LUTs (known as distributed RAMs).the DSP and memory blocks combination is used for efficient map frequency response masking filter architectures to FPGAs.

IV. DESIGN METHODOLOY

We proposed an architecture for time-multiplexed periodic model filters and it is used to achieve low resource utilization when compared to vendor provided time-multiplexed FIR cores were not able to fully utilize the scattered resource. In [9] two architectures were proposed: a none pipelined and a pipelined one, along with an adjustment for odd filter length. For brevity, we will only describe the pipelined architecture for odd filter lengths.

Before discussing about the architecture, some variables are defined. Let $Y_{G,}$,$Y_F$and L denote the filter length of the model filter ,filter length of the model filter respectively. Then Y_{GL} =Y_G. L- L+1 would denote the length of the periodic model filter and F denotes the time-multiplexing factor.There are T-1 cycles between each input of time-multiplexed filters. That implies that the current and previous inputs must be saved in memories. Each of such memories depth is given by V_{dm} =T.While considering the coefficients, instead of one coefficient per multiplier, there are T coefficients in a ROM [9].Coefficient symmetry used by linear phase FIR filters is utilized to further reduces the multiplier count.

A. Architecture –Periodic Model Filter

The design methodology for periodic model filter in [9] is as follows: an array of ROMs holds and an array of ROMs store data. The DSP blocks implement both the convolution function and accumulation as in general, DSP blocks can also support fast accumulation [10].

TABLE 1 EQUATIONS FOR TIME MULTIPLEXED FILTERS [9]

DESCRIPTION	EQUATION
Data Memory Count(R_{dm})	$\frac{Y_{GL}-1}{F*L}$
Coefficient memory count	$\frac{Y_{GL}-1}{F*L*2}$
DSP count	$[\frac{Y_{GL}-1}{T*L*2}]+1$
Data memory count (V_{dm})	$F*L$
Middle memory depth	$\frac{Y_{GL-L*F*(R_{dm}-2)}}{2}$

The pipelined version of this architecture is shown in fig.1 for a 4T+1-tap filter, where the arrows on the data memory indicate the transfer of data between memories.

For the data memories, we use distributed RAMs instead of block RAMs because of the short length required. Each data memory should simultaneously support 1 write and 2 reads. For this operation, one dedicated read port to read the data and one shared read/write port are used to transfer data between memories and write new data. Here each memory is implemented as a circular buffer, where write and read pointers are used to control the read and write operations. To deed symmetry, the data memory is divided into two halves, where one symmetry from each half is combined to form one memory set. Since only non-zero coefficients are stored in the coefficient ROM, the depth is more and data memories less, when compared to a non-sparse filter of same length. To read data, the read counter is incremented by a factor of L to match the coefficient index.

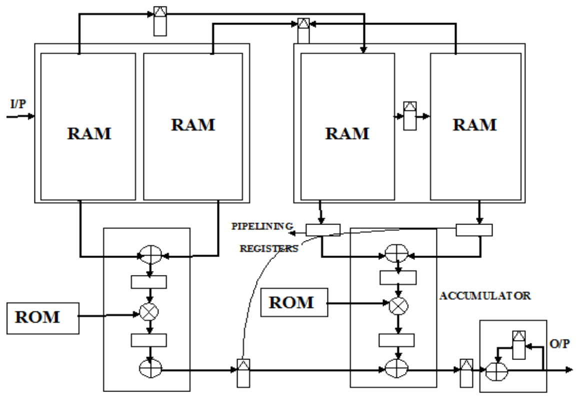

Fig.1 pipelined time-multiplexed architecture for periodic model filters [9]

For pipelining, k pipeline registers are added after the memory set in the memory set M_K, except for the first memory. This is done to balance the pipelining register used at the outputs of tht DSP block. When L is odd one extra middle tap is handled by one extra flip flop between the two halves of the data memory array. This middle tap along with the middle coefficient is fed to the multiplier of the DSP block used as accumulator. This arrangement is shown in fig.2.To properly pipeline this tap,m registers are added after the middle tap in the pipelined architecture with m coefficient memories

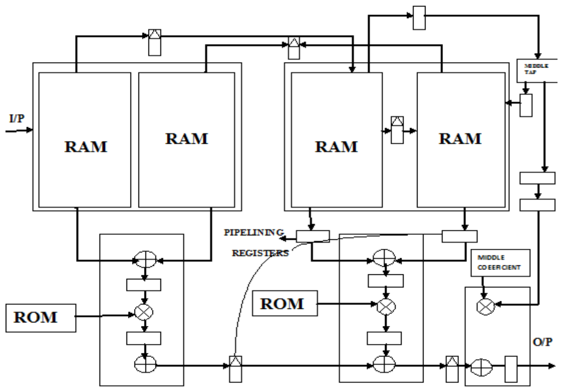

Fig 2 time-multiplexed using the final accumulator DSP block

B.Masking filter

The masking filter has two ways. The first way is to use the same architecture as for the periodic model filter with sparsity factor of one. The other way is to use the vendor provided FIR core as it is more optimized, especially in terms of speed. The intention of this paper is to compare time-multiplexed SSF with FRM filter mainly in terms of power and resource utilization. The masking filter does not have any zero co-efficients; the FIR core is cascaded with design of the periodic model filter. The FIR core is based on the direct form is closely related with our own architecture. The only difference is the data storage.

V. RESULTS

The time multiplexed architecture is implemented using Matlab simulink and results are obtained

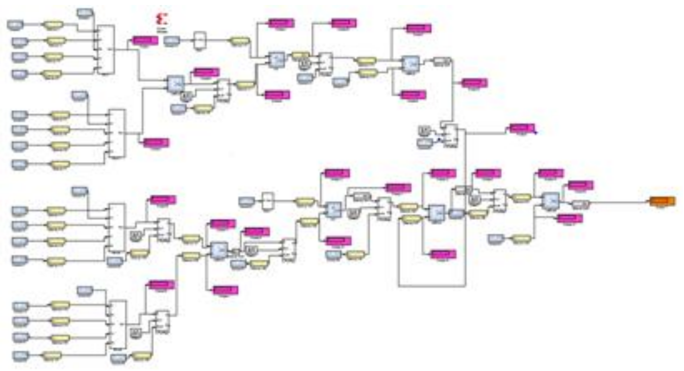

Fig 3. Time multiplexed architecture implemented in Matlab simulink

Optimization of the filters is possible using advanced techniques which could further reduce filters complexity

FUTURE WORKS

By implementing the design in FPGAs power consumption can be further reduced by optimising the design by reducing the number of multipliers and also the logic elements used.

CONCLUSION

In this paper, low power and complexity effective FIR filter is designed using FRM techniques .The low power is achieved by the use of low resources. The time multiplexed architecture provides low resource utilisation which leads to low power consumption.

REFERENCES

[1] S.A. Alam and O.Gustafsson, "*Implementation of Narrow-Band Frequency Response Masking for Efficient Narrow Transition Band FIR Filters on FPGAs*," Norchip, November 2011

[2] S. K. Mitra, Digital Signal Processing, University of California, Santa Barbara: TATA McGraw-Hill, 2006

[3] Y-C. Lim, "Frequency-*Response Masking approach for the synthesis of sharp linear phase digital filters*," IEEE Transactions on Circuits and Systems, volume 33, no. 4, pp. 357-364, 1996

[4] T. Saramiiki, "*Finite Impulse Response Filter Design*," in Handbook for Digital Signal Processing, S. K. Mitra and J. Kaiser, Eds. Wiley-Inter science, 1988, pp. 155-277

[5] T. Saramiiki, T. Neuvo, and S. K. Mitra, "*Design of Computationally Efficient Interpolated FIR filters*," IEEE Transactions on Circuits and Systems, volume 35,no. I, pp. 70-88, 1998

[6] Y. Lian, "*FPGA Implementation of High Speed Multiplierless Frequency Response Masking FIR filters*," in Proceedings of IEEE Workshop on Signal Processing System, Lafayette, LA, 2000, pp. 317-325

[7] Y C. Lim, Y J. Yu, H. Q. Zheng, and S. W. Foo, "*FPGA Implementation of Digital Filters Synthesized using the Frequency-Response Masking Technique*," in Proceedings of IEEE International Symposium on Circuits and Systems, volume 2, Sydney, NSW ,May 2001, pp. 173-176

[8] S. Li and J. Zhang, "*Efficient FPGA Implementation of Sharp FIR Filters using the FRM technique*," IEICE Electronics Express, volume 6, no. 23,pp. 1656-1662, December 2009

[9] S. G. Patronis and L. S. De Brunner, "*Sparse FIR Filters and the Impact on FPGA Area Usage*," in Proceedings of Asilomar Conference on Signals Systems and Computers , Pacific Grove, CA, October 2008, pp. 1862-1866

[10] S. A. Alam and O. Gustafsson, "*Implementation of Time-Multiplexed Sparse Periodic FIR Filters for FRM on FPGAs*," in Proceedings of IEEE International Symposium on Circuits and Systems, Rio de Janeiro, Brazil, May 2011

[11] Xilinx, Virtex-6 FPGA DSP48Ei Slice User Guide, Sep. 2009. [Online].
Available: http://www.xilinx.com/support/documentation/virtex-6.htm

[12] FIR LogiCORE IP FIR Compiler v5.0, 2009. [Online]. Available:
http://www.xilinx.comlsupport/documentation/ipdsp_filter.htm

[13] Y Neuvo, D. Cheng-Yu, and S. Mitra, "Interpolated Finite Impulse Response filters," IEEE Transactions Acoustics Speech Signal Processing, vol. 32, pp. 563-570, May 1984