Ternary and Multi-Bit FIR Filter Area-Performance Tradeoffs in FPGA

KHALIL-UR-RAHMAN DAYO*, AND TAYAB DIN MEMON**

RECEIVED ON 27.08.2012 ACCEPTED ON 31.12.2012

ABSTRACT

In this paper, performance and area of conventional FIR (Finite Impulse Responce) filters versus ternary sigma delta modulated FIR filter is compared in FPGA (Field Programmable Gate Arrays) using VHDL (Verilog Description Language). Two different approaches were designed and synthesized at same spectral performance by obtaining a TIR (Target Impulse Response). Both filters were synthesized on adaptive LUT (Look Up Table) FPGA device in pipelined and non-pipelined modes. It is shown that the Ternary FIR filter occupies approximately the same area as the corresponding multi-bit filter, but for a given specification, the ternary FIR filter has 32% better performance in non-pipelined and 72% in pipelined mode, compared to its equivalent Multi-Bit filter at its optimum 12-bit coefficient quantization. These promising results shows that ternary logic based (i.e. +1,0,-1) filters can be used for huge chip area savings and higher performance.

Key Words: **Ternary, VHDL, FPGA, Sigma-Delta Modulation.**

1. INTRODUCTION

Single-bit ternary FIR filter has been designed, developed and tested in MATLAB and proved to be useful for various applications with an advantage of simple arithmetic for complex DSP (Digital Signal Processing) applications in [1]. Further to this, successful simulation and comparison of LMS (Least Mean Square) like single-bit adaptive algorithm using sigma delta modulation with conventional block LMS algorithms in Matlab® has been reported in [2-3]. LMS-like single-bit adaptive algorithm has been compared using single-bit, ternary and two-bit techniques with conventional adaptive algorithms and shown that these algorithms are able to converge and improve the SNR to certain level.

Despite that, there have been very little direct comparisons between single-bit and conventional techniques in FPGA. The simple arithmetic of ternary filters results in an efficient hardware implementation and can easily be mapped to simple LUT structures within a FPGA. On the other hand, much work has been reported on the efficient hardware implementation of multi-bit FIR filters in FPGAs by using various complex multiplication reduction techniques [4-6]. In this work, we have extended the theoretical work reported in [7-8]. An optimized algorithm has been developed for partial products that lead towards final sum in multi-bit format (described in Section 3 in detail). Ternary and multi-bit FIR filters have been compared based on the same spectral performance.

* Associate Professor, and ** Assistant Professor,
Department of Electronic Engineering, Mehran University of Engineering & Technology, Jamshoro.

2. TERNARY FIR FILTER

While the overall architecture of the ternary FIR filter is same as contemporary multi-bit FIR Filter (Fig. 1), the taps are constrained to the set {+1/0/-1}. The ternary filter output *q(k)* is given as the convolution of bit-stream input *v(k)* and ternary taps *w(k)*:

$$q(k) = \sum_{j=0}^{N} w_j i_{k-j} \tag{1}$$

where N is the order of the filter (number of taps) and w represents the ternary FIR filter coefficients{+1/0/-1}.

In this filter design, we have used sigma-delta modulator to generate ternary taps. The most stable sigma-delta modulator is second order that has been adopted in this filter design as reported in [8-9]. Ternary FIR filter should be generated in MATLAB requires adopting any suitable TIR using optimized algorithms for example least square, equiripple, windowing etc. Remez Exchange Algorithm was used for the generation of TIR of ternary FIR filters. The TIR was generated by Remez exchange algorithm. The passband and stopband attenuation of TIR were set as 0.2 and 0.25 respectively with 55dB of stopband attenuation (Fig. 2).

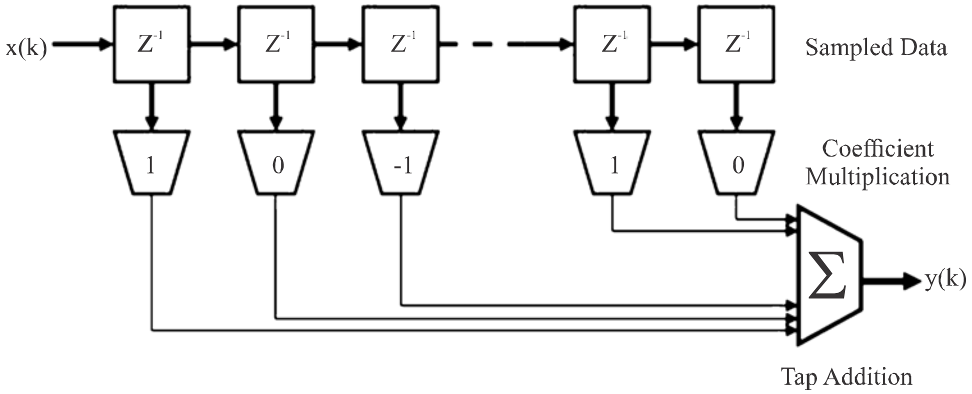

FIG. 1. BLOCK DIAGRAM OF TERNARY FIR FILTER (ADOPTED FROM)

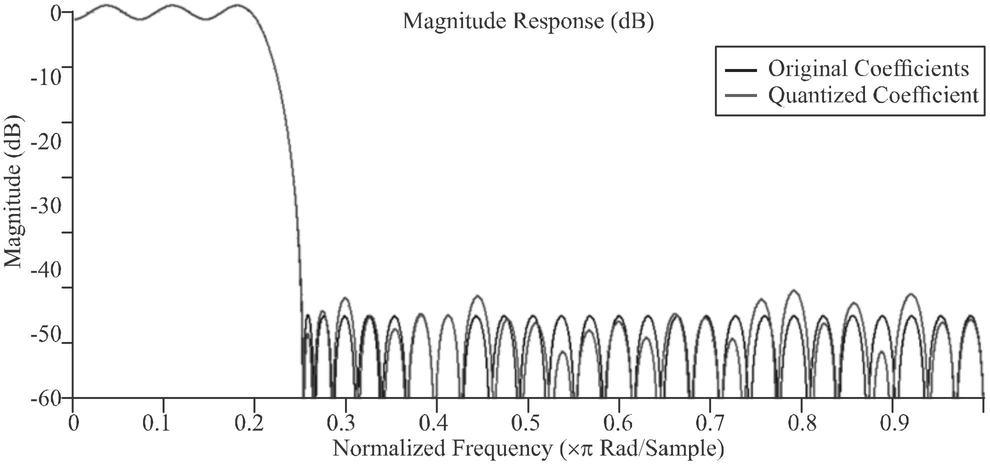

FIG. 2. TARGET IMPULSE RESPONSE WITH OPTIMUM COEFFICIENTS

3. FPGA BASED DESIGN AND ANALYSIS OF PROPOSED FILTER

As shown in Fig. 1, architecture of ternary FIR filter comprises of MCA (Multiply and Accumulates) section i.e. multiplication of ternary taps with bit-stream input that is followed by addition of partial products. The greater noise shaping effect offered by sigma-delta modulator is achieved by using high clock frequency (called OSR (Over Sampling Ratio)) than Nyquist rate. Hence, generation of ternary taps through M requires more taps (i.e. multiplication of higher sampling rate) than Nyquist rate. For example - a FIR filter with 16 multi-bit coefficients will require 512 ternary coefficients at oversampling ratio of 32. We have approached to divide this large coefficients number into taps multiplication blocks followed by an adder tree. Optimized adder tree levels are added by relationship for final accumulation. Much work has been reported on area-performance tradeoffs of adder tree in last few decades. In [5], a FIR filter implementation was described that take the form of a partially systolic array using DA (Distributed Arithmetic). Similarly, [10] has compared the conventional multiplication, distributed arithmetic and Add-and-shift methods for improved performance implementation in a FPGA. A fully pipelined, fully parallel, transposed-form FIR filter implementation has been reported in [4]. To achieve better performance with a smaller number of LUTs, we have focussed on techniques that map efficiently onto FPGA organizations but that are also suitable for ASIC (Application Specific Integrated Circuits) implementation.

Ternary FIR filter with bit-stream input can be mapped on FPGA hardware by using any suitable encoding technique for example - 2's complement, CSD (Canonical Signed Digit) encoding, RBSD (Redundant Binary Signed Digit) etc. In this design, we have taken more general approach and chosen 2's complement format to represent ternary format of coefficients and bit-stream input. By employing 2's complement approach, ternary FIR filter design in FPGA can easily be implemented by simple AND-OR logic that can be mapped on single LUT in most of the FPGA devices [4-5,10]. Hence, overall reduction in conventional multi-bit parallel multiplier complexity mandates more flexibility and reduces chip area that is highly beneficial in ASIC design. In binary adder tree structure (Fig. 3) partial product to final sum stages depends upon the number of ternary coefficients. For example will requires nine stages and a final multi-bit output of . Therefore, overall 11 bits are required to express the full range of 512 ternary coefficients.

Ternary FIR filter design and implementation in VHDL has been achieved as shown in Fig. 3. Adder tree is designed as it reduces adder blocks at each successive level and finally results a multi-bit output. In other words, it expands as we go from bottom to top level. As described above for example we have 512 coefficients, so the number of adder levels will be 9, and the initial adder levels that should be in the range of 2-8 bits can be mapped on single LUT, while rest of the adder stages would need more LUTs to map higher input bits. Obviously, ternary FIR filter design can be improved by using optimized IP blocks but we have taken very general approach as worst-cast scenario may be obtained.

3.1 Conventional FIR Filter in VHDL

There is no essential architectural difference between ternary and conventional filters, the same design method can be used. In a conventional FIR filter, the input and coefficients will be multi-bit rather that single-bit. As a floating point coefficients generated in MATLAB are often difficult to implement in hardware. So an optimum fixed-point coefficient's quantization is realized as reported in [11]. According to the specification of TIR (as discussed in Section 2) and shown in Fig. 2, the 12-bit coefficients quantization would be the optimum solution [2]. For simplicity, the multibit input is assumed as a 12-bit signal.

4. SIMULATION RESULTS AND DISCUSSION

The simulation results of ternary and conventional FIR filters, coded in VHDL and compiled, simulated and synthesized on Stratix-III EP3C120F484C7 FPGA, are given in Table 1. The two filters have been compared based on similar spectral performance as reported in [12]. The specified TIR required 55dB SNR, which can be achieved by 12-bit coefficients quantization [11]. Similarly, 55-dB SQNR with a second order M can be achieved at an OSR of 32 [13]. The above said performance can be achieved by Remez Exchange algorithm with an optimum order of 63. Both the filters were synthesized in pipelined and non-pipelined mode. In pipelined mode, extra registers were used between the adder stages that increased the throughput of the filter and the overall number of registers. Unlike the conventional filter, the performance of the ternary filters

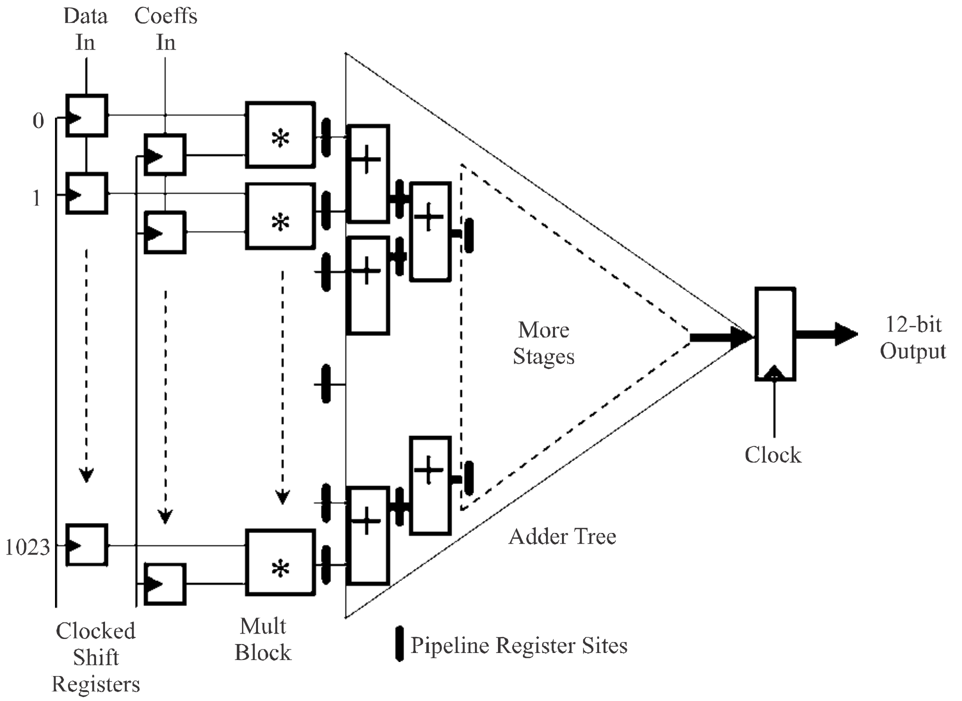

FIG. 3. BALANCED TERNARY FIR FILTER HARDWARE ARCHITECTURE

TABLE 1. SIMULATION RESULTS OF TERNARY AND CONVENTIONAL FILTER

F	N	Non-Pipelined		Pipelined	
		LUT	F_{MAX}	LUT	F_{MAX}
T	2048	12,732 (5%)	100.5	13,846 (5%)	*400.0
C	64	11,734(4%)	68.05	11,199 (4%)	108.25
F is Filter Type, T is Ternary, C is Conventional, N is Order, *I/O Clock Rate					

is greatly increased in pipelined mode as can easily be seen in Table 1.

One way to achieve optimized performance in Quartus-II was to introduce maximum clock frequency constraint during place and route than achievable for the running technology that can be compared across the devices. The flow summary generated by the Quartus-II was used to accumulate data such as number of LUTs used in the design and maximum clock frequency (F_{MAX}) achievable with zero slack on the worse-case critical paths.

Off-course the area-performance results for both designs may vary in real time applications because no account was taken of pin capacitance or specific design optimization such as forcing the use of I/O registers. In case of pipelined mode, it was found that, the ternary filter maximum clock was limited by the maximum possible I/O switching rate of the FPGA (typically below 400MHz). This limit shows the maximum clock frequency obtained by the given technology, under the strict condition that there is not I/O activity. Thus, the clock frequency may become 50% with I/O activity as indicated in the Table 1.

Table 1 compares ternary and conventional filters in pipelined and non-pipelined modes. The number of logic elements utilized by the typical design and maximum clock frequency obtained at equivalent spectral performance has been compared and given in Table 1.

It is clear that the ternary filter is better in performance than conventional filters due to simple single-bit multiplication as compared to the complex given in Table 1. The non-pipelined ternary filter has 32% better performance than its equivalent conventional filter. In pipelined mode this performance gap increases to 73%. In this specific application, the conventional filter requires slightly less area, although the difference amounts to less than 1% of the targeted device. In the case of the conventional FIR filter, the maximum operating frequency (F_{MAX}) reported in non-pipelined mode at 64 taps in [5] is directly comparable with the results given in the Table 1.

5. CONCLUSIONS

Ternary and multi-bit FIR filters area-performance is examined in VHDL using small commercial FPGA devices. We have shown that ternary FIR filters (i.e. SWL (Short Word Length)) techniques) offers superior performance than multi-bit FIR filters at approximately equivalent chip area. Similar performance can be achieved by conventional filters in pipelined mode at the cost of more chip area. The tradeoffs determined in this paper, indicates the characteristics of typical FPGA architectures. We found that 400MHz is easily achievable in a high performance FPGA device. This would easily handle a 6MHz video stream at OSR of 64 in pipelined mode using about 5% of the available FPGA area.

In this work we have taken more general approach and analysed the design without using optimized IP cores or built-in DSP blocks that are more useful for ASIC design. In future we will analyse the area-performance tradeoffs between ternary and conventional adaptive filters.

ACKNOWLEDGEMENT

Authors are highly thankful to Prof. Zahir M. Hussain, RMIT, Melbourne, who helped us throughout this work and provided technical support at all stages.

REFERENCES

[1] Thompson, A.C., "Techniques in Single-Bit Digital Filtering", Ph.D. Thesis, RMIT University, Melbourne, 2004.

[2] Sadik, A., Hussain, Z., and O'Shea, P., "Adaptive Algorithm for Ternary Filtering", Electronics Letters, Volume 42, pp. 420-421, 2006.

[3] Sadik, A.Z., and Hussain, Z.M., "Adaptive LMS Ternary Filtering", IEEE Region 10 TENCON, Melbourne, Australia, 2005.

[4] Macpherson, K.N., and Stewart, R.W., "Area Efficient FIR Filters for High Speed FPGA Implementation", IEE Proceedings of Visual Image Signal Process, Volume 153, pp. 711-720, 2006.

[5] Meher, P.K., Chandrasekaran, S., and Amira, A. "FPGA Realization of FIR Filter by Efficient and Flexible Systolization Using Distributed Arithmetic", IEEE Transaction on Signal Processing, Volume 56, pp. 3009-3017, 2008.

[6] Razak, A.H.A., Zaharin, M.I.A., and Haron, N.Z., "Implementing Digital Finite Impulse Response Filter Using FPGA", Proceedings of APACE, 2007.

[7] Sadik, A.Z., Hussain, Z.M., and O'Shea, P., "An Adaptive Algorithm for Ternary Filtering", IEE Electronics Letters, Volume 42, pp. 420-420, 2006.

[8] Thompson, A.C., O'Shea, P., Hussain, Z.M., and Steele, B.R., "Efficient Single-Bit Ternary Digital Filtering Using Sigma-Delta Modulator", IEEE Letters on Signal Processing, Volume 11, pp. 164-166, 2004.

[9] Memon, T.D., Beckett, P., and Hussain, Z.M., "Design and Implementation of Ternary FIR Filter Using Sigma Delta Modulation", Proceedings of ISCCC, pp. 169-174, 2009.

[10] Li, Y., Peng, C., Yu, D., and Zhang, X., "The Implementation Methods of High Speed FIR Filter on FPGA", Proceedings of ICSICT, pp. 2216-2219, 2008.

[11] Losada, R.A., "Digital Filters with MATLAB", Mathworks Inc pp. 238, May 18, 2008.

[12] Memon, T.D., “Design and Analysis of Short Word Length DSP Systems for Mobile Communication, Ph.D. Thesis, RMIT University, Melbourne, 2012.

[13] Pervez, A.M., Sorensen, H.V., and Spiegel, J.V.D., "An Overview of Sigma-Delta Converters", IEEE Signal Processing Magazine, pp. 61-84, 1996.