Polyphase Comb Filter Based on Dispatching Input Bit-stream and Interlaying Multiplexer Techniques for ΣΔ ADCs

Somayeh Abdollahvand, *Student Member, IEEE*, João Goes, *Senior Member, IEEE*, Nuno Paulino, *Member, IEEE*, and Luis Gomes, *Senior Member, IEEE*

***Abstract*—This paper describes a new design approach for implementing a Polyphase Comb Filter (PCF) based on dispatching input bit-stream and interlaying multiplexer techniques. In order to make our solution more energy efficient in comparison with prior art, we start with a detailed analysis of the drawbacks and advantages of the existing classical techniques. A new structure based on a novel SINC^3 design is proposed. This new design uses a controller unit to activate one sub-filter in each specific time interval. As a consequence, no input registers and switches are required. Since this decimation filter is working with a single-bit output bit-stream, the required multiplication function can be simply done by using interlaying multiplexers (MUXs). By interlaying different levels of MUXs along with the navigation of the input bit stream we can easily emulate the multiplication operation. The implementation in a Xilinx Spartan3 FPGA demonstrates the feasibility and hardware efficiency of our solution. The proposed new filter architecture can be readily applicable to any Sigma-Delta (ΣΔ) ADC with a single-bit output stream and it requires a reduced number of adders and registers when compared with the state-of-the-art approaches.**

***Index Terms*—Decimation filter; Polyphase Comb filter; Sigma-Delta Modulators; Field-Programmable Gate Arrays**

I. Introduction

A Sigma-Delta (ΣΔ) ADC consists of a ΣΔ modulator and a decimation filter, as can be seen in Fig. 1. Since the Digital decimation filters are an indispensable part of ΣΔ converters, the design of a decimation filter with low area and low power consumption can notably increase the efficiency of high bandwidth ADCs. Use of multistage architecture in decimation filter implementation is a proper method to prevent power consumption [1] while reducing the complexity of the first stage of the filter which operates at the highest rate [2]. The multiple stage filtering and downsampling reduce the overall complexity of the decimation filter by simplifying decimation filters of each stage [3].

Decimation is done to reduce the sampling frequency by use of comb filters at the first stage of decimation, followed by more complex filter structures such as FIR filters [4]. The operating frequency of the first stage is the same as the ΣΔ modulator. The oversampling frequency, *Fs*, is much higher than the Nyquist rate and therefore, the power consumption of the first level of the decimation filter comprises a significant portion of the total power consumption of the filter. There are two different perspectives to implement the first stage of the decimation filters: the recursive form and non-recursive form. Recursive form has been extensively used in cascaded integrators and comb (CIC) implementations.

The interesting aspects of the CIC approach are: 1) multiplication-free architecture [4]; 2) simple hardware and more economical hardware implementation [5]; 3) no storage requirement for the filter coefficients [5]. Furthermore, this approach is easily compatible with changing the order (*K*) and the decimation factor (*M*) of the SINC filter. The drawback of this method is that the decimator block is located in the middle stage of the filter. Hence the first half of the filter operates at *Fs* and the second half of the filter operates at *Fs/M*. This leads to higher power consumption when compared with the non-recursive approach.

On the contrary, the non-recursive form has lower power consumption and higher speed. In this implementation, when decimation ratio and filter order increase, power consumption will also slightly increase. These properties make the non-recursive implementation more attractive in low power consumption and high speed designs, especially when the filter order and the decimation frequency is high, the non-recursive implementation is preferred to the recursive ones. There are several ways to implement the non-recursive transfer function. One of the most power efficient comb filter structures is a polyphase decomposition comb decimation. In the Polyphase Comb Filter (PCF) design, the decimation is done before filtering and hence, all registers and functional blocks work at the lowest clock rate, i.e. *Fs/M* [6]. Consequently the power consumption of PCFs is considerably

This work was supported by the Portuguese Foundation of Science and Technology (FCT/MCTES) (CTS-UNINOVA multi-annual funding) through the PIDDAC Program funds and under IMPACT (PTDC/EEA-ELC/101421/2008) project.

The authors are with Centre for Technologies and Systems (CTS) at UNINOVA and Dept. of Electrical Engineering (DEE) of the Faculty of Sciences and technology (FCT) at the New University of Lisbon (UNL), Campus da FCT/UNL, 2829-516 Caparica, Portugal (e-mail: S.Abdollahvand@ieee.org).

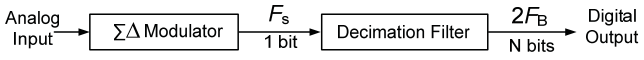

Fig. 1. Sigma-Delta (ΣΔ) modulator ADC.

lower than CIC filters. On the other hand, the use of multiplier-accumulator (MAC) blocks in the design of classical PCF makes its implementation quite expensive in terms of silicon area. As [7] shows to determine the best architecture for filter design the input word length, decimation factor and filter order should be considered.

Focused in the previously mentioned design issues, among which power dissipation and hardware complexity are the most relevant ones, in this paper we propose a new design for a polyphase implementation of a third-order SINC filter ($SINC^3$) with a decimation factor of 8. In order to make our design more power efficient in comparison with the classical polyphase filters, we have employed the counterclockwise commutator technique. By employing properly defined control signals, this design takes advantage of dispatching input bit stream and navigating bits in the related sub-filters. Moreover, by applying a multiplexer interlaying strategy, we are able to implement a multiplier-free polyphase structure. This polyphase filter is applicable to any ΣΔ ADC with a single-bit output stream.

The rest of the paper is organized as follows. In section 2 we review the different comb filter architectures proposed in the literature. Section 3 focuses on the proposed new approach and describes its main building blocks. Simulation results from a Xilinx Spartan 3 implementation are discussed in section 4. Finally, section 5 draws the main conclusions and achievements of this work.

II. State-of-the-Art

As already mentioned, the comb filter can be implemented in two different ways: recursively and non-recursively. The comb filter architectures and their characteristics, advantages and limitations are discussed in this section.

A. Recursive Implementation

The recursive transfer function of a $SINC^K$ filter can be shown as following [8]:

$$H(z) = \left(\frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}\right)^K , \quad (1)$$

where M is decimation factor and K is the order of the SINC filter. The implementation of recursive transfer function can be done in several ways. One realization of the transfer function is FIR-IIR implementation. In this architecture, as shown in Fig. 2(a), the FIR part $(1-z^{-M})^K$ is followed by the IIR part
$(1-z^{-1})^K$. The problem of this design is use of $M\times K$ shift registers [9]. Also this architecture has high power consumption since the FIR and IIR parts work in oversampling frequency.

The IIR-FIR architecture has extensively used in recursive implementation tanks to that they efficiently overcome the above mentioned problems of FIR-IIR implementation. The IIR-FIR architecture (CIC implementation) is shown in Fig. 2(b). In this approach there are K stages of accumulators, followed by an M decimator, followed by K stages of cascaded differentiators [10]. The first K stages operate at high sample rate (Fs) and the second K stages operate at low sample rate (Fs/M).

In IIR-FIR architecture, since the decimation is done after IIR part, it has less power consumption than the FIR-IIR approach and the number of registers in FIR part is reduced to K. Also the word-length of the registers in FIR part in IIR-FIR approach is reduced in comparison to FIR-IIR architecture. On the other hand, the IIR-FIR approach has own its drawbacks. The feedback path of CIC implementation is the source of some of these drawbacks. Unwanted switching power is one of these problems which is caused by registers in the feedback path. Paper [2] resolves this problem by using the accumulators in Retimed fashion. In this approach instead of using registers in the feedback path, registers are used in the forward path. This technique prevents glitches from input and combinational logic in adders from propagating to the next stage and reduces the unwanted switching power. Also CIC architecture is slow duo to the feedback path. To improve the clock rate, the pipeline registers can be used [11]. However, adding the pipeline registers in feedback path is difficult since the pipeline registers would change the transfer function [11].

One of the challenges in recursive implementation of SINC filter is word-length of registers and adders. Since the adders are in feedback path of integrators, there is a high risk of over-flow problem. The sufficient word-length of adders (and registers) in IIR part of CIC architecture is equal to $n+Klog_2(M)$, where n is the number of input bits, K is the filter order, and M is the decimation factor. Using wide (word-length) adders in CIC implementation to avoid overflow, increases the propagation delay of adders. Therefore, these wide adders are difficult to operate at high speed.

Although the CIC implementation is simple and area efficient, it has considerably high power consumption. The use of wide word-length in IIR part, which operates at oversampling frequency, is the reason behind this high power consumption. Such obstacles in front of recursive architectures provoke the use of non-recursive implementation of SINC filter.

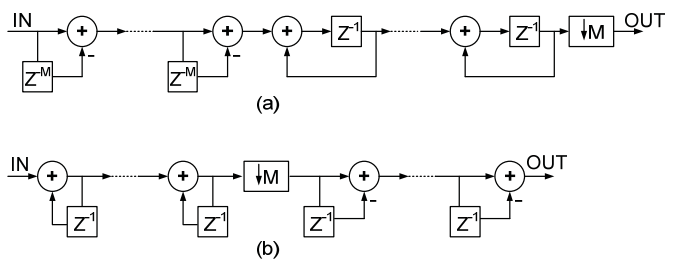

Fig. 2. Recursive architectures of $SINC^K$ filter with M decimation factor. (a) FIR-IIR structure. (b) IIR-FIR structure.

B. Non-Recursive Implementation

There are several ways to interpret the non-recursive transfer function of SINC filter. One realization is shown in (2) [4]. This direct implementation consists of cascading $\log_2 M$ identical FIR filters with a different sampling rate. As Fig. 3(a) depicts, each FIR filter $(1+z^{-1})^K$ in this architecture decimates by 2.

$$H(z) = (1+z^{-1})^K (1+z^{-2})^K ... (1+z^{-2^{(M-1)}})^K \quad (2)$$

This approach attempts to solve the problem of wide adders and registers which work at sampling frequency in CIC architecture while maintaining the simplicity of the structure. Reducing the sampling rates as early as possible helps to reduce the workload and thus the power consumption [4]. The strategy of this approach is based on using a short word-length when the sampling rate is high, and reducing the sampling rate when the word-length increases. Although the first stage of this structure works at the oversampling frequency like CIC approach, the word-length of the first stage is shorter (n+K bits). This leads to improve the frequency limitation of CIC implementation. The reduction of sampling rate through each stage by factor of 2, leads to improve the power consumption in comparison to recursive implementation. But the first stage still needs to work at the oversampling frequency [11]. This drawback makes it unsuitable for high speed applications.

The non-recursive form can be realized by applying polyphase structure too. The transfer function of the polyphase architecture and its implementation is shown in (3) and
Fig. 3(b) respectively. In this structure there are M-1 input registers (the number of bits is equal to input bits number). The polyphase architecture consists of M decimation blocks which are followed by M FIR filters and M-1 adders.

$$H(z) = (1+z^{-1}+z^{-2}+...+z^{-(M-1)})^K \quad (3)$$

An obstacle in front of polyphase structure usage in the SINC filter design is that it requires multiplier for sub-filters implementation. The number of multipliers required is equal to the number of coefficients. Due to the use of multiplier-accumulator (MAC) blocks in sub-filters implementation, the polyphase architecture is not cost efficient in terms of area consumption. Besides, as input word-length grows the calculation time increases linearly [12] which leads to reduced filtering speed. To solve these drawbacks, using multiplier-less implementation of the FIR sub-filters is recommended.

Furthermore, coefficients in the polyphase implementation of SINC filter impose storage requirements, hence the other concern in polyphase implementation is reducing the size of memory. Unlike CIC approach, this structure requires much more adders to obtain the same transfer function [11]. In addition as a result of using multiplication operation, this approach needs large adders. Although polyphase implementation of the first stage in multi-stage decimation filter requires expensive multiplication operations and wide word-length, still the overall power consumption is lower than CIC implementation thanks to the significant reduction of the operation frequency [13].

Since in polyphase architecture the filtering occurs after decimation blocks, all calculations are done in decimating frequency F_s/M, which makes it attractive for first decimation stage implementation of ΣΔ modulator [14]. In this structure all registers work at decimation frequency, except input registers which work at F_s frequency.

Another important factor which affects the area and power consumption and performance of the polyphase filter is how the multiplier is implemented in FIR subfilters. Multiplication circuits generally bear high power consumption. One way to deal with this issue is to implement multiplication by using adders and shift registers, and hence avoiding direct use of multiplication circuits. In the Canonic Sign Digit (CSD) method, the multipliers can be realized using wired shifters, adders and subtractors without using conventional multipliers [15]. In the CSD representation multiplication is more efficient than the binary representation [16]. To implement fast multipliers, it is recommended to use the Wallace tree, which has a minimum logic depth in order to obtain high clock rate and low latency [17].

In other attempts memory based methods have been proposed in order to eliminate the need of multipliers in PCF implementations. Distributed Arithmetic (DA) [12] and Look-Up Tables (LUTs) [18] are two types of implementing memory based methods. These methods have less dynamic power consumption compared to the MAC based approach [12]. Due to serial nature of Distributed Arithmetic, its computation increases linearly with the filter order enhancement [18]. Memory based architecture is also a general approach for implementing multiplier-free FIR filters. DA is basically (but not necessarily) a bit-serial computational operation that forms an inner product of a pair of vectors in a single direct step [19]. DA is computationally more efficient than MAC-based approach when the input vector length is large [12]. Other approaches have been introduced for specific domains, for instance in [11] a parallel CIC filter architecture has been proposed for the design of the SINC filter. However, although relying on a multiplier-free structure, it suffers from higher alias rejection when compared with classical implementation of PCFs.

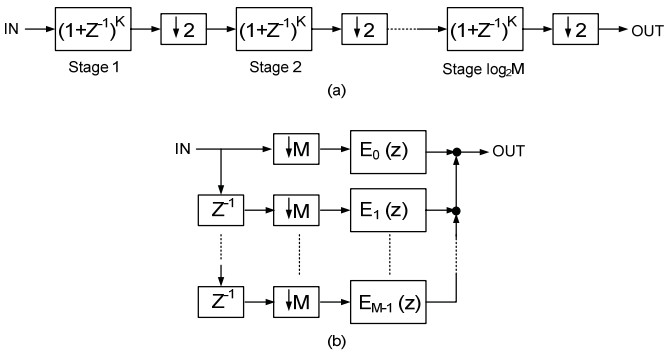

Fig. 3. Block diagram of a non-recursive implementation of a SINC filter. (a) Log_2M stages of FIR filters. (b) Polyphase decomposition architecture.

In addition to the design of a multiplier-free architecture, the paramount issue to be taken into account is the choice of the FIR architecture to implement sub-filters which has an important impact on power consumption [13]. There are two strategies to implement FIR sub-filters: direct form and transposed form. The direct form architecture has the capacity to exploit the symmetry within FIR subfilters coefficients, yet this characteristic is not necessarily efficient in the high-speed decimation filters [20]. This is caused by a long critical path in direct form structure which limits the maximum sampling frequency of the filter [13]. The critical path for the direct form is one multiplier and $\lceil \log_2(K+1) \rceil$ adders (when the additions are rearranged in a binary tree) where K is the FIR filter order [21]. Moreover, increasing the filter order also increases the critical path in direct form.

In the transposed form the critical path is only one multiplier and one adder [21]. The work reported in [20] shows that the transposed form architecture has a lower register complexity except for implementations with one-bit input. Thanks to the shared multipliers structure, this form can handle the multiplier optimization [22]. On the other hand, since the registers are in the accumulation path, the transposed form requires larger word-length for the intermediate registers, compared with direct form. This leads to increased power and area consumption [13]. Another drawback in the transposed form is its high fan-out of the input node as a single node drives several multipliers [22]. Compared to the direct form, the number of registers in the transposed form can be significantly higher in cases that the input word-length is very short [23].

Another important design issue in decreasing the power consumption concerns the timing and the controlling signals in polyphase comb filters. In classical polyphase structure, all sub-filters work at an *Fs*/8 clock rate, but the input registers which are located before down-sampling blocks work at the full sampling frequency. The power consumption of the input registers, in situations that the order of the SINC filter is low, can have severe effects on the power efficiency of the filter [6]. Various strategies are used to tackle this issue. One way to reduce power consumption is using a Counterclockwise Commutator [10]. By using the Counterclockwise Commutator model in implementing the polyphase architecture, the need for input delay line is eliminated. Other approach presented in [24] is based on a time interleaved scheme. This technique generates 8 clock signals to drive sub-filters. In this design there is no need for using input registers but it employs 8 switches, each of which connects to the one of the clock signal. References [6] and [14] propose a clock-and-data distributed technique. This technique tries to change the working frequency of input registers and manipulate them in a way that reduces dynamic power consumption of the filter. For example in [14], the working frequency of all the input registers is close to the decimating frequency, except for one of them which works at *Fs*. In [6] two of the input registers work at the decimating frequency *Fs*/8, four of them work at *Fs*/4, and only one works at *Fs*.

III. Proposed Architecture

The transfer function of SINC^3 filter with the decimating factor of 8 can be written as (4). As Fig. 4 depicts, this architecture consists of 3 main parts: the input registers of the filter; the switching blocks; the FIR sub-filters $E_0(z)\ldots E_7(z)$. According to (4) the delay elements which are shown in the parenthesis, (z^{-i}), corresponds to the input delay chain in the polyphase structure (Fig. 4). The (z^{-i}) in each term, denotes that the corresponding coefficient participates in $E_i(z)$ FIR sub-filter (for i=0,1,...,M-1). For example, coefficients without input registers are used in the $E_0(z)$ implementation, 1, $42z^{-8}$ and $21z^{-16}$, '1', '42' and '21', respectively, is the 1st., the 2nd., and the third term of $E_0(z)$.

$$\begin{aligned} H(z) = {} & 1+(z^{-1})3+(z^{-2})6+(z^{-3})10+(z^{-4})15+(z^{-5})21 \\ & +(z^{-6})28+(z^{-7})36+42z^{-8}+(z^{-1})46z^{-8}+(z^{-2})48z^{-8} \\ & +(z^{-3})48z^{-8}+(z^{-4})46z^{-8}+(z^{-5})42z^{-8}+(z^{-6})36z^{-8} \\ & +(z^{-7})28z^{-8}+21z^{-16}+(z^{-1})15z^{-16}+(z^{-2})10z^{-16} \\ & +(z^{-3})6z^{-16}+(z^{-4})3z^{-16}+(z^{-5})z^{-16} \end{aligned} \quad (4)$$

In this work we propose a novel SINC^3 design based on the Counterclockwise Commutator technique. The new design uses a controller unit to activate one sub-filter in each specific time interval. As a result, in this design no input registers and switches are required. Since this decimation filter is working with a single-bit output bit-stream, the required controller is simple and the required multiplication function can be done by using interlaying MUXs and relying on a simple architecture.

The most relevant idea of our approach is that, by interlaying different levels of MUXs along with navigation of the input bit stream we can easily emulate the multiplication operation. This decimation filter basically comprises 8 Partial Product Generating (PPG) blocks, a Partial Product

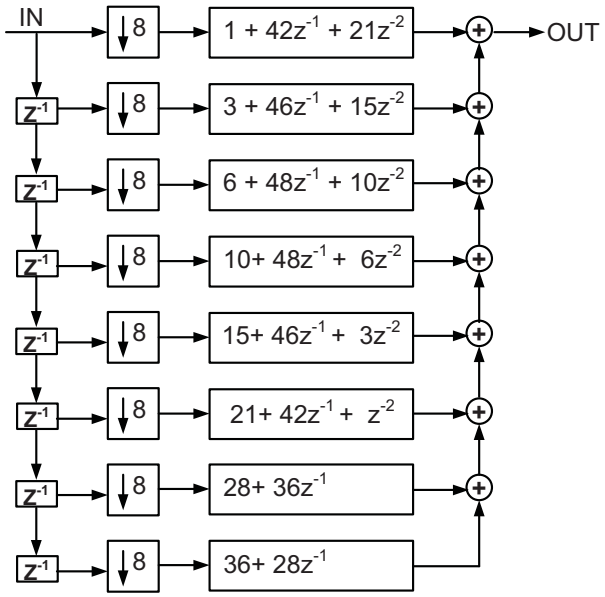

Fig. 4. Classical polyphase decimation filter structure.

Summation (PPS) block and a controller, as shown in Fig. 5. The PPG blocks together with the controlling signals act as a multiplier. The PPS block together with PPG blocks perform the realization of the sub-filters and of the complete filtering function. The description of each individual block is given in the following sub-sections.

A. The Partial Product Generating Blocks

As stated before, the design strategy is based on navigating input bit stream to the related MUXs to generate products of sub-filters in order to calculate filter output. PPG blocks are used to generate products of sub-filters. The number of MUXs and delay elements used in each PPG is equal to the number of products and the maximum delay of each sub-filter, respectively. In order to implement sub-filters with three products, three MUXs and two D-type flip-flops (D-FF) have been used. The internal schema of the PPG blocks for implementation of E_7 and E_0 are shown in Fig. 6(a) and Fig. 6(b), respectively.

The total number of coefficients is equal to 22. However, half of it is merely a repetition of the other half. To store these coefficients we take advantage of this repetition and use only 11 registers. Besides these 11 registers, an additional register is required to store the zero value. Since in each time slice only one of the sub-filters is active, we can share these registers among all sub-filters without having any fan-out problem. The number of bits required for each register is 8 bits (the maximum value of the coefficient in two's complement).

B. The Partial Product Summation Block

Fig. 7 depicts the PPS block is composed of MUXs, adders and registers. The multiplexers in this block are used to implement time division technique in order to share adders among sub-filters. The first MUX and Adder1 altogether calculate sum of first product of all sub-filters. The second and third MUXs, together with Adder2 and Adder3, participate in calculating the sum of the second and third products of the sub-filters, respectively. The maximum word-length of adders output (Adder1, Adder2 and Adder3), is 10 bits (336="0101010000"). Also the maximum word-length of decimation filter's output is 11 bits (512="0101001010"). Again, we have considered a 12 bit representation for two's complement implementation.

C. The Controller Unit and Timing

The role of controller as a dispatcher is enabling desired MUXs and D-FFs. Controlling signals for navigating input bit stream to the PPG blocks and for selecting the multiplication results are E_7 to E_0. These control signals are connected to the delay elements clocks and enable signals in the MUXs. In order to implement the controller unit, a state machine with 8 states is used.

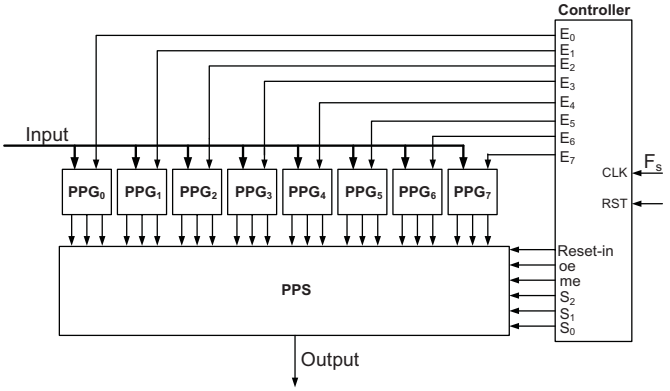

Fig. 5. Proposed Polyphase Decimation filter structure. .

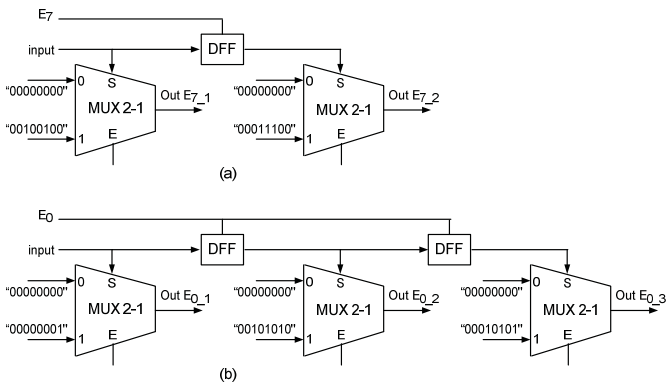

Fig. 6. (a) Partial Products Generating block for $E_7(z)$. (b) Partial Product Generating block for $E_0(z)$.

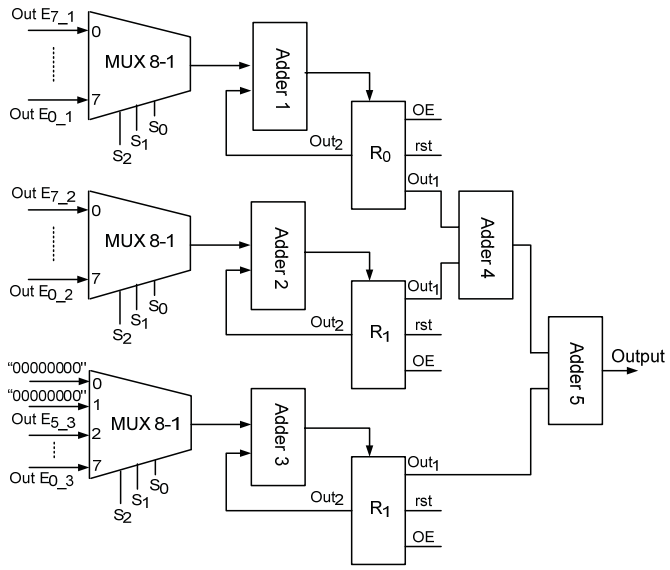

Fig. 7. Partial Product Summation to calculate filter output.

IV. Simulation Results

The simulation result of the VHDL code is shown in Fig. 8. The *Fs* clock frequency is 100 MHz. The implementation in a Xilinx Spartan3 shows that, the utilization summary for FFs, LUTs and Slices is equal to 143, 243 and 138, accordingly, which corresponds to 3%, 6% and 7% of the Spartan3 device, respectively.

Conventional polyphase Comb filter design requires 21 adders operating at *Fs*/8. PCIC design has four CIC filters. In each CIC filter, four stages of accumulators work at *Fs*/4 clock-rate and four cascaded differentiators work at the decimating frequency. Total number of adders working at *Fs*/4 and *Fs*/8 frequency are 16 and 18, respectively. In our proposed filter design, we only require five adders which work at *Fs* frequency but they are properly 'duty-cycled' (i.e., disabled during a large time slot) since each adder is associated to a given sub-filter which is enabled at a given (small) fraction of time.

The adders used in both, the conventional polyphase approach and in our filter have 12 bits. In the proposed approach we use an adder tree in which the adder depth is 3 (without considering the output registers of adders). The adder depth is defined as the number of cascaded adders. Table I summarizes the number of adders used in the conventional polyphase structure, the PCIC and our proposed filter as well as their working frequencies. As the table shows the proposed design has minimum number of adders in comparison to the classical implementation and the PCIC approach. Furthermore, Table II summarizes the required number of registers required for implementing all sub-filter as well as the input registers.

The number of input registers used in the conventional polyphase and PCIC approaches is 8 and 4, respectively, both working at *Fs*. In our proposed approach there is no need for any input registers. On the other hand, the number of registers used for the CIC implementation in PCIC is 32, half of them working at *Fs*/4 frequency and the other half working at *Fs*/8 frequency. All of them have 5-bit length. In our proposed approach, three 12-bit registers are used in each PPS block, working at the maximum sampling rate.

The proposed architecture is more area efficient compared to PCIC and classical implementation of polyphase. The power consumption efficiency gained by the proposed dispatching input bit stream and interlaying Multiplxer techniques is made possible due to the reduction in the number of adders and low adder depth. Adder depth is a significant measure of power consumption. By increasing the adder depth, the glitches are increased due to the longer combinatorial structures [21]. The elements to consider in the power model are full adders and registers. Equations (5) and (6) show the power consumption of the PCIC and proposed implementations respectively.

$$P_{PCIC} = \underbrace{2\times5\times F_s + 32\times5\times(F_s/4 + F_s/8)}_{Adders} + \underbrace{32\times5\times(F_s/4 + F_s/8)}_{Registers} \quad (5)$$

$$P_{proposed-approach} = \underbrace{5\times12\times F_s}_{Adders} + \underbrace{3\times12\times F_s}_{Registers} + \underbrace{2\times(F_s/8) + 6\times2\times(F_s/8)}_{DFFs} \quad (6)$$

TABLE I.
COMPARISONS IN TERMS OF ADDERS

Approach name	Number of Adders	Working Frequency
Conventional Polyphase	21	*Fs*/8
PCIC	34	*Fs*/4 and *Fs*/8
Proposed Approach	5	*Fs*

TABLE II.
COMPARISON IN TERMS OF REGISTERS

Approach name	Input Registers			Sub-filter Registers		
	Number	bits	Frequency	Number	bits	Frequency
PCIC	3	1	*Fs*	32	5	*Fs*/4 and *Fs*/8
Proposed Approach	0	-	-	3	12	*Fs*

V. Conclusions

This paper described a new design approach for implementing a PCF based on dispatching input bit-stream and interlaying multiplexer techniques. A new structure based on a novel $SINC^3$ design is proposed. The proposed new design uses a controller unit to activate one sub-filter in each specific time interval. As a consequence, no input registers and switches are required. Since this decimation filter is working with a single-bit output bit-stream, the required multiplication function can be simply done by using interlaying MUXs.

The implementation in a Xilinx Spartan3 FPGA demonstrated the feasibility and hardware efficiency of our solution. The proposed new filter architecture can be readily applicable to any ΣΔ ADC with a single-bit output stream and it requires a reduced number of adders and registers when compared with the state-of-the-art approaches.

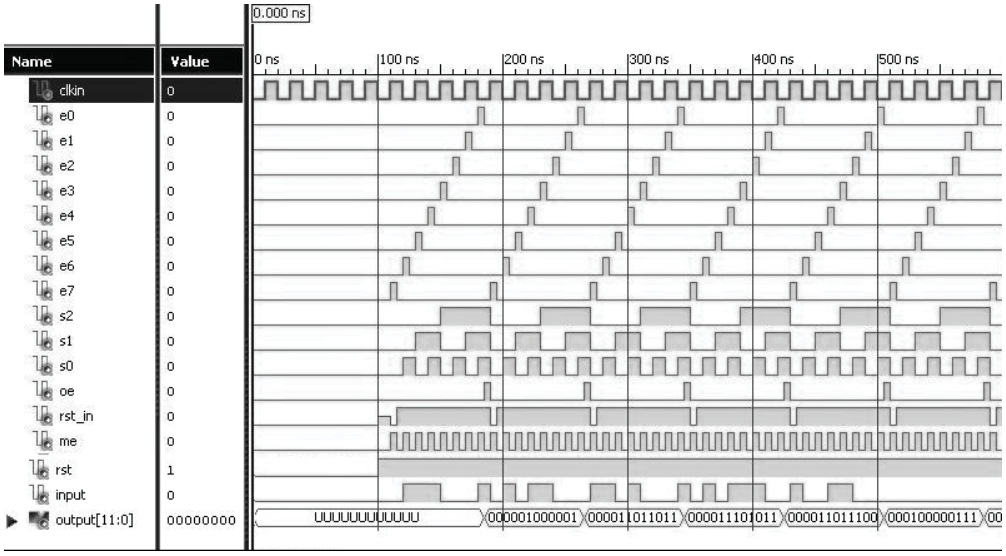

Fig. 8. VHDL simulation result for the proposed Polyphase SINC^3 filter showing output, timing and controlling signals.

REFERENCES

[1] A.Gerosa, and A.Neviani, "A low-power decimation filter for a sigma-delta convertor based on a power-optimised Sinc filter," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2004, vol. 2, pp. 245–248, May. 2004.

[2] S. Parameswaran, and N. Krishnapura, "A 100 μw decimator for a 16 bit 24 kHz bandwidth audio ΔΣ modulator," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2010, pp. 2410–2413, May-June. 2010.

[3] Y. Gao, L. Jia, and H. Tenhunen, "A fifth-order comb decimation filter for multi-standard transceiver applications," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2000, vol. 3, pp.89–92, May. 2000.

[4] Y. Gao, L. Jia, J. Isoaho, and H. Tenhunen, "A comparison design of comb decimators for sigma-delta analog-to-digital converters," Springer J. Analog Integrated Circuits and Signal Processing, vol. 22, no.1, pp. 51–60, Jan. 2000.

[5] E. B. Hougenauer, "An economical class of digital filters for decimation and interpolation," IEEE Trans. Acoustics, Speech and Signal Processing, vol. 2, no. 2, pp. 155–162, Apr. 1981.

[6] N. Younis,M. Ashour, and A. Nassar, "Power-efficient clock/data distribution technique for polyphase comb filter in digital receivers," IEEE Trans. Circuits and Systems Society, vol. 56, no. 8, pp. 639–643, Aug. 2009.

[7] M. Abbas, O. Gustafsson, and L. Wanhammar, "Power Estimation of Recursive and Non-Recursive CIC Filters Implemented in Deep-Submicron Technology," Int. Conf. Green Circuits and Systems, ICGCS, pp. 221–225, Jun. 2010.

[8] F. Yi, and W. Xiaobo, "A novel coefficient automatic calculation method for sinc filter in sigma-delta ADCs," in Proc. IEEE Asia Pacific Conf. Circuits and Syst. APCCAS 2008, pp. 1240–1243, Dec. 2008.

[9] E. Dijkstra, O. Nys, C. Piguet, and M. Degrauwe, "On the use of modulo arithmetic comb filters in sigma delta modulators," Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP-88, vol.4, pp. 2001–2004, Apr. 1988.

[10] S. R. Norsworthy, R. Schreier, and G. C. Temes, Delta-Sigma Data Converters Theory, Design, and Simulation. New York: IEEE Press, 1997, ch. 13, p. 427.

[11] X. Liu, "A high speed digital decimation filter with parallel cascaded integrator-comb pre-filters," 2nd International Congress on Image and Signal Processing, CISP '09, pp. 1–4, Oct. 2009.

[12] E. Ozalevli, W. Huang, P. E. Hasler, and D. V. Anderson, "A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering," IEEE Trans. Circuits and Systems I, vol. 55, no. 2, pp. 510–521, Mar. 2008.

[13] H. Aboushady, Y. Dumonteix, M. Louerat, and H. Mehrez, "Efficient polyphased Decomposition of comb decimation filters in ΣΔ analog-to-digital converters," IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, no. 10, pp. 898–903, Oct. 2001.

[14] N. Y. Ahmed, M. A. Ashour, and A. M. Nassar, "Power efficient polyphase comb decimation filters for ΔΣ modulators in multi-rate digital receivers,"in Proc. European Conf. Circuit Theory and Design, ECCTD 2009, pp. 719–722, Oct. 2009.

[15] M. Yamada, and A. Nishihara, "High-speed FIR digital filter with CSD coefficients implemented on FPGA," in Proc. Asia and South Pacific Conf. Design Automation, ASP-DAC 2001, pp. 7–8, 2001.

[16] A. W. Ruan, Y. B. Liao, P. Li, and J. X. Li, "An ALU-based universal architecture for FIR filters," Int. Conf. Communications, Circuits and Syst. ICCCAS, pp. 1070–1073, Jul. 2009.

[17] O. Gustafsson, and H. Ohlsson, "A low power decimation filter architecture for high-speed single-bit sigma-delta modulation," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2005, vol. 2, pp. 1453–1456, May 2005.

[18] P. K. Meher, "New approach to Look-Up-Table design and memory-based realization of FIR digital filter," IEEE Trans. Circuits and Systems I, vol. 57, no. 3, pp. 592–603, Mar, 2010.

[19] S. A. White, "Applications of distributed arithmetic to digital signal processing: a tutorial review," IEEE ASSPN Mag., vol. 6, no. 3, pp. 4–19, Jul. 1989.

[20] A. Blad, and O. Gustafsson, "Bit-level optimized FIR filter architectures for high-speed decimation applications," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2008, pp. 1914–1917, May. 2008.

[21] O. Gustafssona, J. O. Colemanb, A. G. Dempster, and M.D. Macleod, "Low-complexity hybrid form FIR filters using matrix multiple constant multiplication," in Proc. 38th Asimolar Conf. Signals, Systems and Computers, vol. 1, pp. 77–80, Nov. 2001.

[22] A. Shahein, M. Becker, N. Lotze, and Y. Manoli, "Power aware combination of transposed-form and direct-form FIR polyphase decimators for sigma-delta ADCs," in Proc. 52nd IEEE Int. Midwest Symp. Circuits and Syst. MWSCAS '09, pp. 607– 610, Aug. 2009.

[23] K. Y. Khoo, Z. Yu, and Willson, A.N., Jr., "Design of optimal hybrid form FIR filter," in Proc. IEEE Int. Symp. Circuits and Syst. ISCAS 2001, vol. 2, pp. 621–624, May. 2001.

[24] M. Murozuka, K. Ikeura, F. Adachi, K. Machida, and T. Waho, "Time-interleaved polyphase decimation filter using signed-digit adders," 39th Int. Symp. Multiple-Valued Logic, pp. 245–249, May. 2009.