

An FPGA-Based Implementation of Variable Fractional Delay Filter

Ussanai Nithirochananont, Sorawat Chivapreecha and Kobchai Dejhan
 Department of Telecommunications Engineering, Faculty of Engineering
 King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520 Thailand
 Email: sorawat@telecom.kmitl.ac.th and kobchai@telecom.kmitl.ac.th

Abstract- A variable fractional delay (VFD) filter is widely used in applications such as symbol timing recovery, arbitrary sampling rate conversion and echo cancellation. This paper presents an implementation of variable fractional delay filter on FPGA. The implementation utilizes an efficient structure so called Taylor structure. The main advantage of this structure is to reduce number of multiplier and adder when compared with Farrow structure or modified Farrow structure. The result of implementation will be reported as throughput and area utilization.

I. INTRODUCTION

In communication systems and digital signal processing, some applications such as symbol timing recovery, arbitrary sampling rate conversion and echo cancellation require fractionally delay rather than unit delay. The fractional delay can be achieved by using a variable fractional delay (VFD) filter. A VFD filter will reconstruct the sampled signal and then delay by fraction. Finally it resamples the delayed signal.

A traditional VFD filter is formed in Farrow structure [1]. The advantage of this structure is that sub-filters are fixed for a given order. When order of the VFD filter is higher, the number of multiplier and adder will increase by square of order. The modified Farrow structure is introduced to reduce the number of the operator [2].

Although the modified Farrow structure can save the number of multiplier and adder, the reduced operators are small. The efficient structure so called Taylor structure is then introduced to address this problem. This structure can significantly reduce the number of operator when compared with Farrow structure or modified Farrow structure [3].

This paper presents an implementation of VFD filter using Taylor structure on FPGA. The multiplication which is the core operation in VFD filter will utilize the parallel distributed arithmetic (PDA) method which is improved from traditional distributed arithmetic (DA) method to increase speed of computation.

The paper is organized as following; first, we will briefly review the implementation techniques for the VFD filter in section II. The distributed arithmetic (DA) is then reviewed both serial and parallel DA in section III. In Section IV, an FPGA implementation of VFD filter will be designed and synthesized by using Xilinx ISE tool, and reported as throughput and area utilization. The targeted FPGA is Xilinx's

Spartan-3A XC3S200A. Finally, we will summarize the implementation and future work in the conclusion.

II. VARIABLE FRACTION DELAY FILTER

A variable fractional delay filter utilizes delay the signal where delay is a fractional value and can be varied with time. An ideal discrete-time delay element can be described as

$$H_{id}(z) = z^{-D} \quad (1)$$

where desired delay D is a positive real number which is composed of integer and fractional part. The delay D can be expressed as

$$D = \lfloor D \rfloor + d \quad (2)$$

where $\lfloor \cdot \rfloor$ is the greatest integer function and d is the fractional delay.

When D is integer value, output sample will be one of the previous input sample but we cannot realize (1) when D is non-integer or fraction. Realizable fractional delay needs to approximate a finite-length and causal of (1).

One approach for fractional delay approximation that is most frequently used is maximally-flat FIR fractional delay approximation. This method is equal to the Lagrange interpolation at certain frequency, typically, when $\omega = 0$. Thus, the approximation is to obtain the transfer function of Lagrange interpolation.

There are numerous equivalent derivations to obtain the transfer function of Lagrange interpolation. In the following sub-section we will briefly review the implementation techniques for maximally flat FIR FD approximation those are Farrow structure and its improved structure or modified Farrow structure, and Taylor structure.

A. Farrow Structure

Farrow structure (FS) is a form that expresses an n^{th} -order polynomial in delay D . The transfer function of Farrow structure for Lagrange interpolation is expressed as

$$H_{\text{Farrow}}(z) = \sum_{n=0}^N C_n(z) D^n$$

(3)

where $C_n(z)$ are polynomials in z , and are the transfer function of the FIR sub-filter. The advantage of Farrow structure is $C_n(z)$ are fixed for a given order N , that is no need for updating the sub-filter coefficients.

B. Modified Farrow Structure

Modified Farrow structure (MFS) has been developed to reduce number of operator in Farrow structure. The complexity reduction can be achieved by changing range of delay parameter D . This can be obtained by multiplying sub-filter coefficients matrix by transformation matrix T which is defined as

$$T_{n,m} = \begin{cases} \text{round}\left(\frac{N}{2}\right)^{n-m} \binom{n}{m} & \text{for } n \geq m \\ 0 & \text{for } n < m \end{cases} \quad (4)$$

where $n, m = 0, 1, 2, \dots, N$.

C. Taylor Structure

Unlike the earlier two techniques, Taylor structure (TS) is derived from finite-length discrete-time dual of Taylor series [3]. The transfer function of the Taylor structure is expressed as

$$H_{Taylor}(z) = \sum_{n=0}^N (-1)^n \frac{D^{[n]}}{n!} (1 - z^{-1})^n \quad (5)$$

where $D^{[n]}$ is a falling factorial and $D^{[n]} = D(D-1)(D-2) \dots (D-N+1)$.

D. Complexity Comparison of FS, MFS and TS

The computational complexities of each structure are Farrow with $N^2 + N$ multiplier and N^2 adder, modified Farrow save some operation of Farrow structure and The Taylor structure are $3N - 1$ multiplier and $3N - 2$ adder.

III. DISTRIBUTED ARITHMETIC

Distributed arithmetic (DA) method is an efficient technique for computing the inner product. The inner product of two vectors can be calculated in term of sum of product as

$$Y = \sum_{k=1}^{N-1} A_k x_k \quad (6)$$

where A_k is constant coefficient and x_k is binary input data.

Since x_k are 2's complement binary number, each x_k can be expressed as

$$x_k = -x_{k,n-1} + \sum_{m=1}^{n-1} x_{k,n-1-m} 2^{-m} \quad (7)$$

where $x_{k,m}$ are bits, 0 or 1 and $x_{k,n-1}$ is the sign bit. When substitute (7) into (6), we get

$$\begin{aligned} Y &= \sum_{k=1}^{N-1} A_k \left(x_k = -x_{k,n-1} + \sum_{m=1}^{n-1} x_{k,n-1-m} 2^{-m} \right) \\ &= \sum_{k=1}^{N-1} A_k x_{k,n-1} + \sum_{m=1}^{n-1} \left(\sum_{k=1}^{N-1} A_k x_{k,n-1-m} \right) 2^{-m} \end{aligned} \quad (8)$$

Each $x_{k,m}$ has only 2^N possible values, and can be pre-computed and stored in lookup table (LUT).

Thus, the DA-based calculation of the inner product requires three operations. First, address memory to get the pre-computed values from LUT, then drop the read value into an accumulator. Finally, divide the result that can be realized by shift operator. This process requires n -cycle to complete the inner product of n -bit input data. Figure 1 shows the block diagram of the DA.

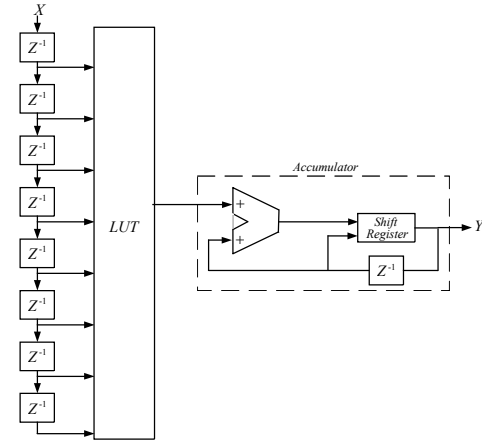


Fig.1 The block diagram of the distributed arithmetic.

Since the input data serially accesses the LUT, the traditional DA can be considered as serial DA (SDA). The speed of computation can be increased by accesses each input data bit in parallel. This improved DA is called parallel distributed arithmetic (PDA) [4]. Figure 2 depicts the parallel DA.

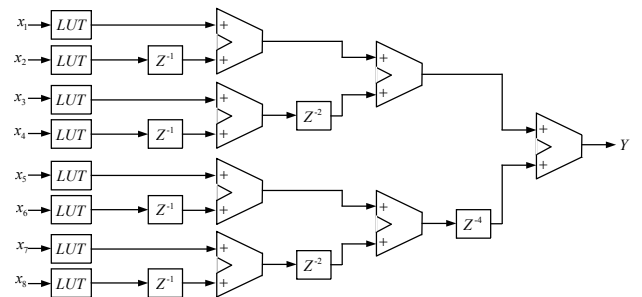


Fig.2 The block diagram of the PDA.

The input data bit accesses LUT in parallel and is scaled down by 2^{-N} , where N is bit position. The all read values are summed. The computational time is reduced to 1 cycle per input data.

IV. FPGA IMPLEMENTATION

According to the advantage in reducing the computational complexity we described in previous section, an implementation of VFD filter will utilize the Taylor structure. The architecture of the Taylor structure VFD filter is shown in Fig. 3 (a). The dash line box is 1-stage VFD filter used for implementation. Detail of 1-stage VFD is shown in Fig. 3 (b).

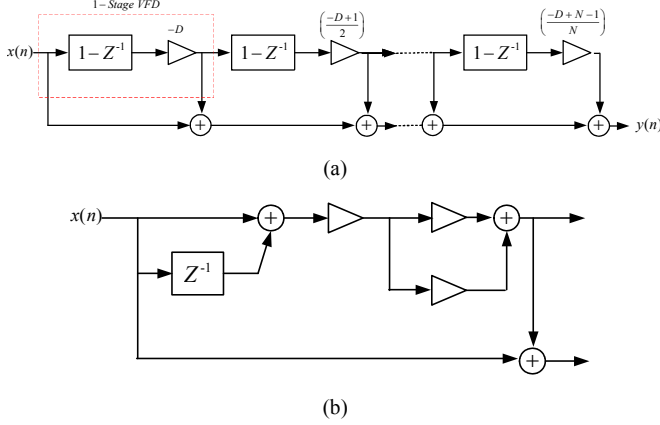


Fig.3 (a) The Taylor structure VFD filter and (b) 1-stage for implementation.

The multiplication which is the core computation of VFD filter will implemented by using the PDA method. Each sub-modules of PDA are synthesized by using Xilinx ISE tool. The details of sub-module are described as following;

A. Bit Splitter

To perform the fully parallel DA, each bit of input data will be split into individual bit by using bit splitter. Every clock when input data arrives to bit splitter, each bit is split. Figure 4 shows the schematic of bit splitter synthesized by Xilinx ISE tool, and its functional simulation result.

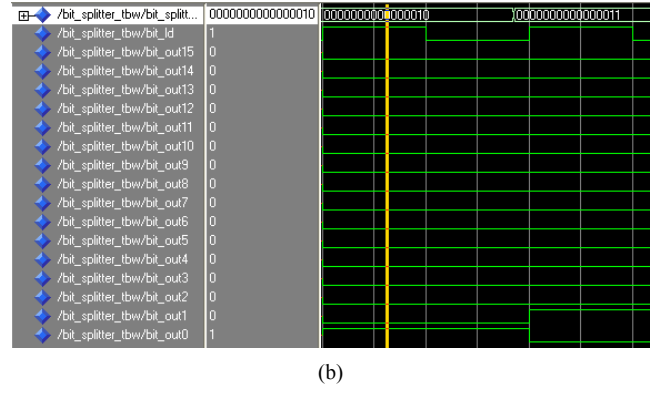
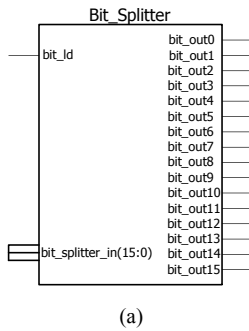


Fig. 4 (a) The schematic symbol of bit splitter and (b) the result of the functional simulation.

B. Lookup table

LUT for the VFD filter is divided into 2 types, which are LUT for constant coefficient and LUT for fractional delay. Both LUT types have 1 input and 1 output in common but the latter type has 1 additional input for selecting the fractional delay value. Both LUT schematics are shown in Fig. 5 and functional simulation results are shown in Fig. 6.

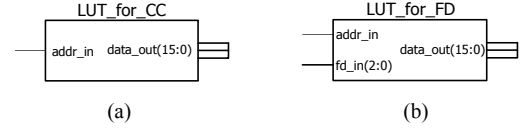


Fig.5 The schematic symbol of (a) LUT for constant coefficient and (b) LUT for fractional delay.

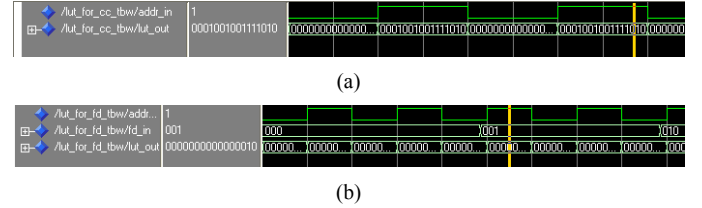


Fig.6 The Result of the functional simulation of (a) LUT for constant coefficient and (b) LUT for fractional delay.

Size of LUT for fractional delay is depending on the number of fractional delay value. For this implementation we use 8 fractional delay values starting from 0.2 to 0.9, increasing by 0.1 each time. Table I shows the detail of the fractional delay value stored in the LUT.

TABLE I
THE LOOKUP TABLE FOR FRACTIONAL DELAY

Fractional Delay Value	Address
0.2	000
0.3	001
0.4	010
0.5	011
0.6	100
0.7	101
0.8	110
0.9	111

C. Shifter

Shifter is used to shift left the read data from LUT. Level of shift-bit is depending on each bit position. The 1-bit shifter schematic is shown in Fig. 7.

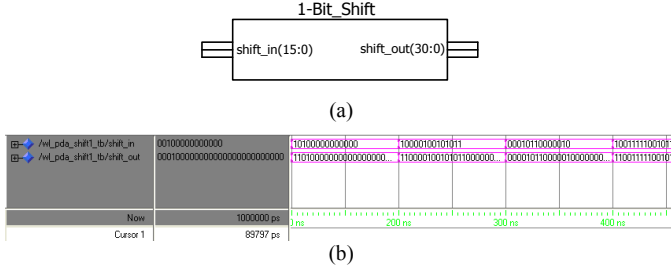


Fig. 7. (a) The schematic symbol of 1-bit shifter and (b) the result of the functional simulation.

D. Truncator

Since the shifter makes the result of multiplication wider than the input data length does, we need to truncate the length to 16 by using truncator. The truncator schematics and its functional simulation result are shown in Fig. 8.

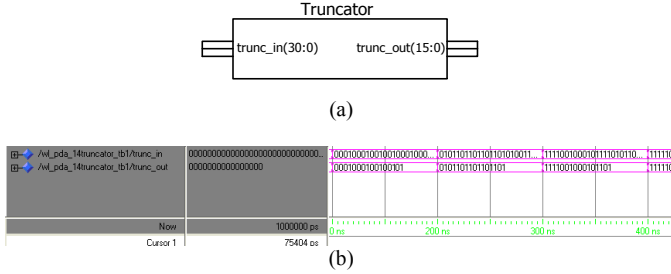


Fig. 8 (a) The schematic symbol of truncator and (b) the result of the functional simulation.

When all sub-modules complete the functional simulation, the sub-modules are then combined to be a core operator of the VFD filter. By using 2 different LUT, the core operator is also divided into 2 types, which are PDA for constant coefficient and PDA for fractional delay. Figure 9 shows the both PDA-based core operator schematics.

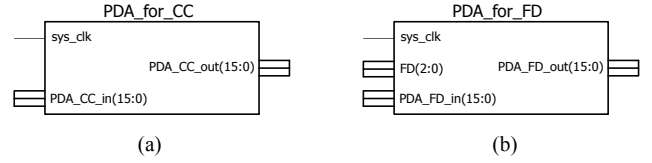


Fig. 9. The schematic symbol of (a) PDA for constant coefficient and (b) PDA for fractional delay.

Since the PDA-based core operator needs only 1 cycle to complete the computation, the increasing speed of computation is achieved.

Then we implement the 5th-order Taylor structure VFD filter with both of PDA-based cores. The schematic of the 1-stage VFD filter is shown in Fig. 10. Finally, the throughput and area utilization of the proposed VFD filter implementation are 38.493 MHz and 1,694 slices respectively.

IV. CONCLUSION

The FPGA-based Taylor structure variable fractional delay filter is implemented with PDA-based core computation. The main advantage of the proposed VFD filter implementation is to reduce the number of operator when compared with VFD filter which utilizes the Farrow or modified Farrow structure. Since the targeted FPGA is the low-cost FPGA, the throughput and area utilization of the VFD filter order show that the implementation is a compromise of cost and performance.

REFERENCES

- [1] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE Int. Symp. Circuit Systems*, 1988, pp.2641-2645.
- [2] V. Valimaki, "A new filter implementation strategy for Lagrange interpolation," in *Proc. IEEE Int. Symp. Circuit Systems*, 1995, pp.361-364.
- [3] C. Candan, "An efficient filtering structure for Lagrange interpolation," *IEEE Signal Process. Letters*, vol. 14, no. 1, pp.17-19, Jan. 2007.
- [4] A. M. Al-Haj, "An FPGA-based parallel distributed arithmetic implementation of the 1-D discrete wavelet transform," *Informatica*, 29: 241-247, Jan. 2005.

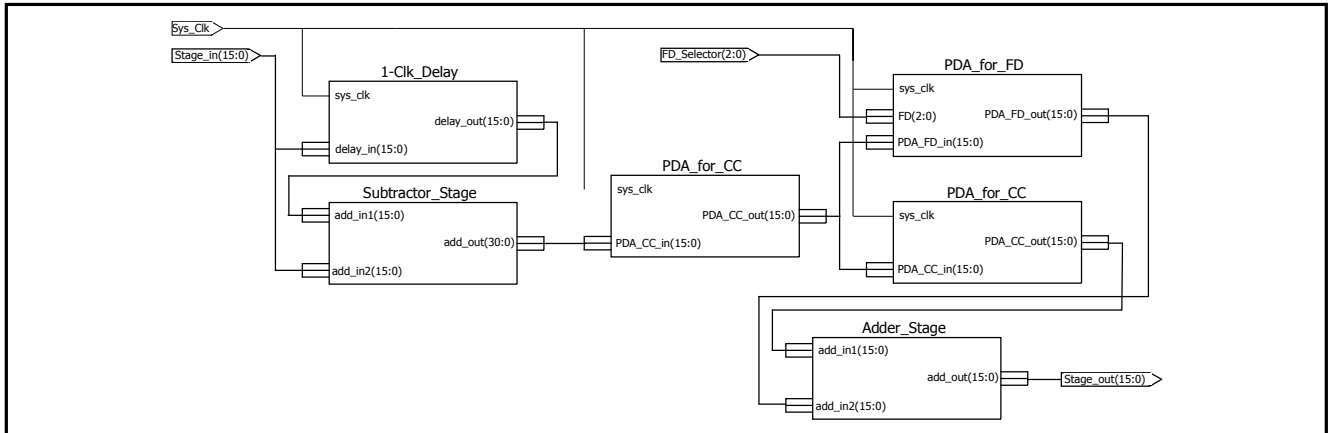


Fig.10 The schematic of 1-stage VFD filter.