A HIGHLY-SCALEABLE SYMMETRIC/ASYMMETRIC FIR PROCESSOR

Wei-Lung Liu, Oscal T.-C. Chen

Signal and Media Laboratories,
Department of Electrical Engineering,
National Chung Cheng University,
Chia-Yi, Taiwan, R.O.C.
Email: oscal@ee.ccu.edu.tw

ABSTRACT

Based on the radix-4 Booth algorithm, we developed a highly-scaleable symmetric/asymmetric finite impulse response (FIR) architecture which comprises a pre-processing unit, data latches, configurable connection units, double Booth decoders, coefficient registers, a path control unit, and a post-processing unit. In order to achieve scaleability, the configurable connection units between data latches and the double Booth decoders have been effectively addressed. The precision of filter coefficients is adjustable by using a path control unit. The double Booth decoder with single and double Booth decoding is efficiently implemented. Especially, the proposed architecture only employs data-path controls to accomplish the scaleable operations without changing word lengths and components of data latches and filter taps. A practical FIR processor, which can accommodate dynamic ranges of 8 and 16 bits of input data and filter coefficients, was implemented by using the COMPASS 5V cell library in the TSMC 0.6 μm CMOS technology. This processor supports ten different operation modes of asymmetric, symmetric, and anti-symmetric filter coefficients at 64, 63, 32, or 16 taps for various industrial applications.

1. INTRODUCTION

Filters of the finite impulse response (FIR) have been widely used in many real-world applications such as communication processing and multimedia computing [1,2]. Efficient FIR implementation has been exhaustively explored in the recent decade. The major operation of a FIR filter is the convolution which can be accomplished by using adders, multipliers, and delay elements. Usually, a multiplier takes a lot of computation time to realize its function as well as high implementation cost. In order to effectively reduce computation time and hardware complexities, high-speed FIR filters without using multipliers have been addressed by many researchers. These multiplierless filters can be classified by a memory-based approach, a canonical signed-digit (CSD) approach, and a Booth-algorithm approach. The simplified multiplierless FIR design in these three approaches allows easy incorporation of programmability. However, the FIR with scaleable dynamic ranges of input data and filter coefficients is not straightforwardly achievable.

Here, we consider to improve the conventional FIR structures with scaleable dynamic ranges of input data and filter coefficients. In the memory-based FIR design, the word length of input data and precision of filter coefficients are usually fixed for one memory configuration. In order to achieve scaleability, we have to configure the memory cells and rearrange connections among taps. Due to a high cost of the original architecture for a large dynamic data range, the memory-based FIR may not be a good candidate for scaleable design. In the CSD FIR design, filter coefficients are easily scaleable but functional units in each tap require the maximum word-length design. All CSD taps are directly addressed by every input datum using the fixed word-length hardware. When considering a large dynamic range of input data, we need partition input data to be a sub-datum sequence. In such a case, there is a need of the complicated tap design to support this FIR computing. The CSD FIRs with scaleable dynamic ranges of input data may not be realized at a low cost. By employing the Booth algorithm in the FIR design, input data are recoded in the bit-level format such that different dynamic ranges of input data can be achieved by configuring connections between an input bit-level sequence and filter taps. In addition, precision of filter coefficients can be scaled by configuring connections among filter taps due to their regular structures. Therefore, the FIR structure using the Booth algorithm can be a cost-effective design to achieve scaleable capability.

In this paper, we propose a new FIR structure which provides users more flexibly to manipulate input data and filter coefficients. Based on this architecture, a practical FIR processor with 8-bit and 16-bit dynamic ranges of input data and filter coefficients was implemented by using TSMC 0.6 μm CMOS technology. It can support ten different operation modes in four groups of (1) 32-tap asymmetric, 63-tap and 64-tap (anti)-symmetric 8-bit filter coefficients at 8-bit input data, (2) 16-tap asymmetric, 31-tap and 32-tap (anti)-symmetric 16-bit filter coefficients at 8-bit input data, (3) 32-tap asymmetric 8-bit filter coefficients at 16-bit input data, and (4) 16-tap asymmetric, 31-tap and 32-tap (anti)-symmetric 16-bit filter coefficients at 16-bit input data.

2. FIR USING THE RADIX-4 BOOTH ALGORITHM

Generally, the adequate architecture used for implementing an FIR is determined by the rate of the input and output sequences and, especially, the speed of a multiplication-accumulation operation is very critical. In order to reduce the complexity of multiplication [3], the radix-4 Booth algorithm is utilized to interpret input data of $\boldsymbol{X}$. Each datum of $\boldsymbol{X_i}$ is partitioned into many 3-bit groups, triplets, each of which has one bit overlapped with the previous group. Hence, the 2's complement of X_i with a wordlength of W can be represented by

$$X_{n-i} = -x_{n-i}^{(W-1)} \times 2^{W-1} + \sum_{j=0}^{W-2} x_{n-i}^{(j)} \times 2^{j}$$
$$= \sum_{l=0}^{\frac{W}{2}-1} \left(-2x_{n-i}^{(2l+1)} + x_{n-i}^{(2l)} + x_{n-i}^{(2l-1)} \right) \times 2^{2l} \qquad (1)$$

where $x_{n-i}^{(j)}$ is the jth digit of X_{n-i}, and $x_{n-i}^{(-1)}$ is equal to zero. When considering a filter coefficient C_i multiplied with X_{n-i}, the equation (1) is modified to

$$C_i \times X_{n-i} = \sum_{l=0}^{\frac{W}{2}-1} \left(-2x_{n-i}^{(2l+1)} + x_{n-i}^{(2l)} + x_{n-i}^{(2l-1)} \right) \times C_i \times 2^{2l}$$
$$= \sum_{l=0}^{\frac{W}{2}-1} B\left(X_{n-i,l}, C_i\right) \times 2^{2l} \qquad (2)$$

According to Eq. (2), $B\left(X_{n-i,l}, C_i\right)$ is the intermediate product which can be represented by 5 different values of *-2C$_j$*, *-C$_j$*, *0*, *C$_j$*, and *2C$_j$*. In addition, the multiplication between an input datum and a filter coefficient requires w/2 summations. For an *N*-tap FIR, the multiplication between X_{n-i} and C_i can be accomplished by the radix-4 Booth algorithm. By applying Eq. (2), we can obtain

$$Y_n = \sum_{i=0}^{N-1} C_i \times D_s^{i}(X_{n-i})$$
$$= \sum_{i=0}^{N-1} \left[\sum_{l=0}^{\frac{W}{2}-1} B\left(D_s^{i \times \frac{W}{2}}(X_{n-i,l}), C_i \right) \times 2^{2l} \right] \qquad (3)$$

where $D_s^{i}(X)$ represents the signal X to be delayed at *i* clock cycles and *s* represents the delay element in the input signal flow According to Eq. (3), we can construct the direct-form FIR architecture requiring accumulations in each tap to sum up intermediate products. In order to improve this architecture, Eq. (3) can be rearranged to obtain the following equation,

$$Y_n = \sum_{l=0}^{\frac{W}{2}-1} \left[\sum_{i=0}^{N-1} B\left(D_s^{i \times \frac{W}{2}}(X_{n-i,l}), C_i \right) \right] \times 2^{2l} \qquad (4)$$

Based on Eq. (4), the modified FIR architecture is shown in Fig. 1. The accumulation in each tap is moved to the post-processing unit such that the word length and hardware components of each tap are optimized. In order to increase the throughput rate, data latches can be inserted to the accumulation flow. However, additional data latches are also required in the input signal flow for the direct-form structure as shown in Fig. 2. When considering the transposed direct-form structure as shown in Fig. 3, data latches of the input signal flow can be correspondingly reduced as described in the following equation.

$$Y_n = \sum_{l=0}^{\frac{W}{2}-1} \left[B\left(X_{n,l}, C_0\right) + B\left(D_s^{\frac{W}{2}}(X_{n-1,l}), C_1 \right) + \ldots \right.$$
$$\left. B\left(D^{(N-1)\times\frac{W}{2}}{}_s\left(X_{n-N+1,l}\right), C_{N-1} \right) \right] \times 2^{2l}$$
$$= \sum_{l=0}^{\frac{W}{2}-1} \left[B\left(X_{n,l}, C_0\right) + \right.$$
$$\left. D_a\left(B\left(D_s^{\frac{W}{2}-1}(X_{n-1,l}), C_1 \right) + D_a\left(B\left(D_s^{2\times(\frac{W}{2}-1)}(X_{n-2,l}), C_2 \right) + D_a(\ldots) \right) \right) \right] \times 2^{2l} \qquad (5)$$

where the symbol, a, of $D_a(\bullet)$ represents the delay element in the accumulation flow. Therefore, the transposed direct-form FIR architecture based on the radix-4 Booth algorithm is better than the direct-form one for high-speed computing. Furthermore, the (anti)-symmetric FIR can be efficiently implemented by using the transposed direct-form architecture as shown in Fig. 4. According to Fig. 4, we can observe that the direction of input signal flow is inverse to that of the accumulation flow, and then data latches in the input signal flow are correspondingly decreased due to pipelining in the accumulation flow. If directions of the input signal flow and accumulation flow are the same, the data latches in the input signal flow are increased. Otherwise, the number of data latches in the input signal flow remains the same for the case of an even number of filter taps. Here, we illustrate a symmetric FIR with an even number of filter taps. Eq. (2) can be modified to

$$Y_n = \sum_{l=0}^{\frac{W}{2}-1} \left[\sum_{i=0}^{\frac{N}{2}-1} B\left(D_s^{i \times \frac{W}{2}}(X_{n-i,l}), C_i \right) + B\left(D_s^{(N-1-i)\times\frac{W}{2}}(X_{n-N+1+i,l}), C_i \right) \right] \times 2^{2l}$$
$$= \sum_{l=0}^{\frac{W}{2}-1} \left[\sum_{i=0}^{\frac{N}{2}-1} \left(D_s^{i \times \frac{W}{2}} \left(-2x_{n-i}^{(2l+1)} + x_{n-i}^{(2l)} + x_{n-i}^{(2l-1)}\right) + \right. \right.$$
$$\left. \left. D_s^{(N-1-i)\times\frac{W}{2}} \left(-2x_{n-N+1+i}^{(2l+1)} + x_{n-N+1+i}^{(2l)} + x_{n-N+1+i}^{(2l-1)}\right) \right) \times C_i \right] \times 2^{2l}$$
$$= \sum_{l=0}^{\frac{W}{2}-1} \left[\sum_{i=0}^{\frac{N}{2}-1} \tilde{B}\left(D_s^{i \times \frac{W}{2}}(X_{n-i,l}), D_s^{(N-1-i)\times\frac{W}{2}}(X_{n-N+1+i,l}), C_i \right) \right] \times 2^{2l} \qquad (6)$$

where the double Booth decoder, $\tilde{B}\left(X_{n-i,l}, X_{n-N+1+i,l}, C_i\right)$, have nine possible values of *-4C$_i$*, *-3C$_i$*, *-2C$_i$*, *-C$_i$*, *0*, *C$_i$*, *2C$_i$*, *3C$_i$*, and *4C$_i$*.

3. PROPOSED FIR ARCHITECTURE

The choice of structure for the implementation of an FIR includes consideration of factors such as hardware complexity, desired throughput and filtering performance. The main challenge would be to optimize flexible architectures for various FIR applications at a low cost. In order to achieve scaleable dynamic ranges of input data and filter coefficients, Eq. (6) is modified by the following way,

$$Y_n = \sum_{l=0}^{\frac{W}{2}-1} \sum_{j=0}^{P} \left[\left(\sum_{i=0}^{\frac{N}{2}-1} \tilde{B}\left(D_s^{i \times \frac{W}{2}}(X_{n-i,l}), D_s^{(N-1-i)\times\frac{W}{2}}(X_{n-N+1+i,l}), C_{i,j} \right) \right) \times 2^{2l} \right] \times 2^{-(j \times k-1)} \qquad (7)$$

where $C_{i,j}$ is the j^{th} sub-precision component of C_i, *P* is the number of sub-coefficients, and *k* represents the bit number of a sub-coefficient. According to Eq. (7), scaleability can be realized in the control of *W* and *P*. Based on the radix-4 Booth algorithm to accomplish multiplication, the proposed highly-scaleable FIR architecture, as shown in Fig. 5, consists of a pre-processing unit, data latches, configurable connection units, double Booth decoders, filter coefficient registers, a path control unit, and a post-processing unit. A detailed description of each unit of the proposed highly-scaleable FIR architecture is as follows.

A pre-processing unit: The pre-processing unit, as shown in Fig. 6, partitions input data to be a triplet sequence according to the radix-4 Booth algorithm. Each triplet datum is recoded in the 2's compliment representation and then pipelined into the data latches.

Data latches: Data latches that form a plurality of sequence units are utilized to store a triplet sequence from the pre-processing unit.

Configurable connection units: Configurable connection units are to determine which of the sequence units are selected for the convolution calculation. In order to efficiently realize scaleable dynamic ranges of input data, simple multiplexors can be used to configure the connection topology as shown in Fig. 5. The paths of each (anti)-symmetric filter tap connected to several input data latches are controlled by two multiplexors for different scaleable ranges. Only two paths are enabled to link an (anti)-symmetric filter tap to the corresponding two latches of input data. One path comes from the configurable connection unit I, and the other is from the configurable connection unit II.

Double Booth decoders: The double Booth decoder, as shown in Fig. 7, receives two selected sequence units of which values are added or subtracted according to the operation of symmetric or anti-symmetric filter taps by using an 4-bit carry-lookahead adder. The added or subtracted result is encoded to be four control signals which are S, Z_0, Z_1, and Z_2. The Z_1 and Z_2 are control signals of a 4-to-1 multiplexor to generate an output result of C_i, $2C_i$, $3C_i$, or $4C_i$ where $3C_i$ is produced by using a half adder to sum $2C_i$ and C_i. This intermediate result with Z_0 are performed by an AND function where Z_0 interprets an output result of zero. The signal S is to determine positive or negative of the final result. If S is logic-1, the result from the AND function is inversed and the following accumulation adder is feed with a carry-in bit of logic-1. Otherwise, the result from the AND function is directly added by the following accumulation adder with a carry-in bit of logic-0. When one input of the double Booth decoder becomes zero, this decoder can function as a single Booth decoder which can be used for all taps of the asymmetric FIR, and the central tap of the (anti)-symmetric FIR with an odd number of filter coefficients

Filter coefficient registers: The filter coefficient registers realized by master-slave latches are to store the values of filter coefficients.

A path control unit: Since the word length of registers for storing filter coefficients is fixed, we can arrange a filter coefficient stored in several registers. A path control unit, as shown in Fig. 5, is to arrange the accumulation relationship among the decoded values of sub-coefficients and input data. In order to effectively to realize the path control unit, a latch is needed in every neighboring taps of the accumulation flow.

A post-processing unit: The post-processing unit is to perform a final accumulation of selected intermediate products for the convolution calculation between the selected sequence units and the filter coefficients. Figure 8 shows the post-processing unit of which major components are adders, shifters, latches, and multiplexors for supporting various dynamic ranges of input data and filter coefficients.

4. AN 8/16-BIT FIR PROCESSOR

An FIR processor with scaleable dynamic ranges of input data and filter coefficients at 8 and 16 bits is designed to support various industrial applications. Based on the proposed architecture, this 8-bit and 16-bit FIR processor is explored to reduce its functional units. Due to the recoding of the radix-4 Booth algorithm, the numbers of data latches required for 8-bit and 16-bit input data are quite different. Hence, we optimized functional units of this processor for ten different operation modes listed in Table 1. In such a design, there is a need of 32 8-bit filter coefficient registers, 32 double Booth decoders, and 268 3-bit data latches.

The proposed FIR processor was designed by using the COMPASS 5V standard cell library in the TSMC 0.6 μm CMOS technology. Figure 9 shows the layout of this FIR processor with a die size of 6.6×6.5 mm^2. The power consumption is around 1.5W at a system clock of 100 MHz. The input/output throughput rate is 25MHz for 8-bit input data and 12.5MHz for 16-bit input data. When considering 8-bit input data and (anti)-symmetric filter coefficients, our FIR processor can have the maximum 64 taps with a computation power of 6.4 billion multiplication-accumulation operations per second. Table 2 lists the specifications of the proposed FIR processor, which illustrates a cost-effective scaleable design.

REFERENCES

[1] A. Oppenheim, and R. Schafer, *Discrete-Time signal Processing*, New Jersey: Prentice Hall, 1989.

[2] A. Antoniou, *Digital Filters*, New York: McGraw-Hill, 1993.

[3] O. T.-C. Chen, W.-L. Liu, H.-C. Hsieh, and J.-Y. Wang, "A highly-scaleable FIR using the radix-4 Booth algorithm," *Proc. of IEEE International Conf. on Acoustic, Speech, and Signal Processing*, vol. 3, pp. 1765-1768, May 1998.

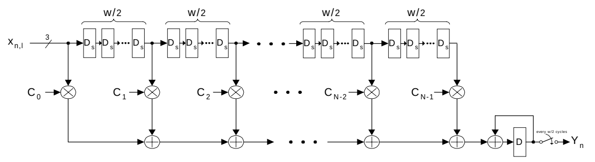

Fig. 1 An N-tap FIR using the radix-4 Booth algorithm.

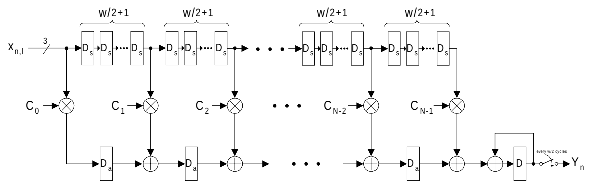

Fig. 2 An N-tap direct-form FIR filter with data latches inserted to the accumulation flow

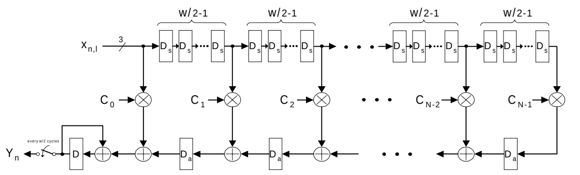

Fig. 3 An N-tap transposed direct-form FIR.

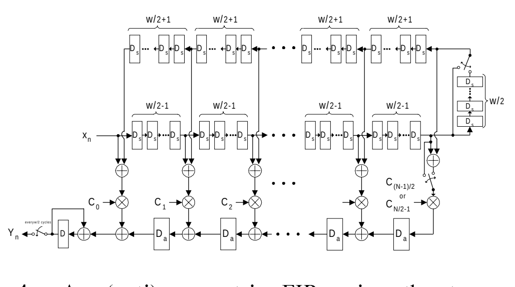

Fig. 4 An (anti)-symmetric FIR using the transposed direct-form architecture.

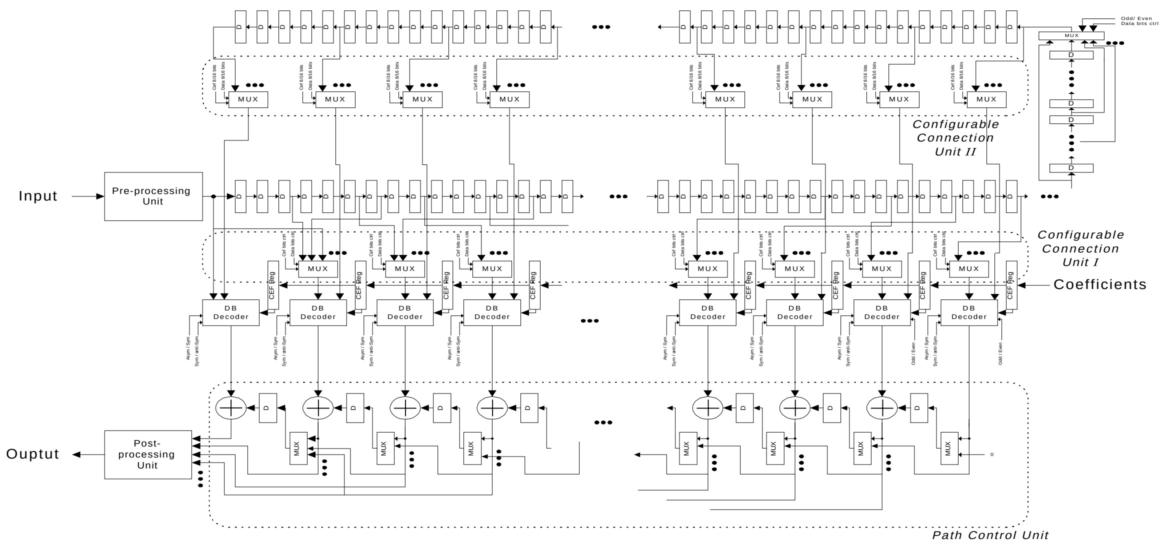

Fig. 5 The highly-scaleable symmetric/asymmetric FIR architecture.

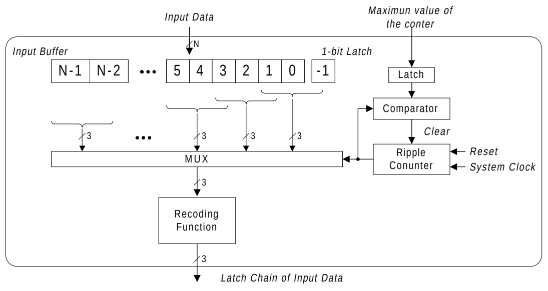

Fig. 6 The pre-processing unit.

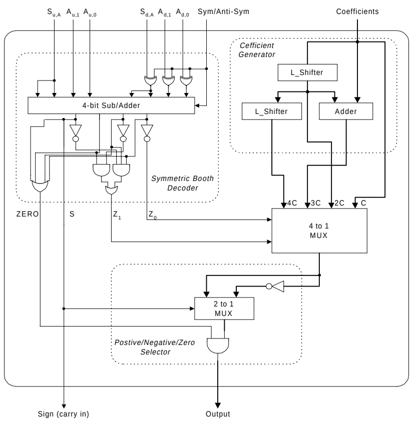

Fig. 7 The double Booth decoder.

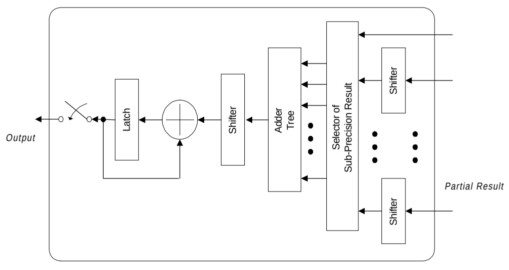

Fig. 8 The post-processing unit.

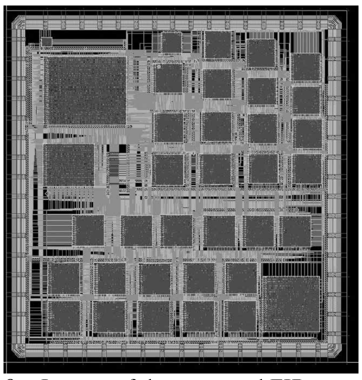

Fig. 9 Layout of the proposed FIR processor.

Table 1 Operation modes of our FIR processor.

Filter Coefficients \ Input Data			8 bits	16 bits
8 bits	Asymmetric		32 taps	32 taps
	(Anti)-Symmetric	even	64 taps	NA
		odd	63 taps	NA
16 bits	Asymmetric		16 taps	16 taps
	(Anti)-Symmetric	even	32 taps	32 taps
		odd	31 taps	31 taps

Table 2 Specifications of our FIR processor.

Chip	Scaleable FIR Processor
Input data & Filter coefficients	8 or 16 bits
Number of taps	16, 31, 32, 63, and 64
Technology	TSMC 0.6 μ m CMOS
Design scheme	COMPASS standard cell library
Supply voltage	5V
Clock rate	100MHz
Die size	6605 μ m $\times$ 6496 μ m
Power consumption	1.5W@100MHz
Throughput rate	25MHz or 12.5MHz samples per second