Microelectronics Reliability 57 (2016) 111–127

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/mr

Design, evaluation and fault-tolerance analysis of stochastic FIR filters

Ran Wang, Jie Han *, Bruce F. Cockburn, Duncan G. Elliott

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

ARTICLE INFO

Article history:
Received 7 July 2015
Received in revised form 23 November 2015
Accepted 29 November 2015
Available online 22 December 2015

Keywords:
Stochastic computing
FIR filter
Throughput per area
Energy per operation
Stochastic sequence length
Fault tolerance

ABSTRACT

Stochastic computing utilizes compact arithmetic circuits that can potentially lower the implementation cost in silicon area. In addition, stochastic computing provides inherent fault tolerance at the cost of a less efficient signal encoding. Finite impulse response (FIR) filters are key elements in digital signal processing (DSP) due to their linear phase-frequency response. In this article, we consider the problem of implementing FIR filters using the stochastic approach. Novel stochastic FIR filter designs based on multiplexers are proposed and compared to conventional binary designs implemented using Synopsys tools with a 28-nm cell library. Silicon area, power and maximum clock frequency are obtained to evaluate the throughput per area (TPA) and the energy per operation (EPO). For equivalent filtering performance, the stochastic FIR filters underperform in terms of TPA and EPO compared to the conventional binary design, although the stochastic design shows more graceful degradation in performance with a significant reduction in energy consumption. A detailed analysis is performed to evaluate the accuracy of stochastic FIR filters and to determine the required stochastic sequence length. The fault-tolerance of the stochastic design is compared with that of the binary circuit enhanced with triple modular redundancy (TMR). The stochastic designs are more reliable than the conventional binary design and its TMR implementation with unreliable voters, but they are less reliable than the binary TMR implementation when the voters are fault-free.

1. Introduction

The importance of finite impulse response (FIR) filters in digital signal processing and the potential benefits of stochastic computing motivated us to investigate the possibility of implementing stochastic FIR filters. Both manufacturing variations and transient errors pose additional challenges to reliable operation. Stochastic computing methods [1,2] can be exploited to address the above issues and thus possibly allow operation with less reliable, leading edge processes in very low voltage and/or high noise operating conditions.

In image processing, Li and Lilja showed that stochastic circuits can outperform conventional binary designs for key image processing algorithms with respect to important design metrics [3]. Specifically, sequential stochastic computational elements were built using finite state machines. Interesting results for a stochastic implementation of the kernel density estimation (KDE)-based image segmentation algorithm are reported in [4]. Alaghi et al. investigated an edge-detection algorithm for real-time image processing [5]. It was shown that the area-delay product of the stochastic edge detection circuit is only 8.7% of that of a conventional binary circuit. Qian and Riedel compared stochastic hardware implementations of polynomial arithmetic [6]. Chang and Parhi investigated novel designs for both FIR and infinite impulse response (IIR) filters based on stochastic logic [7]. Several low-pass and high-pass filters with different cut-off frequencies were considered. In those filters, the coefficients are encoded as stochastic selection signals of multiplexers and XOR gates are used to invert the inputs when the corresponding coefficients are negative numbers. Our new approach works with both unsigned and signed stochastic inputs directly without additional operations, while requiring additional multiplexers as weight generators. Another stochastic FIR filter was designed using a wire selecting method in [8]. The authors claimed that the proposed stochastic FIR filter is more cost-efficient than traditional filters in terms of hardware resource for less than 9-bit implementations. However, it is not clear whether or not the stochastic filter can effectively match the performance of the corresponding traditional filters. In [9], the authors proposed to modify conventional ADCs to convert analog to stochastic streams with minimal overhead to implement IIR filters. Input to output frequency response spectra were computed to demonstrate the idea. However, this method is limited to IIR filters where feedback networks are needed.

In this paper, three different stochastic FIR filter designs are investigated. The conventional weighted average (CWA) design exploits basic stochastic arithmetic elements, such as the XNOR gate for multiplication and the multiplexer for addition. In the hardwired weighted average (HWA) design, the filter coefficients or weights are given by repeating inputs to the multiplexer. Finally, the multi-level weighted average (MWA) design leverages the fact that every input signal is selected with a certain weight determined by the selection inputs to a multiplexer. The MWA design explicitly uses additional multiplexers to generate

* Corresponding author.

http://dx.doi.org/10.1016/j.microrel.2015.11.017
0026-2714/© 2015 Published by Elsevier Ltd.

Notation

$P(S)$	probability encoded by a stochastic sequence S
N_b	the bit resolution for the binary filter
N_s	the number of bits in stochastic sequence S
N_f	the number of taps in the FIR filter
X	a binary number, signed or unsigned
$X[n]$	the binary value of a time-series variable X at time n
$S(X)$	a stochastic binary sequence encoding the probability $\frac{X}{L}$, where L is the sequence length.

filter coefficients as the weights to a weighted adder. In all three designs, the weight of an input signal is the probability of selecting that input signal. That probability is then encoded in the frequency at which the corresponding combination of the selecting signals occurs in the bit streams. It is shown that both HWA- and MWA-based FIR filters have improved performance in terms of area, power and speed, compared to the CWA design.

Different resolutions are considered to determine the threshold (or break-even point) that defines the competitive resolution range for stochastic circuits. 3-bit to 16-bit FIR filters using both the stochastic and binary approaches are implemented initially. Then the minimum required stochastic sequence length that enables the stochastic circuit to filter signals as accurately as the binary circuit is determined empirically. The general metrics of throughput per area (TPA) and energy per operation (EPO) are used to characterize and compare the performance of the stochastic and conventional circuits.

This article is a significant extension of [10]. It contains the following main contributions.

- Stochastic FIR filters based on the HWA and MWA designs are proposed by using multiplexers to implement weighted adders. Different strategies are considered for generating the filter coefficients.
- A detailed analysis is performed to evaluate the accuracy of the binary and stochastic filters. The analytical results provide an estimation of the minimum stochastic sequence length that is required to ensure that the performance of the stochastic filter matches that of a conventional filter.
- A detailed comparison is provided with respect to the fault tolerance of the HWA- and MWA-based stochastic filters. The conventional binary filter and its fault-tolerant triple modular redundancy (TMR) implementation are considered in the comparison.

In the sequel, we first review stochastic computing and binary FIR filter designs in Section 2. In Section 3, the stochastic FIR filter designs are presented. In Section 4, the performance is compared by considering the magnitude responses of both filters. In Section 5, the simulation results from a Synopsys synthesis tool are reported. In Section 6, the accuracy of the FIR filters using both binary and stochastic computing is assessed. The fault tolerance of stochastic and conventional binary circuits are determined and compared in Section 7. Section 8 concludes the paper.

2. Background

2.1. Stochastic computing

Stochastic computing involves processing numbers that are encoded as real values, which are represented using stochastic bit-streams. If there are N_1 1's in a bit stream containing N_s bits, where $N_1 \le N_s$, it represents either the (unsigned) unipolar number N_1 / N_s or the (signed) bipolar number $(2N_1 - N_s) / N_s$ [1,2]. For example, "0001100101" denotes 2/5 in unipolar and $-1/5$ in bipolar for $N_s = 10$ and $N_1 = 4$. Stochastic computing elements can often be built using very small circuits with low power consumption [1]. To encode a binary number containing N_b bits, the minimum sequence length is $N_{s,\min} = 2^{N_b}$. However, the required sequence length N_s is usually made larger for increased accuracy during stochastic processing. Therefore, a performance matching multiplier is introduced as $PMM = N_s/2^{N_b}$. The stochastic sequences are usually generated using linear feedback shift registers (LFSRs) or other similar pseudo-random bit sequence generators. Fig. 1 shows a general stochastic computing system [2].

A stochastic weighted adder can be implemented with a multiplexer. If data inputs X_1 and X_2 are both encoded as unipolar (or bipolar) stochastic sequences, then the output Y of a two-input multiplexer will also be unipolar (bipolar). However, the multiplexer selecting signal A must be encoded as a unipolar sequence. Consider a stochastic implementation of the weighted sum

$$Y = A_1X_1 + A_2X_2, \tag{1}$$

where A_1 and A_2 are positive weights whose sum is one (i.e., $A_1 + A_2 = 1$ and $A_1, A_2 > 0$). Data inputs X_1 and X_2 are encoded as stochastic sequences $S(X_1)$ and $S(X_2)$, and weights A_1 and A_2 are encoded as unipolar stochastic sequences $S(A_1)$ and $S(A_2)$. For the multiplexer, the select input A is driven by unipolar sequence $S(A_1)$, and so this input is 1 with probability $P(S(A_1))$, i.e.,

$$P(S(A_1)) = A_1 = 1 - A_2. \tag{2}$$

The multiplexer output Y will be 1 with probability $P(S(Y))$, i.e.,

$$P(S(Y)) = A_1 \cdot P(S(X_1)) + A_2 \cdot P(S(X_2)). \tag{3}$$

First consider the case of unipolar data inputs. By definition the numbers encoded by $S(X_1)$, $S(X_2)$ and $S(Y)$ are exactly $P(S(X_1)) = X_1$, $P(S(X_2)) = X_2$ and $P(S(Y)) = Y$, respectively. Thus the computed output value Y will be the desired weighted sum $A_1X_1 + A_2X_2$.

Now consider the case of bipolar data inputs. By definition, the numbers encoded by $S(X_1)$, $S(X_2)$ and $S(Y)$ are exactly $\frac{X_1+1}{2}$, $\frac{X_2+1}{2}$ and $\frac{Y+1}{2}$, respectively. Thus the output sequence $S(Y)$ will be 1 with probability $P(S(Y))$, i.e.,

$$P(S(Y)) = \frac{A_1X_1 + A_2X_2 + A_1 + A_2}{2} = \frac{A_1X_1 + A_2X_2 + 1}{2}. \tag{4}$$

From the definition of the bipolar encoding, sequence $S(Y)$ encodes the number

$$Y = 2 \cdot P(S(Y)) - 1 = A_1X_1 + A_2X_2, \tag{5}$$

which is again the desired weighted sum.

The same argument can be generalized to weighted sums containing $N \ge 2$ inputs implemented with N-input multiplexers, where the weights $A_1, A_2, \ldots, A_N$ sum up to 1.

2.2. Encoding numbers as unipolar and bipolar stochastic sequences

Stochastic number generators (SNGs) are typically based on pseudo-random bit generators such as linear feedback shift registers (LFSRs). For example, to generate the stochastic sequence for a 4-bit unsigned binary number, the SNG in Fig. 2(a) is implemented with a 4-bit LFSR [2]. The SNG in Fig. 2(a) converts a 4-bit unsigned binary number x to a stochastic number (sequence) of length 16. The all-zero state must be

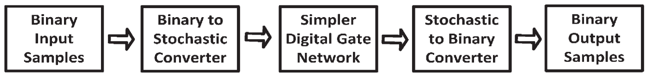
Binary Input Samples
Binary to Stochastic Converter
Simpler Digital Gate Network
Stochastic to Binary Converter
Binary Output Samples

Fig. 1. A basic stochastic computing system.