# HARDWARE-EFFICIENT DISTRIBUTED ARITHMETIC ARCHITECTURE FOR HIGH-ORDER DIGITAL FILTERS

*Heejong Yoo and David V. Anderson*

Center for Signal and Image Processing,
School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, GA 30332

**ABSTRACT**

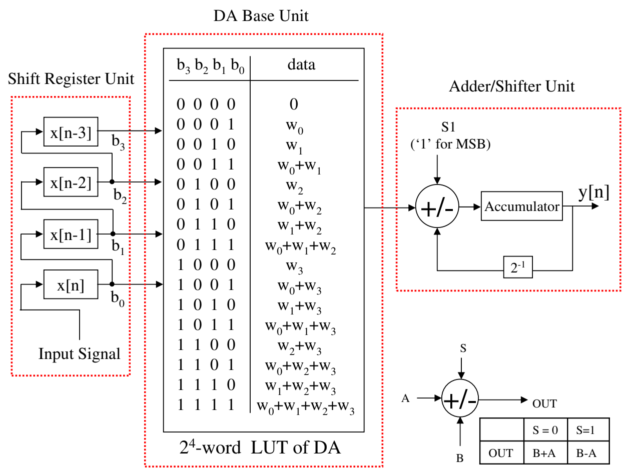
This paper presents a new memory-efficient distributed arithmetic (DA) architecture for high-order FIR filters. The proposed architecture is based on a memory reduction technique for DA look-up-tables (LUTs); it requires fewer transistors for high-order filters than original LUT-based DA, DA-offset binary coding (DA-OBC), and the LUT-less DA-OBC. Recursive iteration of the memory reduction technique significantly increases the maximum number of filter order implementable on an FPGA platform by not only saving transistor counts, but also balancing hardware usage between logic element (LE) and memory. FPGA implementation results confirm that the proposed DA architecture can implement a 1024-tap FIR filter with significantly smaller area usage ($< 50\%$) than the original LUT-based DA and the LUT-less DA-OBC.

## 1. INTRODUCTION

A discrete-time linear finite impulse response (FIR) filter generates the output $y[n]$ as a sum of delayed and scaled input samples $x[n]$. In other words,

$$y[n] = \sum_{i=0}^{K-1} w_i x[n-i]. \tag{1}$$

A direct VLSI implementation requires $K$ multiply-and-accumulate (MAC) operations, which are expensive to implement in hardware due to logic complexity and area usage. Alternatively, the MAC operations may be replaced by a series of look-up-table (LUT) accesses and summations, known as distributed arithmetic (DA) [1, 2, 3]. DA is a bit-serial operation that implements a series of fixed-point MAC operations in a fixed number of steps, regardless of the number of terms to be calculated. One problem with original DA architecture is that its LUT size ($2^K$-words) grows exponentially as the filter order $K$ increases. Several techniques have been proposed to address this memory problem of DA. The partial sum technique [1, 4], or multiple memory bank technique, breaks a $K$-tap FIR filter into $m$ smaller filters each having $k$-tap DA base units ($K = m \times k$). Here it is assumed that $K$ is not prime. The total memory requirement for a $K$-tap FIR filter, which is divided into $m$ smaller filters each having $k$-tap DA base units, is $m \times 2^k$ memory elements. The total number of clock cycles increases to $B + \lceil \log_2(m) \rceil$ from $B$ clock cycles of original DA. DA offset binary coding (DA-OBC) [4] can be used to alleviate exponentially increasing memory burden of DA. For a $K$-tap FIR filter, DA-OBC requires a $2^{K-1}$-word LUT at the cost of marginally increased control logic. The modified DA-OBC [5] can reduce the LUT size from $2^{K-2}$ to as low as 2 by exploiting the observation



| $b_3\ b_2\ b_1\ b_0$ | data |
|---|---|
| 0 0 0 0 | 0 |
| 0 0 0 1 | $w_0$ |
| 0 0 1 0 | $w_1$ |
| 0 0 1 1 | $w_0+w_1$ |
| 0 1 0 0 | $w_2$ |
| 0 1 0 1 | $w_0+w_2$ |
| 0 1 1 0 | $w_1+w_2$ |
| 0 1 1 1 | $w_0+w_1+w_2$ |
| 1 0 0 0 | $w_3$ |
| 1 0 0 1 | $w_0+w_3$ |
| 1 0 1 0 | $w_1+w_3$ |
| 1 0 1 1 | $w_0+w_1+w_3$ |
| 1 1 0 0 | $w_2+w_3$ |
| 1 1 0 1 | $w_0+w_2+w_3$ |
| 1 1 1 0 | $w_1+w_2+w_3$ |
| 1 1 1 1 | $w_0+w_1+w_2+w_3$ |

| | S = 0 | S=1 |
|---|---|---|
| OUT | B+A | B-A |

**Fig. 1.** Original LUT-based DA implementation of a 4–tap ($K = 4$) FIR filter. The DA architecture consists of three small units: the shift register unit, the DA base unit, and the adder/shifter unit.

that if the single term inside the LUT can be relocated outside the LUT, then the lower half of the LUT is mirrored version of the upper half of the LUT with only the signs reversed [5].

In this paper, a hardware-efficient DA architecture is presented. A recursive LUT reduction to the original DA decreases the LUT size by half at every iteration and eventually the LUT-less DA architecture can be achieved. It will be shown that the proposed DA architecture is more area efficient than the original DA; the DA-OBC; and the LUT-less DA-OBC, which is the more area-efficient version of the modified DA-OBC. Thus the proposed DA architecture enables more high-order FIR filter implementation on a given FPGA platform.

This paper is organized as follows. Section 2 reviews the basic of DA and Section 3 presents the new DA architecture. The performance of the proposed DA architecture is compared with previous methods in Section 4; and conclusions are given in Section 5.

## 2. REVIEW OF DISTRIBUTED ARITHMETIC

### 2.1. Distributed Arithmetic (DA)

Let the input samples be represented as scaled $B$–bit two's complement binary numbers such that $|x[n-i]| < 1$,

$$x[n-i] = -b_{i,B-1} + \sum_{l=1}^{B-1} b_{i,B-1-l} 2^{-l}, \quad 0 \le i \le K-1, \tag{2}$$

where $b_{i,B-1-l} \in \{0,1\}$, and $b_{i,B-1}$ is the most significant bit (MSB) of $x[n-i]$. Substituting (2) into (1) and changing the order of the summations yields

$$y[n] = -\left[\sum_{i=0}^{K-1} b_{i,B-1} w_i\right] + \sum_{l=1}^{B-1}\left[\sum_{i=0}^{K-1} b_{i,B-1-l} w_i\right] 2^{-l}. \quad (3)$$

For a given set of $w_i$ ($i = 0, \ldots, K-1$), the terms in the brackets may take one of $2^K$ possible values that can be stored in an LUT. Original LUT-based DA implementation of a 4-tap ($K = 4$) FIR filter is shown in Fig. 1. The DA architecture consists of three small units: the shift register unit, the DA base unit, and the adder/shifter unit. The DA base unit can be more complicated than just a single LUT, but both the shift register and the adder/shifter units are common for different DA architectures.

## 3. PROPOSED ARCHITECTURES

### 3.1. LUT-less DA-OBC

For comparison purpose with the LUT-less DA architecture, which is explained in the following section, LUT-less DA-OBC is presented here. The LUT-less DA-OBC, shown in Fig. 2, is the result of applying LUT reduction technique of [5] one more time to the smallest 2-word LUT-based DA-OBC. However, the LUT-less DA-OBC in this form is still area inefficient, when implemented with the partial sum technique, since a global adder/shifter unit can't be used for the following two reasons. First, the adder with subtractor selector (drawn in gray color) in Fig. 2 is XORed with input signal spanned from the shift register unit. Second, the adder/shifter unit is multiplexed with $D_{init}$, which is the function of filter coefficients stored in each base unit.

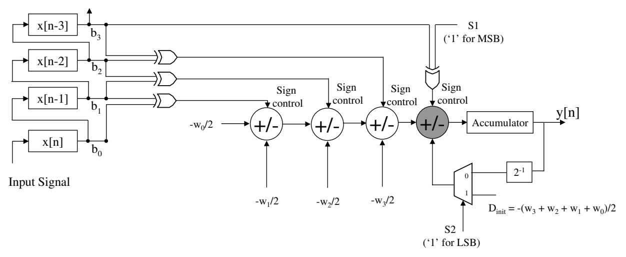


**Fig. 2**. Block diagram of LUT-less DA-OBC architecture for a 4-tap FIR filter.

Figure 3 shows the final LUT-less DA-OBC architecture re-organized to be area efficient when used with the partial sum technique for high-order filter implementation. Now, the single global adder/shifter unit can be used since the adder in the adder/shifter unit is no longer controlled by either input signal or $D_{init}$.

### 3.2. Proposed DA Architectures

The proposed DA architectures exploits the following symmetry of the LUT of the original DA

$$\mathrm{LUT}(1, b_{k-2}, \cdots, b_1, b_0) = \mathrm{LUT}(0, b_{k-2}, \cdots, b_1, b_0) + w_{k-1}, \quad (4)$$

where $k$ is the number of the address lines connected to the LUT. In Fig. 1, it can be seen that the lower half of LUT (locations whose

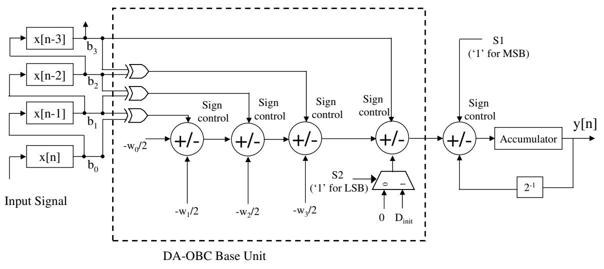


**Fig. 3**. Block diagram of the LUT-less DA-OBC for a 4-tap FIR filter. The architecture is re-organized to be area efficient for multiple memory bank technique.

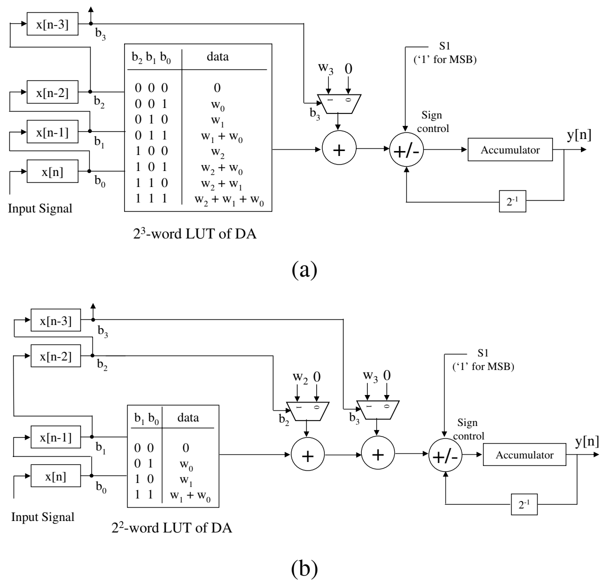


**Fig. 4**. Proposed DA architectures for a 4-tap FIR filter. (a) $2^3$-word LUT implementation of DA. (b) $2^2$-word LUT implementation of DA.

addresses have a 1 in the MSB) is the same with the sum of the upper half of LUT (locations whose addresses have a 0 in the MSB) and $w_3$ term. Hence, LUT size can be reduced by a factor of 2 with an additional 2x1 MUX and a full adder, as shown in Fig. 4(a). The LUT size can be further reduced to Fig. 4(b) by the same LUT reduction procedure since the LUT of Fig. 4(a) still satisfies symmetry property of (4). After several iterations of the LUT reduction, final LUT-less DA architecture for a 4-tap FIR filter implementation can be achieved as shown in Fig. 5.

## 4. PERFORMANCE ANALYSIS

Hardware complexity of the proposed DA architecture is compared with the original LUT-based DA, the DA-OBC, and the LUT-less DA-OBC using both transistor count estimate and FPGA synthesis result.

Comparing transistor count is a commonly used technique to evaluate silicon area of custom VLSI chips for different architectures. Digital logic functions, however, can be implemented in many ways depending on design optimization goals such as sili-

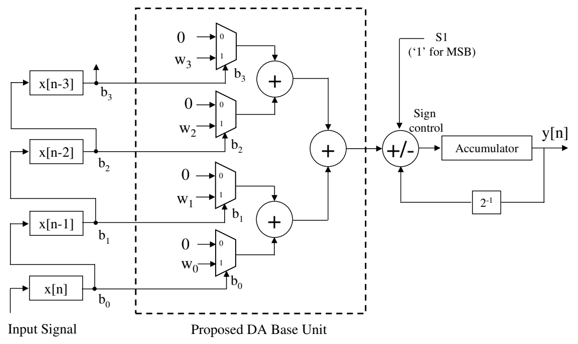


**Fig. 5**. Proposed LUT-less DA architecture for a 4-tap FIR filter.

con area, power consumption, and maximum frequency. For example, XOR logic can be implemented with 4 NAND gates, which are equivalent to 16 transistors, or with a full static CMOS logic, which is equivalent to 8 transistors [6, 7].

Table 1 lists the transistor count assumed in this paper for each logic function. Transistor count is estimated based on the area-optimized implementation of digital logic and it is the same as the assumptions made in [5]. Cost functions in Table 1 are defined as $C(a,b)=4(2\sum_{i=a}^{b-1}(b-i)+2^{b-a+1})$ and $D(a,b,c)=2^{b-a+1}\times c$.

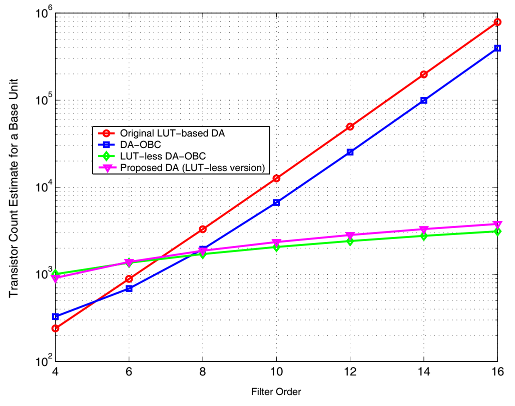
When one of the three inputs to the full adder has a fixed value (we assume fixed input is $C_{in}$ for convenience), 12 and 14 transistors for each $C_{in}=0$ and $C_{in}=1$ are used instead of 30 transistors for an ordinary full adder [7]. The occurrence rate of $C_{in}=0$ and $C_{in}=1$ is statistically assumed to be the same as $B_c/2$ bits out of $B_c$ bits. The adder with subtractor selection needs 8 more transistors for an extra 2x1 MUX and an inverter than the original full adder [7].

**Table 1**. Transistor counts for various digital logic functions.

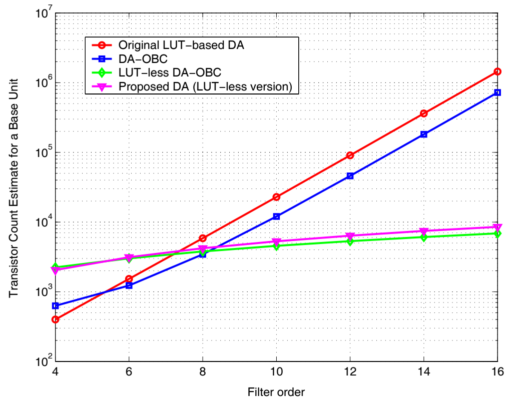
| Logic | | Transistor count |
|---|---|---|
| INV (1 bit) | | 2 |
| XOR (1 bit) | | 8 |
| 2x1 MUX (1 bit) | | 6 |
| Adder (fixed $C_{in}$) | $C_{in}$=0 ($B_c/2$ bit) | $12(B_c/2)$ |
| | $C_{in}$=1 ($B_c/2$ bit) | $14(B_c/2)$ |
| Adder ($B_c$ bit) | | $30B_c$ |
| Adder/Sub. (fixed $C_{in}$) | $C_{in}$=0 ($B_c/2$ bit) | $(12+8)(B_c/2)$ |
| | $C_{in}$=1 ($B_c/2$ bit) | $(14+8)(B_c/2)$ |
| Adder/Sub. ($B_c$ bit) | | $30B_c+8B_c$ |
| $2^k\times B_c$ (ROM) | Decoder | $C(1,k)$ |
| | Data | $D(1,k,B_c)$ |
| Register (1 bit) | | 16 |

Table 2 shows the area comparison of single base unit for various DA structures with $B_c$ and $k$ representing the word lengths of the original LUT and base unit size, respectively. The shift register and the adder/shifter units are not considered in Table 2 since they are common for all structures.

Figure 6 shows the transistor counts for various filter sizes $k$ from 4 to 16 with $B_c = 8$ and $B_c = 18$. In Fig. 6, both (a) and (b) show similar patterns with different orders of transistor



(a)



(b)

**Fig. 6**. Transistor count comparison plot regenerated from Table 2 for various filter size $k$. (a) $B_c = 8$. (b) $B_c = 18$.

counts. The hardware reduction of the LUT-less DA-OBC and the proposed DA architecture (LUT-less version) occurs around 6 to 8 filter taps and the hardware reduction rate grows significantly as the filter size increases. Figure 6 also shows that both the LUT-less DA-OBC and the proposed DA architectures are hardware efficient for high-order FIR filters compared to the original DA and the DA-OBC.

To illustrate the merits of the proposed DA architecture, the LUT-based DA, the LUT-less DA-OBC, and the proposed LUT-less architecture are physically implemented on an Altera Stratix EP1S80F1508C6 FPGA chip and the results are listed in Table 3. Randomly generated FIR filters of which filter orders vary from 4 to 1024 taps are tested for the case in which word length of the LUT, $B_c$, is 18 and base unit size, $k$, is 4.

Table 3 shows that the proposed DA architecture requires fewer logic elements (LEs) and memory than the original LUT-based DA; and fewer LEs and same amount of memory than the LUT-less DA-OBC for all filter orders. For instance, the proposed LUT-less DA architecture only requires 48% of the LEs and 16% of the memory of the original LUT-based DA implementation for a 1024-tap FIR filter. The addition clock cycles are only 2 assuming the adders inside the $k$-tap base unit are pipelined. Likewise, the proposed DA architecture also requires 37% of the LEs compared to the LUT-less DA-OBC for a 1024-tap FIR filter implementation with the same memory usage and throughput.

**Table 2**. Transistor count estimates for various $k$-tap base units. The shift register and the adder/shifter units are not considered since they are common for all structures.

| Logic functions | LUT-based DA (Fig. 1) | DA-OBC | The LUT-less DA-OBC (Fig. 3) | The proposed method (Fig. 5) |
|---|---|---|---|---|
| ROM decoder | $C(1,k)$ | $C(2,k)$ | 0 | 0 |
| ROM data | $D(1,k,B_c)$ | $D(2,k,B_c)$ | 0 | 0 |
| XOR | 0 | $8k$ | $8(k-1)$ | 0 |
| 2x1 MUX | 0 | $6B_c$ | $6B_c$ | $6k \times B_c$ |
| Register | 0 | $16B_c$ | $16B_c$ | 0 |
| Adder | 0 | 0 | 0 | $(k-1)\times 30B_c$ |
| Adder/Sub. $C_{in}=0$ | 0 | 0 | $(k-1)\times 10B_c$ | 0 |
| Adder/Sub. $C_{in}=1$ | 0 | 0 | $(k-1)\times 11B_c$ | 0 |
| Adder/Sub. | 0 | 0 | $38B_c$ | 0 |

**Table 3**. Comparison of FPGA implementation results for various filter sizes $K$ with $k=4$ and $B_c=18$.

| | | Filter size ($K$) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 4 | 16 | 64 | 128 | 256 | 512 | 1024 |
| LUT-based DA (Fig. 1) | LE | 272 | 551 | 1639 | 3056 | 5890 | 11547 | 22862 (100%) |
| | Memory | 344 | 1376 | 5504 | 11008 | 22016 | 44032 | 88064 (100%) |
| The LUT-less DA-OBC (Fig. 3) | LE | 300 | 667 | 2104 | 3984 | 7746 | 15259 | 30286 (132%) |
| | Memory | 56 | 224 | 896 | 1792 | 3584 | 7168 | 14336 (16%) |
| The proposed DA (Fig. 5) | LE | 210 | 367 | 887 | 1569 | 2946 | 5659 | 11086 (48%) |
| | Memory | 56 | 224 | 896 | 1792 | 3584 | 7168 | 14336 (16%) |

It should be noted that the LUT-less DA-OBC is slightly more area efficient in terms of transistor count, as shown in Fig. 6, but the proposed LUT-less DA architecture is much more area efficient than the LUT-less DA-OBC in terms of LE usage, as shown in Table 3. This result is not surprising since the transistor count comparison assumes that the LUT-less DA-OBC uses simplified adders to save transistor count.

## 5. CONCLUSIONS

New hardware-efficient DA architectures and a LUT-less DA-OBC architecture for high-order filters are presented. The proposed DA architectures reduce the memory usage by half at every iteration of LUT reduction exploiting the symmetry property of the LUT at the cost of the limited increase of the control circuit.

It is shown that the proposed DA architectures are hardware efficient for both custom VLSI and FPGA implementations, while the LUT-less DA-OBC is hardware efficient only for custom VLSI implementation. FPGA implementation result shows that, for a 1024-tap FIR filter implementation with $k=4$ and $B_c=18$, the proposed DA architecture (LUT-less version) saves 52% of LEs and 84% of memory over the original LUT-based DA, and also saves 63% of LEs over the LUT-less DA-OBC.

## 6. REFERENCES

[1] S. Yu and E. E. Swartzlander, "DCT implementation with distributed arithmetic," *IEEE Transactions on Computers*, vol. 50, no. 9, pp. 985–991, Sept. 2001.

[2] T.-S. Chang, C. Chen, and C.-W. Jen, "New distributed arithmetic algorithm and its application to IDCT," *IEE Proceedings Circuits, Devices and Systems*, vol. 146, no. 4, pp. 159–163, Aug. 1999.

[3] T.-S. Chang and C.-W. Jen, "Hardware-efficient implementations for discrete function transforms using LUT-based FPGAs," *IEE Proceedings Circuits, Devices and Systems*, vol. 146, no. 6, pp. 309–315, Nov. 1999.

[4] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Magazine*, vol. 6, pp. 4–19, July 1989.

[5] J. Choi, S. Shin, and J. Chung, "Efficient ROM size reduction for distributed arithmetic," in *Proceedings of the IEEE ISCAS*, Geneva, Switzerland, May 2000, vol. 2, pp. 61–64.

[6] U. Ko, P. T. Balsara, and W. Lee, "Low-power design techniques for high-performance CMOS adders," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 3, no. 2, pp. 327–333, June 1995.

[7] J. M. Rabaey, A. Chandrakasan, and B. Nilolic, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall, Upper Saddle River, NJ, 2002.