electronics

Article

A Variation-Aware Design Methodology for Distributed Arithmetic

Yue Lu, Shengyu Duan *, Basel Halak and Tom Kazmierski

Department of Electronics and computer science, University of Southampton, Southampton SO17 1BJ, UK; yl15g13@soton.ac.uk (Y.L.); bh9@ecs.soton.ac.uk (B.H.); tjk@ecs.soton.ac.uk (T.K.)
* Correspondence: sd5g13@soton.ac.uk; Tel.: +44-238-059-3520

Received: 13 December 2018; Accepted: 16 January 2019; Published: 18 January 2019

Abstract: Distributed arithmetic (DA) brings area and power benefits to digital designs relevant to the Internet-of-Things. Therefore, new error resilient techniques for DA computation are urgently required to improve robustness against the process, voltage, and temperature (PVT) variations. This paper proposes a new in-situ timing error prevention technique to mitigate the impact of variations in DA circuits by providing a guardband for significant (most significant bit) computations. This guardband is initially achieved by modifying the sign extension block and carefully gate-sizing. Therefore, least significant bit (LSB) computation can correspond to the critical path, and timing error can be tolerated at the cost of acceptable accuracy loss. Our approach is demonstrated on a 16-tap finite impulse respons (FIR) filter using the 65 nm CMOS process and the simulation results show that this design can still maintain high-accuracy performance without worst case timing margin, and achieve up to 32% power savings by voltage scaling when the worst case margin is considered with only 9% area overhead.

Keywords: PVT variations; robustness; distributed arithmetic VLSI; hardware design

1. Introduction

In recent years, much attention has been placed on Internet-of-Things (IoT) technologies [1,2], which is expected to bring benefits to numerous application areas including industrial wireless sensor node systems, and healthcare systems manufacturing. Specifically, an IoT network is created by integrating smart sensors onto a multitude of devices that share their data with other devices. Small-size sensors are needed to be installed in a multitude of places with cost-efficient hardware devices capable of performing Digital Signal Processing (DSP) using extremely low levels of energy [3,4].

Among state-of-the-art circuit techniques, distributed arithmetic has been widely used in the area-efficient and low-cost signal processing applications for convolution [5,6], transforms [7,8] and filtering [9,10]. Besides, DA-based architecture is also exploited as an excellent technique for implementing approximate computing [11], which has recently emerged as a promising approach to the energy-efficient design of IoT-related systems [12]. With the drastic scaling of CMOS technology, the design of a robust system is becoming a major concern [13–15]. There are still few approaches that can specifically solve the reliability issue on DA circuits, although DA is commonly regarded as a promising technique in current very-large-scale integration (VLSI) design. In Khairy's work [16], a novel N-modular redundancy (NMR) algorithm based on the maximum a posteriori (MAP) and the statistics of output bit-failure rate was proposed to tolerate faults. However, it is not cost-efficient in terms of hardware implementation. Ting et al. proposed an approximate distribute arithmetic architecture to improve the robustness at the cost of computation accuracy [17], which clock-gates the whole system when the timing error is predicted. If the circuit suffers from serious process variation or ageing effects, the circuit performance would be shaped drastically, as only $\frac{N}{2}$-bit effective computations remains for an N-bit arithmetic.

As for the conventional circuits, a large number of error resilient techniques [18] have been proposed to solve this reliability issue, such as algorithmic noise tolerance (ANT) [19], noise reduction unit (NRU) [20], RAZOR [21], and adaptive latency technique [22–24]. Among them, adaptive latency technique is a very popular fault tolerant technique which addresses device variability by tuning architectural latencies. In Choi's work [22], a novel FIR filter synthesis technique based on common-subexpression-elimination (CSE) algorithm is proposed where the fact that not all filter coefficients are equally important to obtain a "reasonably accurate" computation results are exploited. In this design, the critical paths of the computations involving the important coefficients are constrained to take a fixed number of adders while the later computational steps compute only the less important coefficient outputs. In this case, only the less important outputs are affected by process variation and voltage scaling. However, this approach does not provide an inherent error-detection mechanism and only works in the CSE-based circuit. ARM research group proposed a new error-resilient approach called path delay shaping (PDS) [23], where the critical paths are ensured to correspond to a group of least significant bit (LSB) result registers by using modified carry-merge adder and device sizing. While PDS is limited with choice of arithmetic unit, only those have minimum delay difference between the LSB and most significant bit (MSB) path would be suitable, such as Kogge-Stone adder. If the same idea is demonstrated on conventional carry-save adder tree or ripple adder, the circuit overhead would be considerable. In [24], Tiwari et al. transferred the time slack of the faster stages to the slow ones by skewing clock arrival times to latching elements in a pipeline processor. Therefore, timing violations due to process variations in one stage can be prevented by borrowing some extra time from another stage.

In this paper, the idea of adaptive latency computation is exploited on the distribute arithmetic for the first time. An error-resilient approach is proposed to bound the magnitude of timing errors with the presence of timing violations and demonstrated on a 16-tap FIR filter using 65-nm complementary metal oxide semiconductor (CMOS) process. The rest of the paper is organized as follows. Section 2 shows a brief review about distributed arithmetic based computation. Section 3 describes the principle of the proposed error-resilient approaches and VLSI implementation. The corresponding simulation results are analysed in Section 4, and Section 5 summarizes and concludes the paper.

2. MSB-First Distributed Arithmetic (DA) Computation

Distributed arithmetic (DA) computes the inner product of two vectors (one of which is a constant) in parallel, that is a bit-serial operation in nature. The main advantage of DA computation is that no multiply operations are required, which are replaced by precomputed look-up tables. The DA-based multiplication and accumulation (MAC) can be expressed as:

$$y_k = \sum_{k=0}^{L-1} h_k * x_k \tag{1}$$

where x_k and y_k represent the input and output data respectively. If we consider that each input data is an N-bit two's complement binary number and can be represented as:

$$x_k = -b_{k(N-1)}2^{N-1} + \sum_{i=1}^{N-1} b_{k(N-1-i)}2^{N-1-i} \tag{2}$$

where b_{ki} is the ith bit of the input data x_k. Accordingly, b_{k0} and $b_{k(N-1)}$ are the least significant bit and most significant bit (sign bit) respectively. Substituting Equation (1) in Equation (2):

$$y_k = -\sum_{k=0}^{L-1} h_k b_{k(N-1)}2^{N-1} + \sum_{i=1}^{N-1}\left[\sum_{k=0}^{L-1} h_k b_{k(N-1-i)}2^{N-1-i}\right] \tag{3}$$

If we replace the term in brackets by

$$q_i = \sum_{k=0}^{L-1} h_k * b_{k(N-1-i)} \tag{4}$$

The MAC operation can be represented by Equation (5), which shows the *i*th intermediate result of a MSB-first DA operation in terms of hardware implementation, the bit-serial input data $b_{0i}, b_{1i}, b_{2i}, ..., b_{ki}$, is used to form the look-up tables address and the data of filter coefficients h_k are stored on look-up tables. When the DA computation is implemented MSB first, it performs stochastically monotonic successive approximation characteristic. In other words, each successive intermediate value is closer to the final value in a probabilistic sense.

$$y_k = -q_{N-1}2^{N-1} + \sum_{i=1}^{N-1} q_i 2^{N-1-i} \tag{5}$$

Figure 1 shows a detailed example of a DA computation. This structure consists of a look-up table with 16 × N bits, each of the look-up table entries has corresponding registers storing input data, which are repackaged to form the memory address, most significant bit (MSB) first. The low right shows a sign extension block and $2N-1$-bit accumulator. The N-bit data fetched from the look-up table is scaled to $2N-1$-bit with $N-1$-bit sign extension. The accumulation register adds two times of its previous value to the current look-up table contents every clock cycle. The signal T_s is activated when the read-only memory (ROM) is addressed by the first bit of input data (sign). In this case, the adder subtracts the current ROM contents from the accumulator state. After N clock cycles, the $2N$-bit computation result is generated.

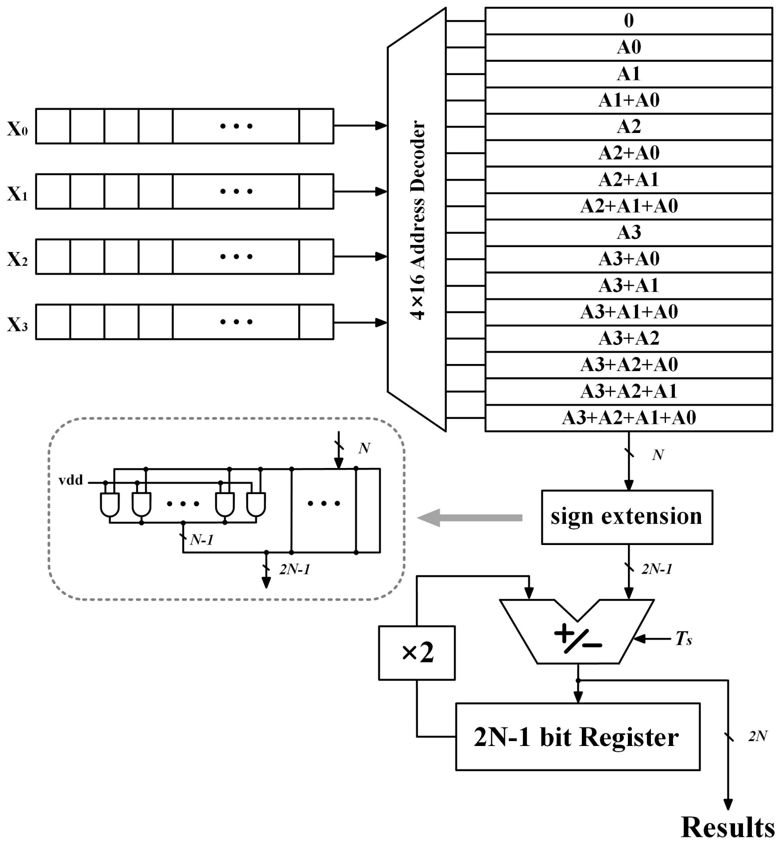

Figure 1. Conventional most significant bit (MSB)-first distributed arithmetic ROM and accumulator (RAC) structure.

As the DA computation is realized using the same arithmetic unit in each clock cycle, delay imbalance of the DA circuit is very close, which significantly differs from the conventional arithmetic circuits. Figure 2 shows the results of a static timing analysis. The longest and fastest path delay for each cycle is almost same and the difference between the mean path delay is very tiny.

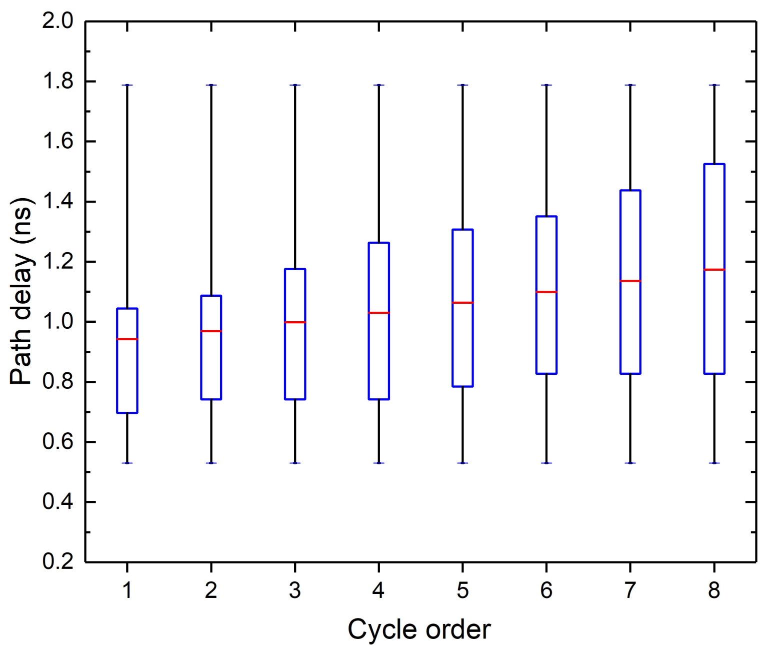

Figure 2. Distribution of path delay in each clock cycle.

3. Proposed Variation-Aware DA Computation

The above observations show us that the DA circuit consumes various clock cycles to generate the result bit in order and the corresponding critical path in each cycle has a very small delay difference. Based on this characteristic, we propose to trade a small amount of area overheads to increase the difference between the critical (corresponds to the LSB computation) and other paths, thereby bounding the magnitude of timing errors with the presence of variations.

3.1. Dynamic Sign Extension

The proposed approach relies on achieving a small guard-band between the critical path delay in the different clock cycles, this is firstly realized by modification of the sign extension block. As described in Section 2, the sign extension block is used to scale the N-bit data to $2N-1$-bit for the further shift accumulation. Since the $2N-1$-bit shift accumulation involves significant path delay every clock cycle, the delay difference is negligible in the conventional DA design. In order to create the expected delay imbalance, a dynamic sign extension block is proposed.

It can be found that only one increment bit is generated at each stage of the accumulation. Hence, there is no need to implement $2N-1$-bit accumulation every cycle. If the bit-width of the accumulate operation can be extended in sequence, the redundant additions can be avoided. To better illustrate, the accumulator circuit for an 8-bit DA computation can be seen in Figure 3. In the first cycle, the 8-bit ROM content is subtracted from zero and the result is stored in the registers, while the correct results would be acquired by a 15-bit addition in the eighth cycle.

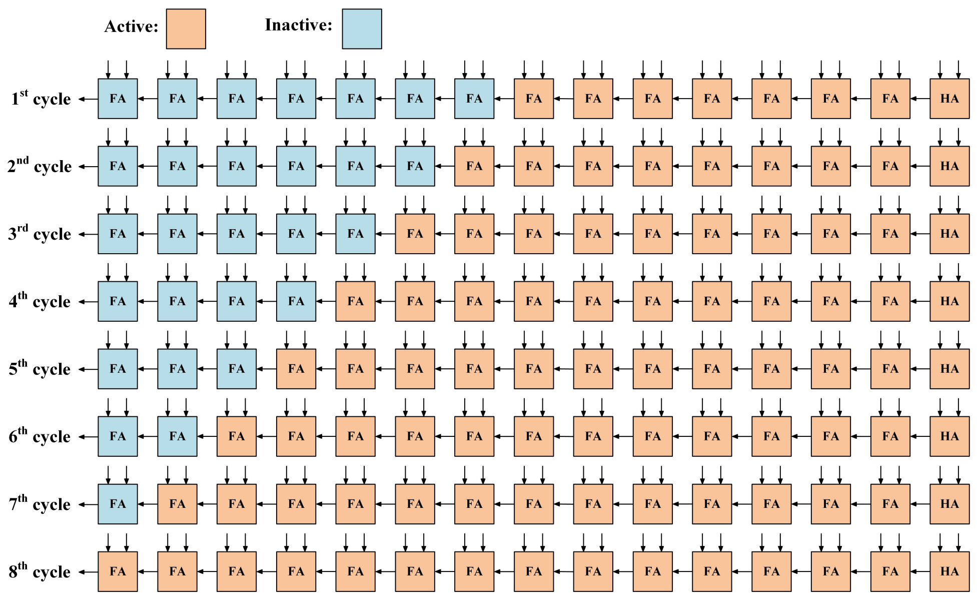

Figure 3. The state change of full adder in the accumulator circuit for an 8-bit distributed arithmetic (DA) computation.

The VLSI implementation of the proposed DA circuit can be seen in Figure 4. A log_2N-bit counter is used to record the cycle order, the counter signals $b_0, b_1, ..., b_{log_2N}$ are fed into a decoder block in parallel to generate an $N-1$-bit output signal named *select*. The logic function of the proposed decoder can be realized by several combinational logic gates and clearly illustrated by an 8-bit case study as shown in Table 1. Each bit of the signal *select* is connected to an individual AND gate respectively, while another input of theses gates is the sign bit of the current lookup-table content. Therefore, the signal *select* can activate the extended sign bit during each clock cycle, until the final cycle, the fetched data would be scaled to $2N-1$-bit.

Table 1. Binary representation of the signals *Counter* and *Select* along with the cycle order.

Cycle Order	Counter ($b_0b_1b_2$)	Signal Select [6:0]	Select [Cycle Order − 1]
1	000	0000000	$b_0 + b_1 + b_2$
2	001	0000001	$b_0 + b_1$
3	010	0000011	$(\overline{b_0} * b_1 * b_2) + b_0$
4	011	0000111	b_0
5	100	0001111	$b_0 * (b_1 + b_2)$
6	101	0011111	$b_0 * b_1$
7	110	0111111	$b_0 * b_1 * b_2$
8	111	1111111	*None*

The modification in the circuit block alter the path delay distribution, the critical path delay increases linearly along with the cycle order, as shown in Figure 5. The largest delay difference can reach up to 0.78 ns, which is nearly a 31% reduction.

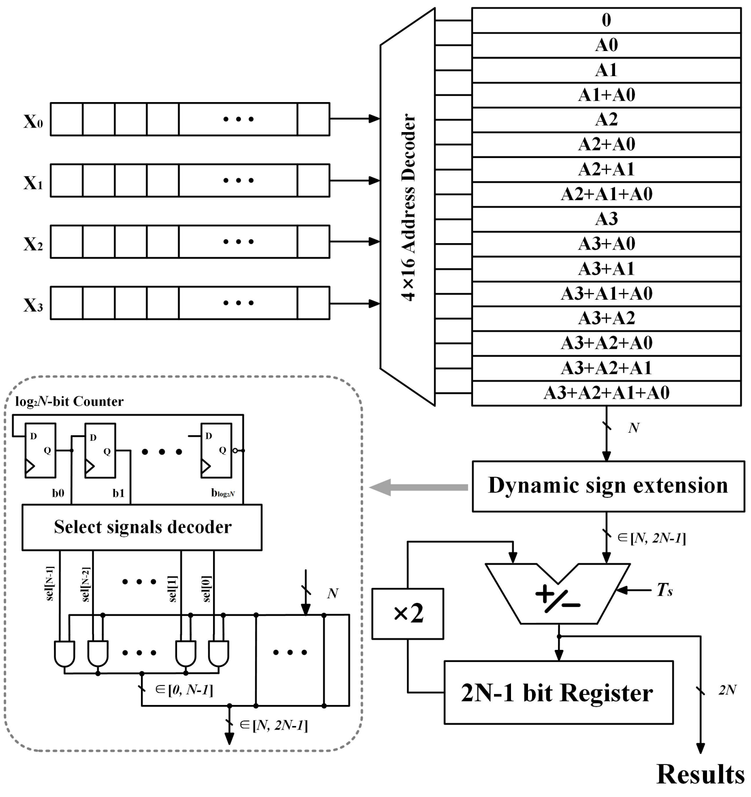

Figure 4. DA circuit with the modified dynamic sign extension block.

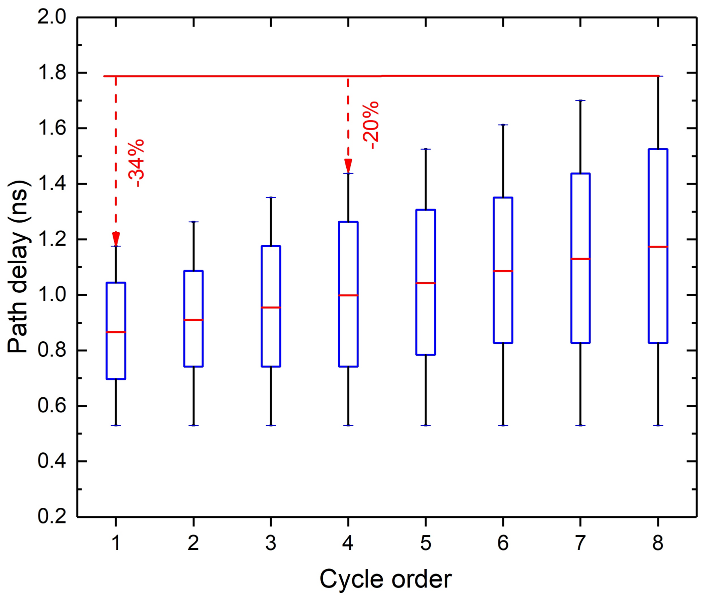

Figure 5. Distribution of path delay in each clock cycle with the proposed dynamic sign extension block.

3.2. Synthesis for Further Altering Path Delay Distribution

In the previous section, we proposed a dynamic sign extension block to create the delay imbalance. The next step is to further alter the path delay distribution at the synthesis level. The main idea is to replace some of the ripple adders by faster alternatives, such as carry-lookahead adders (CLA) and to realize the least area overhead by downsizing the rest ripple adders.

Assume an 8-bit DA circuit is required to have an $8 \times m$-bit multiplication accuracy. In such a case, any bit errors occurring on the most significant m bits of the result, due to the variability, are considered to be unacceptable. We denote the most significant m bits as $\text{MSB}_{m-1:0}$. We then give MSB_{m-1} and MSB_0 as the most and the least significant bits of $\text{MSB}_{m-1:0}$, respectively. As the errors are generally caused by the timing violations, a preset timing margin (*e.g.* n% of the current circuit delay) can be applied to compensate for the possible timing variations. We propose a synthesis method, presented in Algorithm 1, to ensure a desired accuracy for a DA circuit, considering the possible variations. Specifically, a required circuit delay (D_{req}) is firstly computed based on the maximum delay that may cause timing violations, and the preset timing margin, n%. As has been described, a longer signal path is constructed for the computation of a less significant bit, allowing that the results of the more significant bits take the precedence to be produced. In this way, if the arrival time of MSB_0 is not greater than D_{req}, we can conclude that the arrival times of all bits of $\text{MSB}_{m-1:0}$ are not greater than D_{req}, and thus have more than n% timing margin. For this reason, we only evaluate the arrival time of MSB_0, denoted as t, in the following steps. The optimization process is initiated if t is larger than the required delay, D_{req}. Two phases are included in the optimization process: adder replacement and gate downsizing.

Algorithm 1 Synthesis for accuracy specification

```
Require: the arrival times of $\text{MSB}_{m-1:0}$ have more than $n$% timing margin;
  procedure SYN()
      $D_{req} \leftarrow$ max delay/(1+$n$%);
      $t \leftarrow$ signal arrival time of $\text{MSB}_{m-1}$;
  // Adder replacement
      if $t > D_{req}$ then
          $\boldsymbol{Adder} \leftarrow$ all ripple adders on the critical path ending at $\text{MSB}_0$;
          $i \leftarrow 2$;                ▷ the number of 1-bit adders that can be replaced by a CLA
          repeat
              Replace $i$ 1-bit adders of $\boldsymbol{Adder}$ by a CLA
              $i$ increases;
          until $t \leq D_{req}$;
  // Gate downsizing
      $\boldsymbol{G_{inactive}} \leftarrow$ all gates that are irrelevant to $\text{MSB}_{m-1:0}$;
      Downsize $\boldsymbol{G_{inactive}}$.
```

During the adder replacement phase, the ripple adder located at the critical path arriving at MSB_0 are replaced by a CLA. A ripple adder consists of number of 1-bit adders and we denote all 1-bit adders of a ripple adder as $\boldsymbol{Adder}$. Here we partially replace the ripple adder to find the design with the least area increase satisfying D_{req}. Thus, the optimization starts by replacing the least number of the 1-bit adders. In this case, we assume the least number of the 1-bit adders that can be replaced by a CLA is two, while it could be different based on the minimum length of a CLA provided in a library. Afterwards, the arrival time of MSB_0 on the replaced critical path is checked: if t is still larger than D_{req}, the number of i is increased, indicating more 1-bit adders would be replaced by a CLA. The above steps are repeated until t becomes less than or equal to D_{req}, which suggests the arrival times of all bits of $\text{MSB}_{m-1:0}$ have more than n% timing margins.

By replacing the ripple adders with CLAs, the overall area of the circuit is increased. The area overhead can be reduced by downsizing some gates that do not affect the circuit accuracy. For instance, according to Figure 6, a 15-bit accumulator is required for an 8-bit computation, where m full adders are inactive when computing the most significant $8 - m$ bits. In such a case, these m adders can be downsized while the circuit can still at least obtain an $8 \times m$-bit accuracy. Thus, during the gate downsizing phase, we firstly identify the inactive gates based the given accuracy specification. We denote $\boldsymbol{G_{inactive}}$ as all gates that do not affect the result of $\text{MSB}_{m-1:0}$. The gates of $\boldsymbol{G_{inactive}}$ are then replaced by those having the same functions but smaller dimensions. Accordingly, the path delay distribution can be modified with the synthesis approach, example cases with $m = 4$ and $m = 6$ are presented in Figure 7.

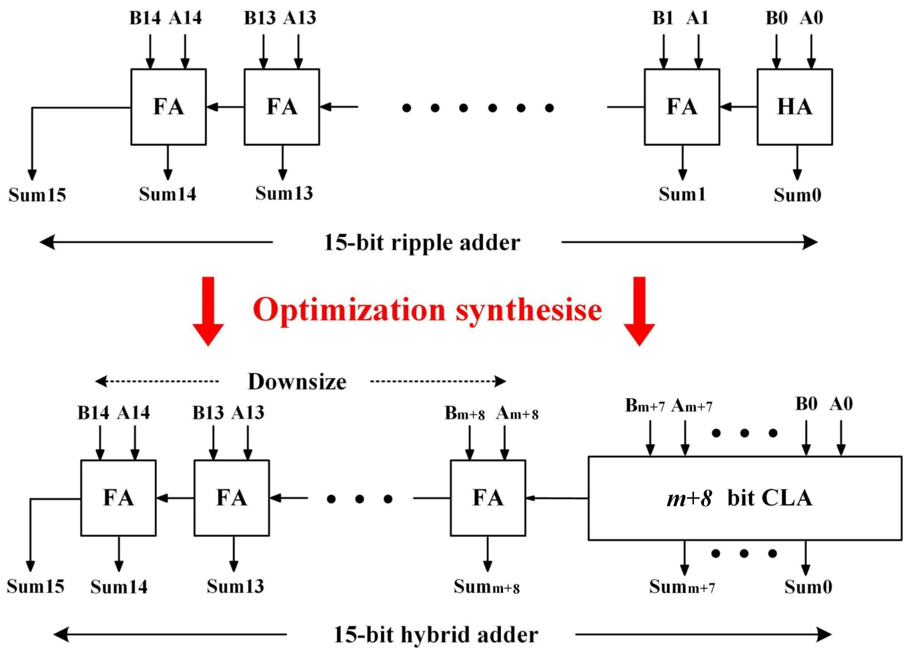

Figure 6. An example of the proposed synthesis approach.

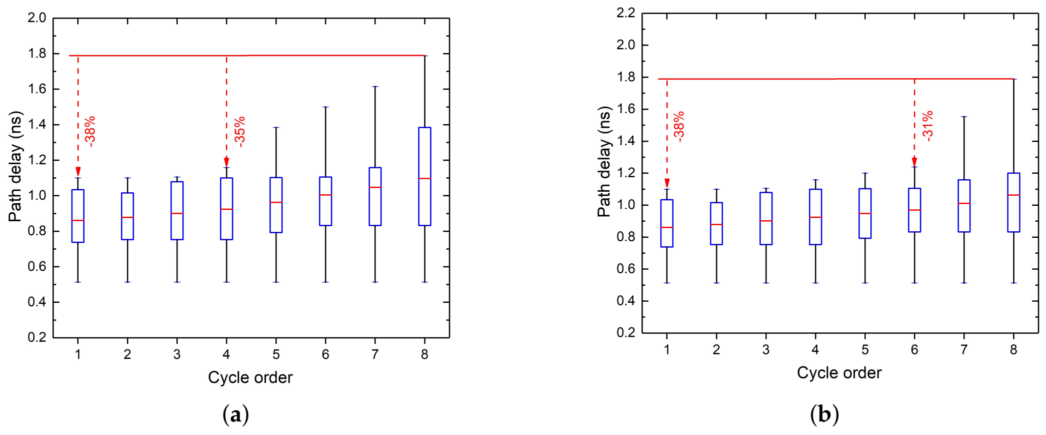

Figure 7. Distribution of path delay in each clock cycle with synthesis optimization, when (**a**) $m = 4$ (**b**) $m = 6$.

4. Case Study: FIR Filter VLSI Implementation

As mentioned in the previous section, the proposed approach can improve circuit robustness against timing errors at the cost of hardware area. To demonstrate this, the proposed architecture was applied on a digital FIR filter. This design is synthesized using the Synopsys Design Compiler with the ST 65 nm CMOS process.

We synthesize the FIR filter with the following coefficient set $\{-0.0215, 0.0106, 0.0270, 0.0084, -0.0261, -0.0147, 0.0390, 0.04225, -0.04285, -0.0905, 0.0490, 0.3138, 0.4507, 0.3138, 0.0490, -0.0905, -0.0428, 0.0422, 0.0390, -0.0147, -0.0261, 0.0084, 0.0270, 0.0105, -0.0215\}$. Both the proposed DA-based and conventional designs are simulated in the worst case (slow corner, 125°C) without any timing margin. The corresponding filter characteristics are shown in Figure 8, compared with the magnitude response of the original floating-point filter. The simulation results show that our proposed design can maintain an approximately correct function while the conventional design suffers from significant accuracy loss. In this design, the proposed approach trades extra hardware cost for error resilience over a given timing guard-band. For a 16-tap implementation, this approach results in approximately 9% extra hardware area.

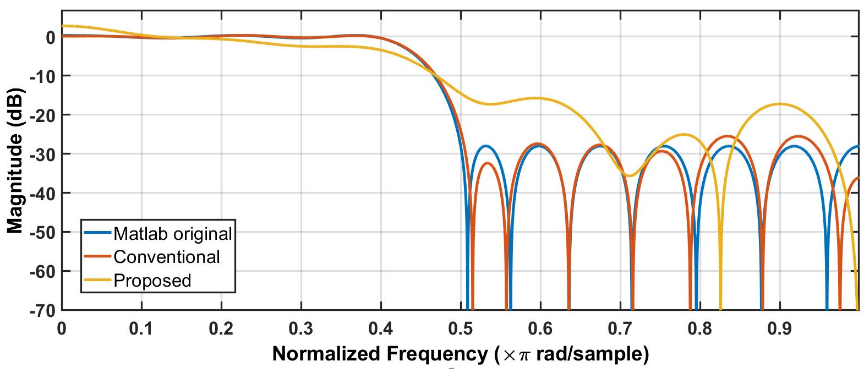

Figure 8. Frequency response of conventional design versus proposed design in the worst case (slow-slow (SS) process corner and 125 °C temperature).

Because the simulation results of power consumption and computation accuracy are dependent on the input samples and the coefficients, random stimulus is implemented over the full input dynamic range. The power consumption savings in our proposed FIR filter are shown in Figure 9 under different circumstances of slow-slow (SS), typical-typical (TT), fast-fast (FF) corners, and −40 °C, 25 °C, 125 °C. Compared with the TT process corner, 25 °C, and 1.2 V supply voltage, the proposed approach achieves 21%, 18%, and 15% power savings for best (SS corner, −40 °C), nominal (TT corner, 25 °C) and worst (FF corner, 125 °C) cases, respectively (at 1.00 V). When the voltage is reduced to 0.90 V, a maximum power reduction of up to 32% can be obtained. To better present the trade-off between the power consumption and computation accuracy, the error probability (P_e) serves as the measurement metric to evaluate the accuracy loss, which means it is probable that the output of computation differs from the correct one. Figure 9 also shows the error probability versus supply voltage, starting from the nominal 1.10 V, which includes the full worst-case design margin. If this margin is removed in the nominal case (TT, 25 °C) using DVS, high-accuracy operations can still be retained. The point of first failure (PoFF) occurs at 1.06 V, where the paths from the LSB group start to fail, but the corresponding error rate is extremely low. Beyond 0.90 V, MSBs paths also begin to fail, which leads to a rapid decline in computation accuracy.

To implement a tentative comparison, the conventional DA-based design is up-sized to achieve the same hardware cost as the proposed one. Hence, more timing guardband is provided via gate sizing for the conventional design. The error rate comparison between the gate-sized conventional and proposed design with 0.9 V supply voltage using 1000 test samples can be seen in Figure 10. The conventional gate-sized design would suffer from more than 50% accuracy loss some time while at least 75% computation accuracy can still remain using the proposed error-mitigation method.

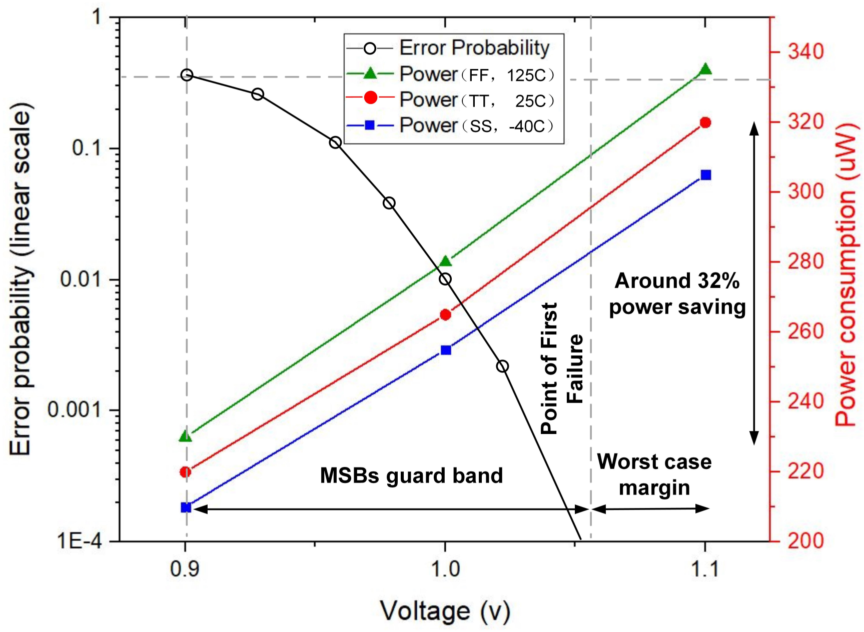

Figure 9. Power dissipation and error probability versus supply voltage with different process corners and temperature.

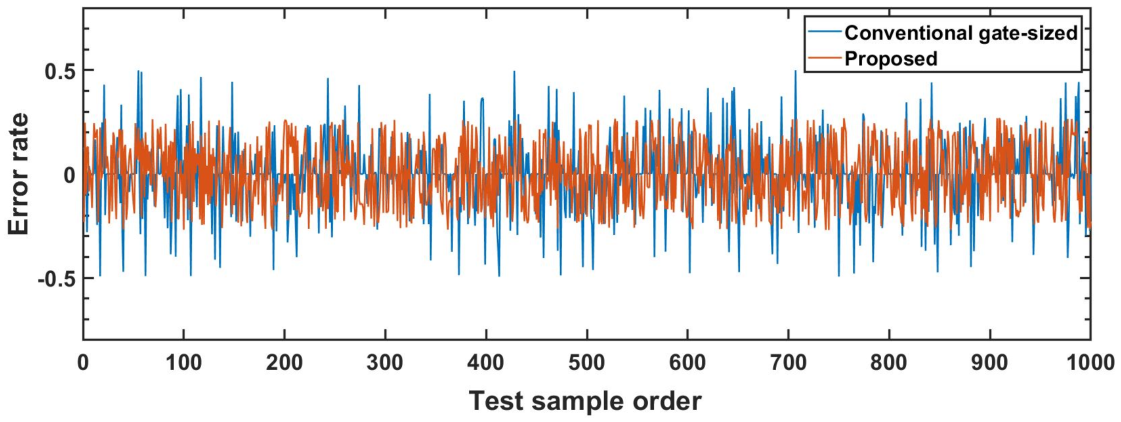

Figure 10. The error rate comparisons between the gate-sized conventional and proposed design with 0.9 V supply voltage using 1000 test samples.

5. Conclusions

In this paper, a novel approach to distributed arithmetic (DA) is proposed to mitigate the effects of process, voltage, and temperature (PVT) variations. The bit-slice accumulation of DA is modified as MSB-first, it can alter the path delay distribution, where the MSB path has the shortest delay. Additionally, extra timing slack is provided for computations of the MSB group by a modified sign extension block and carefully gate-sizing, thereby providing extra guardband against the variation-induced timing violations. A case study of FIR implementation shows that this design outperforms the traditional gate-sized counterpart in terms of computation accuracy with the same hardware overhead (9%) while it can also achieve significant power savings up to 32% under different operating conditions when the worst case margin is included.

Author Contributions: Y.L. and S.D. designed, simulated and measured the circuits described in this manuscript under the supervision of T.K., and B.H contributed to the technical discussions during the design time.

Funding: This research received no external funding

Acknowledgments: The authors would like to thank Mengjia Ye and Park Lu, for partially supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Muhic, I.; Hodzic, M. Internet of Things: Current Technological Review and New Low Power Wireless Sensor Network Protocol Proposal. *Southeast Eur. J. Soft Comput.* **2014**, *3*. [CrossRef]
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
3. Golanbari, M.S.; Tahoori, M.B. Runtime adjustment of IoT system-on-chips for minimum energy operation. In Proceedings of the 55th Annual Design Automation Conference, San Francisco, CA, USA, 24–28 June 2018; p. 145.
4. Kiamehr, S.; Ebrahimi, M.; Golanbari, M.S.; Tahoori, M.B. Temperature-aware dynamic voltage scaling to improve energy efficiency of near-threshold computing. *IEEE Trans. Very Large Scale Integr. Syst.* **2017**, *25*, 2017–2026. [CrossRef]
5. Licciardo, G.D.; Cappetta, C.; Di Benedetto, L.; Vigliar, M. Weighted Partitioning for Fast Multiplierless Multiple-Constant Convolution Circuit. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 66–70. [CrossRef]
6. Panwar, M.; Padmini, J.; Acharyya, A.; Biswas, D. Modified distributed arithmetic based low complexity CNN architecture design methodology. In Proceedings of the 2017 European Conference on Circuit Theory and Design (ECCTD), Catania, Italy, 4–6 September 2017; pp. 1–4.
7. Xie, J.; Meher, P.K.; He, J. Hardware-efficient realization of prime-length DCT based on distributed arithmetic. *IEEE Trans. Comput.* **2013**, *62*, 1170–1178. [CrossRef]
8. Martina, M.; Masera, G.; Roch, M.R.; Piccinini, G. Result-biased Distributed-Arithmetic-based filter architectures for approximately computing the DWT. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2015**, *62*, 2103–2113. [CrossRef]
9. Singhal, S.K.; Mohanty, B.K. Efficient Parallel Architecture for Fixed-Coefficient and Variable-Coefficient FIR Filters Using Distributed Arithmetic. *J. Circuits Syst. Comput.* **2016**, *25*, 1650073. [CrossRef]
10. Park, S.Y.; Meher, P.K. Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic. *IEEE Trans. Circuits Syst. II Express Briefs* **2013**, *60*, 346–350. [CrossRef]
11. Venkatachalam, S.; Ko, S.B. Approximate Sum-of-Products Designs Based on Distributed Arithmetic. *IEEE Trans. Very Large Scale Integr. Syst.* **2018**, *26*, 1604–1608.
12. Mittal, S. A survey of techniques for approximate computing. *ACM Comput. Surv.* **2016**, *48*, 62. [CrossRef]
13. Radfar, M.; Singh, J. A yield improvement technique in severe process, voltage, and temperature variations and extreme voltage scaling. *Microelectron. Reliab.* **2014**, *54*, 2813–2823. [CrossRef]
14. Islam, A.; Hasan, M. A technique to mitigate impact of process, voltage and temperature variations on design metrics of SRAM Cell. *Microelectron. Reliab.* **2012**, *52*, 405–411. [CrossRef]
15. Golanbari, M.S.; Gebregiorgis, A.; Moradi, E.; Kiamehr, S.; Tahoori, M.B. Balancing resiliency and energy efficiency of functional units in ultra-low power systems. In Proceedings of the 23rd Asia and South Pacific Design Automation Conference, Jeju, Korea, 22–25 January 2018; pp. 637–644.
16. Khairy, M.S.; Gholamipour, A.; Kurdahi, F.J.; Eltawil, A.M. Reliable low power Distributed Arithmetic filters via N-Modular Redundancy. In Proceedings of the 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, USA, 4–7 November 2012; pp. 621–625.
17. Ting, Y.H.; Lin, T.J.; Chang, C.C.; Hu, C.C.; Yeh, C.; Wang, J.S. Approximate Distributed Arithmetic for Variable-Latency Table Lookup. In Proceedings of the 2017 New Generation of CAS (NGCAS), Genova, Genoa, 6–9 September 2017; pp. 137–140. doi:10.1109/NGCAS.2017.39 [CrossRef]
18. Alam, M. Reliability-and process-variation aware design of integrated circuits. *Microelectron. Reliab.* **2008**, *48*, 1114–1122. [CrossRef]
19. Zhang, S.; Shanbhag, N.R. Embedded algorithmic noise-tolerance for signal processing and machine learning systems via Data Path Decomposition. *IEEE Trans. Signal Process.* **2016**, *64*, 3338–3350. [CrossRef]
20. Vaseghi, S.V. *Advanced Digital Signal Processing and Noise Reduction*; John Wiley & Sons: Chichester, West Sussex, UK; 2008, ISBN 978-0-470-75406-1 .

21. Ernst, D.; Kim, N.S.; Das, S.; Pant, S.; Rao, R.; Pham, T.; Ziesler, C.; Blaauw, D.; Austin, T.; Flautner, K.; et al. Razor: A low-power pipeline based on circuit-level timing speculation. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, Washington, DC, USA, 3–5 December 2003; p. 7.
22. Choi, J.H.; Banerjee, N.; Roy, K. Variation-aware low-power synthesis methodology for fixed-point FIR filters. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2009**, *28*, 87–97. [CrossRef]
23. Whatmough, P.N.; Das, S.; Bull, D.M.; Darwazeh, I. Circuit-level timing error tolerance for low-power DSP filters and transforms. *IEEE Trans. Very Large Scale Integr. Syst.* **2013**, *21*, 989–999. [CrossRef]
24. Tiwari, A.; Sarangi, S.R.; Torrellas, J. *ReCycle: Pipeline Adaptation to Tolerate Process Variation*; ACM SIGARCH Computer Architecture News; ACM: New York, NY, USA, 2007; Volume 35, pp. 323–334.