A Multi-Stage Fault-Tolerant Multiplier with Triple Module Redundancy (TMR) Technique

Ping-Yeh Yin*, Yuan-Ho Chen†, Chih-Wen Lu‡, Shian-Shing Shyu§, Chung-Lin Lee†, Ting-Chia Ou‡, and Yo-Sheng Lin*  
*Institute of Nuclear Energy Research Atomic Energy Council, Taiwan.  
†Department of Electrical Engineering, National Chi Nan University, Taiwan.  
‡Department of Engineering and System Science, National Tsing Hua University, Taiwan.  
§Department of Information & Computer Engineering, Chung Yuan Christian University, Taiwan.  
Email: yhchen@ice.cycu.edu.tw

Abstract—This study proposes a multistage fault-tolerant (MSFT) scheme for fixed-width array multipliers. The proposed MSFT multipliers divide the array multiplier into multiple stages, and implement a single processing element (PE) by regarding multiple computation cycles to achieve a low area design. To tolerate the fault that occurs in the integrated circuit, three redundancy replicas of PE (TMR-PE) architecture are proposed. Thus, the MSFT multiplier employs the TMR-PEs to achieve a low-cost fault-tolerant design. The TMR-PEs are designed by using compressors with multiple operands, such as 4-2 compressors or other compressors with more operands, to reduce computation cycles and speed up the execution time. Because of implementation with a 0.18-μm CMOS process, the long word-length MSFT multiplier saves a significant amount of the circuit area. The proposed 64 × 64 MSFT multiplier has only 13% of the circuit area and 3% of the delay overhead of the original multiplier. Based on the measurements of the area-delay product (AT) metric, the value of the 64 × 64 MSFT multiplier is only 0.21 fold of the value of the original multiplier. Consequently, the proposed MSFT multipliers achieve a low-cost fault-tolerant design.

Keywords—Fixed-width array multiplier, Multistage fault-tolerant (MSFT) multiplier, Triple module redundancy.

I. INTRODUCTION

The multiplier is an important component for digital signal processing (DSP) systems [1]-[4], such as fast Fourier transform (FFT) [3] and the finite impulse response (FIR) filter [4]. Because of the large transistors that are integrated in a chip to achieve high-speed computing in the advanced very-large-scale integration (VLSI) process, any fault damages the function of the operation circuit. In this manner, high reliability becomes a critical issue in VLSI design.

In VLSI systems, the high-speed computing circuit does not transient faults easily. Therefore, these faults must be detected simultaneously with the operation of the circuit. Generally, self-checking circuits [5]-[9] and built-in fault detection and correction circuits [10]-[20] can detect transient faults. Because of the built-in fault detection and correction circuits, transient faults can be tolerated in the run-time.

In recent years, numerous researchers work on fault-tolerant techniques for multipliers [10]-[20]. The REcom-puting with Circularly shifted Operands (RECO) technique is applied to multipliers to detect errors in the run-time [15]. Because of the extra bit-slices added to accommodate multiplier shifts, the circularly-shift approach is unsuitable for array multipliers. Tolerant array multipliers are introduced in [16]-[17]. Namba et al. present a defect-tolerant Wallace multiplier [16], and Chen et al. use a bi-directional operation (BIDO) scheme to achieve concurrent error detection in array multipliers [17]. Time-shared fault-tolerant multipliers are introduced in [18]-[20]. The time-shared TMR method (TSTMR) is presented in [19]. An alternative fault-tolerant multiplier using a partitioning technique is presented in [20]. Chen et al. divide the multiplier in to m parts and adopt the TMR technique to achieve error correction with good area-delay performance.

This paper proposes a multistage fault-tolerant multiplier. The proposed MSFT divides the multiplier operation into multiple stages, and employs a single processing element (PE) to implement the main computation stage as a low-cost design. To tolerate the transient faults, the PE is reproduced as three replicas called TMR-PEs to achieve a fault-tolerant design. The TMR-PEs utilize 3-2 compressors to achieve a low-cost design. However, the delay overhead is increased because of the extra multiplexors and voter in the multiple computation cycles. Thus, more operands compressors [6] and [21]-[24] are adopted to implement the proposed TMR-PE. In this paper, the TMR-PEs are designed by using 4-2 compressors as an example. Based on the circuit implementation result, the 64 × 64 MSFT multiplier achieve extremely low area cost with only a slight delay penalty in comparison to the original multiplier. A superior performance in the value of area-delay product (AT) is observed in the proposed MSFT multipliers. The 64 × 64 MSFT multiplier is only 0.21 fold of the original multiplier for the AT value. Consequently, the proposed MSFT multipliers achieve superior performance in area cost, with a slight delay penalty for the fault-tolerant design.

The remainder of this paper is organized as follows: In Section II, multistage fault-tolerant (MSFT) multipli-
ers are described, including the background of fixed-width multiplier, MSFT with 3-2 and 4-2 compressors, and the systematic steps of the MSFT design. Comparisons and discussions are presented in Section III, and Section IV offers a conclusion.

II. PROPOSED MULTISTAGE FAULT-TOLERANT (MSFT) MULTIPLIER

A. Fixed-width array multiplier

The $2L$-bit products $P$ can be expressed in unsigned representation, as follows:

$$X = \sum_{i=0}^{L-1} x_i \cdot 2^i$$

$$Y = \sum_{i=0}^{L-1} y_i \cdot 2^i$$

$$P = X \times Y$$

By expressing the multiplier operation, the partial products of $L \times L$ fixed-width multiplier are shown in Fig. 1. A regular structure is observed in the partial product arrays. The $L$ product rows shift one bit for each neighbor product row. The proposed MSFT multiplier adopts this phenomenon to insert fault tolerance into the multiplier design.

B. Proposed MSFT multiplier with 3-2 compressors

Because of the phenomenon of regular structure in fixed-width array multipliers, the $L \times L$ multiplier can be divided into $L-2$ computation stages, as shown in Fig. 2 for the $8 \times 8$ multiplier as an example. Each computation stage shown in Fig. 2 compresses three operands into two operands, which are the sums and carries. The same operation in each computation stage occurs in this multistage multiplier.

Therefore, by exploiting one operation block by considering $L-2$ clock cycles, the product results can be obtained.

Generally, the fault-tolerant with the triple module redundancy (TMR) method increases area by more than threefold compared with the original circuit. Therefore, the proposed MSFT multiplier employs a single processing element (PE) to achieve a small area design. To alleviate the uncertain fault, the PE is implemented in three replicas and followed by a voter designed in a manner similar to the TMR method. The overall architecture of the proposed MSFT multiplier is illustrated in Fig. 3(a).

1) Pre-processing (Pre-PE) module: Figure 3(b) shows the proposed pre-processing (Pre-PE) module for the MSFT multiplier. Three AND2 rows (each row contains $L$ two input AND gates) produce the partial products for each shading block in Fig. 3(b). In the first cycle, Pre-PE generates $\{p_{k,0}, p_{k,1}, p_{k,2}\}$ ($0 \leq k \leq L-1$) three partial product rows and feeds them into the TMR stage. However, during cycles $2 \sim (L-2)$, only one partial product row must be generated, and two AND2 row gates are idled to save computation power.

2) TMR-PE module: The triple module redundancy (TMR) technique is applied into the main computation with three PE replicas. Figure 3(c) illustrates the PE module, which consists of full-adders (FAs) and half-adders (HAs). In the first cycle, triple PE modules (TMR-PEs) sum three partial product rows $\{p_{k,0}, p_{k,1}, p_{k,2}\}$ ($0 \leq k \leq L-1$), and sum another partial product row and other operands, which is the sums and carries of the previous cycle, during cycle $2 \sim (L-2)$.

3) Carry propagation adder (CPA) module: The first $2L$-bit registers store the sums and carries from the voter after TMR-PE. In the final cycle, the sums and carries from the first $2L$-bit registers feed into the second $2L$-bit registers. Thus, the CPA can use the next $(L-2)$ cycle period to execute the final summation to avoid delay overhead. The CPA module uses the parallel pre-fixed adder to achieve a high-speed computation in the final cycle.

C. Proposed MSFT multiplier with 4-2 compressors

The proposed MSFT multiplier employs 3-2 compressors, which compress three operands into two operands, to implement the TMR-PEs module in the above section. Therefore, $L-2$ cycles are required to complete the multiplier operation and produce one partial product row for each cycle. Injecting the fault tolerance into the proposed MSFT, the speed of the proposed MSFT multiplier degrades because of the extra multiplexors and voter. Thus, the delay of the MSFT multiplier is increased because of the cycles of multistage computation. To reduce the cycles of the computation, the TMR-PEs are implemented by using 4-2 compressors that compress four operands into two operands. In this manner, the computation cycle is reduced with a small area penalty because of the more gate area in the 4-2 compressor.
compared with the 3-2 compressor. The entire architecture of the proposed MSFT multiplier with 4-2 compressors is illustrated in Fig. 4.

D. Design of the proposed MSFT multipliers

To implement MSFT multipliers, four systematic steps are proposed:

1) Choose the main component to implement TMR-PEs. The main component can be the 3-2 compressor, the 4-2 compressor, or another compressor with more operands.

2) Partition the partial products into a multistage structure, such as in Figs. 2 and 5, for 3-2 and 4-2 compressors, respectively.

3) Design the Pre-PE, TMR-PEs, and voter according to the choice of Step 2).

4) Finally, use two 2L-bit registers to store the result from the voter, and used the (L + 1)-bit CPA to produce the final products.

With these four systematic steps, the proposed MSFT multipliers can be implemented easily, and it requires (L – 2) or \(\lceil(L – 2)/2\rceil\) cycles to complete the \(L \times L\) multiplier with TMR-PEs designed with 3-2 or 4-2 compressors, respectively. The ceiling function \(\lceil\cdot\rceil\) maps \(\cdot\) to the smallest following integer. For speed consideration, compressors with more operands can be applied to TMR-PEs to reduce the computation cycles. Regarding as an example \(N\)-2 compressors, which compress \(N\) operands to two operands, the computation cycle can be reduced to \(\lceil(L – 2)/(N – 2)\rceil\) cycles.

III. COMPARISONS AND DISCUSSIONS

This section presents a discussion of important issues, such as area, delay, and the power of the proposed MSFT multiplier.

A. Circuit characteristics comparisons

Comparisons of circuit characteristics, area, delay, and power for the proposed MSFT multiplier with the original (unused fault-tolerant multiplier) and TMR multipliers are shown in Table I. The proposed MSFT multiplier can be directly observed to have the smallest circuit area compared with the TMR multiplier. Particularly in long word-length, the MSFT multiplier has only 13% area cost compared with the original multiplier’s \(L = 64\). However, a 75% delay overhead is incurred because of the multistage fault-tolerance inserting. To handle the delay overhead, an MSFT
Figure 4. Architecture of the $8 \times 8$ MSFT multiplier with 4-2 compressors.

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Original with TMR</th>
<th>MSFT with 3-2 Compressors</th>
<th>MSFT with 4-2 Compressors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L = 8$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area ($\mu m^2$)</td>
<td>4737</td>
<td>100%</td>
<td>14689</td>
<td>310%</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>4.49</td>
<td>100%</td>
<td>4.73</td>
<td>105%</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>2.39</td>
<td>100%</td>
<td>7.37</td>
<td>308%</td>
</tr>
<tr>
<td>AT</td>
<td>1.00</td>
<td></td>
<td>3.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$L = 16$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area ($\mu m^2$)</td>
<td>20544</td>
<td>100%</td>
<td>62590</td>
<td>305%</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>9.71</td>
<td>100%</td>
<td>9.95</td>
<td>102%</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>15.31</td>
<td>100%</td>
<td>46.54</td>
<td>304%</td>
</tr>
<tr>
<td>AT</td>
<td>1.00</td>
<td></td>
<td>3.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$L = 32$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area ($\mu m^2$)</td>
<td>85582</td>
<td>100%</td>
<td>256661</td>
<td>302%</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>19.33</td>
<td>100%</td>
<td>19.57</td>
<td>101%</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>80.90</td>
<td>100%</td>
<td>244.33</td>
<td>302%</td>
</tr>
<tr>
<td>AT</td>
<td>1.00</td>
<td></td>
<td>3.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$L = 64$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area ($\mu m^2$)</td>
<td>348287</td>
<td>100%</td>
<td>1048694</td>
<td>301%</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>39.43</td>
<td>100%</td>
<td>39.67</td>
<td>101%</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>425.00</td>
<td>100%</td>
<td>1278.80</td>
<td>301%</td>
</tr>
<tr>
<td>AT</td>
<td>1.00</td>
<td></td>
<td>3.03</td>
<td></td>
</tr>
</tbody>
</table>

Table I: Comparisons of area, delay, AT, and power with other methods.
using compressors with more operands, such as 4-2 compressors, is proposed. The delay overhead is reduced to 23% with only 4% area penalty for the 64-bit MSFT multiplier. Consequently, a trade-off is required between area cost and computation time. The value of area-delay product (AT) is a good metric to evaluate the circuit performance. The AT values of the fault-tolerant multipliers normalized to the original multiplier, which is not inserted with a fault-tolerant design, are shown in Table I. The proposed MSFT multipliers with 4-2 compressors outperform other multipliers in AT values. Consequently, the proposed MSFT multipliers achieve a low-cost design with only a slight delay penalty.

### B. Chip implementations

For the proposed MSFT multiplier, the Synopsys Design Compiler is applied to synthesize the RTL design of the 64 × 64 MSFT multiplier, and the Cadence SoC Encounter is adopted for placement and routing (P&R). Implemented in a 1.8-V TSMC 0.18-μm 1P6M CMOS process, the proposed 64 × 64 MSFT multiplier is operated at 641 MHz. Because of the 31 computation cycles, the proposed 64 × 64 MSFT multiplier consumes 48.36 (= 1.56 × 31) ns to complete a multiplier operation, and the power consumption is 25.7 mW. The core layout and characteristics are shown in Fig. 6 and Table II, respectively. The test module (TM) in the layout block tests the proposed MSFT multiplier. In test mode, the external data can input into and output from the TM serially. The TM feeds and captures the input and output data of the MSFT multiplier paralleled in the function mode.

### IV. Conclusions

This paper proposes a multistage fault-tolerant multiplier. The proposed \( L \times L \) MSFT multiplier employs single stage PE by considering \( [(L - 2)/2] \) clock cycles to achieve a low area design. The MSFT utilizes three PE modules called as TMR-PEs to achieve a fault-tolerant design. In this manner, a low-cost fault-tolerant multiplier is achieved. For the 64 × 64 MSFT multiplier, circuit area is only 13% of the area of the original multiplier. The value of AT in the 64 × 64 MSFT multiplier with 4-2 compressors is only 0.21 normalized to the value of the original multiplier. Therefore, the proposed MSFT multipliers achieve a superior performance in area with only a slight delay penalty for a fault-tolerant design.

---

**Figure 5.** Stages of the 8 × 8 MSFT multiplier with 4-2 compressors.

**Figure 6.** Chip layout of the 64 × 64 MSFT multiplier.

**Table II**

<table>
<thead>
<tr>
<th>Chip Characteristics of the Proposed 64 × 64 MSFT Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Technology</td>
</tr>
<tr>
<td>Supply Voltage</td>
</tr>
<tr>
<td>Critical Delay</td>
</tr>
<tr>
<td>Core Area</td>
</tr>
<tr>
<td>Gate Counts</td>
</tr>
<tr>
<td>Power Consumption</td>
</tr>
</tbody>
</table>

**ACKNOWLEDGMENT**

The authors would like to thank the National Chip Implementation Center (CIC), Taiwan, for providing the electronic design automation tools.

**REFERENCES**


