Design and Implementation of Cost-Effective Probabilistic-Based Noise-Tolerant VLSI Circuits

I-Chyn Wey, Student Member, IEEE, You-Gang Chen, Student Member, IEEE, Chang-Hong Yu, An-You Andy Wu, Member, IEEE, and Jie Chen, Senior Member, IEEE

Abstract—As the size of CMOS devices is scaled down to nanometers, noise can significantly affect circuit performance. Because noise is random and dynamic in nature, a probabilistic-based approach is better suited to handle these types of errors compared with conventional CMOS designs. In this paper, we propose a cost-effective probabilistic-based noise-tolerant circuit-design methodology. Our cost-effective method is based on master-and-slave Markov random field (MRF) mapping and master-and-slave MRF logic-gate construction. The resulting probabilistic-based MRF circuit trades hardware cost for circuit reliability. To demonstrate a noise-tolerant performance, an 8-bit MRF carry-lookahead adder (MRF_CLA) was implemented using the 0.13-μm CMOS process technology. The chip measurement results show that the proposed master-and-slave MRF_CLA can provide a 7.00 × 10⁻¹⁰ bit-error rate (BER) under 10.6-μV signal-to-noise ratio, while the conventional CMOS_CLA can only provide 8.84 × 10⁻⁹ BER. Because of high noise immunity, the master-and-slave MRF_CLA can operate under 0.25 V to tolerate noise interference with only 1.9 μW/MHz of energy consumption. Moreover, the transistor count can be reduced by 42% compared with the direct-mapping MRF_CLA design [1].

Index Terms—Cost-effective hardware design, Markov random field (MRF), master-and-slave MRF mapping, noise-tolerant circuit, probabilistically based circuit.

I. INTRODUCTION

W ith the progress of VLSI process technology, power consumption and power density increase as design complexity and transistor density in systems-on-chip (SoCs) increase. Therefore, low-power design becomes a major design challenge. In general, lowering the supply voltage is the most efficient way to save power since power consumption is proportional to the square of supply voltage. Future electronic devices are expected to operate at much a lower supply voltage than traditional designs, which is very useful for portable electronics and biomedical implants. Unfortunately, when supply voltage decreases, noise does not decrease proportionally. As a result, one of the major challenges in ultralow-power and deep-submicrometer circuit design is noise fluctuation [2]–[8]. Ultralow-submicrometer VLSI circuits are expected to operate at much smaller noise margins, and thus, VLSI circuits are more sensitive to noise.

Many low-noise or noise-tolerant circuit techniques were proposed recently [9]–[13], but they all focus on dealing with specific types of noise, such as crosstalk noise and substrate noise. These designs cannot effectively solve the random intrinsic noise in ultralow-submicrometer or future nanoscale VLSI systems [14], [15]. The triple-majority-redundant (TMR) methods [16], [17] are popular solutions to device failure. If any one of the three circuits fails, the other two circuits can still correct faults. However, intrinsic noise can significantly interfere with each signal, and the noisy signals often render wrong voting results in TMR. Therefore, TMR cannot effectively solve the random intrinsic noise that often occurs in nanoscale circuits. The error correction code (ECC) techniques are pervasively and successfully used to correct the transmission data in communication systems. However, we do not have prior knowledge about what types of errors could occur in nanoscale circuitry, and thus, it is very difficult to select the best type of error coding scheme. Moreover, ECC achieves higher error-correction capability at the expense of higher latency. ECC techniques applied in communication systems may not be the ideal choice to solve intrinsic noise in nanoscale circuits. The neural network structure [18] is another approach, which provides continuous adaptation to errors. However, training sequences are difficult to obtain in advance, and the neural network is also difficult to implement.

Because intrinsic noise is random and dynamic in nature, conventional deterministic logic design methods used in CMOS circuits are insufficient to handle these faults. Probabilistic approaches are more desirable to handle this problem [1], [19]–[21]. In probabilistic-based noise-tolerant approaches, the noise energy is shared and averaged by the whole system; therefore, the noise signal level can be greatly reduced. As a result, noise tolerance can be enhanced by using probabilistic-based approaches.

In this paper, the probabilistic-based noise-tolerant circuits are designed based on the Markov random field (MRF) theory [19]. The basic idea of MRF design is as follows: Under the probabilistic framework, we cannot expect logic values (‘0’ or ‘1’) in a circuit at a particular time to be correct. We can only expect the probability distribution of the values to have
the highest likelihood of being in a correct logic state. The appropriate mathematical framework for this type of analysis is the MRF, which was initially developed in [19] to support the optimization of the values of a large set of random variables so that their overall joint probability has a global maximum. A possible MRF circuit design was presented in [20] and [21], where the MRF theory was mapped onto logic circuits. An MRF silicon chip design was further presented in [1] to prove the design concept of noise-tolerant MRF circuits. As demonstrated in [1], [20], and [21], the probabilistically based circuits can achieve much better noise immunity than their CMOS counterparts. However, the hardware overhead of MRF direct-mapping circuits in [1] is much larger than that of deterministically based CMOS circuits.

In this paper, a cost-effective design is proposed to provide nearly the same noise-tolerant performance, but with lower hardware overhead through MRF network simplification. Our simplification method is based on master-and-slave MRF mapping and master-and-slave MRF logic-gate construction. To demonstrate the design methodology, we implemented an 8-bit MRF carry-lookahead adder (MRF_CLA) using 0.13-μm CMOS process technology. The number of transistors in the proposed cost-effective MRF_CLA can be significantly reduced by 42% as compared with the initial direct-mapping MRF design in [1].

The major contributions of this work are as follows.

1) Design cost-effective MRF circuit by master-and-slave MRF mapping and master-and-slave MRF logic-gate construction. Instead of directly implementing all valid states in the MRF graph, we separate the truth table into two groups according to its output logic states and then implement them separately. All output states in logic “1” are collected as group “1,” while logic “0” states are collected as group “0.” The minterms in groups “1” and “0” are implemented separately in a master fashion. The feedback stage is constructed in a slave fashion to decrease hardware cost. The hardware complexity in the proposed master-and-slave design can be greatly reduced, and the hardware overhead does not increase exponentially when the number of nodes in the MRF network increases. The number of transistors in the proposed cost-effective MRF_CLA can be significantly reduced by 42% as compared with the initial direct-mapping MRF design in [1].

2) Decompose the MRF network into small subnetworks for practical implementation. At the architectural level, we propose a practical solution to divide the whole MRF network into many small-scale MRF subnetworks to reduce overall hardware complexity. Within individual MRF subnetworks, we implement the noise-tolerant logic circuit using valid minterm feedback convergence loops. All these feedback loops are constructed by neighborhood nodes in the same MRF subnetwork to achieve high noise tolerance. At the architectural level, we can combine these subnetworks and thus provide a practical architectural mapping solution for any computing systems. The proposed practical MRF architectural mapping can achieve good tradeoffs between hardware complexity and noise-tolerance performance.

3) Map the MRF theory to a real silicon adder circuit (with complete measurement results) as proof of design concept. References [19]–[21] applied the MRF theory to solve noise interference issues at the algorithmic and logic levels, and we extend the design idea to the architectural level for chip implementation. To prove the design concept in silicon and demonstrate the noise-tolerance capability, we implemented an 8-bit MRF_CLA with the 0.13-μm CMOS process technology. Some portions of the chip implementation results have been reported in our previous paper [22]. In this paper, we give more details about our design and present some new experiments to compare circuit performance in terms of noise tolerance and hardware cost. From the measurement results, the proposed cost-effective MRF_CLA can provide a 7.00 × 10^{-5} bit-error rate (BER) under 10.6-dB signal-to-noise ratio (SNR), while the conventional CMOS_CLA can only provide an 8.84 × 10^{-3} BER. Because of high noise immunity, the master-and-slave MRF_CLA can tolerate noise interference to operate under 0.25 V with only 1.9 μW/MHz of energy consumption.

This paper is organized as follows. In Section II, we review the MRF circuit design and illustrate its superiority in noise tolerance. In Section III, we propose a cost-effective MRF noise-tolerant circuit design by using master-and-slave MRF mapping and master-and-slave MRF logic-gate construction. In Section IV, we present a practical MRF architectural mapping solution through MRF local network mapping. In Section V, we address the proof-of-concept chip implementation and measurement results. Finally, we draw a conclusion in Section VI.

II. REVIEW OF PROBABILISTIC-BASED NOISE-TOLERANT CIRCUIT DESIGN

A. Review of MRF Noise-Tolerant Design Methodology

The MRF is a probabilistic framework that has been commonly used in many fields, including pattern recognition and communications [23]–[26]. The MRF theory is first applied to enhance noise tolerance in VLSI circuit design in [20]. The MRF approach is further extended to the design of a Hamming decoder for the protection of memory against single event upsets in [27]. The main idea in MRF is that output results of the decoder for the protection of memory against single event upsets are uniquely determined by circuit input or output values. Therefore, the noise signal level can be greatly reduced. In this way, we can judge the correct logic states precisely and enhance the reliability of computing circuits. The other main reason for selecting the Markov random network as the basis for our design is that its operation does not depend on perfect devices or perfect input signals. The following is a brief review of how MRF works.

The most important feature associated with the MRF is the Markovian property: The probability of a given variable depends only on its neighbors. In an MRF network, variables $x_i$ represent the states of logic gates in a circuit. These logic states are uniquely determined by circuit input or output values. Arcs or edges in an MRF network represent conditional probabilities relating each circuit node to its neighboring nodes.
The MRF, in principal, can encode arbitrary logic. Taking an example circuit as shown in Fig. 1(a), we can explain the MRF mapping as follows. Each logic signal terminal can be viewed as a node of the MRF network. The MRF network is constructed based on its logic combinations. The example circuit can be mapped onto an MRF, as shown in Fig. 1(b).

In this application, the MRF is defined by a set of random variables $V = \{x_0, x_1, \ldots, x_6\}$. The nodes in MRF correspond to logic signal terminals $\{x_0, x_1, \ldots, x_6\}$, and $v_i P(x_i) > 0$. The Markovian property corresponds to the logic interactions between nodes, i.e., $P(x_6) = P(x_0|x_5)P(x_5)$. The conditional probability $P(x_4|V - x_4)$ of a node state in terms of its neighborhood can be formulated in terms of cliques. For instance, the conditional probability of $P(x_4|x_2, x_3)$ can be formulated in terms of the clique of $\{x_2, x_3, x_4\}$. A clique is a set of fully connected nodes, and each node connects to all the remaining nodes in the clique as its neighbors.

In a logic circuit, hundreds of internal logic signal paths exist from its inputs to its outputs. However, only one set of variable combination is correct. The correct set corresponds to the maximum joint probability. Noise tolerance is achieved because the MRF network is updated iteratively in terms of the state changes of individual nodes through the network. Ultimately, the network converges to a stable set of states, which corresponds to the correct logic states. Successful operation only requires that the energy of correct states is lower than the energy of incorrect states [19].

Next, we review the mechanism of an MRF noise-tolerant design via a simple example.

**B. Mapping MRF Graph Onto Logic Circuits**

To demonstrate how to map an MRF network onto a logic circuit, we consider a two-input NAND gate with inputs $\{x_0, x_1\}$ and an output $\{x_2\}$ in Fig. 2(a) as an example. To map the logic function onto an MRF network $V = \{x_0, x_1, x_2\}$, we first list all possible states in a truth table and mark its valid states apart from its invalid states, as in Fig. 2(b) [20], [21].

The design goal is to ensure that the circuit stays in the correct states. Based on the MRF theory, the logic output state can be influenced by its neighboring nodes, and the correct states correspond to its valid minterms. In the NAND example, $[\overline{x}_0 \overline{x}_1 x_2]$, $[\overline{x}_0 x_1 \overline{x}_2]$, $[\overline{x}_0 \overline{x}_1 \overline{x}_2]$, and $[x_0 x_1 \overline{x}_2]$ are four valid minterms $\{x_0 x_1 x_2\} = \{001\}, \{011\}, \{101\}, \{110\}$

In the example circuit, the valid minterms are collected in a group and connected back to the corresponding input nodes. For example, all three minterms $[\overline{x}_0 x_1 x_2]$, $[\overline{x}_0 x_1 x_2]$, and $[\overline{x}_0 x_1 x_2]$ contain the logical variable $x_2$; thus, their outputs are feedback connected to $x_2$. By following the same design rule, the other five feedback paths can be connected to input nodes $x_0$, $x_1$, $\overline{x}_1$, and $\overline{x}_2$, as shown in Fig. 3.
To show how an MRF NAND works, the pattern \( \{ x_0 = 0, x_1 = 0, x_2 = 1 \} \) is taken as an example for illustration. The AND gate at the very top is ‘on’ and feeds a logic ‘1’ back to the input of the OR gates shown at the bottom of Fig. 3. The other AND gates are ‘off’ and feed back logic ‘0’. These valid minterm values are grouped by OR gates and fed back to all input nodes. In this case, the input and output values are \( \{ x_0 = 0, x_1 = 0, x_2 = 1 \} \) and the overall circuit latches into the correct state \( \overline{x_0} x_1 x_2 \). The other correct configurations, such as \( \{ x_0 = 0, x_1 = 1, x_2 = 1 \} \), \( \{ x_0 = 1, x_1 = 0, x_2 = 1 \} \), and \( \{ x_0 = 1, x_1 = 1, x_2 = 0 \} \) or \( \overline{x_0} x_1 \overline{x_2}, x_0 \overline{x_1} x_2, \) and \( \overline{x_0} x_1 \overline{x_2} \), all work in the same fashion. However, the incorrect configurations, such as \( \{ x_0 = 0, x_1 = 0, x_2 = 0 \} \), are not stable and will converge to a correct state through valid minterm feedback loops, such as \( \overline{x_0} x_1 \overline{x_2} \) converging to \( \overline{x_0} x_1 x_2 \). In other words, the MRF can automatically recover from incorrect states and achieve noise tolerance. However, if incorrect states occur in a conventional CMOS NAND gate, such as \( \{ x_0 = 0, x_1 = 1, x_2 = 0 \} \), \( \{ x_0 = 0, x_1 = 1, x_2 = 0 \} \), \( \{ x_0 = 1, x_1 = 0, x_2 = 0 \} \), and \( \{ x_0 = 1, x_1 = 1, x_2 = 1 \} \) or \( \overline{x_0} x_1 \overline{x_2} \), \( \overline{x_0} x_1 \overline{x_2} \), \( \overline{x_0} \overline{x_1} x_2 \), and \( \overline{x_0} x_1 \overline{x_2} \), the circuit cannot recover automatically from its incorrect states. The NAND gate is the basic logic combinational gate in computing systems. The other basic MRF logic gates or complicated MRF combinational logic can also be designed in the same way.

C. Noise-Tolerant Performance of MRF Circuits

From Fig. 3, we can observe that MRF circuits require more transistors than their counterpart conventional CMOS designs. MRF implementations, however, provide significant advantages over standard CMOS designs in terms of noise tolerance when supply voltage is significantly decreased. Fig. 4 shows the simulation results by comparing an MRF NAND with a CMOS NAND gate. The simulations were carried out in SPICE using the 70-nm CMOS library from Berkeley [29] at room temperature. The threshold \( V_{TH} \) values are 0.2 V for nMOS and 0.22 V for pMOS. We used the supply voltage \( V_{DD} = 0.15 \) V, in which the gate operates in the subthreshold region \( (V_{DD} < V_{TH}) \). The outputs of an MRF NAND and a CMOS NAND gate in the presence of very noisy input signals are shown in Fig. 4. The top two curves are the input signals \( \{ x_0, x_1 \} \), which are generated by adding a Gaussian noise with zero mean and a variance of 3.4 mV to logic “1” (0.15 V) and to logic ‘0’ (0.0 V). Noise with 3.4-mV variance is picked to illustrate and compare the noise-tolerance capability between MRF and conventional CMOS design. If the noise variation is too large, the MRF circuit will also malfunction. The noise model that we used can be found in [30]. The noise on the input signal causes the standard CMOS NAND gate to switch between correct and incorrect output values, as shown in the curve second to the bottom in Fig. 4. In other words, the deterministic CMOS design simply cannot function when the circuit operates under the subthreshold condition. The probabilistic-based MRF NAND gate, on the other hand, provides excellent noise immunity even in the subthreshold region, as shown in the curve at the bottom of Fig. 4.

III. Cost-Effective MRF Noise-Tolerant Logic Circuit Design

A. Master-and-Slave MRF Mapping

In [1], the first working silicon chip was fabricated to prove the design concept of a noise-tolerant circuit using the MRF theory. The circuit in Fig. 1(a) can be mapped onto an MRF graph according to its logic combination as shown in Fig. 1(b). If we implement the MRF noise-tolerant circuit following the procedures outlined in [1], the hardware complexity grows exponentially as the scale of the MRF network increases. More specifically, as shown in Fig. 5, the valid function of this sample circuit is determined by the nodes from \( X_0 \) to \( X_6 \), and thus, the hardware complexity is proportional to \( 2^7 \) (or the exponentiation of the number of MRF nodes).

To reduce the hardware complexity, some area-efficient MRF designs have been proposed in [31] and [32]. In [31], the hardware complexity of an MRF logic gate is reduced through Boolean simplification and valid minterm reduction. By mapping the MRF logic gate using the simplified Boolean equation proposed in [31], the circuit area can be greatly reduced. In particular, high fan-in circuits can be constructed without exponentially increasing the circuit area. In [32], some techniques, including clique variable sharing, implied dependence, and supergate techniques, were proposed to reduce the transistor count of large-scale multilevel combinational circuits built within the MRF design. The simplified technique in [31]
is to reduce the MRF hardware complexity in each simple logic gate. Nevertheless, the technique presented in [32] is to merge the multilevel logical gates into simplified ones, which is a systematic approach. This approach cannot always result in hardware saving because it depends heavily on detailed circuit topology.

In this paper, a more general and cost-effective MRF design is proposed and implemented to further reduce the MRF hardware complexity in each simple logic gate. Our simplification method is based on master-and-slave MRF mapping and master-and-slave MRF logic-gate construction. Instead of directly implementing all valid states in the MRF graph, the truth table is separated into two groups according to its output logic states and then implemented separately in the master-and-slave MRF fashion. As shown in Fig. 6(a), the truth table of the sample circuit in Fig. 1(a) is separated into two groups. All output states in logic “1” are collected as group “1,” while logic “0” states are collected as group “0.” Next, minterms in groups “1” and “0” are implemented separately. These two groups compete to determine the circuit outputs. We call this first-stage design a master MRF network. To let the circuit have a better chance of staying in the correct states, the outputs are connected back to inputs in order to reinforce the correct states in the master MRF network. The feedback stage is named the slave MRF network. The hardware complexity in the proposed master-and-slave design is proportional to $2^7$ (or two groups times two stages instead of $2^7$ in the direct-mapping MRF design), as shown in Fig. 6(b). Using this design, the hardware overhead does not increase exponentially while the noise-tolerance capability is preserved.

### B. Master-and-Slave MRF Logic-Gate Construction

All combinatorial circuits can be derived from basic logic gates, such as NAND, NOR, and inverters. To revisit the construction of an MRF circuit, we consider a simple two-input NAND gate with inputs $\{x_1, x_2\}$ and an output $\{x_3\}$ as an example. The MRF logic gate is constructed by the valid minterm generator and the feedback connector. The hardware complexity is determined by its input and output numbers.

In the direct-mapping design in [1], the MRF NAND gate is constructed by a valid minterm generator (with six inputs and four minterms) and a feedback connector (with four inputs and six outputs), as shown in Fig. 7(a). In the proposed design, as shown in Fig. 7(b), we separate the valid minterm generator into two groups to fix the minterm number to two. In this way, we can prevent the exponential increase of hardware complexity in the valid minterm generator. Moreover, no matter how complex the valid minterm generator circuit is, the hardware complexity in the feedback connection is fixed. Therefore, the hardware complexity can be greatly reduced, particularly in the case of a logic function with high fan-in.

---

**Fig. 6.** Proposed cost-effective MRF mapping method. (a) Truth table is separated into two groups. (b) Output is determined by the master-and-slave MRF network.

**Fig. 7.** Construction of an MRF NAND gate. (a) Direct-mapping MRF NAND architecture. (b) Proposed cost-effective MRF NAND architecture.
When summing all valid minterms, the compatibility function of a two-input NAND gate can be obtained by

\[ U(X_0, X_1, X_2) = X'_0X'_1X_2 + X'_0X_1X_2 + X_0X'_1X_2 + X_0X_1X'_2, \]

(1)

According to the concept of Boolean difference, a Boolean expression can always be separated into two groups. By applying Boolean difference, (1) can be re-expressed as

\[ U(X_0, X_1, X_2) = (X'_0 + X'_1)X_2 + (X_0X_1)X'_2, \]

(2)

The MRF NAND circuit in [1] is implemented by directly following (1), and the design is shown in Fig. 8(a). The direct-mapping implementation needs four AND gates and six OR gates. In [31], the MRF logic gates are constructed through Boolean simplification to save hardware cost. The method in [31] is to merge several valid minterms into one complex term. This way, the number of valid minterms can be reduced. In the proposed cost-effective MRF logic gates, the number of valid minterms can be further reduced to two because we separate the valid minterm generator into two groups.

Fig. 8(b) shows the proposed cost-effective MRF NAND circuit. The four valid minterms in (1) are merged into two terms \((X'_0 + X'_1)\) and \((X_0X_1)\), as shown in (2). The master MRF NAND network is constructed using an AND gate and an OR gate. In the simplified design, about half the number of hardware components to implement the valid minterms can be saved compared with the direct-mapping implementation. Based on DeMorgan’s Law, we can separate all inputs \(X_0, X_1, X_2, X'_0, X'_1,\) and \(X'_2\) individually, and each input signal appears only once in each logic gate. In this case, connecting the feedback paths to the output of the master MRF network achieves the same function as connecting them back to the inputs of the master MRF network. Therefore, we only need to construct a slave network to realize the feedback function in the MRF network. Based on this design, the hardware complexity to implement the feedback function of an MRF network is fixed to two NAND gates and two inverters.

By comparing this simplified design with the initial MRF NAND design in [1], the hardware complexity in the feedback connector is reduced from six OR gates to two NAND gates plus two inverters. The total transistor count in the simplified two-input MRF NAND gate can be reduced from 60 to 28, a 53.3% reduction. Following the same design methodology, we can build the simplified MRF NOR gate, XOR gate, etc.

In order to show that the hardware overhead can be significantly reduced from exponentially increasing in the direct-mapping approach, a three-input NAND gate is taken as another example to compare the hardware complexity between the direct-mapping MRF design and the proposed master-and-slave MRF design. In the direct-mapping MRF design, the three-input NAND gate is implemented according to

\[ U(X_0, X_1, X_2) = X'_0X'_1X'_2X_3 + X'_0X'_1X_2X_3 + X'_0X_1X'_2X_3 + X_0X'_1X'_2X_3 + X_0X_1X'_2X_3 + X_0X_1X_2X'_3, \]

(3)

In the proposed master-and-slave MRF design, the three-input NAND gate is implemented according to

\[ U(X_0, X_1, X_2) = (X'_0 + X'_1 + X'_2)X_3 + (X_0X_1X_2)X'_3. \]

(4)

The direct-mapping implementation needs eight four-input AND gates and eight four-input OR gates. In the proposed master-and-slave MRF design, the number of valid minterms \(U(X_0, X_1, X_2)\) can be reduced from eight terms to two terms because we separate the valid minterm generator into two groups based on the Boolean difference. Moreover, the hardware complexity to implement the feedback function of a three-input MRF NAND gate can be further simplified from eight four-input OR gates to two NAND gates and two inverters by following DeMorgan’s Law and using the slave MRF network. As a result, the total transistor count in the simplified three-input MRF NAND gate is significantly reduced from 144 to 34, a 76.4% reduction. In general, the minterms of the direct-mapping MRF grow exponentially as input increases, but the minterms only increase linearly in the master–slave approach.

C. Comparison of Hardware Cost

To illustrate the effect on area saving by adopting the proposed cost-effective master-and-slave MRF circuit, the transistor counts in the multiple-input NAND gates and Hamming decoders using the direct-mapping MRF approach in [1], the state-of-art cost-effective MRF design in [31], and the proposed master-and-slave MRF approach are compared. The comparison results are listed in Table I.
As illustrated in Table I, the transistor counts in the proposed master-and-slave MRF design can be reduced from 53.3% to 94.5% in the multiple-input MRF noise-tolerant NAND gates as compared with the direct-mapping MRF design [1]. Moreover, the hardware overhead does not increase exponentially as the fan-in increases in the NAND gate circuits. In the (6, 3) Hamming decoder, our cost-effective design can save 46.4% transistor counts as compared with the MRF design in [1]. In the (8, 4) Hamming decoder, the transistor count reduction can be further increased to 72.7%. As a result, when the fan-in of an MRF circuit increases, the effect on area saving by adopting the proposed master-and-slave MRF circuit is more significant. To construct the NAND gates, the transistor counts in the proposed master-and-slave MRF design and the state-of-art cost-effective MRF design in [31] are nearly the same. Nevertheless, in the Hamming decoders, our cost-effective design can save 28% of the transistors as compared with the state-of-art cost-effective MRF design in [31]. As a result, when the complexity of an MRF circuit increases, the area saving by adopting the proposed master-and-slave MRF circuit is also more significant.

### IV. PRACTICAL MRF NOISE-TOLERANT ARCHITECTURE DESIGN

In the previous section, the mapping of simple logic gates onto MRF networks was discussed, and various simulations to show that MRF circuits perform better than CMOS circuits in terms of noise tolerance were presented. The master-and-slave MRF mapping approach is to construct an MRF functional block such as NAND, NOR, and XOR gates through Boolean simplification. This design is good for constructing basic MRF logic functions but is not suitable for constructing a complex and irregular Boolean function. Local network mapping, on the other hand, is applied to divide a complicated MRF function into many simple MRF functions. We apply these approaches for different conditions according to the circuit characteristics. In Section IV, we present a practical case of mapping an MRF network onto a computing system in order to meet the requirements of both performance and hardware cost.

#### A. Local Network Mapping of MRF Noise-Tolerant Circuit

We remap the M3 circuit in Fig. 1(a) onto a practical MRF network, as shown in Fig. 9. The dashed lines in Fig. 9 representing two nets are connected by a simple wire connection. The MRF subnetworks are only constructed within those nets that are connected by solid lines. The difference between Fig. 1(b) and Fig. 9 is that global mapping is used in Fig. 1(b) while local mapping is used in Fig. 9. Global mapping means that the whole logic function is directly mapped onto the MRF network. Local mapping means that the whole logic circuit is divided into many submodules, which are then mapped onto MRF networks individually. By using the local mapping approach, the hardware
complexity to implement an MRF noise-tolerant circuit can be greatly reduced.

To demonstrate the concept of local network mapping in an MRF network, a one-bit full adder is taken as an example for illustration. The schematic of a one-bit full adder is shown in Fig. 10(a). We map the one-bit full adder onto an MRF network using global and local mappings in Fig. 10(b) and (c), respectively. The circuit complexity to implement the MRF network is determined by the number of nodes (or the possible valid states) in an MRF network defined as $N$. If an MRF network is implemented by global mapping, all possible valid states in the MRF network are $2^N$, which means $2^N$ feedback loops must be constructed in order to construct the MRF network. Because the valid minterm feedback loop includes the feedforward and feedback parts, the hardware overhead in each feedback loop path is twice as large as the original CMOS design. As a result, the circuit complexity to implement the MRF network is $2^N$, which is an exponential function of $N$. If the MRF network is divided into many smaller MRF subnetworks, the node number in each subnetwork can be smaller than $N$. The hardware complexity to implement the whole MRF network using the local mapping can thus be greatly reduced.

As shown in Fig. 10(b), $2^7 = 256$ feedback loops must be constructed to implement a one-bit MRF full adder using the global mapping, while only $5 \times 2^2 = 20$ feedback loops need to be constructed to implement a one-bit MRF full adder using the local mapping in Fig. 10(c). Therefore, the local mapping technique is a more practical architectural mapping solution than the global mapping design to implement an MRF noise-tolerant computing system. To save hardware cost, the submodules can be subdivided as small as possible. As shown in Fig. 10(d), a
one-bit MRF full adder using the local optimization only needs 20 feedback loops (four feedbacks for each submodule and five submodules in the adder).

**B. Analysis of the Noise-Tolerant Performance and Hardware Complexity of an MRF Network**

To illustrate why the noise-tolerant performance can still be preserved when the local mapping is applied to the MRF network, the simple circuit in Fig. 11(a) is used. All possible valid states for the example circuit using MRF local and global network mappings are listed in Fig. 11(b) and (c), respectively. In total, there are eight valid states in the example circuit using MRF local mapping. In the MRF global mapping case, there are a total of 16 valid states, where eight states (unmarked states) are the same as in the local mapping case and the other eight states (marked by dashed and solid rectangles) are different. The number of valid states in the global mapping case is twice that of the local mapping case, where more redundant states can achieve better noise immunity. However, among these eight discrepancy states in the global mapping, there are four states that lead to the same output states as in the local mapping case (marked by dashed rectangle). The occurrence probability of another four states (marked by solid rectangles) is only 1/8 as compared with the eight states in the local mapping case because the internal node \( X \) has been protected by the MRF network \( V = \{ A, B, X \} \). Therefore, the noise tolerance in the MRF global mapping design is improved by only 6.25% (\( 4 \times (1/8)/8 = 0.0025 \)) as compared with the MRF local mapping approach. However, the hardware complexity is twice that of the MRF local network mapping design.

In Fig. 12, the noise-tolerant performances of global and local MRF mappings in the simple circuit in Fig. 11(a) are compared. The simulations were carried out in SPICE using the Taiwan Semiconductor Manufacturing Company (TSMC) 0.13-\( \mu \)m CMOS process with 1.2-V supply voltage. The noise source is additive white Gaussian noise (AWGN). By adjusting the noise power, we can compare the BERs of the global MRF network mapping and the proposed local MRF network mapping under various SNRs. As an illustration, the global MRF
network mapping method can achieve a superior noise-tolerance performance with a lower BER since all possible valid states are constructed and converged into a stable state. The local network mapping technique, on the other hand, cannot achieve the same noise-tolerant ability as the global network mapping because some possible internal valid states in the MRF network are removed. However, the degradation of the noise-tolerant ability in local network mapping is negligible. Moreover, the design complexity and the critical path in the computing systems can be greatly reduced because the design complexity of the MRF network increases exponentially with the number of network nodes. As a result, implementing the MRF noise-tolerant circuit with the local network mapping approach can provide a good tradeoff between noise-tolerant performance and design complexity and, thus, is a practical solution to implementing the MRF algorithm.

C. Architectural Mapping of an 8-bit MRF_CLA Circuit

In this paper, an 8-bit MRF_CLA is implemented as the proof-of-concept design to show its noise tolerance. A conventional 8-bit CMOS_CLA [33] is shown in Fig. 13. To realize the 8-bit MRF_CLA through global MRF network mapping, 44 nodes must be used to construct the whole MRF network. In other words, the implementation complexity of the MRF network is $2^{44}$. Such a design is impractical for real implementation. On the other hand, a practical 8-bit MRF_CLA mapping architecture realized with local network mapping is shown in Fig. 14, where the design complexity is only $27 \times 2^3 + 11 \times 2^4$.

The architecture of the proposed 8-bit MRF_CLA is shown in Fig. 15. In the MRF_CLA, the MRF subnetwork is constructed within only basic logical modules, such as MRF_INV, MRF_NAND/MRF_AND, MRF_NOR/MRF.OR, and MRF_XNOR/MRF_XOR circuits. Then, the whole 8-bit
MRF_CLA circuit is hierarchically constructed using these MRF submicromodules. The CLA consists of three basic logic generators: the PG, carry, and sum generators. In Fig. 15, the basic logic circuits labeled “MRF” denote logic circuits that are implemented by the MRF algorithm and represented as MRF subnetworks of the whole 8-bit MRF_CLA. In this way, local optimization in the output state of individual MRF submodules is achieved. The local optimized signal can propagate stage by stage until the output states converge to a stable state.

V. CHIP IMPLEMENTATION AND MEASUREMENT RESULTS

A. Chip Implementation

To demonstrate the proof-of-concept design, a master-and-slave MRF chip, i.e., an 8-bit MRF_CLA, was fabricated using the TSMC 0.13-μm CMOS process technology. The cost-effective MRF design was implemented using the 0.13-μm CMOS technology process to show its performance under lower supply voltage. The conventional and MRF CLA designs have also been implemented using the 0.18-μm CMOS technology process to demonstrate and compare their performance in terms of noise tolerance, power, speed, and area [1]. For different CMOS technologies, BER can be measured and compared with the same SNR to establish a fair baseline. For different CMOS technologies, BER can be measured and compared with the same SNR to establish a fair baseline. For different CMOS technologies, BER can be measured and compared with the same SNR to establish a fair baseline. For different CMOS technologies, BER can be measured and compared with the same SNR to establish a fair baseline. For different CMOS technologies, BER can be measured and compared with the same SNR to establish a fair baseline.

B. Measurement Results

In the previous sections, the superiority of the MRF probabilistic-based noise-tolerant circuits in terms of noise immunity was demonstrated using simulation results. However, some restrictions and artifacts remain when the computer-aided design tool was used to model behavior and simulate circuit performance under noisy environments. Therefore, an 8-bit MRF_CLA chip was implemented to demonstrate the proof-of-concept design. By implementing the MRF design onto silicon and measuring the chip performance in real time, it cannot only demonstrate the superiority in the aspect of noise-immunity enhancement but also help further clarify the noise-tolerant mechanism in the probabilistic-based approach.

1) Measurement Setup to Test the MRF Circuit: To compare the noise tolerance between the MRF and the CMOS design, noisy signals were injected into the inputs of an 8-bit master-and-slave MRF_CLA, an 8-bit direct-mapping MRF_CLA, and an 8-bit CMOS_CLA. These noisy signals consist of noise-free signals generated by a pattern generator (Agilent 33250A) and noise signals from a power supply. The measurement setup is shown in Fig. 17. Signal timing and circuit function can be verified by a logic analyzer.

2) Measurement of the MRF Circuit Single-Bit Waveforms Under Noise Interference: Through the use of an oscilloscope (Agilent Infinium Oscilloscope DSO8104A), the waveforms of each single bit in the CLA output can be observed to compare its stability under noise interference. In both panels of Fig. 18(a) and (b), the upper signals are the input signals, and the lower signals are the corresponding output signals. As shown in Fig. 18, under the same noisy input signal with 14.1-dB-SNR AWGN interference, the output waveform of MRF_CLA in Fig. 18(b) is more stable than the conventional CMOS_CLA one in Fig. 18(a). In addition to the output waveform, the input signal of the MRF_CLA circuit is also more stable than the conventional CMOS_CLA one. The reason is that the MRF_CLA circuit is constructed as a network in which all nodes in the MRF network, no matter inputs or outputs, can jointly share the noise energy to make all nodes in the network stay in a stable state. The measurement results support our arguments that the noise energy can be jointly shared in the probabilistic-based MRF_CLA chip and that the noise signal level can be reduced.

3) Measure Reduplicating Signals of the MRF Circuit Under Noise Interference: The output waveform of circuits under the
interference of random noise by continually reduplicating “0” and “1” signals under random noise interference is also observed. By comparing the reduplicating signals in Fig. 19 under the case of AWGN interference with 14.1-dB SNR, the noise tolerance of the proposed master-and-slave MRF noise-tolerant circuit is shown to be much better than the CMOS_CLA circuit. The noise margin or the difference between the worst “0” and “1” is 1 V in the MRF_CLA circuit under the 0.13-μm process with 1.2-V supply voltage, while the noise margin in the CMOS_CLA circuit is ~1.2 V under the 0.18-μm process with 1.8-V supply voltage. This indicates that MRF_CLA can operate correctly with 14.1-dB SNR AWGN interference, but the CMOS_CLA cannot (a negative noise margin leads to computing error).

4) Measurement of BER Under Various SNR in the MRF Circuit: By adjusting the power levels of noise and supply, the BERs of the proposed master-and-slave MRF_CLA, the direct-mapping MRF_CLA, and the CMOS_CLA circuits under various SNR conditions can be measured. In Fig. 20, the supply power is fixed, and only the noise power level is adjusted. In Fig. 20, the power supply voltage is 1.8 V for the 0.18-μm process and 1.2 V for the 0.13-μm process, respectively. Based on the real-time measurement results shown in Fig. 20, one can see that the noise immunity of the MRF_CLA can be greatly enhanced compared to the CMOS_CLA. The high-noise-immunity feature of the MRF circuit is very useful to low-power silicon circuits when they operate under a very low VDD condition. In the 0.18-μm process, the direct-mapping MRF circuit can achieve a $2.88 \times 10^{-6}$ BER under a 10.6-dB SNR and sub-0.5-V supply-voltage condition. The direct-mapping MRF_CLA circuit consumes only 8 μW/MHz under a 0.45-V supply. The proposed master-and-slave MRF_CLA can provide a $7.00 \times 10^{-5}$ BER under a 10.6-dB SNR, while the conventional CMOS_CLA can only provide an $8.84 \times 10^{-5}$ BER. The master-and-slave MRF_CLA can operate under 0.25 V with only 1.9 μW/MHz of energy consumption. The simulation results using the 70-nm process [29] even show that the MRF circuit can operate with an 8-dB low SNR and 0.15-V supply voltage.

C. Chip Performance Summary

The performance comparisons of the MRF_CLA chip and the conventional CMOS_CLA chip are listed in Table II. As a comparison, the transistor count in the conventional CMOS design is only 384, while the direct-mapping MRF design [1] needs 5040 transistors and the proposed master-and-slave MRF_CLA needs 2940 transistors. The measurement results show that the
TABLE II

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>UMC 0.18μm</td>
<td>UMC 0.18μm</td>
<td>TSMC 0.13μm</td>
</tr>
<tr>
<td>Area</td>
<td>110μm²*50μm</td>
<td>280μm²*270μm</td>
<td>140μm²*140μm</td>
</tr>
<tr>
<td>Transistor</td>
<td>384</td>
<td>5040</td>
<td>2940</td>
</tr>
<tr>
<td>$V_{DD_{min}}$</td>
<td>0.9V</td>
<td>0.45V</td>
<td>0.25V</td>
</tr>
<tr>
<td>Energy @ $V_{DD_{min}}$</td>
<td>95 μW/MHz @ $V_{DD}=1.8$</td>
<td>724 μW/MHz @ $V_{DD}=1.8$</td>
<td>84 μW/MHz @ $V_{DD}=1.2$</td>
</tr>
<tr>
<td>Delay</td>
<td>0.55ns @ $V_{DD}=1.8$</td>
<td>4.65ns @ $V_{DD}=1.8$</td>
<td>2.55ns @ $V_{DD}=1.2$</td>
</tr>
<tr>
<td>BER @ Various SNR</td>
<td><a href="mailto:8.84E-03@10.6dB">8.84E-03@10.6dB</a> SNR</td>
<td><a href="mailto:2.88E-06@10.6dB">2.88E-06@10.6dB</a> SNR</td>
<td><a href="mailto:7.00E-05@10.6dB">7.00E-05@10.6dB</a> SNR</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:8.64E-04@11.9dB">8.64E-04@11.9dB</a> SNR</td>
<td><a href="mailto:1.95E-07@11.9dB">1.95E-07@11.9dB</a> SNR</td>
<td><a href="mailto:1.21E-05@11.5dB">1.21E-05@11.5dB</a> SNR</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:5.33E-05@13.3dB">5.33E-05@13.3dB</a> SNR</td>
<td><a href="mailto:2.50E-09@13.3dB">2.50E-09@13.3dB</a> SNR</td>
<td><a href="mailto:1.25E-08@13.7dB">1.25E-08@13.7dB</a> SNR</td>
</tr>
</tbody>
</table>

REFERENCES


ACKNOWLEDGMENT

The authors would like to thank the National Chip Implementation Center, Taiwan, for the chip implementation support.

noise-tolerance ability of MRF design design, however, is an improvement of $10^2$–$10^4$ times under various AWGN interference compared to the CMOS. Compared with the direct-mapping MRF design, the master-and-slave MRF design can reduce the transistor count by 42% while maintaining noise immunity. Compared with the conventional CMOS design, the hardware complexity in the master-and-slave MRF design is still higher, and the speed is slower; however, the MRF design can operate under the subthreshold conditions to save power. The conventional CMOS circuit, on the other hand, simply cannot operate under such a low-power environment. With the 0.13-μm process, the proposed 8-bit MRF_CLA can achieve under 0.25 V of supply voltage with only 1.9 μW/MHz of energy consumption. In comparison, the conventional 8-bit CMOS_CLA chip can only operate correctly under 0.9 V of supply voltage.

The unavoidable worsening of noise interference in nanoscale circuits is a problem that conventional CMOS circuit techniques cannot possibly mitigate. As a result, the proposed cost-effective probabilistic-based noise-tolerant design can achieve high noise immunity with reasonable design overhead, which is suitable for future nanoscale computing systems.

VI. CONCLUSION

In this paper, we have presented the design and implementation of a cost-effective probabilistically based noise-tolerant circuit, an 8-bit MRF_CLA, using the TSMC 0.13-μm CMOS process. The proposed master-and-slave MRF_CLA can provide a $7.00 \times 10^{-5}$ BER under a 10.6-dB SNR, while the conventional CMOS_CLA can only provide an $8.84 \times 10^{-3}$ BER. The master-and-slave MRF_CLA can operate under 0.25 V to tolerate noise interference with only 1.9 μW/MHz of energy consumption. Moreover, the transistor count can be reduced by 42% as compared to the direct-mapping MRF_CLA design [1].
IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 56, NO. 11, NOVEMBER 2009

[16] J. Han, J. Gao, Y. Qi, P. Jonker, and J. A. B. Fortes, “Toward hard-

ware-redundant, fault-tolerant logic for nanoelectronics,” IEEE Des.


2001, pp. 216–221.


probabilistic approach to nano-computing,” in Proc. IEEE Non-Silicon


[21] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky,

“Designing logic circuits for probabilistic computation in the presence


hardware-efficient probabilistic-based noise-tolerant circuit design and

implementation with 24.5 dB noise-immunity improvement,” in Proc.


Markov random field model and error backpropagation network,” in


circuit based error concealment method for video communication in an

error prone environment,” in Proc. IEEE Int. Conf. Acoust., Speech,


[27] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky,

“Designing MRF based error correcting circuits for memory elements,”


distribution to hidden Markov modeling in isolated word recognition,”


[29] [Online]. Available: http://www-device.eecs.berkeley.edu/~ptm/


Bahar, “Thermally-induced soft errors in nanoscale CMOS circuits,”


[31] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky,

“Optimizing noise immune nanoscale circuits using principles of

Markov random fields,” in Proc. Great Lakes Symp. VLSI, Apr. 2006,

pp. 149–152.

[32] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky,

“Techniques for designing noise-tolerant multi-level combinational cir-


[33] K. K. Pathi, VLSI Digital Signal Processing Systems: Design and Im-


I-Chyn Wey was born in Taipei, Taiwan, in 1979. He received the B.S.
and M.S. degrees in electronics engineering from Chang Gung University, Taoyuan, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree in electronics engineering from National Taiwan University, Taipei, in 2008. He is currently an Assistant Professor with Chang Gung University. His research interests include VLSI CMOS circuit design, noise-tolerant CMOS circuits, and ultralow-power CMOS circuit design.

You-Gang Chen was born in Tainan, Taiwan, in 1980. He received the B.S. and M.S. degrees in electronics engineering from Chang Gung University, Taoyuan, Taiwan, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree in electronics engineering at National Taiwan University, Taipei, Taiwan.

His research interests include VLSI CMOS circuit design, noise-tolerant CMOS circuits, and I/O circuit design.

Chang-Hong Yu received the Ph.D. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2007. He is currently with the College of Information and Communication Engineering, Zhejiang Gongshang University, Hangzhou, China.

An-You (Andy) Wu (S’91–M’96) received the B.S. degree from National Taiwan University (NTU), Taipei, Taiwan, in 1987 and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1992 and 1995, respectively, all in electrical engineering.

In August 2000, he joined the faculty of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, where he is currently a Professor. Since August 2007, he is on leave from NTU and serves as the Deputy General Director of SoC Technology Center (STC), Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan. His research interests include low-power/high-performance VLSI architectures for DSP and communication applications, adaptive/multirate signal processing, reconfigurable broadband access systems and architectures, and System-on-chip (SoC)/network-on-chip (NoC) platform for software/hardware co-design.

Dr. Wu now serves as the Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS.

Jie Chen (S’96–M’98–SM’02) received the Ph.D. degree from the University of Maryland, College Park, in 1998.

He is an Associate Professor with the Department of Electrical and Computer Engineering and the Department of Biomedical Engineering, University of Alberta, Edmonton, AB, Canada. He is also a Research Officer with the National Institute of Nanotechnology, Ottawa, ON, Canada. He has published about 85 peer-reviewed papers and is the holder of seven patents. His research interest is in the area of nanoscale electronics and cross-disciplinary biomedical nanotechnology.

Dr. Chen has been serving as an Associated Editor for several IEEE magazines and journals. He was the recipient of the Distinguished Lecturer Award from the IEEE Circuits and Systems Society in 2003. His cross-disciplinary research was also listed as one of 2006’s Reader’s Digest Canadian medical breakthroughs. He was the recipient of the Canadian Foundation of Innovation Leaders’ Opportunity Award in February 2008 and the Best Student Paper Award in the IEEE/National Institutes of Health (NIH) 2007 Life Science Systems & Applications Workshop in November 2007 at NIH, Bethesda, MD.