Quantum Boolean Circuits are 1-Testable

Yao-Hsin Chou, *Student Member, IEEE*, I-Ming Tsai, and Sy-Yen Kuo, *Fellow, IEEE*

***Abstract*—Recently, a systematic procedure was proposed to derive a minimum input quantum circuit for any given classical logic with the generalized quantum Toffoli gate, which is universal in Boolean logic. Since quantum Boolean circuits are reversible, we can apply this property to build quantum iterative logic array (QILA). QILA can be easily tested in constant time (C-testable) if stuck-at fault model is assumed. In this paper, we use Hadamard and general controlled-controlled NOT gates to make QILA 1-testable. That is, for any quantum Boolean circuit, the number of test patterns is independent of both the size of the array and the length of the inputs.**

***Index Terms*—C-testable, design for testability (DFT), iterative logic array (ILA), M-testable, quantum circuit, quantum computation, reversible circuit.**

I. Introduction

NANOTECHNOLOGY has made the semiconductor industry to keep up with the growth of consumers' performance–capacity demands. Sophisticated semiconductor fabrication techniques are used for the production of nanoscale structures. By nanometer technologies, there are more transistors fabricated on a single chip with increasing integration scale, thus reducing the cost per transistor.

However, nanometer-scale devices have much higher manufacturing fault rates and are more sensitive to failures of transistors and wires owing to many external factors. Consequently, the difficulty of testing each transistor increases as the complexity of devices increases. Testing such highly complex and dense circuits becomes very difficult and expensive.

On the other hand, when devices are getting smaller and smaller, the quantum effect appears. With nanoscale phenomenon such as superposition or entanglement, we can perform *quantum computation* to accomplish some tasks that are classically impossible. Some of these examples are Shor's factorization and Grover's search algorithm [1], [2].

It is interesting to note that these nanophenomena can be used to solve the circuit testing problem as well. Previous study showed that any classical circuit can be implemented by a straightforward replacement algorithm with auxiliary qubits as intermediate storage. Recently, a construction procedure of minimum input quantum Boolean circuit was proposed [3]–[8]. The constructed quantum Boolean circuits are reversible. Such circuits are of interest for several reasons.

One of the important reasons is energy saving. Reversible circuits are information lossless and hence tend to dissipate relatively little energy. According to Landauer's principle, it is possible to construct a computer using reversible circuits that can compute with arbitrarily small amounts of energy [9].

Another reversible circuit's property of particular interest is that the Boolean function of a reversible circuit is bijective (one-to-one and onto); hence, it can be used to form an *iterative logic array* (ILA), a system that consists of identical modules arranged in a geometrically regular interconnection pattern. ILA is well known to be easily testable [10]–[20].

In this paper, we study the testing properties on quantum ILA (QILA). QILA consists of quantum circuits constructed from a library of universal reversible gates, including quantum NOT (N), controlled NOT (CN), generalized controlled-controlled NOT (CCN), and Hadamard gates. Under stuck-at fault and cell fault model, we show that any QILA and quantum Boolean circuits are 1-testable. That is, the circuits can be tested with only one test pattern.

The rest of this paper is organized as follows. Section II introduces basic notations and preliminaries about the quantum circuits and the fault models we used in this paper. Section III illustrates and proves the testability of quantum Boolean circuits. Finally, conclusions are given in Section IV.

II. Preliminaries and Notations

A. Testability and Fault Model

Although the general logic testing problem is known to be *NP-complete* [17], it is much easier to deal with the testing problem for *ILAs*. The definitions and properties of ILA architectures can be found in [18]. ILA is a well-known and demonstrated test scheme for systolic-like array architectures. It provides a systematic way to test all cells in an array with only the same number of test patterns for a single cell. There is an assumption that the cell's behavior in the array is invariant over time, even if it is faulty. A faulty cell's function may deviate from the correct one in any manner, as long as it remains combinational. In other words, we are testing for permanent combinational faults only. If all cells of an ILA have two directional input/output pins, we can construct a two-dimensional ILA architecture with these cells. A one-dimensional ILA architecture is illustrated in Fig. 1.

We call an ILA a *C-testable* array if it is testable with a constant number of test patterns. An *M-testable* array is an array

Manuscript received November 29, 2007; revised January 10, 2008. This work was supported by the National Science Council, Taiwan under Grant NSC 95-2218-E-002-004. The review of this paper was arranged by Associate Editor K. Likharev.

Y.-H. Chou is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: gugu@lion.ee.ntu.edu.tw).

I-M. Tsai is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, and also with Chunghwa Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan 32601, Taiwan, R.O.C. (e-mail: imtsai@cht.com.tw).

S.-Y. Kuo is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C., and also with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan, R.O.C. (e-mail: sykuo@cc.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNANO.2008.926369

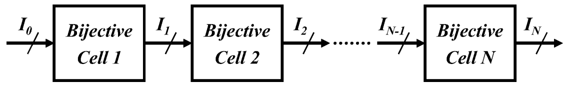

Fig. 1. 1-D ILA architecture.

testable with a constant number of test patterns, and in addition, this number is a minimum. Therefore, M-testable techniques are more desirable than C-testable techniques. In the past, the *single-cell fault model* was used as the array-level fault model by most researchers. In this paper, two assumptions are made for the fault models used in the quantum Boolean circuit and QILA.

First, the fault on the cell level is either stuck-at zero or stuck-at one. That is, we use the *stuck-at fault model* in this paper. The stuck-at fault model is a logical fault model successfully used for decades. A stuck-at fault influences the logic state on a line in a logic circuit. The correct value on a faulty line is transformed into a constant logic value, which seems to be stuck, either at logic 0 or at logic 1, referred to as a *stuck-at-0* (SA0) or *stuck-at-1* (SA1) fault, respectively. Because of the reversible and bijective property of the quantum Boolean circuit, a stuck-at fault will propagate from gate to gate and from cell to cell. Without loss of generality, in the following sections, the stuck-at faults are assumed to be permanent (time invariant) on lines of a logic cell and can always be observed on the primary outputs of an array. This assumption is reasonable for quantum computation in a semiconductor-based system since any tiny piece of dust or irregular geometric shapes will cause a defect under nanoscale. In addition, process variations or manufacturing imperfections can result in defects. Process variations affect threshold voltage and channel length of transistors as well as interconnect width and thickness of the metal. Imperfections at random locations will result in resistive short between metal lines or open in metal lines. Any one of these defects may results in stuck-at faults.

Second, verifying a cell function involves generating inputs for the cell (i.e., controlling the cell) and propagating faults from the cell to the outputs (i.e., observing the cell). That is, it is sufficient to verify the function of every cell in the array to complete the testing of an array. This model is called the *cell fault model*, and it has been used as the array-level fault model by most researchers in this field. Verifying the Boolean function of a quantum circuit cell should only involve generating input and propagating faults for any quantum circuit before the quantum measurement. Thus, for a quantum circuit, this assumption is also true as long as it is not being measured. Basically, a quantum circuit fault caused by a defect (interaction with the environment) is just like measuring the qubits, once collapsed, the fault remains and propagates as in a classical circuit without nanophenomenon (quantum property). Many implementations such as optical- or semiconductor-based quantum computer may have permanent faults like short or leakage caused by imperfect materials or broken wires. Hence, these two assumptions are both reasonable for quantum Boolean circuits.

B. Quantum Computation, Gate and Circuit

In a two-level quantum system, each bit can be represented by a basis consisting of two eigenstates denoted by $|0\rangle$ and $|1\rangle$. Any state can be written as a linear combination of these two orthonormal eigenvectors as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $\alpha, \beta \in \mathcal{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. Upon a measurement, the system is projected to one of its basis (i.e., either $|0\rangle$ or $|1\rangle$). The overall probability of getting a particular state is given by the absolute square of its amplitude. That is, $|\alpha|^2$ and $|\beta|^2$ for getting $|0\rangle$ and $|1\rangle$, respectively. To distinguish the system described before from classical binary logic, a bit in a quantum system is called a quantum bit, or *qubit*. Multiple qubits can form a quantum system jointly. For example, the space of an uncorrelated two-qubit system is the tensor product ($\otimes$) of their spaces. As a result, the state of two-qubit system is spanned by the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. For example, $|\psi'\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$. In general, the space of an n-qubit system can be modeled as a 2^n-dimensional complex vector space.

A quantum system can be manipulated by unitary transformations called *quantum gates*. An example of quantum gates is the quantum NOT (N) gate. When a qubit $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ goes through a quantum NOT gate, the state changes to $|\phi'\rangle = \alpha|1\rangle + \beta|0\rangle$. Another commonly used quantum gate is the *Hadamard* (H) gate, which changes $|0\rangle \to (1/\sqrt{2})|0\rangle + (1/\sqrt{2})|1\rangle$ and $|1\rangle \to (1/\sqrt{2})|0\rangle - (1/\sqrt{2})|1\rangle$. Applying H twice is equivalent to an identity operation (I). That is,

$$H(|0\rangle) = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = |0\rangle \tag{1}$$

$$H(|1\rangle) = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \qquad H\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |1\rangle. \tag{2}$$

Note that the single-qubit state described before exhibits an interesting phenomenon in quantum mechanics called *superposition*. For example, if a qubit in state $|1\rangle$ is being measured, the outcome will be $|1\rangle$ for certain; but if a Hadamard gate is applied before the measurement, there will be a probability of $0.5 = |1/\sqrt{2}|^2$ that the qubit is found in $|0\rangle$ and a probability of $0.5 = |-1/\sqrt{2}|^2$ in $|1\rangle$ state. However, if this qubit is measured along the x-axis, which means the system is measured by the basis $\{|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}\}$, it is 100% for sure that the qubit is in the state $|-\rangle = 0 \cdot |+\rangle + 1 \cdot |-\rangle$.

An important example of two-qubit gate is the CN gate. It consists of a *control* qubit c and a *target* qubit t. The target bit is inverted when control bits is in $|1\rangle$ state. The CN gate is also called the XOR gate. In other words, the gate changes the state from $|x\rangle_c|y\rangle_t$ to $|x\rangle_c|x \oplus y\rangle_t$, where "$\oplus$" denotes EXCLUSIVE OR and subscript c and t indicate the control bit and target bit, respectively. The symbols for the N, H, CN gates, and quantum measurement are shown in Fig. 2.

Similarly, an example of three-qubit gate is the CCN (or Toffoli) gate. It consists of two control bits, x and y, which do not change their values, and a target bit z, which changes its value only if $x = y = 1$. The bit-wise operation can be written as $\text{CCN}(|x, y, z\rangle_{c,c,t}) = |x, y, (x \cdot y) \oplus z\rangle_{c,c,t}$, where "$\cdot$" (dot)

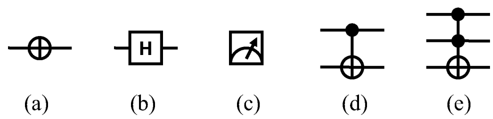

Fig. 2. Symbols for (a) N, (b) H, (c) measurement, (d) CN, and (e) CCN gates.

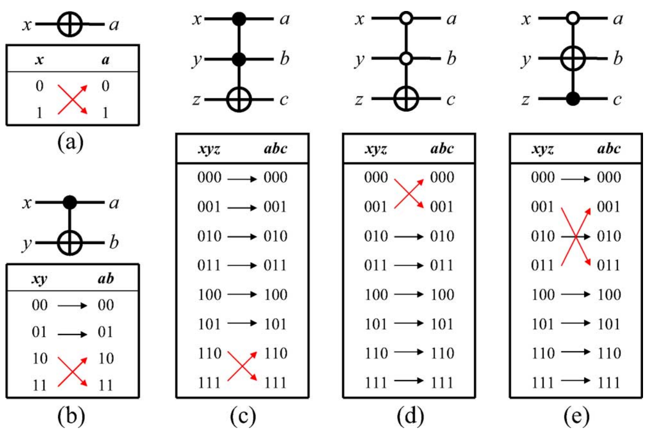

Fig. 3. Symbols and Boolean functions for (a) N, (b) CN, and (c) CCN gate. (d) Zero-controlled CCN gate. (d) and (e) Generalized CCN gates.

stands for bit-wise AND operation. The Boolean functions of N, CN, CCN, and some generalized CCN gates are shown in Fig. 3.

III. Testability of Quantum Boolean Circuits

A. C- and M-Testabilities of Quantum Boolean Circuit

In this section, we study the testing properties of quantum Boolean circuit that is an important subclass of quantum reversible circuits constructed from a library of quantum reversible gates called k-controlled NOT ($C^{\otimes^k}$ NOT) gates. A $C^{\otimes^k}$ NOT gate has $k+1$ inputs and $k+1$ outputs. It keeps the first k input signals unchanged, and inverts the last input signal if and only if the first k inputs are all 1. Clearly, this input/output mapping is reversible. If $k=0$, a $C^{\otimes^k}$ NOT is a simple quantum N (NOT) gate. $C^{\otimes^1}$ NOT and $C^{\otimes^2}$ NOT gates are also known as the CN and CCN gates, respectively.

Definition 1: A quantum Boolean circuit is a quantum circuit that implements Boolean functions. The input and output spaces are only the eigenspace of the quantum system.

Lemma 1: Any classical circuit can be transformed into a quantum Boolean circuit.

Proof: This is true since quantum Toffoli gate is a universal gate for classical circuit. By setting target bit to $|1\rangle$ at the third input, we can get $\mathrm{CCN}(|x,y,1\rangle) = |x,y,(x\cdot y)\oplus 1\rangle = |x,y,\overline{x\cdot y}\rangle$, which is the NAND function. NAND gate is well known as a universal gate in classical computation. Also, we can use the CN gate in quantum Boolean circuit as the fanout in classical circuit. By setting target bit to $|0\rangle$ in a CN gate, we have $\mathrm{CN}(|x,0\rangle) = |x, x\oplus 0\rangle = |x,x\rangle$, which is a fanout. Similarly, AND, OR, and NOT functions can be easily replaced by $C^{\otimes^k}$ NOT gates. Following the fact that {AND, OR, NOT} is universal, any classical circuit can be transformed into quantum Boolean circuit. ■

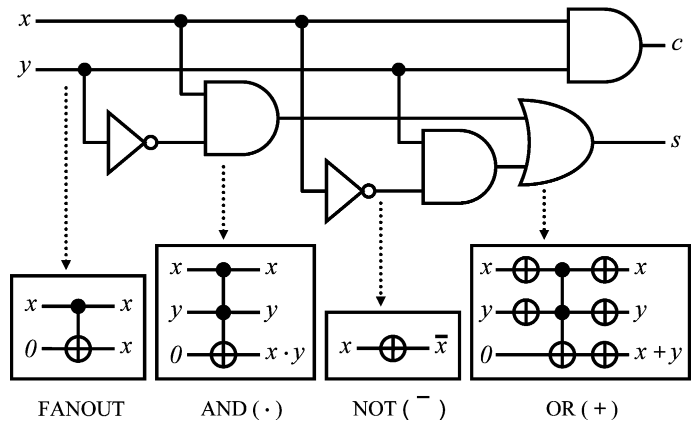

Fig. 4. Straightforward replacement for classical gates in a half adder.

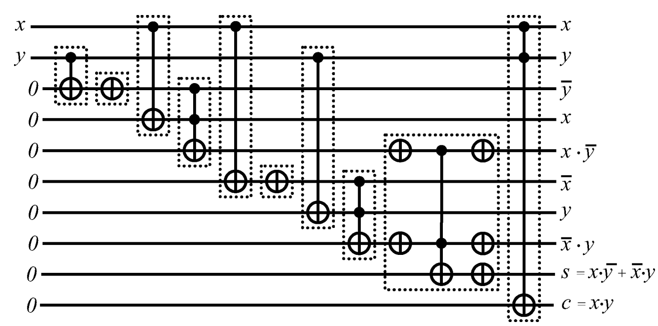

Fig. 5. Direct implementation of a quantum half adder with eight auxiliary qubits.

For example, Figs. 4 and 5 show the quantum circuit implementation with direct replacement for classical gates in a half adder, where the inputs of the half adder are x, y, and the output *sum* and *carry* are $s = \bar{x}\cdot y + x\cdot\bar{y}$ and $c = x\cdot y$, respectively.

Theorem 1: Any classical circuit can be transformed into quantum Boolean circuit with minimal space.

Proof: Previous study showed that any classical circuit can be implemented by the straightforward replacement algorithm with auxiliary qubits as intermediate storage [5], [6]. However, this inefficient implementation causes a large number of auxiliary qubits to be used (see Fig. 5). In our previous work [7], [8], a systematic procedure was proposed to obtain a minimum space quantum circuit for a given classical Boolean function. We first formulate the problem of transforming an m-to-n bit classical Boolean logic into a t-bit unitary quantum operation. The eligible solution set is then constructed such that a solution can be easily found by selecting any member from this set. We also showed that the algorithm is optimal in terms of space. A constructed minimum space quantum half adder is shown in Fig. 6.

In general, for an m-bit input and n-bit output classical circuit, we need $t = \max(m, p+n+z)$ input/output qubits to realize the same Boolean function of the classical circuit, where p is the number of preserved input bits and z is the number of bits for distinguishing the same value in the outputs. It has been proved that t is the minimum value, mainly because that if $t < \max(m, p+n+z)$, the number of qubits will not be sufficient to construct a reversible quantum circuit. ■

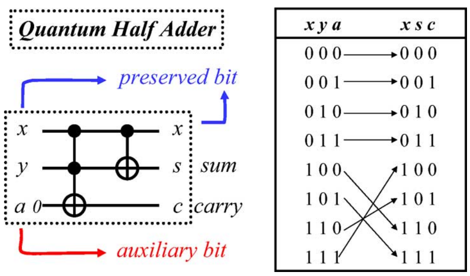

Fig. 6. Minimum space quantum half adder with one preserved qubit.

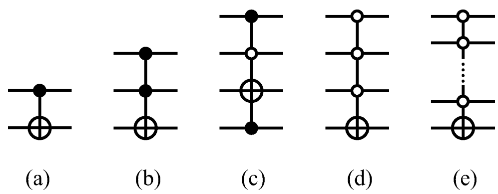

Fig. 7. Different kinds of gates in a *generalized* $C^{\otimes^k}$ NOT gate family. (a) $C^{\otimes^1}$ NOT (CN) gate. (b) $C^{\otimes^2}$ NOT (CCN) gate. (c) and (d) generalized $C^{\otimes^3}$ NOT gate. (d) and (e) ZCN gates.

Here, we can conclude that at most $t = m + n + \log_2 n$ bits are needed to convert an m-to-n bit classical circuit to a reversible t-bit input/output quantum circuit. Comparing with Fig. 5 that uses eight auxiliary bits, Fig. 6 shows a minimum space quantum Boolean circuit of the half adder where $m = n = 2$ and $p = 1, z = 0$. Note that this three-qubit circuit is a minimal space quantum half adder since we cannot distinguish the same output $(c, s) = (0, 1)$ from two different inputs $(x, y) = (0, 1)$ and $(x, y) = (1, 0)$ if we have not preserved any input bit. That is, $z = 0$ iff $p >= 1$, but if $p = 0$, $z = 1$.

Definition 2: A *generalized* $C^{\otimes^k}$ NOT gate has $k+1$ input/output lines, one of which is the target bit and the others are the control bits. Given $t \in \{0, 1, 2, \ldots, k\}$ as the wire number of the target bit and $C = (c_1, c_2, \ldots, c_k)$ where $c_i \in \{0, 1\}$ as the values of control bits.

For example, if $k = 2$, $t = 2$, $C = \{1, 1\}$ or $k = 1$, $t = 1$, $C = \{1\}$, f is the function of CCN or CN gate, respectively. The quantum circuit of $k = 3, t = 2, C = \{1, 0, 1\}$ was shown in Fig. 7(c). Note that if C equals to zero, $C = \{0, 0, \ldots, 0\}$, these gates are usually called as *zero-controlled* NOT (ZCN) gates.

Corollary 1: Any quantum Boolean circuit consists of only *generalized* $C^{\otimes^k}$ NOT gates.

Proof: It has been shown that we can transform any classical circuit into its quantum version with CCN gate only [5], [6], and into *minimal space* quantum Boolean circuits by using only quantum N, CN, and generalized CCN gates [7], [8]. Since these gates are all of a subset of *generalized* $C^{\otimes^k}$ NOT gates, as a result, quantum Boolean circuits can consist of only *generalized* $C^{\otimes^k}$ NOT gates. ■

Definition 3: A cell function $f : \Sigma \rightarrow \Delta$ where $\Sigma = \{0, 1\}^I$ and $\Delta = \{0, 1\}^O$ for $I, O \in N$ is *injective* if $\forall i_1 \neq i_2, f(i_1) \neq f(i_2)$. If a function is injective and $\Sigma = \Delta$, then the function is *bijective*. A gate or a circuit is called *bijective* if its Boolean function is *bijective*.

Lemma 2: Any *generalized* $C^{\otimes^k}$ NOT gate is bijective.

Proof: Let t be target bit number and $C = (c_1, c_2, \ldots, c_k)$ be the values of control bits of a *generalized* $C^{\otimes^k}$ NOT gate. The Boolean function of the gate is $f_{C,t}(\mathbb{X}) : \{0, 1\}^{k+1} \rightarrow \{0, 1\}^{k+1}$ that maps $f_{C,t}(x_1, x_2, \ldots, x_t, x, x_{t+1}, \ldots, x_k)$ to $(x_1, x_2, \ldots, x_t, \bar{x}, x_{t+1}, \ldots, x_k)$, if $\forall i \in \{1, 2, \ldots, k\}$, $x_i = c_i$; otherwise, $f_{C,t}(\mathbb{X}) = \mathbb{X}$. In other words, it only swaps two of the input patterns with each other at the output, i.e., from $(c_1, c_2, \ldots, c_t, x, c_{t+1}, \ldots, c_k)$ to $(c_1, c_2, \ldots, c_t, \bar{x}, c_{t+1}, \ldots, c_k)$ and *vice versa*. Other input patterns remain unchanged. Clearly, this function is bijective as well. Hence, *generalized* $C^{\otimes^k}$ NOT gates are bijective. ■

Theorem 2: Any quantum Boolean circuit is bijective.

Proof: By *Corollary* 1 and *Lemma* 2, a quantum Boolean circuit can be made to only consist of a sequence of n generalized k-controlled NOT gates, denoted as $\{G_1, G_2, \ldots, G_n\}$ where $G_i \in$ *generalized* $C^{\otimes^k}$ NOT, for every $i \in \{1, 2, \ldots, n\}$ and $k \in \mathbb{N}$. Let $\{f_1, f_2, \ldots, f_n\}$ be the Boolean function of $\{G_1, G_2, \ldots, G_n\}$, respectively, then the quantum Boolean circuit output of $G_n \cdots G_2 G_1 |x\rangle$ equals to the value of Boolean function $f_1 \circ f_2 \circ \cdots \circ f_n(x)$.

Assume that $f = f_1 \circ f_2 \circ \cdots \circ f_n$ and g are bijective functions from Σ to Δ and Δ to Ω, respectively, then $\Sigma = \Delta = \Omega$ and f, g are injective. Because f is injective, if $x_1 \neq x_2$, $f(x_1) \neq f(x_2)$; also if $f(x_1) \neq f(x_2)$, $g(f(x_1)) \neq g(f(x_2))$ since g is injective. Thus, $g(f(x))$ is also injective since $g(f(x_1)) \neq g(f(x_2))$, if $x_1 \neq x_2$. That is, the function $f \circ g = g(f(x))$ from Σ to Ω is injective and $\Sigma = \Omega$. Therefore, $f \circ g$ is a bijective function. Consequently, if f_{n+1} is also bijective and we let $g = f_{n+1}$, then $f_1 \circ f_2 \circ \cdots \circ f_n \circ f_{n+1}$ is bijective as well. By mathematical induction, $\circ_{i=1}^{k} f_i$ is bijective $\forall k \in \mathbb{N}$.

Note that if bijective functions $f : \{0, 1\}^n$ and $g : \{0, 1\}^m$ are not of the same dimension, say $n < m$, we can make a function f' of the dimension m and keep it bijective by letting $f'(x \cdot 2^{m-n} + y) = f(x) \cdot 2^{m-n} + y$, $\forall x \in \{0 \sim 2^n - 1\}$ and $y \in \{0 \sim 2^{m-n} - 1\}$ where x, y are binary encoded. It is like an m-input gate that interacts with only n input lines, and keeps intact with the other $m - n$ lines. After that, $\forall x_1 \neq x_2, f'(x_1) \neq f'(x_2)$. Thus, f' and $f' \circ g$ are still bijective.

Finally, by *Corollary* 1 and *Lemma* 2, any quantum Boolean circuit can consist of only bijective gates. That is, the Boolean functions $\{f_1, f_2, \ldots, f_n\}$ of quantum gates $\{G_1, G_2, \ldots, G_n\}$ in a quantum Boolean circuit are all bijective. As a result, quantum Boolean circuits with finite numbers of gates are all bijective. Since $\circ_{i=1}^{n} f_i = f_1 \circ f_2 \circ \cdots \circ f_n$ is bijective $\forall n \in \mathbb{N}$, any quantum Boolean circuit is bijective. ■

Definition 4: An ILA is a regularly structured circuit that consists of an array of identical cells.

Corollary 2: Any ILA with bijective cell functions is bijective.

Proof: The proof is similar to *Theorem 2.* A compound function of many bijective functions is bijective as well. The Boolean function after regularly structured bijective cells is still bijective. Hence, the ILA with bijective cell functions is bijective. ■

Definition 5: A C-testable array is an array testable with a constant number of test patterns independent of the size of the array.

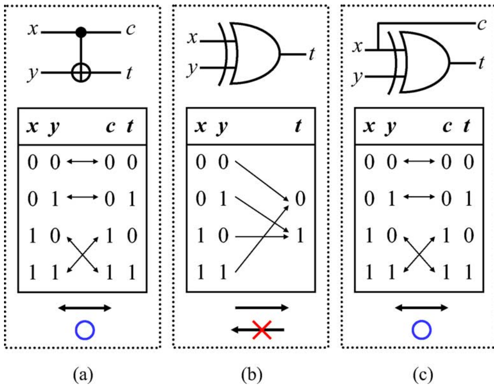

Fig. 8. (a) Quantum XOR (CN) gate. (b) Irreversible classical XOR gate. (c). Making the classical XOR gate reversible by preserving one input bit. The symbol "×" indicates the irreversible function of the circuit, whereas "◯" represent the reversible ones.

Theorem 3: Any ILA with bijective cell functions is C-testable.

Proof: If a so-called "complete" or "exhaustive input sequence" for a cell, which is an input sequence including all input combinations of the cell $\{0, 1, \sim, 2^n - 1\}$, $(0, 0, \ldots, 0), (0, 0, \ldots, 1), \ldots, (1, 1, \ldots, 1)$, is entering a perfect cell of an ILA, the output sequence must be "complete" since the Boolean function of the cell is bijective. The complete sequence from the first cell's output will then be the input of the second cell and will propagate on and on (see Fig. 1). The main reason of why an ILA with bijective cell functions is C-testable is that any fault can automatically propagate to the end of the array and to some observable primary outputs. It has been shown that a d-dimensional ILA ($d \in \mathbb{N}$) is C-testable if it has bijective cell functions [18]. In this paper, we only consider about 1-D arrays. ■

Definition 6: An m-input, m-output, totally specified Boolean function $f(X)$, $X = \{x_1, x_2, \ldots, x_m\}$ is reversible if it maps each input assignment to a unique output assignment [21]. An n-input, n-output gate or circuit is reversible if it realizes a reversible function.

Corollary 3: Any reversible Boolean circuit is bijective.

Proof: If a Boolean logic gate is reversible, every distinct input must yield a distinct output, and the number of inputs and outputs must be equal. Hence, the Boolean function of the gate, the mapping of inputs to outputs, is bijective. The remaining part of the proof is also similar to *Theorem 2*; a function of another bijective function is bijective. Then, the following can be easily proved by contradiction. Assume that there is a reversible circuit that is not bijective, i.e., there must be two or more inputs which map to the same one output. Thus, it will be irreversible that contradicts the assumption (→←). Hence, all reversible Boolean circuits are bijective. ■

For example, Fig. 8 illustrates the comparison of the quantum, classical, and reversible XOR gates.

Definition 7: A reversible ILA (RILA) is a regularly structured circuit that consists of an array of reversible Boolean circuit cells.

Corollary 4: Any RILA is C-testable.

Proof: By *Corollaries* 2 and 3 and *Theorem 3*, RILA is C-testable. ■

Other testing properties of reversible circuits are also elaborated in [19]–[25].

Definition 8: An M-testable array is an array testable with the minimal number of test patterns independent of the size of the array.

Definition 9: A QILA is a regularly structured circuit that consists of an array of quantum Boolean circuit cells.

Theorem 4: Any QILA is C-testable and M-testable.

Proof: A QILA is a regularly structured ILA composed of arrays of cells and each cell of the QILA is a quantum Boolean circuit. Thus, by *Theorem 2*, each of the cell functions of the QILA is bijective. Hence, by *Theorem 3*, the QILA is C-testable. Note that, unlike classical ILAs, each cell of the QILA needs not to be identical. Similar proof is given in the third paragraph of the proof of Theorem 2. Furthermore, since QILA is C-Testable and by *Theorem 1*, any classical circuit can be transformed into its quantum version with minimal input/output lines, the transformed quantum circuits are also bijective and suitable for constructing an ILA with the minimum number of test patterns independent of the size of the array, which is by definition M-testable. ■

Now, we have some good properties about quantum Boolean circuits. Given classical Boolean circuits, we can transform them into quantum Boolean circuits or/and arrange them to form some QILAs. They are all C-testable and M-testable.

B. 1-Testability of Quantum Boolean Circuit

It is well known that transferring classical circuits into a C-testable ILA is not an easy task. It needs many design-for-testability techniques and huge overhead. Besides, for an m-bit input and n-bit output classical circuit, say $m = 128$ and $n = 100$, we can hardly deal with so many testing patterns for all 2^m to 2^n possible mappings. The number of test patterns grows exponentially with the number of inputs. Though the number of test patterns is a constant number or even a minimum number such as 2^{64}, an *exhaustive testing*, which applies all 2^m possible input patterns for testing stuck-at faults, is not practical when m increases.

However, by using quantum properties, such as Walsh–Hadamard transformation, to achieve quantum parallelism, $H^{\otimes^m}|0\rangle^{\otimes^m} = (1/\sqrt{2^m})\sum_{x=0}^{2^m-1}|x\rangle$ and $H^{\otimes^m}((1/\sqrt{2^m})\sum_{x=0}^{2^m-1}|x\rangle) = |0\rangle^{\otimes^m}$, we can test the whole array of circuits with one test pattern $|0\rangle^{\otimes^n} = |000\cdots0\rangle$. We call it "1-testable" since it needs only one single test pattern and one output response. It is noteworthy that the number of test patterns is independent of the length of the inputs as well as the size of the array.

Taking a circuit of two qubits as an example, the initial state $|00\rangle = |0\rangle|0\rangle = |0\rangle \otimes |0\rangle$ after applying the Hadamard gates is

$$H^{\otimes^2}|00\rangle \quad (3)$$

$$= (H \otimes H)|0\rangle \otimes |0\rangle \quad (4)$$

$$= H|0\rangle \otimes H|0\rangle \quad (5)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{6}$$

$$= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{7}$$

This is a superposition of all combinations of possible inputs with equally weighted amplitude. If a circuit under test (CUT) without any fault is tested, a circuit of bijective function is added that does not change the state. After applying the Hadamard gates once again at the end of CUT, the quantum state will become

$$(H \otimes H)\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \tag{8}$$

$$= (H \otimes H)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{9}$$

$$= H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \tag{10}$$

$$= |0\rangle \otimes |0\rangle = |00\rangle \tag{11}$$

which returns to the initial state.

However, if the CUT has a stuck-at zero fault on the first qubit, the state in the tested cell will become

$$|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle). \tag{12}$$

After the circuit is tested and the Hadamard gates are applied, it will become

$$(H \otimes H)\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \tag{13}$$

$$= (H \otimes H)|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{14}$$

$$= H|0\rangle \otimes H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \tag{15}$$

$$= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle. \tag{16}$$

Hence, at the end of circuit testing, if we measure these two qubits, the second qubit will be in state $|0\rangle$ while the first qubit may be either in state zero or in state one with equal probability.

Similarly, we can easily make a superposition consisting of all $2^3 = 8$ classical test vectors with only three qubits. Assuming that we want to test a quantum half adder (the dashed box in Fig. 9), we can just apply a test pattern in the beginning

$$|0\rangle \otimes |0\rangle \otimes |0\rangle = |0\rangle|0\rangle|0\rangle = |000\rangle \tag{17}$$

as shown in stage (a) of Fig. 9.

The joint state after Hadamard gates and before the CUT (in this case, the quantum half adder) will become

$$H^{\otimes^3}|000\rangle \tag{18}$$

$$= (H \otimes H \otimes H)|0\rangle \otimes |0\rangle \otimes |0\rangle \tag{19}$$

$$= H|0\rangle \otimes H|0\rangle \otimes H|0\rangle \tag{20}$$

$$= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{21}$$

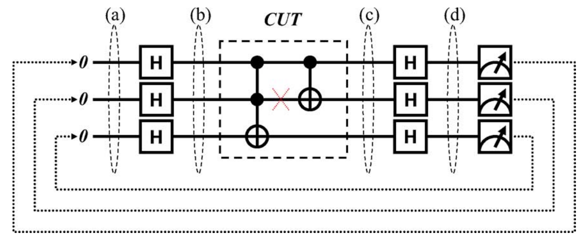

Fig. 9. Testing a quantum half adder with one test pattern.

$$= \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{22}$$

at stage (b) in Fig. 9.

If the input state of this quantum half adder is

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{23}$$

the joint state of output will be

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |\mathbf{110}\rangle + |\mathbf{111}\rangle + |\mathbf{101}\rangle + |\mathbf{100}\rangle). \tag{24}$$

Note that the Boolean function of any quantum Boolean circuit is bijective, so this quantum half adder is bijective as well (see Fig. 6). Thus, the testing pattern (an equally weighted superposition) will not be changed since all of the $8(= 2^3)$ eigenvectors used here will remain with equally weighted amplitude under the commutative linear system. Hence, the input state and output state will be the same.

As a result, if a quantum half adder circuitry without any fault is tested, after applying the second column of Hadamard gates at the end of CUT, the quantum state will become

$$(H \otimes H \otimes H)\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |110\rangle + |111\rangle + |101\rangle + |100\rangle) \tag{25}$$

$$= (H \otimes H \otimes H)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{26}$$

$$= H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \tag{27}$$

$$= |0\rangle \otimes |0\rangle \otimes |0\rangle = |000\rangle. \tag{28}$$

Now, if there is a stuck-at-one fault occurring in the second qubit, the joint state becomes

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{29}$$

$$= \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{30}$$

$$= \frac{1}{\sqrt{4}}(|010\rangle + |011\rangle + |110\rangle + |111\rangle) \tag{31}$$

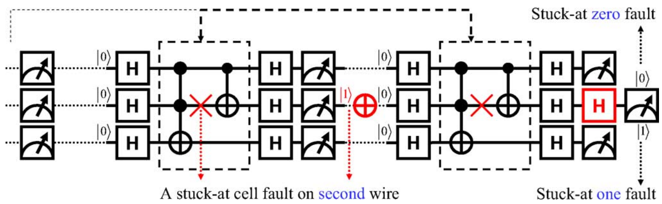

Fig. 10. Identifying the faulty wires and types of stuck-at faults.

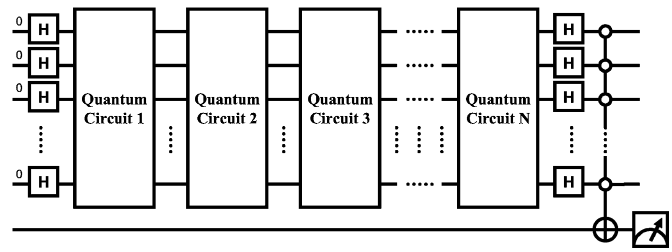

Fig. 11. Sample QILA with only one test pattern and one readout.

at stage (c) of Fig. 9. After applying the Hadamard gates, it becomes

$$(H \otimes H \otimes H)\frac{1}{\sqrt{4}}(|010\rangle + |011\rangle + |110\rangle + |111\rangle) \quad (32)$$

$$= (H \otimes H \otimes H)\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (33)$$

$$= H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes H|1\rangle \otimes H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \quad (34)$$

$$= |0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle \quad (35)$$

at stage (d) in Fig. 9. After the measurement, both the first and third qubits must be in the zero state. However, the second qubit will be either in state $|0\rangle$ or in state $|1\rangle$ with equal probability ($|1/\sqrt{2}|^2 = |-1/\sqrt{2}|^2 = 0.5$ and $|1/\sqrt{2}|^2 + |-1/\sqrt{2}|^2 = 1$).

As a result, if one or more $|1\rangle$ states are measured at the end of testing, there must be some faults in the CUT. That is, each time we have a 50% detection rate under the single-fault model. Hence, by simply repeating the trial for t times, there will be a probability of $1 - (1 - 1/2)^t$ to detect if there is a single fault. That is, if $t = 10$, we have a confidence over 0.999 (three nines) of fault detection and we need only $t = 20$ trials to have a six nines reliability (over 99.9999%). Besides, in the cases of multiple faults on the same or different wires, the confidence will be the same or even higher. By adjusting t, the probability of false positive decreases exponentially. Given any arbitrary small positive number ε, we can choose $t \geq \lceil \log 1/\varepsilon \rceil$ to make the detection rate greater than $1 - \varepsilon$. Hence, the detection rate would be ≈ 1 with high probability (w.h.p).

This solution not only detects the fault, but also points out the line on which the fault has occurred. In the example mentioned before, the faulty wire will result in a superposition state $1/\sqrt{2}(|0\rangle + |1\rangle)$ or $1/\sqrt{2}(|0\rangle - |1\rangle)$, which is also known as the $|+\rangle$ or $|-\rangle$ state. Although the syndromes of both stuck-at zero and stuck-at one faults are the same (the $|1\rangle$ state), if we run the testing one more time right after the fault is detected and measure the faulty qubits along the x-axis again, we can identify the type of fault. When the measured result is in $|+\rangle$ or $|-\rangle$ state, there must be a stuck-at zero or stuck-at one fault occurring, respectively, as shown in Fig. 10.

Also, we could identify the faulty cells with 2-D array or by using the *bisection method*. We can identify a faulty cells in $O(\log_2 N)$ steps, where N is the length of the array, or the number of cells.

Definition 10: A 1-testable circuit is an array of logic cells testable with only one single test pattern that is independent of both the size of the array and the length of the inputs.

Theorem 5: Any QILA and quantum Boolean circuit are 1-testable.

Proof: According to our testing method earlier, by utilizing the superposition property of quantum computation to achieve quantum parallelism and using multiple Hadamard gates to compact the output response, QILA can be tested by one single test pattern only. Besides, no matter how many cells in an ILA and how many millions of gates in a cell (regardless of the width of each cell), this array still can be tested with one test pattern. In other words, the number of test patterns is independent of both the size of the array and the length of the inputs, which is 1-testable. Particularly, if a QILA comprises only one cell, it is undoubtedly 1-testable. That is, any quantum Boolean circuit is also 1-testable. ∎

Theoretically, any quantum Boolean circuit with input size of w can be 1-testable by using w Hadamard gates to perform a superposition of a combination of all 2^w possible inputs. In practice, however, it is better to partition the quantum circuit into smaller cells of a QILA. Nevertheless, the partitioned cells do not need to be equal in length since operations of tensor product for joint quantum system will not change each cell's bijective property in an ILA. In the future, different partitioning methods either new or used in classical design for testability (DFT) techniques are worthy investigating.

To evaluate the overhead of this proposal, we know that we first increase the number of qubits by $t - m$ for transferring classical circuits into quantum Boolean circuits and then apply $2m$ Hadamard gates. Although these two steps increase the number of bits and elementary gates that is proportional to the number of inputs, fortunately we have a superexponential speedup for the circuit testing. Also, the effective fault coverage rate will be 100% since we can perform all functional testing by applying exhaustive testing techniques, which is very hard to achieve in classical logic testing. Furthermore, we do not need extra effort to perform test pattern generation, fault simulation, output verification, and result analysis as used in classical testing.

The 1-testability of QILA only requires one test pattern and one readout, as shown in Fig. 11. Without measuring all the output bits, we can use a ZCN gate to detect the fault. Note that in a general ZCN gate, the target bit is inverted only if all its control bits are set to 0. If we found the target bit of ZCN gate was not inverted, there must be some faults in this circuit.

IV. Conclusion

There are two reasons why we propose such a novel idea on quantum Boolean circuit testing. First, since the conventional device architecture will eventually reach its physical limits, we have to take advantage of quantum physics at nanometer scale and continue the computer hardware evolution. Second, since any quantum gate transformation is a thermodynamically reversible process, a general-purpose computer implemented using nanoscale devices and quantum operations can theoretically operate with arbitrary little energy, which is an attractive feature.

It has been proved that any Boolean circuit can have its quantum version. In this paper, we showed that quantum Boolean circuits can be arranged into a 1-testable QILA. Furthermore, with superposition, which is a nanoscale phenomenon in the quantum computation, any quantum Boolean circuit and any QILA can be tested in just a few steps for any given classical Boolean function. By utilizing nanotechnology, these methods provide a smooth migration to the next generation circuit design and may intrinsically change the IC design flow in the future.

Circuit testing is now over 40 years and poses many considerable challenges. The circuits of modern electrical appliances become more and more complicated and the cost of circuit testing is rapidly increasing along with the complexity of the chip. According to International Technology Roadmap for Semiconductors 2001/1997, test cost will become critical part of the total cost in ten years. It is indispensable to control these costs and provide a cost-effective solution. Therefore, it is important to develop efficient testing approaches. Testing is also the key to many fault tolerance approaches that improve product reliability. Nowadays, every electronic device/electrical appliance we use is built by Boolean circuits. In this paper, a novel method is proposed to perform logic testing for Boolean circuit by utilizing the quantum computation. In the future, any Boolean circuit can be tested easily, quickly, and cost-effectively, so that reliable and inexpensive products can be acquired by everyone.

References

[1] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 1994, pp. 124–134.

[2] L. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 212–219.

[3] T. Toffoli, *Automata, Languages and Programming*, J. W. de Bakker and J. van Leeuwen, Eds. New York: Springer-Verlag, 1980, pp. 632–644.

[4] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, 1995.

[5] D. P. DiVincenzo, "Quantum gates and circuits," *Proc. Roy. Soc. Lond. A*, vol. 454, pp. 261–276, 1998.

[6] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[7] I.-M. Tsai and S.-Y. Kuo, "An algorithm for quantum Boolean circuit construction," in *Proc. 2001 IEEE Conf. Nanotechnol.*, pp. 111–116.

[8] I.-M. Tsai and S.-Y. Kuo, "An algorithm for minimum space quantum Boolean circuits construction," *J. Circuits, Syst., Comput.*, vol. 15, no. 5, pp. 719–738, Oct. 2006.

[9] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Dev.*, vol. 5, no. 3, pp. 183–191, 1961.

[10] W. H. Kautz, "Testing for faults in combinational cellular logic arrays," in *Proc. 8th Annu. Symp. Switching Automata Theory*, 1967, pp. 161–174.

[11] P. R. Menon and A. D. Friedman, "Fault detection in iterative arrays," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 524–535, May 1971.

[12] A. D. Friedman, "Easily testable iterative systems," *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1061–1064, Dec. 1973.

[13] S. K. Lu, J. C. Wang, and C. W. Wu, "C-testable design techniques for iterative logic arrays," *IEEE Trans. VLSI*, vol. 3, no. 1, pp. 146–152, Mar. 1995.

[14] M. A. Breuer, "Fault detection in a linear cascade of identical machines," in *Proc. 9th Annu. Symp. Switching Automata Theory*, 1968, pp. 235–243.

[15] C.-H. Sung, "Testable sequential cellular arrays," *IEEE Trans. Comput.*, vol. C-25, no. 1, pp. 11–18, Jan. 1976.

[16] C. Y. Su and C. W. Wu, "Testing iterative logic arrays for sequential faults with a constant number of patterns," *IEEE Trans. Comput.*, vol. 43, no. 4, pp. 495–501, Apr. 1994.

[17] H. Fujiwara and S. Toida, "The complexity of fault detection problems for combinatioinal logic circuits," *IEEE Trans. Comput.*, vol. C-31, no. 6, pp. 555–560, Jun. 1982.

[18] C. W. Wu and P. R. Cappello, "Easily testable iterative logic arrays," *IEEE Trans. Comput.*, vol. 39, no. 5, pp. 640–652, May 1990.

[19] X. Ma, J. Huang, C. Metra, and F. Lombardi, "Testing reversible 1D arrays for molecular QCA," in *Proc. IEEE Symp. Defect Fault Tolerance VLSI Syst. (DFT 2006)*, pp. 71–79.

[20] A. Chakraborty, "Synthesis of reversible circuits for testing with universal test set and C-testability of reversible iterative logic arrays," in *Proc. 18th Int. Conf. VLSI Design (VLSID 2005)*, pp. 249–254.

[21] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Proc. Des. Autom. Conf. (DAC 2003)*, Anaheim, CA, pp. 318–323.

[22] K. Patel, J. Hayes, and I. Markov, "Fault testing for reversible circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 23, no. 8, pp. 1220–1230, Aug. 2004.

[23] V. Shende, A. Prasad, I. Markov, and J. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 6, pp. 710–722, Jun. 2003.

[24] P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 11, pp. 2317–2330, Nov. 2006.

[25] K. lwama, Y. Kambayashi, and S. Yamashita, "Transformation rules for designing CNOT-based quantum circuits," in *Proc. Des. Autom. Conf. (DAC 2002)*, New Orleans, LA, pp. 419–424.

[26] L. T. Wang, C. W. Wu, and X. Wen, *Design for Testability: VLSI Test Principles and Architectures*. San Francisco, CA: Elsevier (Morgan Kaufmann), 2006.

[27] I.-M. Tsai and S.-Y. Kuo, "Digital switching in the quantum domain," *IEEE Trans. Nanotechnol.*, vol. 1, no. 3, pp. 154–164, Sep. 2002.

[28] I.-M. Tsai, S.-Y. Kuo, and D. S. L. Wei, "Quantum Boolean circuit approach for searching an unordered database," in *Proc. 2002 IEEE Conf. Nanotechnol.*, pp. 315–318.

[29] Y.-H. Chou, I.-M. Tsai, and S.-Y. Kuo, "Quantum Boolean circuit is 1-testable," in *Proc. 2007 IEEE Conf. Nanotechnol. (IEEE-NANO 2007)*, pp. 1292–1295.

[30] Y.-H. Chou, I.-M. Tsai, and S.-Y. Kuo, "QBIST: Quantum built-in self-test for any Boolean circuit," *Proc. 26th IEEE VLSI Test Symp. (VTS 2008)*, pp. 261–266.

Yao-Hsin Chou (S'06) received the B.S. degree from the Department of Business Mathematics (Department of Financial Engineering and Actuarial Mathematics), Soochow University, Taipei, Taiwan, R.O.C., and the M.S. degree in computer science and information engineering from the National Central University, Taoyuan, Taiwan, in 1998 and 2001, respectively. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

His current research interests include quantum information science (QIS) and electronic design automation (EDA).

I-Ming Tsai received the B.S. degree in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, R.O.C., the M.S. degree in telecommunications from the University of Pittsburgh, Pittsburgh, PA, and the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1988, 1993, and 2003, respectively.

Since 1993, he has been with Chunghwa Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan, where he is engaged in research on circuit design, broadband switching, information security, and network architecture evolution. He is also with the Department of Electrical Engineering, National Taiwan University. His current research interests include experimental and theoretical aspects of quantum computing and quantum communication.

Sy-Yen Kuo (S'85–M'88–SM'98–F'01) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., the M.S. degree in electrical and computer engineering from the University of California, Santa Barbara, and the Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, in 1979, 1982, and 1987, respectively.

He is currently the Chair Professor and the Dean of the College of Electrical and Computer Engineering, National Taiwan University of Science and Technology, Taipei. He is also a Distinguished Professor in the Department of Electrical Engineering, National Taiwan University, where he is currently taking a leave of absence and was the Chairman from 2001 to 2004. He spent his sabbatical years as a Visiting Professor in the Computer Science and Engineering Department, the Chinese University of Hong Kong from 2004 to 2005, and as a Visiting Researcher at AT&T Laboratories Research, NJ, from 1999 to 2000, respectively. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan, from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering, University of Arizona, from 1988 to 1991, and an Engineer at Fairchild Semiconductor and Silvar-Lisco, from 1982 to 1984. During 1989, he was a Summer Faculty Fellow at the Jet Propulsion Laboratory, California Institute of Technology. His current research interests include dependable systems and networks, software reliability engineering, mobile computing, and reliable sensor networks. He is the author or coauthor of more than 270 papers published in journals and conferences, and also holds several patents.

Prof. Kuo received the Distinguished Research Award between 1997 and 2005 consecutively from the National Science Council, Taiwan, where he is currently a Research Fellow. He was also the recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/Association for Computing Machinery (ACM) Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.