Simulation-Based Equivalence Checking for Reversible Circuits

專題學生：陳柏宏
指導教授：江介宏 教授
台灣大學電機系
113 年 1 月 2 日

Abstract :

Reversible circuits realize logic functions that bijectively map every input to a unique output. Due to their advantageous characteristics that prevent information loss, reversible circuits play important roles in low power CMOS, optical computing, quantum computing, and various other fields. They are especially important in quantum computing, as they are the only digital logic functions implementable in quantum circuits. However, as an essential component of circuit design, equivalence checking for reversible circuits is a difficult problem that scales quickly with the number of gates and lines. In this work, we propose an innovative approach to this problem that performs significantly better than the state-of-the-art by abusing the inherent properties of reversible circuits and the emerging tools for quantum circuit simulation. By examining the properties of reversible circuits, we reduce the problem to linear number of simulations with superposition states. Meanwhile, prominent BDD-based quantum circuit simulation techniques can be modified to simulate reversible circuits efficiently. Ultimately, experiment results show that our methodology produces comparable results to both state-of-the-art QMDD and SAT solutions.

Keywords: **Reversible circuit, Equivalence checking, Quantum, Simulations**

1. Introduction

Reversible circuits are a class of digital logic circuit that, in contrast to conventional logic circuits, preserves all information so that none can be lost in the calculation process. In classical reversible computing, the circuit is carefully designed so that each input state maps to a unique output state. In other words, reversible functions are bijective, which makes retrieving the unique input state from the output state possible, thus invertible. This property led to advantages such as less heat generation and low power consumption[5].

While there are multiple elements in the circuit design process, we focus on the equivalence checking for reversible circuits. It has long been known that the equivalence checking for conventional logic circuits is coNP-complete, and equivalence checking for reversible circuits is coNP-complete as well. In previous works, two approaches are mainly discussed: constructing the function matrix with decision diagrams or utilizing SAT-solvers. The first approach to this problem is proposed in [1] and involves describing and evolving the matrix described by a QMDD. The second approach involves formulating the problem into either pure CNF form [1] or XOR-CNF form [2].

In this work, we propose a different take. Instead of considering reversible circuits as elementary quantum circuits or complex conventional circuits, we exploited unique characteristics of reversible circuits for the first time and developed a robust verification method specifically designed for them. Our methodology is derived by three important characteristics of reversible circuits:

- Reversible circuits are inherently invertible. Therefore, if two reversible circuits C_1 and C_2 are functionally equivalent, the concatenation of C_1 and the inverse of C_2, or a *reversible miter*, should represent the identity function. This allows fast detection of non-equivalence by inserting different stimuli, or input state, and check whether the output state is identical to the input state.
- All reversible circuits can be presented in quantum circuit form. The quantum circuit model is an effective tool to represent reversible circuits since it guarantees reversibility in contrast with the classical AND/OR gate model. This allows us to execute the reversible circuit by quantum circuit simulators, even with input states that do not seem possible in classical circuits, e.g. superposition states.
- Reversible functions represent a bijective mapping. Every input maps to a single output, and no two inputs produce the same output. Thus, if a set of inputs go through the reversible function and produced a set of outputs, then the input set and the output set must have the same size. Furthermore, any input that does not fall in the input set corresponds to an output outside the output set as well. This allows us to eliminate output possibilities for untested inputs, leading to potential extrapolation of the corresponding output for the untested input.

These characteristics unearth the potential of a simulation-based equivalence checking scheme. In our proposed method, the reversible miter is first constructed by the two test circuits. Next, we apply specially designed superposition states as input states to the miter and simulate the circuit with advanced quantum

simulation techniques. Finally, we examine the results and determine whether the two test circuits are equivalent. Instead of testing each possible input states individually, we are able to formally verify the equivalence of two reversible circuits with just n simulations (with n being the number of bits). Experimental results show that our method outperforms other previous methods in most, especially complex cases, and our method also provides immediate advantage when parallel computing is considered.

The remainder of this paper is structured as follows: section 2 provides the necessary backgrounds. Section 3 presents the proposed methodology on equivalence checking. Section 4 elaborates on handling constant inputs and garbage outputs. Experimental results are provided in section 5 and the paper is concluded in section 6.

2. Background

In this section, we introduce the necessary backgrounds of reversible circuits and briefly discuss the main concepts of quantum computing. The equivalence checking problem Decision diagrams are then introduced as a means for state representation and manipulation methods.

2.1 Reversible circuits

A logic function is reversible if and only if it maps each input assignment to a unique output assignment, in other words, it represents a bijection. Therefore, its input size must be identical to its output size, and it can be represented as a cascade of reversible gates. The circuit can thus be denoted as $C=g_1*g_2*...*g_m$, with g_1, g_2, ..., g_m being reversible gates. In this work, we focus on Toffoli gates since they are universal and compatible with quantum circuit design, an important application of reversible circuits. Toffoli gates include:

- X gate: inverts a single bit.
- Controlled-NOT gate (CNOT): inverts the target bit if the control bit is 1.
- Multi-controlled Toffoli gate (MCT): inverts the target bit if the control bits are all 1.

These gates are also implementable in quantum circuit, with the operation being done on qubits instead of bits. When considering Toffoli gates as the basic component of reversible circuits, constructing the inverse of them are trivial. All Toffoli gates are their own inverse, i.e. $\forall g \in \{X,CNOT, MCT\}$, $g^{-1}=g$. Thus, for a circuit $C=g_1*g_2*...*g_m$ with $g_1, g_2, ..., g_m \in \{X,CNOT, MCT\}$, its inverse C^{-1} can be found by

$$C^{-1} = g_m \cdot g_{m-1} \cdot \ldots \cdot g_1.$$

Since C^{-1} is constructed by reversible gates, it is still a reversible circuit. This follows the fact that reversible circuits are inherently invertible, and Toffoli circuits simplifies the process. Furthermore, the concatenation of two reversible circuits is still a cascade of reversible gates, thus a reversible circuit.

2.2 Quantum Computing

In quantum computing, the computational unit is the qubit. In contrast to the classical bit that can only be either in the 0 state or the 1 state, qubits can also be in an arbitrary superposition of these states. The state of a qubit can be described as $|\varphi\rangle=\alpha|0\rangle+\beta|1\rangle$, with $\alpha, \beta \in C$ being *amplitudes* such that $\alpha^2+\beta^2=1$ and $|0\rangle$, $|1\rangle$ being the *computational basis states* written in Dirac notation. For a system consisting of n qubits, the state of the system can be described as

$$|\phi\rangle = \sum_{i=0}^{2^n-1} a_i|i\rangle = [a_0 \;\; a_1 \;\ldots\; a_{2^n-1}]^T$$

To manipulate a state of a quantum system in the circuit model, *quantum gates* are required. Quantum gates can be described as unitary matrices, and the reversible gates of interest, the Toffoli gates, are also valid quantum gates. Throughout this paper, we will treat reversible circuits commutatively as quantum circuits.

2.3 Binary Decision Diagram

Binary decision diagram (or BDD) is a graphical representation of a Boolean function, with vertices being decision nodes controlled by variables and leaves being constants 0 and 1. Furthermore, it can also be utilized as an advanced data structure for quantum state representation. By recursively dividing the vector, we can represent a vector with indexes 0 or 1 with a single BDD.

3. Problem Formulation

The main purpose of this work is to determine whether two reversible circuits realize the same functionality. If we can prove that the two circuits have the same functionality under all possible inputs, we can claim that the two circuits are equivalent. Sometimes, we do not need the two circuits to be completely equivalent, and we allow some flexibility on equivalence by introducing constant inputs and garbage outputs. Constant inputs are inputs with designated logic value, and garbage outputs are outputs that are discarded after the circuit. Under these circumstances, we only have to make sure that under all possible inputs on not-constant bits, the outputs excluding garbage outputs are the same.

4. Proposed Method

In this section, we present our simulation based approach which reduces the equivalence checking problem to n simulations of the test circuits, with n being the number of bits under the completely specified case. We also show that by augmenting and modifying our method, we can construct the full functionality of two reversible circuits and determine equivalence under constant input and garbage output constraints.

4.1 Segmented Mapping

As mentioned in section 2, reversible circuits map each input assignment to a unique output assignment. In quantum circuit terms, a reversible circuit maps each computational basis state to another unique computational basis state. Furthermore, since both the inverse of a reversible circuit and a concatenation of two reversible circuits are reversible circuits, $C_1*C_2^{-1}$ is reversible as well, and it maps each computational basis

state to another unique computational basis state.

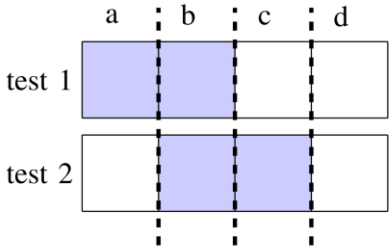

Figure 1. a, b, c, d are subsets and the colored portions represent correct mappings

Hence, if we treat reversible circuits as quantum circuits and set the input state as an equal superposition of computational basis states, the output would also be an equal superposition of computational basis states. Note that we can consider a superposition of computational basis states as a set of such states. If the reversible miter is identity, the output state would be identical to the input state. Thus, for every test that the output state is equal to the input state, we call that a "correct mapping", meaning that the test is successful and the miter may be identity. We can consider a superposition of computational basis states as a set of such states. For multiple correct mappings, we can deduce more information by comparing the two correct mappings. Consider Figure 1, where we are able to deduce that subsets a, b, c, d maps to itself with the bijective property considered.

By choosing the set of computational basis states wisely, we can therefore segment the output mapping space, hence the name segmented mapping. To segment the output space of 2^n, we can choose each input state to cover exactly half of the input space. By doing so, the output space would be divided by two each time, and we can separate the 2^n outputs in just n simulations. In other words, we feed states $|0 + +...\rangle$, $| + 0 + ...\rangle$, ..., $| + +..0\rangle$ into the reversible circuit which was treated as a quantum circuit. If all the tests are correct mappings, we can conclude that the circuit resembles identity. An example can be seen in Figure 2, where we use two tests to confirm that the 2-qubit circuit is identity.

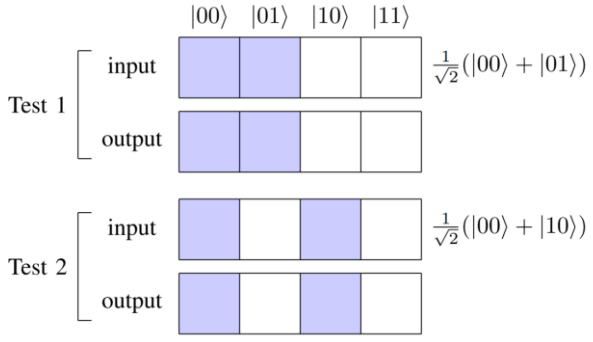

Figure 2. Two tests are used to determine whether the circuit is identity.

4.2 BDD Simulation

To perform segmented mapping, we need an efficient method of simulating quantum circuits. In our work, we choose BDD as the simulation method for its simplicity and suitability. In section 2, we introduced binary decision diagram (or BDD) as a means of representing and manipulating a quantum state. The canonicality of BDD is also an important asset, because it allows us to determine the equivalence of two states with a single pointer check.

For general quantum states, we need 4r BDDs along with a variable to store the constant value k. However, some redundancy can be removed for our case, and our simulation data structure becomes a single BDD. For instance, if we consider an input state in segmented mapping for a 3-qubit system, $|+ 0 +\rangle$, its corresponding vector is 1/sqrt(2)*[1 1 0 0 1 1 0 0]. By neglecting the constant value 1/sqrt(2), we can represent the vector in a single BDD with its corresponding Boolean function being q_2', where q_2 is the second variable of the system.

To manipulate the BDD state with gates, we can refer to the quantum state manipulation methods introduced in [3], which involves cofactoring as well as other Boolean operations on the BDD.

4.3 Output Function Construction

The BDD simulation method also provides an additional advantage. During segmented mapping, we can halt the program and report non-equivalence when a test fails. However, to perform further diagnosis or repair the bugged circuit, we require the full functionality of the miter. Fortunately, we are able to obtain the functionality of the miter easily if we finish the n simulations. Let the reversible miter of interest be M, and the output BDDs of the n input states be a_1, a_2, ..., a_n . Note that each of them is essentially a Boolean function since they are stored as a single BDD, and they can also be comprehended as a set of computational basis states. Assume that we want to know which output state the input state $|b_1\ b_2\ ...\ b_n\rangle$, with $b_i \in \{0,1\}\ \forall i$ maps to. For i=1, 2, ..., n, there are two possible values of b_i :

- If b_i=0, then we know that the input state must belong to the input set with the first qubit as 0, and the corresponding output set would fall within a_i.
- If b_i=1, then the input state would not belong to the input set with the first qubit as 0, and thus the corresponding output state must fall outside a_i, or in other words, inside $\neg a_i$.

Therefore, we can conclude that the output state must be contained by the $a_1^{\neg b1}$, $a_2^{\neg b2}$, ..., $a_n^{\neg bn}$, where $a_i^0 = \neg a_i$ and $a_i^1 = a_i$. The possible choices of the output state would be the intersection of these sets. Furthermore, the concatenation of $a_1^{\neg b1}$, $a_2^{\neg b2}$, ..., $a_n^{\neg bn}$ represents a single computational basis state. Hence, we are able to construct the output function of the reversible circuit by

$$f(b_1, b_2, ..., b_n) = a_1^{\neg b_1} \cdot a_2^{\neg b_2} \cdot ... \cdot a_n^{\neg b_n}$$

which can be represented as a BDD with 2n variables, with n variables being the input variables and the other n variables used to represent the output state. This shows that the outputs of segmented mapping carries the full information of the reversible circuit. Detailed diagnosis of

the miter can be done with the information, and correction procedures can also be applied. Constant inputs and garbage outputs can also be solved by this construction.

4.4 Handling Constant inputs and Garbage outputs

While equivalence of completely-specified reversible circuits can be determined by comparing reversible miters to identity with the segmented mapping method, we can not do that when constant inputs and garbage outputs are considered, because we consider different circuits with the same partial functionality equivalent under these cases. Thus, we need to construct the full functionality of two circuits and determine if the required outputs of all possible inputs are the same for both circuits. For this purpose, we can construct the output sets for individual circuits instead of the miter to obtain their respective functions. The proposed method is modified as follows:

- Constant inputs:

For n-n' constant inputs, we only need 2n' simulations to construct the function. When a high ratio of constant inputs is encountered, we can perform segmented mapping on the actual input qubits instead of performing it on all qubits and reduce them afterwards. That is, if qubits 1 to n' are actual inputs and qubits n'+1 to n are constant inputs with values $c_{n'+1}$, $c_{n'+2}$, ..., c_n, then the input states of segmented mapping becomes $|0 + ... + c_{n'+1} ... c_n\rangle$, $|+ 0 + ... + c_{n'+1} ... c_n\rangle$, ..., $|+ ... + 0\ c_{n'+1} ... c_n\rangle$. We denote a_1, a_2, ..., $a_{n'}$ as the respective output sets generated by testing. However, in order to construct the output function, the negation of output sets are needed, and we cannot simply negate the obtained output set because we are not working on the full input space. Instead, we are working on the subspace restricted by the constant inputs. Thus, more output sets have to be generated. To obtain the negation of the output sets under the subspace, we perform segmented mapping again but with input states $|1 + ... + c_{n'+1} ... c_n\rangle$, $|+ 1 + ... + c_{n'+1} ... c_n\rangle$, ..., $|+ ... + 1\ c_{n'+1} ... c_n\rangle$ instead, as they serve as the counterpart of the subspace restricted by the constants. Let the output sets generated by the latter testing be a_1', a_2', ..., $a_{n'}'$, then the output function becomes

$$f(b_1, b_2, ..., b_{n'}) = A_1^{\neg b_1} \cdot A_2^{\neg b_2} \cdot ... \cdot A_{n'}^{\neg b_{n'}}$$

with $A_i^0=a_i'$ and $A_i^1=a_i$. Overall, 2n' simulations are required to generate an output function for a circuit. If the output function is identical for both circuits, then they are equivalent.

- Garbage Outputs:

After the output function is derived, we need to remove the garbage outputs from the function. Let $G=\{g_1, g_2, ..., g_t\}$ be the set of all garbage outputs. If the output function returns a state containing either g_i or $\neg g_i$, both literals can be removed. This equates to finding the existential quantification over variable g_i, i.e. $h=\exists g_i.f$, which is defined as

$$h = f(x_1, x_2, ..., 0, ..., x_n) \vee f(x_1, x_2, ..., 1, ..., x_n).$$

We can thus remove the garbage output by finding the existential quantification of the output function over all variables in the garbage output set G.

5. Experimental Results

For this experiment, we compared our approach(Seg) to the SAT method and the QMDD method implemented in Qcec. The experimental values are the runtime of the equivalence checking in seconds, and TO represents a timeout of no results after over 10 minutes. Most of the testbenches are from Revlib[4], and others are randomly generated circuits with Toffoli gates labeled "random_<#qubits>".

Circuit	#qubits	#gates	Seg	SAT	Qcec
urf1_149	9	11554	3.8	TO	3.0
urf6_160	15	10740	48.76	TO	32.53
apex5_290	1025	2909	263.56	2.22	TO
cps_292	923	2763	169.45	2.12	TO
random_10	10	50000	39.25	TO	41.27
random_100	100	500	TO	0.73	TO
random_1000	1000	100	21.43	0.36	TO

6. Conclusions

In this work, we proposed a brand new approach to check the equivalence of reversible circuits with the power of linear number of simulations. Furthermore, we dealt with constant inputs and garbage outputs under the scope and prove that segmented mapping harnesses the full output function with the simulations, regardless of whether the two circuits are equivalent or not. The experimental results show that our method produces decent results and demonstrates more versatility compared to the SAT method and the QMDD method with a wide variety of cases. However, it still struggles when the circuit complexity rises under large number of qubits, or lines. Future work can be done by replacing the BDD simulation method, which scales rather poorly. With the potential developments of classical quantum simulation techniques, this method can be further improved in the future as the scalability of classical quantum simulation increases and benefit the reversible circuit design flow.

References

[1] Wille, Robert, et al. "Equivalence checking of reversible circuits." *2009 39th International Symposium on Multiple-Valued Logic*. IEEE, 2009.

[2] Amarú, Luca, et al. "Exploiting inherent characteristics of reversible circuits for faster combinational equivalence checking." *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Ieee, 2016.

[3] Tsai, Yuan-Hung, Jie-Hong R. Jiang, and Chiao-Shan Jhang. "Bit-slicing the Hilbert space: Scaling up accurate quantum circuit simulation." *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021.

[4] Wille, Robert, et al. "RevLib: An online resource for reversible functions and reversible circuits." *38th International Symposium on Multiple Valued Logic (ismvl 2008)*. IEEE, 2008.

[5] Landauer, Rolf. "Irreversibility and heat generation in the computing process." *IBM journal of research and development* 5.3 (1961): 183-191.