Fault Tolerant Quantum Computation with a New Reversible Gate

D. P. Vasudevan, P. K. Lala and J. P. Parkerson

Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville, AR 72701, USA
E-mail: {dvasude,lala}@uark.edu and jpp@engr.uark.edu

ABSTRACT

Fault tolerance plays a major role in quantum computer design. As the quantum environment is not stable enough when the information is read from it, better designs with error correcting capability have to be designed to overcome the information loss due to decoherence in quantum circuits. This paper presents one such fault tolerant design using a new reversible gate.

Keywords: Fault tolerance, Quantum computer model, Reversible gates, Qubits

1 INTRODUCTION

Current CMOS technology will remain dominant only for the next two decades (or less) until it reaches its physical limit. Once the atomic level of system design becomes a mature technology, Quantum mechanics will become successful for computation models to be incorporated in designs. Several computational processes that are exponentially time consuming can be achieved in lesser time with quantum computation. But the unreliable physical implementation of the system is the main barrier for reliable quantum computer design. The quantum information seems to be lost at every step of computation due to physical processes like decoherence, interference etc., but fault tolerant system design seems to be a promising field to aid the design of reliable quantum computers. Quantum error correcting codes are being developed to implement fault tolerant quantum computers. Several elementary quantum gates are being constructed which can form the basic building blocks of fault tolerant quantum computers. This paper deals with the construction one of such elementary quantum gate.

Quantum computers, being in the theoretical level of modeling, have attracted lots of attention in the past few years. Quantum computation is not digital like conventional binary computation. It involves a special state called the superposition state, which makes quantum computation more powerful than conventional bit computation. An n-bit Quantum register, unlike a conventional register (which can store only the data of size n at a time), can store all possible 2^n data at a time. Thus quantum computers can process several data in parallel.

Quantum error correcting codes basically are derived from the conventional error correcting codes. In this paper we used a Self Dual Code to construct the quantum error correcting codes [1]. The k qubit system is mapped to an n qubit system so that t errors occurring in the n qubit system can be tolerated to reconstruct the k qubit system [4].

In this paper, a quantum computer model [2], which allows additional measurement steps during the computation is used to construct quantum error correcting codes discovered by Calderbank and Shor and by Steane [3,4]. Shor [1] showed how errors in quantum information encoded in the quantum error correcting codes can be corrected using slightly noisy gates without introducing more error than is eliminated. He also showed how to perform Boolean linear operations and certain π/2 rotations. He showed how to construct the Toffoli gate (which is a reversible gate) from the ancillary state to make it an elementary gate for quantum computation, which is discussed further in this paper. We have constructed a new reversible gate by following the approach [1] mentioned by Shor and making it an elementary gate for developing Quantum error correcting code. This gate is also constructed using the ancillary state used by Shor in [1]. One more elementary quantum gate has been constructed and it can be used in the encoding operation for developing Quantum error correcting codes.

2 REVERSIBLE LOGIC

Landauer has shown that for every bit of information lost in logic computations that are not reversible, KT*log2 joules of heat energy is generated, where k is Boltzmann’s constant and T the absolute temperature at which computation is performed [5]. Bennett showed that zero power dissipation in logic circuits is possible only if a circuit is composed of reversible logic gates [6]. A gate is reversible, if there is a distinct output assignment for each distinct input. Thus, a reversible gate’s inputs can be uniquely determined from its outputs. A reversible logic gate must have the same number of inputs and outputs [6]. In an n-output reversible gate the output vectors are permutation of the numbers 0 to 2^n-1. Reversible gates are balanced, i.e. the outputs are 1s for exactly half of the inputs. A circuit without constants on its inputs and composed of reversible gates realizes only balanced functions. It can realize non-balanced functions only with
garbage outputs. Some of the major problems with reversible logic synthesis are [7]:

i) Fan-outs are not allowed
ii) Feedback from gate outputs to inputs are not permitted

A logic synthesis technique using reversible gates should have the following features [7]:

i) Use a minimum number of garbage outputs
ii) Use a minimum number of input constants
iii) Keep the length of cascading gates to a minimum
iv) Use a minimum number of gates

3 QUBITS AND QUANTUM GATES

The quantum circuit model in [2] is used as the quantum computation model in this paper. The quantum computation carried out here is in quantum state space of n qubits. A qubit is a quantum system in which the Boolean states 0 and 1 are represented by a prescribed pair of normalized and mutually orthogonal quantum states labeled as \( |0\rangle \) and \( |1\rangle \). The two states form a computational basis and any state of the qubit can be represented as a superposition \( \alpha |0\rangle + \beta |1\rangle \) for some \( |\alpha|^2 + |\beta|^2 = 1 \) [8]. A group of n such qubits is called quantum registers of size n. A quantum register can be assumed to be the same as a binary register. But the bits ‘0’ and ‘1’ are stored in superposition states in these quantum registers, which enables all the possible combinations of a n-bit register are available at the same time. Thus \( |x\rangle \) denotes a tensor product \( |x_1\rangle \otimes |x_0\rangle \otimes \ldots \otimes |x_n\rangle \), where \( x_i \in \{0,1\} \). The number 4 is represented by the register in state \( |1\rangle \otimes |0\rangle \otimes |0\rangle \). Any manipulations of these qubits can be performed only by unitary operations. A quantum logic gate is a device, which can perform unitary operations on specific qubits in a fixed period of time and a quantum network is a device consisting of a group of quantum logic gates whose computational steps are synchronized in time [8]. The most common example of a unitary operation is the Hadamard Transformation. A Hadamard gate is a single qubit gate performing the unitary Hadamard Transformation as shown below.

\[
\begin{pmatrix}
1 & 1 \\
1 & -1
\end{pmatrix}
\]

Some other elementary quantum gates:

- \( I = |0\rangle \langle 0| + |1\rangle \langle 1| \) identity
- \( X = |0\rangle \langle 1| + |1\rangle \langle 0| \) not
- \( X = P \ (P_i) \)
- \( Y = XZ \)

4 CONSTRUCTION OF TOFFOLI GATE

Shor constructed a fault tolerant Toffoli gate given a set of ancillary quantum bits, known to be in encoded state \( |0\rangle \langle 0| |0\rangle \langle 0| |0\rangle \). The Toffoli gate transforms qubits by negating the third qubit if and only if the first two are 1’s. First a gate

\[
\begin{align*}
|s_0s_0\rangle & \rightarrow |s_0s_0s_0\rangle \\
|s_0s_1\rangle & \rightarrow |s_0s_1s_0\rangle \\
|s_1s_0\rangle & \rightarrow |s_1s_0s_0\rangle \\
|s_1s_1\rangle & \rightarrow |s_1s_1s_1\rangle
\end{align*}
\]

that makes following transformation;

by taking two encoded qubits to three encoded qubits was built. This gate acts like a conventional AND gate giving the third qubit output as ‘1’, only when the first two qubits are ‘1’ otherwise ‘0’. To construct this gate, ancilla \( |A\rangle \) was appended to the first two qubits. Then, by XORing the third qubit in to the first and the fourth qubit in to the second produce the transformation:

\[
\begin{align*}
|s_0s_0\rangle |A\rangle & \rightarrow \frac{1}{2} (|s_0s_0s_0s_0\rangle + |s_0s_1s_0s_1\rangle + |s_1s_0s_0s_1\rangle + |s_1s_1s_1s_1\rangle) \\
|s_0s_1\rangle |A\rangle & \rightarrow \frac{1}{2} (|s_0s_0s_0s_1\rangle + |s_0s_1s_0s_1\rangle + |s_1s_0s_0s_1\rangle + |s_1s_1s_1s_1\rangle) \\
|s_1s_0\rangle |A\rangle & \rightarrow \frac{1}{2} (|s_0s_0s_1s_0\rangle + |s_0s_1s_1s_0\rangle + |s_1s_0s_1s_0\rangle + |s_1s_1s_1s_0\rangle) \\
|s_1s_1\rangle |A\rangle & \rightarrow \frac{1}{2} (|s_0s_0s_1s_1\rangle + |s_0s_1s_1s_1\rangle + |s_1s_0s_1s_1\rangle + |s_1s_1s_1s_1\rangle)
\end{align*}
\]

Measuring the first and second qubits, and if it is at the state \( |s_0s_0\rangle \>, pulling out the relevant elements of the superposition, the expression:

\[
\begin{align*}
|s_0s_0\rangle & \rightarrow |s_0s_1s_0\rangle \\
|s_0s_1\rangle & \rightarrow |s_0s_0s_0\rangle \\
|s_1s_0\rangle & \rightarrow |s_1s_1s_1\rangle \\
|s_1s_1\rangle & \rightarrow |s_1s_0s_0\rangle
\end{align*}
\]

was obtained. This transformation was converted to the required one by applying a controlled NOT from the first to the third qubit and then applying a NOT to the second qubit. The resulting transformation is shown below:
\[
|s_0s_0\rangle \rightarrow |s_0s_1s_0\rangle \rightarrow |s_0s_1s_0\rangle \\
|s_0s_1\rangle \rightarrow |s_0s_0s_0\rangle \rightarrow |s_0s_0s_0\rangle \\
|s_1s_0\rangle \rightarrow |s_1s_1s_1\rangle \rightarrow |s_1s_1s_1\rangle \\
|s_1s_1\rangle \rightarrow |s_1s_0s_1\rangle \rightarrow |s_1s_0s_1\rangle \\
|s_0s_1\rangle \rightarrow (|s_0\rangle + |s_1\rangle)/2, |s_1\rangle \rightarrow (|s_0\rangle - |s_1\rangle)/2
\]

Finally by applying the transformation:

This gives the transformation:

\[
|s_0s_0s_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_0s_0s_0\rangle (|s_0\rangle + |s_1\rangle)) \\
|s_0s_1s_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_0s_1s_0\rangle (|s_0\rangle + |s_1\rangle)) \\
|s_1s_0s_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_1s_0s_0\rangle (|s_0\rangle + |s_1\rangle)) \\
|s_1s_1s_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_1s_1s_0\rangle (|s_0\rangle + |s_1\rangle)) \\
|s_0s_0s_1\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_0s_0s_1\rangle (|s_0\rangle - |s_1\rangle)) \\
|s_0s_1s_1\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_0s_1s_1\rangle (|s_0\rangle - |s_1\rangle)) \\
|s_1s_0s_1\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_1s_0s_1\rangle (|s_0\rangle - |s_1\rangle)) \\
|s_1s_1s_1\rangle \rightarrow \frac{1}{\sqrt{2}}(|s_1s_1s_1\rangle (|s_0\rangle - |s_1\rangle))
\]

If we observe the final state obtained with $|s_1\rangle$, there is a phase change in the resulting state. To set this state, the following transformation is applied:

\[
|s_a s_b s_c\rangle \rightarrow (-1)^a b (-1)^c |s_a s_b s_c\rangle
\]

NOT from the third qubit to the newly introduced qubit was then applied. Finally by applying $|s_0\rangle = (|s_0\rangle + |s_1\rangle)/2$, $|s_1\rangle = (|s_0\rangle - |s_1\rangle)/2$ to the original third qubit, the resulting transformation gives the Toffoli gate as shown in Figure 5.

The construction of ancillary state is briefed by Shor in [1].

5 CONSTRUCTION OF NEW REVERSIBLE GATE

The circuit for the new reversible gate is shown below (Fig 1). It's a three input and three output reversible gate, as is the Toffoli gate. It gives an EX-OR of the first two inputs as the first output. The first input is passed as the second output. Third output of the gate is the third input only when the other two inputs are `1'.

![Figure 1. New Reversible Gate](image)

To get the complete Toffoli gate, it was started with three qubits for which Toffoli needed to be applied and the following transformations to first two qubits. A controlled NOT from the third qubit to the newly introduced qubit was then applied. Finally by applying $|s_0\rangle = (|s_0\rangle + |s_1\rangle)/2$, $|s_1\rangle = (|s_0\rangle - |s_1\rangle)/2$ to the original third qubit, the resulting transformation gives the Toffoli gate as shown in Figure 5.

The construction of ancillary state is briefed by Shor in [1].

5 CONSTRUCTION OF NEW REVERSIBLE GATE

The circuit for the new reversible gate is shown below (Fig 1). It's a three input and three output reversible gate, as is the Toffoli gate. It gives an EX-OR of the first two inputs as the first output. The first input is passed as the second output. Third output of the gate is the third input only when the other two inputs are `1'.

![Figure 2. Shor's Fault Tolerant Toffoli Gate](image)

In the construction of this new reversible gate, the steps followed in the construction of Toffoli are followed until the Toffoli output pattern is obtained. Then a NOT gate is applied to the third qubit. This is followed by a CNOT from first qubit to the second qubit. Then a CNOT from second qubit to first qubit is applied. Again a CNOT is applied from the first qubit to the second qubit and then the second
qubit to the first qubit. This repetitive process of applying CNOT from the first to the second and the second to the first qubit ensures a binary swap of the first qubit and second qubit. Thus produces the required first and second outputs x ⊕ y and x. The NOT gate applied to the third qubit produces the third output z' ⊕ xy. Thus the new reversible gate is a binary transformation of the fault tolerant Toffoli gate designed by Shor [1] as shown in Fig. 3. The transformations are shown below.

| s₀s₀s₀ | → | s₀s₀s₀ | → | s₀s₀s₀ |
| s₀s₁s₀ | → | s₀s₁s₀ | → | s₀s₁s₀ |
| s₁s₀s₀ | → | s₁s₀s₀ | → | s₁s₀s₀ |
| s₁s₁s₀ | → | s₁s₁s₀ | → | s₁s₁s₁ |
| s₀s₀s₁ | → | s₀s₀s₀ | → | s₀s₀s₁ |
| s₀s₁s₁ | → | s₀s₁s₀ | → | s₀s₁s₁ |
| s₁s₀s₁ | → | s₁s₀s₀ | → | s₁s₀s₁ |
| s₁s₁s₁ | → | s₁s₁s₀ | → | s₁s₁s₁ |


\[ \rightarrow s_0 s_0 s_1 \rightarrow s_0 s_0 s_1 \rightarrow s_0 s_0 s_0 \]
\[ \rightarrow s_1 s_0 s_0 \rightarrow s_1 s_0 s_0 \rightarrow s_1 s_0 s_0 \]
\[ \rightarrow s_0 s_0 s_1 \rightarrow s_0 s_0 s_1 \rightarrow s_0 s_1 s_0 \]
\[ \rightarrow s_1 s_0 s_0 \rightarrow s_1 s_0 s_1 \rightarrow s_1 s_0 s_1 \]

6 CONCLUSION

A new fault tolerant reversible gate has been presented. This work focused on realizing this new gate by binary transformations of the fault tolerant Toffoli gate. Further work will be focused on realizing the gate from the Ancilla. As stated by Shor [1] the analysis in the paper is also purely asymptotic. This work further can be extended to find the amount of fault tolerance offered, the specific number of gates required, and the quantum cost behind this realization.

REFERENCES