S

Reversible Logic as a Stepping Stone to Quantum Computing

J. E. Rice
University of Lethbridge, Canada

INTRODUCTION

In 1965 Intel co-founder Gordon Moore predicted that the number of devices per chip would double every year (Moore, 1965). His prediction has held true for nearly 50 years as consumers and industry have demanded smaller and lower power computing devices. However the Semiconductor Industries Association predicts that this will soon end: "It is forecasted that by the end of the next decade it will be necessary to augment the capabilities of the CMOS process by introducing multiple new devices that will hopefully realize some properties beyond the ones of CMOS devices". This same report indicates clearly that future architectures must allow for new technologies such as quantum and reversible computing (ITRS, 2009).

Quantum computing is an area of research with great potential; however significant change would be required for industry to move to a quantum paradigm. Reversible computing may provide a stepping-stone from current technologies and approaches to that of quantum computing. Reversible circuits can be viewed as a special case of quantum circuits (Nielsen & Chuang, 2000); thus research into reversible logic can offer a path to fully quantum computing in a way that maintains some of the familiar processes and techniques as well as the effectiveness inherent in the tried and true methods. Further motivation is provided in articles such as (Frank, 2005).

This article introduces and discusses reversible logic with focus on synthesis of reversible logic circuits. We aim to provide background and an overview of the current state-of-the-art in these areas. For those unfamiliar with this field Pan and Nalasani (2005) offer an introductory work on reversible logic in general.

BACKGROUND

In 1961 Landauer defined a device as being logically irreversible if the output of the device did not uniquely define the inputs. He argued that logical irreversibility implied physical irreversibility, which unavoidably would lead to heat dissipation. However Landauer thought that the extra computation and/or memory required to achieve reversibility would offset the inherent advantages (Landauer, 1961). Bennett refuted this in 1973 and demonstrated that it was possible to design a reversible equivalent that was not significantly more complex than an equivalent irreversible device. Bennett also demonstrated that for power not to be dissipated it is *necessary* that a binary circuit be constructed from reversible gates (Bennett, 1973).

Following in Landauer and Bennett's footsteps, Toffoli wrote that "...it appears possible to design circuits whose internal power dissipation, under ideal physical circumstances, is zero". Toffoli also demonstrated that it is *always* possible to construct an arbitrary function by means of an invertible function (Toffoli, 1980). The current literature in reversible logic can almost entirely be traced back to these papers.

Basic Terminology

In this article all operators are Boolean operators and standard symbols are used. While it is common practice to refer to "inputs" and "outputs" of a reversible function, this is not technically correct. A reversible function may be implemented using some quantum technology where wires and traditional technologies cannot be used and where inputs/outputs are really starting/ending states of some technological entity implementing the function. A more accurate term might be "qubits", which is used by some authors; however many refer instead to lines, variables, or inputs and outputs of the

DOI: 10.4018/978-1-4666-5888-2.ch715

Figure 1. (A) The truth table for a reversible function; (B) The truth table for the traditional Boolean AND operator, which is irreversible; (C) The truth table for a half-adder; and (D) The truth table for a reversible embedding of a half-adder

(A)

x_2x_1	f_2f_1
00	00
01	11
10	10
11	01

(B)

x_2x_1	$x_2 \bullet x_1$
00	0
01	0
10	0
11	1

(C)

x_2x_1	cs
00	00
01	01
10	01
11	10

(D)

gx_2x_1	$g'cs$	gx_2x_1	$g'cs$
000	000	100	100
001	001	101	011
010	101	110	111
011	010	111	110

circuit, assuming that these will be implemented by some reversible technology equivalent.

Reversible Functions

A function is reversible if there is a one-to-one and onto (bijective) mapping from the inputs to the outputs of the function. For example, the function shown in Figure 1(A) is reversible, as it is bijective, while the function shown in Figure 1(B) is not. Any irreversible function can be embedded within a reversible function, as shown by Bennett (1973). Figure 1(C) gives the truth table for a half-adder, with two bits (x_2 and x_1) that are added together to produce a carry and sum output. The half-adder is irreversible, since there are two output rows that are not unique. By including an additional input (*g*) and output (*g'*) we obtain a reversible function, shown in Figure 1(D), which embeds the half-adder functionality in the first four rows of the truth table. In this example the values for the garbage line have been arbitrarily assigned with the only goal being to ensure uniqueness in each output row. Determining the most optimal assignment is an area of research addressed by works such as (Miller et al., 2009).

Reversible Gates

Definition 1.1 *A gate is reversible if the (Boolean) function it computes is bijective (Shende et al. 2003).*

The majority of traditional logic gates such as the AND, NOT, OR and exclusive-OR (XOR) gates are not reversible, as they are not bijective. The exception to this is the traditional inverter, or NOT gate. In fact, any (potential) physical implementation of a bijective function could be considered to be a reversible gate. However the most commonly used reversible gates include the NOT gate, SWAP gate, and variations on these, as listed in Table 1. In this table the lines, or inputs to the gates are labeled as *x*, *y*, and if needed, *z*. It is common to extend these behaviours to any number of *control* lines. A control line is an input, or line, which controls the behaviour of the gate. Lines whose values may change are referred to as *target* lines.

Toffoli Gates

Because it is a universal gate and easy to extend, many researchers have focused on the Toffoli gate and its variations in logic synthesis approaches. A Toffoli gate may be referred to as a multiple control Toffoli gate, or TOF*n*, where *n* is the total number of signal lines the gate is operating on. A TOF*n* gate may also

Table 1. The behaviour of a selection of reversible gates

gate	behaviour
NOT	$(x) \rightarrow (x \oplus 1)$ (see Figure 2A)
Feynman (Tof2)	$(x, y) \rightarrow (x, x \oplus y)$ (see Figure 2B)
Toffoli	$(x, y, z) \rightarrow (x, y, xy \oplus z)$ (see Figure 2C)
SWAP	$(x, y) \rightarrow (y, x)$
Fredkin	$(x,y,z) \rightarrow (x,z,y)$ iff $x = 1$