Technical Report: Projects & Background in Reversible Logic

TR-CSJR1-2005
J. E. Rice
University of Lethbridge
j.rice@uleth.ca

1 Introduction - What is Reversible Logic?

In order to discuss new trends and projects in the area of reversible logic one must first have an understanding of what this is. First of all, we'll restrict our discussion of logic functions to two-valued functions describing switching logic. Reversible multiple-valued functions are also possible, but are beyond the scope of this report. We will be discussing functions that describe Boolean logic circuits. These functions are generally built using logic *gates*. According to Shende *et. al.* [16],

Definition 1.1 *a gate is reversible if the (Boolean) function it computes is bijective.*

Bijective means one-to-one and onto; or, for those of us who forget our mathematics terminology, there must be the same number of inputs as outputs, and for each output value there is a unique input value that leads to it.

Table 1 shows the truth table for a 3x3 reversible function. Note that a reduced representation can be obtained from the truth table by simply listing the row numbers correspoinding to the binary expansions represented by the inputs and by the outputs. If we assume that the inputs are given in numerical order from 0 to 2^{n-1} then we can list simply the decimal numbering of the outputs, in this case $0, 1, 3, 2, 4, 5, 7, 6$.

Since, by our previous definition, we need reversible gates in order to build reversible logic we must next define some reversible gates and how they behave. Again according to Shende *et. al.* [16],

Definition 1.2 *A k-CNOT is a $(k+1) \times (k+1)$ gate. It leaves the first k inputs unchanged, and inverts the last* iff *all others are 1.*

There are many types of k-CNOT gates, and they are referred to in the literature in a variety of ways. Here is a few of the names given to the varieties:

- A 0-CNOT gate is just an inverter, referred to as a NOT gate.

	xyz	x'y'z'	
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	100	4
5	101	101	5
6	110	111	7
7	111	110	6

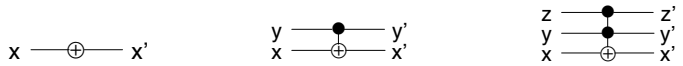
Table 1: The truth table of a 3x3 reversible function.

gate	behaviour
Not	$(x) \rightarrow (x \oplus 1)$
Feynman	$(y, x) \rightarrow (y, x \oplus y)$
Toffoli	$(z, y, z) \rightarrow (z, y, x \oplus yz)$
swap	$(x, y) \rightarrow (y, x)$
Fredkin	$(z, y, z) \rightarrow (z, x, y)$ iff $z = 1$

Table 2: The behaviour of a selection of more commonly used reversible logic gates.

- A 1-CNOT gate is called a controlled-NOT, or CNOT (this is also known as a Feynman gate).
- A 2-CNOT gate is called a TOFFOLI gate.

Other gates include the SWAP gate and the FREDKIN gate. Table 2 lists the behaviour of each of the most commonly used reversible gates. Figure 1 illustrates the symbols usually used for each of the gates. Each of the Toffoli,



(a) the NOT gate (b) the Feynman gate (c) the Toffoli gate

(d) the swap gate (e) the Fredkin gate

Figure 1: Symbols for some of the more commonly used reversible logic gates.

Fredkin and CNOT gate are universal gates [10]; that is, they each can be used to create any logic circuit without the addition of any other type of gate. In traditional logic the AND gate is a universal gate.

There are many different notations being used to denote these and other gates. The symbols in Figure 1 are from [4]. Dueck *et. al.* [4] also use the following notation:

- TOF(C; T) denotes a Toffoli gate, where C is a set of 0 or more control inputs and T is the input to be inverted,
- FRE(C; T) denotes a Fredkin gate, where C is a set of 0 or more control inputs and T is the inputs to be swapped, and the NOT can then be written TOF(; T) and a SWAP can then be written FRE(; T).

2 Motivation - Why Reversible Logic?

In addition to understanding the background of a topic it is equally important to understand why a particular topic is of interest. According to Frank [6]

> ...computers based mainly on reversible logic operations can reuse a fraction of the signal energy that theoretically can approach arbitrarily near to 100% as the quality of the hardware is improved...

Many researchers believe that Moore's law is at an end. We can't keep increasing performance as we have previously done, in order to meet consumer demands, because we simply can't keep up with the power requirements. For an extremely convincing explanation of why this is so the reader is directed to section 1.1 of [6]. If this doesn't convince you then possibly Bennet's statement that "loss of information implies energy loss" [2] and Perkowski *et. al.'s* convictions that "every future technology will have to use reversible gates in order to reduce power" and "[our reversible techniques are] useful for arbitrary reversible technology, *e.g.* quantum, CMOS, DNA, optical, *etc.*" [14] may convince us of the usefulness of pursuing research in the area of reversible logic.

3 Issues

This is a relatively new area of work. One prominent researcher has identified four big problems in the area. Frank [6] states that

1. we need to develop fast and cheap switching devices with adiabatic energy coefficients well below those of transistors;
2. we also need clocking systems that are themselves of very high reversible quality;
3. it is also essential that we pursue research into the design of highly-optimized reversible logic circuits and algorithms.
4. Finally, the area faces an uphill social battle in overcoming the enormous inertia of the established semiconductor industry.

I have chosen three projects, as described below, that fit into these problem areas.

4 Projects

Much of my prior work has been in the are of traditional logic synthesis. Thus it is logical for me to extend this knowledge to problem 3 in Frank's list; the problem of developing highly-optimized reversible logic circuits and algorithms. This is, of course, logic synthesis.

Definition 4.1 *Reversible logic synthesis: given a truth table or other specification of a reversible Boolean logic function, how do we generate (what is usually) a cascade of reversible gates to implement the function(s)?*

4.1 Project 1 - Sequential Logic Synthesis for Reversible Logic Circuits

Nearly all the literature on reversible logic synthesis has addressed only the aspect of combinational logic. Researchers that have addressed the problem of sequential logic synthesis for reversible logic are as follows:

- M. Frank, in "Approaching the Physical Limits of Computing" [5],
- Kwon *et. al.* in "A three-port nRERL register file for ultra-low-energy applications" [9],
- Picton, in "Multi-Valued Sequential Logic Design Using Fredkin Gates" [15], and
- Thapliyal *et. al.* in "A Beginning in the Reversible Logic Synthesis of Sequential Circuits" [17].

Part of the problem is that the traditional description of reversible logic gates seems to contradict the requirements for building sequential circuits. However, in order to make reversible logic a feasible tool there must be some type of revrsible memory gate or object, and tools for sequential logic design. The first mention of sequential design for reversible logic was by Picton. He describes the use of Fredkin gates to build clocked D-type latches, which can then be combined to form more complex memory elements required in sequential logic design. Since then Frank (in various presentations, including [5]) and Thapliyal *et. al.* have considered this issue. I include Kwon *et. al.* in the above list since in order to design a register, by nature sequential logic must be used. However synthesis techniques are not mentioned at all in this work; indeed it appears that the design is a one-off hand-tooled design, not suitable for general logic synthesis or what we today know as CAD.

I propose to investigate this problem with the goal of developing a logic synthesis technique, possibly combined with techniques discussed in the following section, that is tailored towrards reversible logic solutions and will take into account the requirements of sequential logic.

4.2 Project 2 - Decision Diagram-based Synthesis for Reversible Logic Circuits

Existing research into reversible logic synthesis seems to fall into two main categories:

1. transform-based techniques, such as those introduced by Dueck *et. al.* [4, 13, 12], and

2. decomposition techniques, such as those introduced by Perkowksi *et. al.* [14], De Vos *et. al.* [18] and Miller [11].

Other researchers in the area include Kerntopf [8, 7], Agrawal *et. al.* [1], and Shende *et. al.* [16]. These latter works rely more on heuristics for deciding how to build reversible circuits.

Any of us familiar with traditional logic synthesis will likely be aware of the impact Bryant's [3] binary decision diagrams have had on the area. In my reading I have found it interesting to note that there are little to no details on how thse useful tools may be extended to reversible logic synthesis. Perkowski does certainly mention the use of PKDDs [14] but gives no information on how the dd structure is then converted into a reversible gate layout. I intend to provide these details, and also make use of the various decomposition and transformation-based techniques in the design of heuristics combining the best aspects of each of these methods.

4.3 Project 3 - Transforming a Traditional Gate-level Layout to a Reversible Logic Layout

This section cannot begin with a list of existing work because in this area, there is none. The motivation of this particular topic is to provide a bridge from existing traditional logic designs to the novel reversible logic designs. Let's face it; academia and industry has put a lot of time and effort into the existing designs we have. If we want to transition to reversible logic we must plan for it.

I intend to begin with traditional gate layouts such as (N)AND-(N)OR implementations and devise algorithms and/or heuristics for generating reversible equivalents. This, of course, is the simplest of traditional logic designs; muxes, registers, and other more complex devices must be incorporated as the work progresses. Clearly the work on sequential reversible logic synthesis from above will provide help in this area as well.

5 Conclusion

This report is intended to fulfill two purposes; firstly to provide the reader with an overview of the area of reversible logic and a list of references from the area, and secondly to give a very brief overview of the projects I intend to pursue in the area.

References

[1] A. Agrawal and N. K. Jha. Synthesis of Reversible Logic. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2004.

[2] Bennet. Logic reversibility of computation. *IBM J. of Research & Development*, 1973.

[3] R. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. on Comp.*, C-35(8):677–691, Aug. 1986.

[4] G. W. Dueck, D. Maslov, and D. M. Miller. Transformation-based synthesis of networks of toffoli/fredkin gates. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, 2003.

[5] M. P. Frank. Approaching the Physical Limits of Computing. In *Proceedings of the International Symposium on Multiple-Valued Logic (ISMVL)*, 2005.

[6] M. P. Frank. Introduction to reversible computing: motivation, progress, and challenges. In *Proceedings of the 2nd Conference on Computing Frontiers*, pages 385–390, 2005.

[7] P. Kerntopf. Synthesis of Multipurpose Reversible Logic Gates. In *Proceedings of the Euromicro Symposium on Digital System Design (DSD)*, 2002.

[8] P. Kerntopf. A New Heuristic Algorithm for Reversible Logic Synthesis. In *Proceedings of the Design Automation Conference (DAC)*, pages 834–837, 2004.

[9] J.-H. Kwon, J. Lim, and S.-I. Chae. A three-port nRERL register file for ultra-low-energy applications. In *Proceedings of the 2000 International Symposium on Low Power Electronics and Design (ISLPED '00)*, pages 161–166, 2000.

[10] Jon Marshall. http://www.themilkyway.com/quantum/finalreport/quantumgates, 2000.

[11] D. M. Miller. Spectral and Two-Place Decomposition Techniques in Reversible Logic. In *Proceedings of the IEEE Midwest Symposium on Circuits and Systems (MWCAS)*, pages II493–II496, 2002.

[12] D. M. Miller, G. W. Dueck, and D. Maslov. A Synthesis Method for MVL Reversible Logic. In *Proceedings of the 34th International Symposium on Multiple-Valued Logic (ISMVL)*, 2004.

[13] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In *Proceedings of the 40th Design Automation Conference*, 2003.

[14] M. Perkowski, L. Joziwak, A. Mixhchenko, A. Al-rabadi, A. Coppola, A. Buller, X. Song, M. Khan, S. Yanushkevich, V. P. Shmerko, and M. Chrzanowska-Jeske. A general decomposition for reversible logic. In *Proceedings of the International Workshop on Methods and Representations (RM)*, 2001.

[15] P. Picton. Multi-Valued Sequential Logic Design using Fredkin Gates. *Multiple-Valued Logic*, pages 241–251, 1996.

[16] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes. Reversible logic circuit synthesis. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 353–360, 2002.

[17] H. Thapliyal and M. B. Srinivas. A Beginning in the Reversible Logic Synthesis of Sequential Circuits. In *Proceedings of Military and Aerospace Programmable Logic Devices (MAPLD) International Conference*, 2005. online Sept. 2005, http://klabs.org/mapld05/abstracts/1012_thapliyal_2_a.html.

[18] A. De Vos and Y. Van Rentergem. Reversible computing: from mathematical group theory to electronical circuit experiment. In *Proceedings of the 2nd conference on Computing frontiers*, 2005.