Design of Reversible Number Generator using Finite State Automation Realization

Vikash Kumar*, Prashant Gupta, Shashank Kumar Ranu, Manish Kumar Pandey, Aminul Islam

Department of Electronics and Communication Engineering, Birla Institute of Technology (Deemed University), Mesra, Ranchi – 835215, Jharkhand, India; vikashkr@bitmesra.ac.in, prashant1485.11@bitmesra.ac.in,shashank1411.11@bitmesra.ac.in, manish1468.11@bitmesra.ac.in, aminulislam@bitmesra.ac.in

Abstract:

Objective: The objective of this paper is to realize an n-bit reversible number generator. Several designs of number generator and their corresponding constraints with feedback controller and initial state have also been discussed. **Method/Analysis:** The proposed design is validated by simulating the results in Verilog HDL. **Findings:** In this paper a design is presented to implement an n bit number generator as a finite state automation, using reversible logic for the implementation of the transition network. A realizable circuit is designed to implement the n-bit number generator using feedback controllers. **Novelty/Improvement:** With the use of reversible logic, the designed circuit reaches all the possible states and consumes no power.

Keywords: Finite State Automation, Number Generator, Reversible Computing, Reversible Logic Synthesis

1. Introduction

Due to recent advances in semiconductor technologies, there is a demand for circuits requiring very low power consumption and reduce the information loss caused due to energy dissipated in irreversible logic gates. Reversible logic has attained wide consideration in the last few years considering their capability to minimize the power consumption. Reversible circuits preserve information, thereby taking care of the energy dissipated due to information loss and are reversible in nature i.e., they can potentially run backwards. A reversible gate contains equivalent number of elements in the input and output vectors and has a bijective mapping between the vectors.

The concept and applications of reversible computing is being considered by a reasonable segment of researcher community. Reversible computing is being utilized in forthcoming technology advancements like low-power CMOS design¹, optical computing², quantum computing³ and nanotechnology⁴. Several papers on reversible logic systems have been published so far^{5–12}. Out of which, papers^{5–9} suggests designs for reversible memory elements essential for sequential systems with enhancement regarding number of gates, garbage outputs and delay. Other paper^{10–12} presents algorithms for synthesis of reversible logic circuits and sequential systems. The proposals and results presented in the paper¹³ indicate that systems designed using reversible logic have lower power consumption.

In this paper, a generic synthesis technique for realizing an n-bit number generator using state diagram description is presented. The work focuses on realizing such systems with the aid of traditional simplification techniques to attain the state equations, such that the prerequisites for reversible computation are satisfied. Moreover a practical number generator is presented so that the reversible circuit can be employed in computer architecture design.

2. Background

2.1 Definitions

Definition 1: An injective function is a function $f : A \rightarrow B$ such that every element of set B is mapped to by at most one element of the set A, thereby f(x) = y. (The function is both injective and surjective.)

Definition 2: A surjective function is a function $f: A \rightarrow B$ such that every element of set B is mapped to by at least one element of the set A, thereby f(x) = y. (The function is both injective and surjective.)

Definition 3: A bijective function is a function $f: A \rightarrow B$ such that every element of set B is mapped to by exactly one element of the set A, thereby f(x) = y. (The function is both injective and surjective.)

Definition 4: A gate is reversible provided it implements a bijective mapping of all its inputs to outputs. A reversible function $f_1A(a_1, a_2, a_3, ..., a_m) \rightarrow B(b_1, b_2, b_3, ..., b_m)$ fulfills the requirements of one-to-one and onto mapping amidst the input and output vectors.

2.2 Fan-Out

Fan-out is used to replicate output signals from a circuit to drive multiple successive circuits such that:

$$\begin{array}{l} a \to p \\ a \to q \end{array} \tag{1}$$

Where a is the input and p, q are the outputs.

It is evident from the truth table provided in Figure 1 that fan-out is not surjective¹⁴, thus cannot be used to copy the output signal in reversible computation.

2.3 Fenyman Gate

The Feynman Gate (FG)¹⁵ is a 2-input 2-output reversible gate which follows the mapping:

$$\begin{array}{l} a \to p \\ b \to q = a \oplus b \end{array} \tag{2}$$



Figure 1. (a) Fan-Out Function (b) Truth Table of the Function.

Where a, b are the inputs and p, q are the outputs. Figure 2 (a) and Figure 2 (b) present the symbolic notation of Feynman gate and its truth table. The Feynman gate can be employed for replicating the input signal there by averting the fan-out issue in reversible logic displayed in Figure 2 (c).

2.4 Reversible Master-Slave D Flip-Flop

Master-Slave D flip-flop using reversible logic^{16.17} is presented in Figure 3. The data is inputted into the flipflop onto the rising edge of the Clk signal, but the output does not follow the input state until the falling edge of the Clk signal. The advantage of using this strategy aids in reducing the inversion process of the Clk signal, thus eliminating a NOT gate. Table 1 summarizes the operation of the master-slave flip-flop.

2.5 Fredkin Gate

The Fredkin gate¹⁸ is a 3-input 3-output reversible gate which follows the mapping:

$$a \rightarrow p = a$$

$$b \rightarrow q = \overline{ab} + a.c$$

$$c \rightarrow r = a.b + \overline{a.c}$$
(3)



Figure 2. (a) Feynman Gate (b) Truth Table of the Function (c) Feynman Gate used to Replicate Signal thus Avoiding Fan-Out.



Figure 3. Reversible Master-Slave D Flip-Flop (FG represents Feynman Gate).

Where a, b, c are the inputs and p, q, r are the outputs. Figure 4(a) and Figure 4(b) show the symbolic notation of Feynman gate and its truth table respectively.

3. Synthesis

3.1 Synthesis of Transition Network

The proposed synthesis procedure is outlined in Figure 5. For reversible sequential circuit, it is necessary that the combinational as well as the sequential elements must also be reversible¹⁹. The state diagram indicates the transitions of the finite state machine. The state table consists of the present and the next state of the machine. The present state shows the state of the machine at any time t. The next state exhibits the state of the machine when the machine makes a transition from the current state. The designer must ensure that the machine has the following characteristics:

Table 1. Truth Table of Master-Slave D Flip-Flop

Clock (Clk)	D	Q(t)
Pos-edge ↑	0	0
Pos-edge ↑	1	1
Neg-edge↓	Х	Q(t-1)



Figure 4. (a) Fredkin Gate (b) Truth Table of the Function.



Figure 5. Synthesis Procedure.

- Each state must transit to only one state. (It ensures the injective nature of the transition network).
- The machine must transit through 2n states where n are the number of bits in the number generator sequence. (It ensures subjectivity of the transition network).

3.2 Feedback Controller

The feedback controller as described in Section II can be designed using two techniques - irreversible element and reversible element. In our work we have designed the controller using reversible delay element in the form of reversible master-slave D flip-flop, details of which are mentioned in Section II.

Though the option of level-triggered D latch was available, the latch will cause the output to oscillate with period equal to the delay of the transition network when the clk signal is HIGH. To use D latch as feedback controller element, constraints on either propagation delay (Δ t) or clk signal must be imposed such that:

$$t_p < \Delta t < T \tag{4}$$

Where t_p is the period during which clk signal remains HIGH (or LOW depending on the type of level triggering) and T is the clock period.

3.3 Seed Input

The Seed or the initial state must be provided to the machine after which the machine automatically runs. To input the Seed, we employed Fredkin gate⁴ as the reversible

multiplexing element with two garbage outputs. The gate provides the scope of using negative logic (output line $3(S_0.I_0 + \overline{S_0}.I_0)$ must be used in such a case) and providing same selection line to multiple devices.

Figure 6 illustrates the structure of the reversible 3-bit number generator with master-slave D flip-flops as delay elements.

3.4 Cascade Structure

The number generators can be placed in cascade to achieve variable transitions in the output signal of the system, in the form of N stage number generator system. Due to the prerequisites of reversible computation, all the generators in the cascade structure must be of equal bit length to provide the next stage with Seed of same bit length. The cntrl signal is essential for determining the Seed value of an individual generator. The output signal is coupled from one generator to another generator using a global cntrl signal or using individual cntrl signal for controlling Seed input, for each generator. Figure 7 illustrates the design of cascade structure.

4. Simulation and Verification

The designs for number generator with feedback control element along with 2 stage cascade structure are validated



Figure 6. 3-Bit Number Generator with Master-Slave D Flip-Flops as Delay Elements and Fredkin Gate based 2:1 Multiplexer.



Figure 7. Cascade Structure.



Figure 8. Simulation of 3-Bit Number Generator.

by coding and simulating the structures in Verilog HDL. The test benches for the number generators were created to verify their working. Model Sim Simulator is used to functionally verify the Verilog HDL codes. The waveforms presented in this paper are achieved using Model Sim Verilog simulator. Figure 8 shows the result of the number generator. When cntrl = 1, the Seed is latched to the generator through the 2:1 multiplexing element and the feedback lines through the master-slave D flip-flops are cutoff. When cntrl = 0, the Seed lines are cutoff from the generator circuit and the feedback is applied to the circuit. The Fredkin based multiplexer provides the option of using negative logic for cntrl signal as mentioned in Section III. At negative edge of clk, the generator transits to the next stage. For the rest of the clk period, the generator maintains its current state. In our Verilog simulations, we have used (0, 0, 0) as the Seed value. Arrows A (0:7) show the eight stages of the 3-bit number generator based on the finite state machine diagram shown in Figure 5. When cntrl = 1, the input Seed value causes the transition to the next stage at negative edge of clk. Since no signal is applied to the generator (initially cntrl = 0), output from the generator will be undefined (as evident from Figure 8).

5. Conclusion

In this paper, a technique is presented for realizing an n-bit number generator using reversible logic. In particular, a finite automation is transformed into a Boolean function representing the sequential behavior. Afterwards, the transition function is realized using combinational synthesis methods. We have discussed various designs of number generators and corresponding constraints with feedback controller and Initial State (Seed) Latching Mechanism. Sequential computing structures designed using reversible logic, allow the system to reach a feasible states, thereby reducing the overall power consumption. The presented work will led the basis of effective system testing which can be employed in nano-computing systems.

6. References

- 1. Schrom, G. Ultra-Low Power CMOS Technology. PhD thesis. Technische Universitaet Wien; 1998.
- Knill E, Laflamme R, and Milburn GJ. A scheme for efficient quantum computation with linear optics. Nature. 2001. p. 46–52.
- 3. Nielsen MI, Chuang. Quantum Computation and Quantum Information. Cambridge Univ. Press; 2000.
- Merkle RC. Two types of mechanical reversible logic. Nanotechnology. 1993; 4(2): 114–131.
- 5. Rice JE. A new look at reversible memory elements. Proceedings of 2006 IEEE International Symposium on Circuits and Systems; ISCAS. 2006. p. 243–6.
- Nandal A, Vigneswaran T, RanaAshwani K. Booth multiplier using reversible logic with low power and reduced logical complexity. Indian Journal of Science and Technology. 2014; 7(4): 525–9.
- Bhagyalakshmi HR, Venkatesha MK. Design of sequential circuit elements using reversible logic. World Applied Programming. 2012; 2: 263–271.
- Mamun MSA, Mandal I and Hasanuzzaman M. Efficient design of reversible sequential circuit. IOSR Journal of Computer Engineering. 2015; 5(6): 42–7.

- 9. Sayem ASM, Ueda M. Optimization of reversible sequential circuits. Journal of Computing. 2012; 2(6): 208–14.
- Miller DM, Maslov D, Dueck GW. A transformation based algorithm for reversible logic synthesis. Proceedings of Design Automation Conference; 2003. p. 318–323.
- Gupta P, Agrawal A, Jha NK. An algorithm for synthesis of reversible logic circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2006; 25(11): 2317–30.
- Morrison M, Ranganathan N. Design of a Moore finite state machine using a novel reversible logic gate, decoder and synchronous up-counter. Proceedings of 11th IEEE Conference on Nanotechnology (IEEE-NANO); 2011. p. 1445–49.
- Mahapatro M, Panda SK, Satpathy J, Saheel M, Suresh M, Panda AK, Sukla MK. Design of arithmetic circuits using reversible logic gates and power dissipation calculation. International Symposium on Electronic System Design (ISED). 2010. p. 85–90.
- Toffoli T. Reversible Computing. Cambridge, MA: MIT Lab. For Comp. Sci., MIT; 1980. Report No.: MIT/LCS/ TM-,1980.
- Feynman R. Quantum mechanical computers. Optics News; 1985. p. 11–20.
- Thapliyal H, Ranganathan N. Design of reversible sequential circuits optimizing quantum costs, delay, and garbage outputs, ACM J. Emerg. Technol. Comput. Syst. 2010; 6(4): 14.1–14.31.
- Hari SKS, Shroff S, Mahammad SK, Kamakoti V. Efficient building blocks for reversible sequential circuit design. 49th IEEE International Midwest Symposium on Circuits and Systems; 2006. p. 437–41.
- Fredkin E, Toffoli T. Conservative logic. Int. J. Theor Phys. 1982. p. 219–253.
- ChuangM-L, Wang, C-Y. Synthesis of reversible sequential elements. ACM J. Emerg. Technol. Comput. Syst. 2008; 3(4): 14.1–14.19.