RESEARCH ARTICLE OPEN ACCESS

Implementation of Effective Code Converters using Reversible Logic Gates

Ponnuru Koteswara Rao[#1], P Raveendra[#2], Kankata Venkateswara Rao[#3]

[#1,2,3]*Assistant Professor, SRK Institute of Technology, Enikepadu, Vijayawada*

ABSTRACT
The development in the field of nanometer technology leads to minimize the power consumption of logic circuits. Reversible logic design has been one of the promising technologies gaining greater interest due to less dissipation of heat and low power consumption. In the digital design, the code converters are widely used process. So, the reversible logic gates and reversible circuits for realizing code converters like as Binary to Gray code, Gray to Binary code, BCD to Excess 3 code, Excess 3 to BCD codes using reversible logic gates is proposed. Designing of reversible logic circuit is challenging task, since not enough number of gates are available for design. Reversible processor design needs its building blocks should be reversible in this view the designing of reversible code converters became essential one. In the digital domain, data or information is represented by a combination of 0's and 1's. A code is basically the pattern of these 0's and 1's used to represent the data. Code converters are a class of combinational digital circuits that are used to convert one type of code in to another. The proposed design leads to the reduction of power consumption compared with conventional logic circuits.

Keywords: Reversible logic, efficient code converters, low power logic gates & VLSI design.

I .INTRODUCTION

Reversible computing is a model of computing where the computational process to some extent is reversible, i.e., time invertible. A necessary condition for reversibility of a computational model is that the relation of the mapping states of transition functions to their successors should at all times be one-to-one. Reversible computing is generally considered an unconventional form of computing. There are two major, closely related, types of reversibility that are of particular interest for this purpose: physical reversibility and logical reversibility. A process is said to be physically reversible if it results in no increase in physical entropy; it is isentropic.

These circuits are also referred to as charge recovery logic or adiabatic computing. Although in practice no stationary physical process can be exactly physically reversible or isentropic, there is no known limit to the closeness with which we can approach perfect reversibility, in systems that are sufficiently well-isolated from interactions with unknown external environments, when the laws of physics describing the system's evolution are precisely known. Probably the largest motivation for the study of technologies aimed at actually implementing reversible computing is that they offer what is predicted to be the only potential way to improve the energy efficiency of computers beyond the fundamental von Neumann-Landauer limit of kTln energy dissipated per irreversible bit operation.

As was first argued by Rolf Landauer of IBM, in order for a computational process to be physically reversible, it must also be logically reversible. Landauer's principle is the loosely formulated notion that the erasure of n bits of information must always incur a cost of nkln in thermodynamic entropy. A discrete, deterministic computational process is said to be logically reversible if the transition function that maps old computational states to new ones is a one-to-one function; i.e. the output logical states uniquely defines the input logical states of the computational operation. For computational processes that are nondeterministic (in the sense of being probabilistic or random), the relation between old and new states is not a single-valued function, and the requirement needed to obtain physical reversibility becomes a slightly weaker condition, namely that the size of a given ensemble of possible initial computational states does not decrease, on average, as the computation proceeds forwards.

The reversibility of physics and reversible computing Landauer's principle (and indeed, the second law of thermodynamics itself) can also be understood to be a direct logical consequence of the underlying reversibility of physics, as is reflected in the general Hamiltonian formulation of mechanics and in the unitary time-evolution operator of quantum mechanics more specifically.

In the context of reversible physics, the phenomenon of entropy increase (and the observed arrow of time) can be understood to be consequences of the fact that our evolved predictive capabilities are rather limited, and cannot keep perfect track of the exact reversible evolution of complex physical systems, especially

since these systems are never perfectly isolated from an unknown external environment, and even the laws of physics themselves are still not known with complete precision. Thus, we (and physical observers generally) always accumulate some uncertainty about the state of physical systems, even if the system's true underlying dynamics is a perfectly reversible one that is subject to no entropy increase if viewed from a hypothetical omniscient perspective in which the dynamical laws are precisely known. The implementation of reversible computing thus amounts to learning how to characterize and control the physical dynamics of mechanisms to carry out desired computational operations so precisely that we can accumulate a negligible total amount of uncertainty regarding the complete physical state of the mechanism, per each logic operation that is performed. In other words, we would need to precisely track the state of the active energy that is involved in carrying out computational operations within the machine, and design the machine in such a way that the majority of this energy is recovered in an organized form that can be reused f subsequent operations, rather than being permitted to dissipate into the form of heat. Although achieving this goal presents a significant challenge for the design, manufacturing, and characterization of ultra-precise new physical mechanisms for computing there is at present no fundamental reason to think that this goal cannot eventually be accomplished, allowing us to someday build computers that generate much less than 1 bit's worth of physical entropy (and dissipate much less than kTln 2 energy to heat) for each useful logical operation that they carry out internally.

The motivation behind much of the research that has been done in reversible computing was the first seminal paper on the topic, which was published by Charles H. Bennett of IBM research in 1973. Today, the field has a substantial body of academic literature behind it. A wide variety of reversible device concepts, logic gates, electronic circuits, processor architectures, programming languages, and application algorithms have been designed and analyzed by physicists, electrical engineers, and computer scientists. This field of research awaits the detailed development of a high-quality, cost reversible logic device technology, one that includes highly energy-efficient clocking and synchronization mechanisms. This sort of solid engineering progress will be needed before the large body of theoretical research on reversible computing can find practical application in enabling real computer technology to circumvent the various near-term barriers to its energy efficiency, including the von Neumann bound. This may only be circumvented by the use of logically reversible computing, due to the Second Law of Thermodynamics.

1.1 BASIC DEFINITIONS OF REVERSIBLE LOGIC

In this section, basic definitions and ideas related to reversible logic are presented.

Definition 1 A Reversible gate is a k-input, k-output circuit that produces unique output for each possible input. Reversible gates are circuits in which the number of input is equal to the number of output and there is one to one correspondence between the vector of inputs and outputs.

Definition 2 The unused output of a reversible gate is known as Garbage output. More it known also, the outputs, which are needed only to maintain reversibility, are called garbage outputs.[1]

Definition 3 The simple and basis Reversible gate is conventional NOT gate and is a 1*1 gate. The block diagram is given in Figure1. The cost of Reversible NOT gate is 1.

Definition 4 The 2*2 Feynman gate, also known as Controlled NOT gate. This gate is one through because it passes one of its inputs. Every linear reversible function can be built by using only 2 * 2 inverters. Since this is a 2 * 2 gate, the cost is 1. The reasons to use this gate in reversible circuits are:

(i) Make the copy of an input (by putting any of the input a constant 0);

(ii) To invert an input bit (by putting any of the input a constant 1).

II.PROPOSED REVERSIBLE CODE CONVERTERS

Designing of reversible logic circuit is challenging task, since not enough number of gates are available for design. Reversible processor design needs its building blocks should be reversible in this view the designing of reversible code converters became essential one. In the digital domain, data or information is represented by a combination of 0's and 1's. A code is basically the pattern of these 0's and 1's used to represent the data. Code converters are a class of combinational digital circuits that are used to convert of code in to another. Some of the most prominently used codes in digital systems are Natural Binary Sequence, Binary Coded Decimal, Excess-3 Code, Gray Code, ASCII Code etc. Like any combinational digital circuit, a code converter can be implemented by using a circuitry of AND, OR and NOT gates.

Here this paper focuses more on conversion of code between binary to gray and BCD to excess. Binary to Gray code converters used to reduce switching activity by achieving single bit transition between logical sequences.

2.1 Binary To Gray converter Using FG gate

If Input vector is I(D,C,B,A) then the output vector O (Z,Y,X, W). The circuit is constructed with the help of Feynman Gate (FG) gate[5], figure 2.1 & 2.2

shows the circuit diagram of reversible Binary to Gray code converter & Gray to Binary code converters.

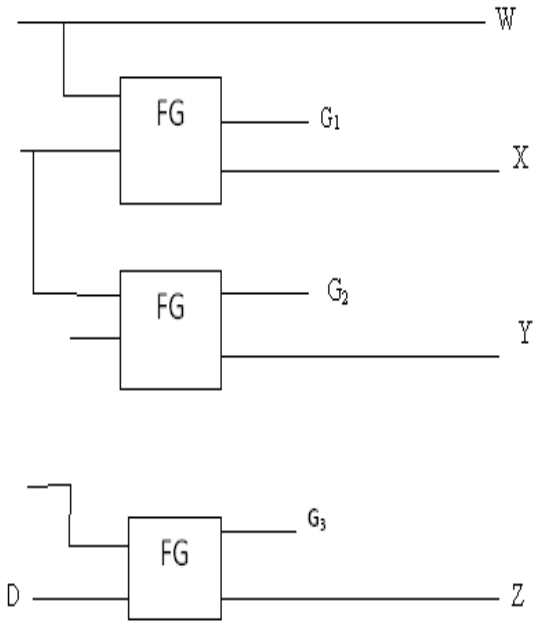


fig 2.1 Binary to Gray converter

2.2 Gray To Binary converter Using FG gate

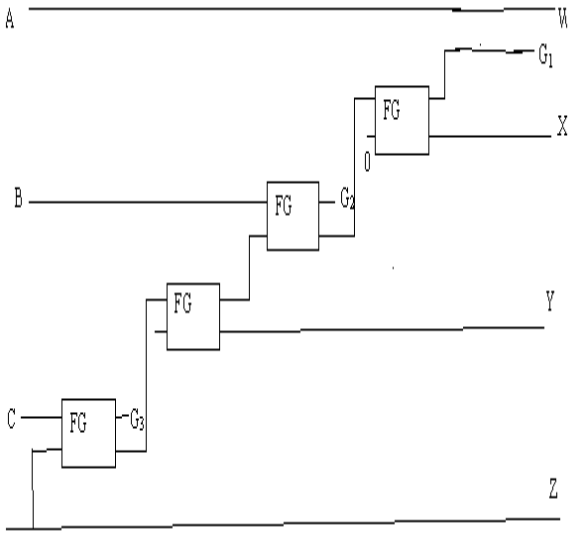


fig 2.2 Gray to Binary converter

2.3 Binary To Gray converter Using TG gate

If Input vector is I(D,C,B,A) then the output vector O (Z,Y,X,W). The circuit is constructed with the help of Toffoli Gate (TG) gate[5], figure 2.3 & 2.4 shows the circuit diagram of reversible Binary to Gray code converter & Gray to Binary code converters.

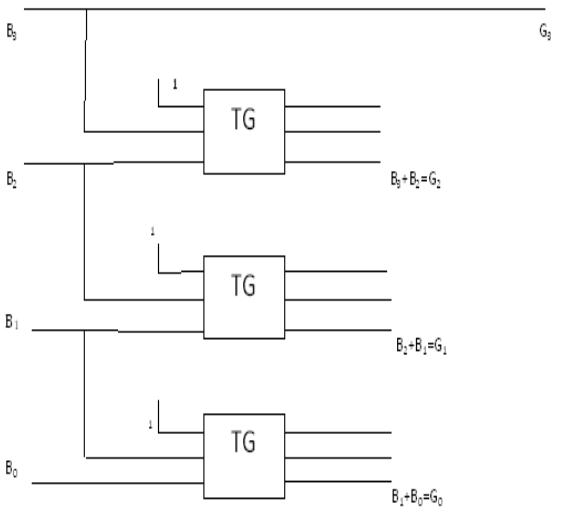


fig 2.3 Binary to Gray converter

2.4 Gray To Binary converter Using TG gate

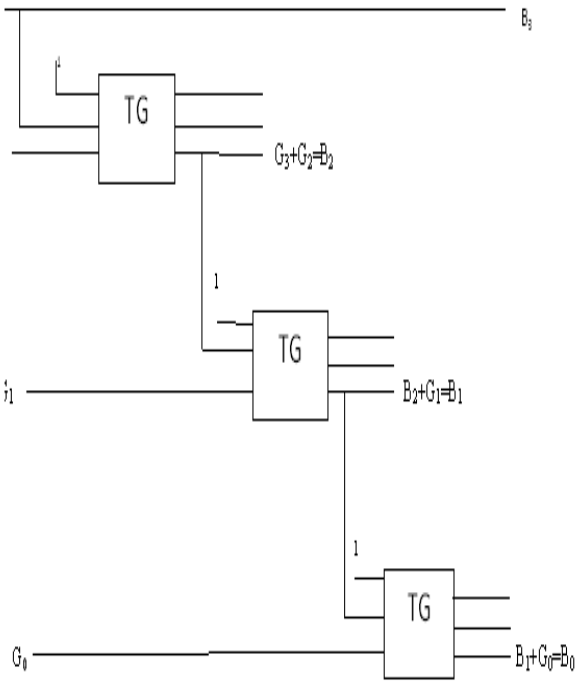


fig 2.4 Gray to Binary converter

2.5 Binary To Gray converter Using PG gate

If Input vector is I (D, C, B, A) then the output vector O (Z, Y, X, W). The circuit is constructed with the help of Peres Gate (PG) gate[5], figure 2.5 & 2.6 shows the circuit diagram of reversible Binary to Gray code converter & Gray to Binary code converters.

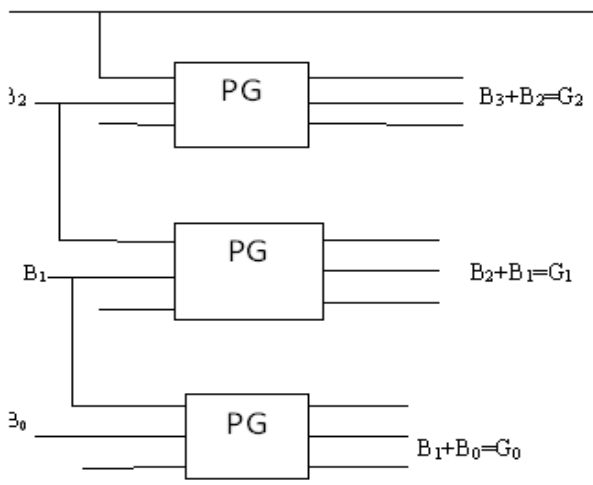


fig 2.5 Binary to Gray converter

2.6 Gray To Binary converter Using PG gate

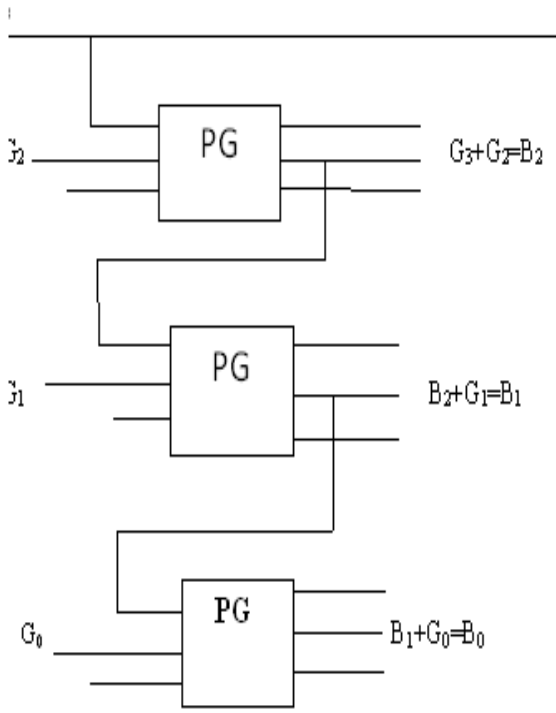


fig 2.6 Gray to Binary converter

Excess-3 to BCD code converters are used in arithmetic operational circuits to reduce the overall hardware complexity. Fig 2.7 shows the circuit diagram of Excess-3 to BCD code converter using URG gate.

2.7 Excess-3 To BCD code converter

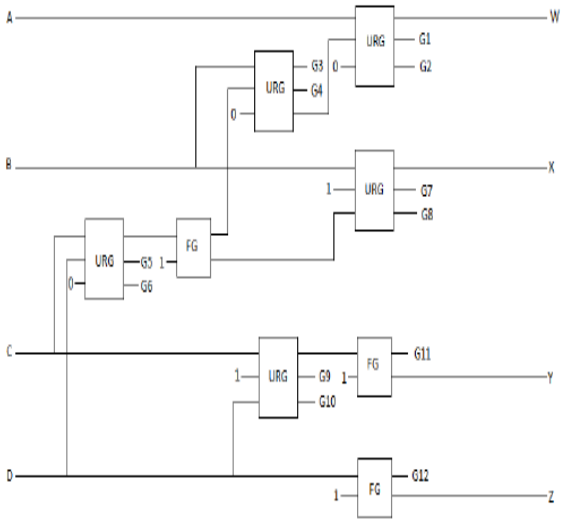


fig 2.7 Excess-3 to BCD converter

The proposed reversible code converters are more efficient than the conventional code converters.

III.RESULT

Design Statistics for binary to gray converters

Cell Usage:

# BELS	: 3
# LUT2	: 3
# IO Buffers	: 8
# IBUF	: 4
# OBUF	: 4
Selected Device	: 3s50pq208-5
Number of Slices	: 2 out of 768
Number of 4 input LUTs	: 3 out of 1536
Number of IOs	: 8
Number of bonded IOBs	: 8 out of 124
Maximum combinational path delay	: 7.824ns
Total memory usage	: 140512 kilobytes

Design Statistics for gray to binary converters

Cell Usage:

# BELS	: 3
# LUT2	: 1
# LUT3	: 1
# LUT4	: 1
# IO Buffers	: 8
# IBUF	: 4
# OBUF	: 4

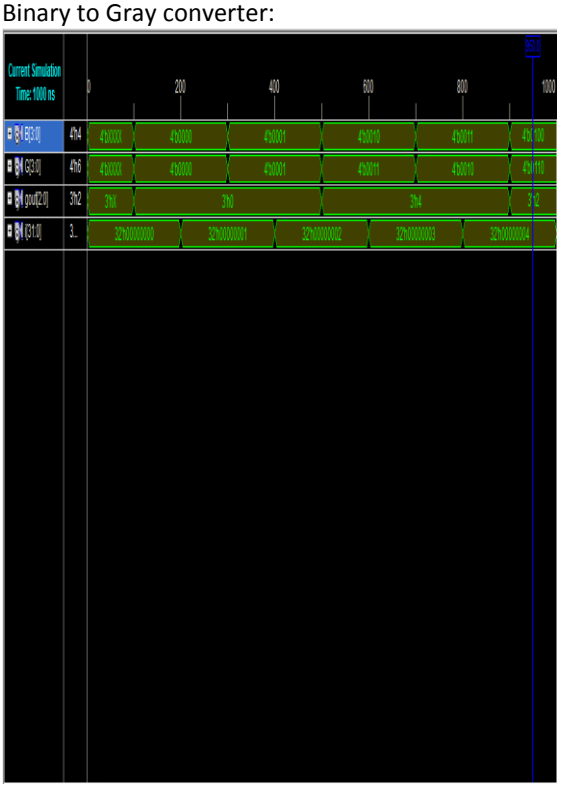
Device utilization summary:

Selected Device	: 3s50pq208-5
Number of Slices	: 2 out of 768

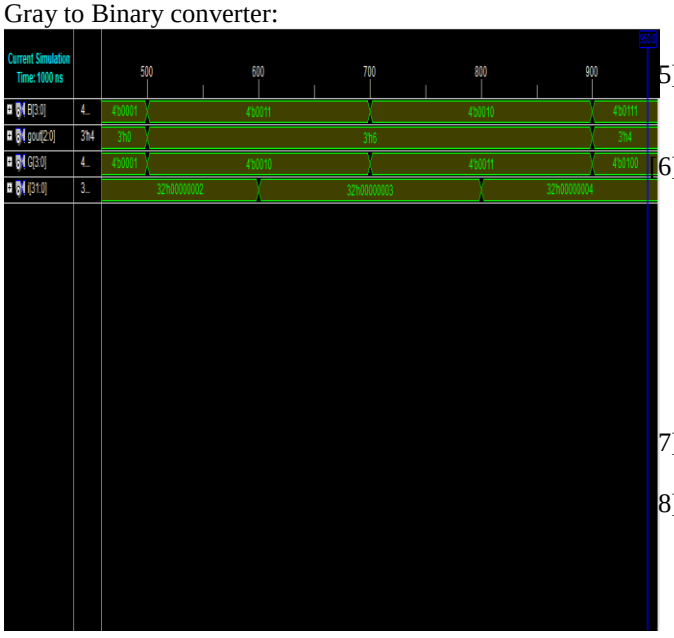
Number of 4 input LUTs	:	3 out of 1536
Number of IOs	:	8
Number of bonded IOBs	:	8 out of 124

Maximum combinational path delay: 7.850ns
Total memory usage is 140512 kilobytes

Binary to Gray converter:



Gray to Binary converter:



IV.CONCLUSION

This paper has introduced and proposed reversible logic gates and reversible circuits for realizing different code converters like BCD to Excess-3, Excess- Gray to Binary using reversible logic gates. The proposed design leads to the reduction of power consumption compared with conventional logic circuits, the design proposed is implemented with FG and URG gates only in near future with the invent of new RLG the power consumption may reduced to little more greater extent, not only that there will be a chance of implementing different logic circuits using reversible logic gates and which intern helps to increase the energy efficiency to a greater extent.

REFERENCES

[1]. Landauer, R., 1961. Irreversibility and heat generation in the computing process, IBM J.Research and Development, 5 (3): 183-191.

[2]. Bennett, C.H., 1973. Logical reversibility of computation, IBMJ. Research and Development, 17: 525-532.

[3]. Kerntopf, P., M.A. Perkowski and M.H.A. Khan, 2004. On universality of general reversible multiple valued logic gates, IEEE Proceeding Of the 34th international symposium on multiple valued logic (ISMVL'04), pp: 68- 73.

[4]. Perkowski, M., A. Al-Rabadi, P. Kerntopf, A.Buller, M. Chrzanowska- Jeske, A. Mishchenko, M.Azad Khan, A. Coppola, S. Yanushkevich, V.Shmerko and L. Jozwiak, 2001A general decomposition for reversible logic, Proc. RM'2001, Starkville, pp: 119-138.

5]. Saravanan.M, *Energy Efficient Code Converters using Reversible Logic Gates*, 2013.

6]. Thapliyal Himanshu, and M.B. Srinivas, 2005.Novel reversible TSG gate and its application for designing reversible carry look ahead adder and other adder architectures, Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC 05). Lecture Notes of Computer Science, Springer-Verlag, 3740: 775-786.

7]. Feynman, R., 1985. Quantum mechanical computers, Optics News, 11:11-20.

8]. M.Saravanan. M., Cholan K., Abhishek G, 2010. Design of Noval Reversible Multiplier Using MKG Gate in Nanotechnology, Proceedings of National Conference on Automation Control and Computing.