

Online Testable Decoder using Reversible Logic

Hemalatha.K.N.¹, Manjula B.B.², Girija.S.³

¹Lecturer, Electronics & Communications Engg Dept, Dr.Ambedakar Institute of technology, Bangalore, Karnataka

²Sr.Lecturer, Electronics & Communications Engg Dept, East West Institute of technology, Bangalore, Karnataka

³Sr. Lecturer, Electronics & Communications Engg Dept, Dr.Ambedakar Institute of technology, Bangalore, Karnataka

¹knhemalatha@gmail.com, ²manjulabb@gmail.com, ³sb_girija@yahoo.com

Abstract - The project proposes to design and test 2 to 4 reversible Decoder circuit with arbitrary number of gates to an online testable reversible one and is independent of the type of reversible gate used. The constructed circuit can detect any single bit errors and to convert a decoder circuit that is designed by reversible gates to an online testable reversible decoder circuit. Conventional digital circuits dissipate a significant amount of energy because bits of information are erased during the logic operations. Thus if logic gates are designed such that the information bits are not destroyed, the power consumption can be reduced. The information bits are not lost in case of a reversible computation. Reversible logic can be used to implement any Boolean logic function.

Keywords - Reversible logic, Feynman gate, NOT Gate, Fredkin Gate, Deduced reversible gate (DRG), Testable reversible gate (TRC), Test Cell (TC).

1. INTRODUCTION TO REVERSIBLE LOGIC

The new mantra of success has been fabricated by computing revolution. Every day we see it penetrating in to new application areas with the performance –per –unit – power consumption of digital technology improving as Moore had predicted. When the conventional approach will run out of steam, reversible computing will then be the only way to rapidly improve performance.

Reversible computing is the application of principles of recycling to computing. It means computing using a physical mechanism that is thermodynamically reversible and logically reversible as well. They are adiabatic system that recycle their energy and emit very little heat.

Reversible logic has gained importance in the recent past. The rapid decrease in the size of the chips has lead to the exponential increase in the transistor count per unit area. As a result, the energy dissipation is becoming a major barrier in the evolving computing era.

Researchers like Landauer have shown that for irreversible logic computations, each bit of information lost generates $KT \ln 2$ joules of heat energy, where K is Boltzmann's constant & T the absolute temperature at which computation is performed .Bennett showed that $KT \ln 2$ energy dissipation would not occur, if a computation is carried out in a reversible way, since the amount of energy dissipated in a system bears a direct relation ship to the number of bits erased during computation[1][2] .Reversible circuits are those circuits that do not lose information and reversible computation in a system can be performed only

when the system comprises of reversible gates. These circuits can generate unique output vector from each input vector and vice versa, a one to one mapping between input and output vectors.

Reversible circuits can be viewed as a special case of quantum circuits because quantum evolution must be reversible. Classical (non-quantum) reversible gates are subject to the same “circuit rules,” whether they operate on classical bits or quantum states. In fact, popular universal gate libraries for quantum computation often contain as subsets universal gate libraries for classical reversible computation. While the speed-ups which make quantum computing attractive are not available without purely quantum gates, logic synthesis for classical reversible circuits is a first step toward synthesis of quantum circuits. Moreover, algorithms for quantum communications and cryptography often do not have classical counterparts because they act on quantum states, even if their action in a given computational basis corresponds to classical reversible functions on bit-strings.

1.1 The Concept

Reversibility in computing implies that no information about the computational states can ever be lost. We can recover any earlier stage by computing backwards or uncomputing the result. This is termed as logical reversibility. The benefits of logical reversibility can be gained only after employing physical reversibility. Physical reversibility is a process that dissipates no energy to heat. Absolutely perfect physical reversibility is practically unachievable.

Computing systems give off heat when voltage levels change from positive to negative bits from zero to one. Most of the energy needed to make that change is given off in the form of heat. Rather than changing voltages to new levels, reversible circuit elements will gradually move charge from one node to the next. This way one can only expect to lose a minute amount of energy on each transition.

Reversible computing strongly affects digital logic designs. Reversible logic elements are needed to recover the state of inputs from the outputs. It will impact instruction sets & high level programming languages as well. Eventually these will also have to be reversible to provide optimal efficiency.

1.2 Need for Reversible Computing

High performance chips releasing large amounts of heat impose practical limitation on how far can we improve the performance of the system. Reversible circuits that conserve information by uncomputing bits instead of throwing them away, will soon after the only physical possible way to keep improving performance.

Reversible computing will also lead to improvement in energy efficiency. Energy efficiency will fundamentally affect the speed of circuits such as nanocircuits and therefore the speed of most computing applications. To increase the portability of devices again reversible computing is required. It will let circuit element sizes to reduce to atomic size limits and hence devices will become more portable.

Although the hardware design costs incurred in near future may be high but the power cost & performance being more dominant than logic hardware cost in today's computing era, the need of reversible computing cannot be ignored

Two conditions must be satisfied for reversible computation.

The First Condition

For any deterministic device to be reversible its input and output must be uniquely retrievable from each other.

- This is called logical reversibility.

The Second Condition

The device can actually run backwards, i.e., in another term it can be said that each operation converts no energy to heat and produces no entropy

- This is called physical reversibility.

- Second Law of Thermodynamics guarantees that no heat is dissipated

1.3 Limitations of Reversible Gates:

- 1) Fan-out is not permitted.
- 2) Loops are not allowed.

Fan-out and feedback can be achieved using copying gate –Feynman and Double Feynman gates [12][13].

1.4 Properties of Reversible Logic Gate:

1. Minimum input constants.
2. Minimum number of gates.
3. Minimum number of garbage outputs.

Garbage outputs are those outputs which are not used further for any computation.

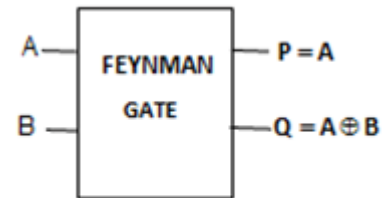
1.5 Synthesis of Reversible Gates:

Synthesis of reversible logic is different from conventional logic [20]. Synthesis can be carried out from the input towards the outputs or from the output towards the inputs. In reversible logic there is one more factor, which is more important than the number of gates used i.e. the number of garbage outputs. The Unutilized outputs from a reversible gate/circuit are called “garbage”. Though every synthesis method engages them producing less number of garbage outputs, but sometimes garbage outputs are unavoidable [5].

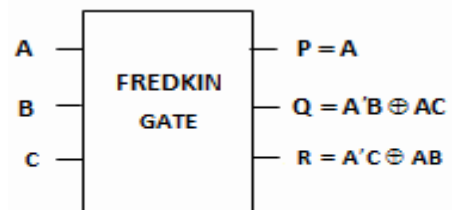
2. DIFFERENT TYPES OF REVERSIBLE GATES

2.1 Basic reversible logic gates

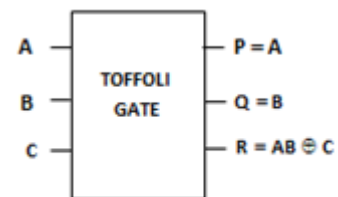
The important basic reversible logic gates are Feynman gate which is the only 2*2 reversible gates which is as shown in the figure and it is used most popularly by the designers for fan-out purposes. There is also a double Feynman gate, Fredkin gate and Toffoli gate, New Gate, Peres gate, all of which can be used to realize important combinational functions and all are 3*3 reversible gates and are as shown in the figure respectively.



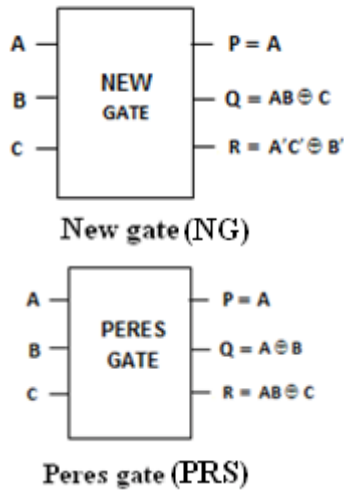
Feynman gate (FG)



Fredkin gate



Toffoli gate (TG)

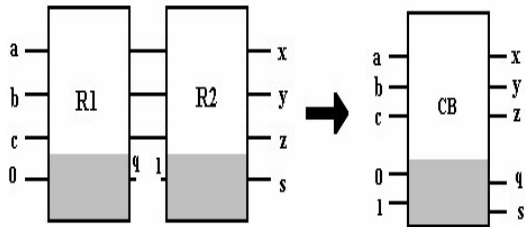


2.2 Applications of Reversible Circuits:

The most prominent application of reversible logic lies in,
 Quantum computers.
 Low power CMOS design.
 Optical Computing.
 Nanotechnology

2.3 Reversible Gates with a Built-in Testability:

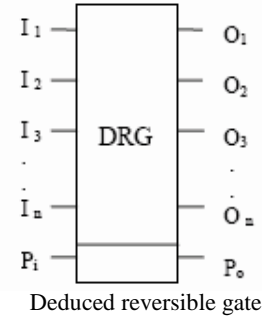
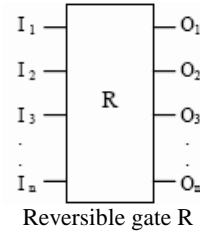
A testable logic block can be formed by cascading R1 and R2, as shown in Fig.. In this configuration, gate R2 is used to check online whether there is a fault in R1 or in itself. If R1 is fault free, its parity output q and the parity output s of R2 should be complementary; otherwise, the presence of a fault is assumed. Thus, during a normal operation, the presence of a fault in the logic block can be detected.



2.4 Deduced Reversible Gate (DRG):

Let the $n \times n$ reversible gate R is as shown in figure 4.1 with the input vector $I=[I_1, I_2, I_3, \dots, I_n]$ and the output vector $O=[O_1, O_2, O_3, \dots, O_n]$. As the gate is reversible we have one-to-one mapping between vector I and O. A deduced reversible gate of R, DRG(R) as shown in fig, is constructed by adding an extra input bit P_i and the corresponding output bit P_o to the gate R. Where :

$P_o = F + P_i$, Where $F = O_1 \oplus O_2 \oplus O_3 \dots \oplus O_n$, realized in terms of. $I_1, I_2, I_3, \dots, I_n$

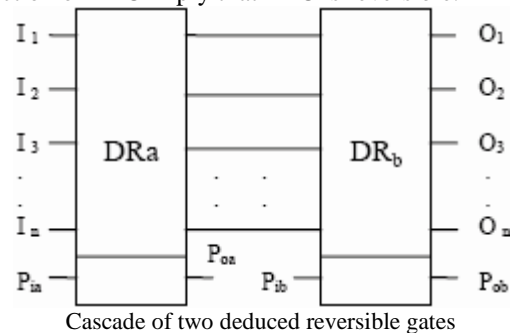


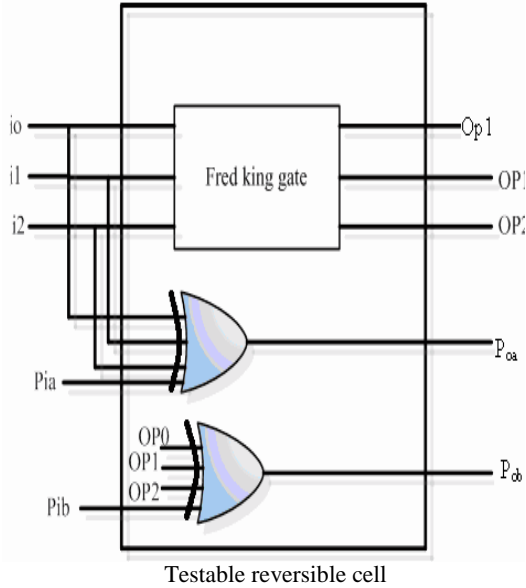
2.5 Testable Reversible Cell (TRC):

To construct a Testable reversible cell, TRC(R) as shown in fig, for a given reversible gate R. Consider a $n \times n$ gate X such that all the inputs to the gate X are mapped to the outputs without any change. It is obvious that the gate X is reversible. Let R be an $n \times n$ reversible gate. Let $DR_a = DRG_I$ and $DR_b = DRG(X)$. DR_a and DR_b are $(n+1) \times (n+1)$ gates. DR_a and DR_b are reversible gates.

Cascade the gates DR_a and DR_b as shown in figure by connecting the first n outputs of DR_a to the first n inputs of DR_b in order. The resultant gate can be viewed as a $(n+2) \times (n+2)$ gate as shown in figure, which we denote as the testable reversible cell of R, TRC. The input vector of TRC is defined as $[I, P_{ia}, P_{ib}]$, where I is the input vector of gate R and P_{ia}, P_{ib} are the added one bit inputs to the gates DR_a and DR_b respectively. Similarly, the output vector is defined as $[O, P_{oa}, P_{ob}]$, where O is the output vector of gate R and P_{oa}, P_{ob} are the added output bits to gate DR_a and Gate DR_b respectively.

The fact that DR_a and DR_b are reversible and the construction of TRC imply that TRC is reversible.





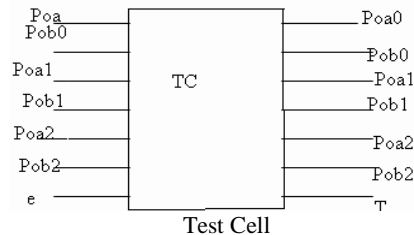
3. CONSTRUCTION OF ONLINE TESTABLE DECODER

3.1 Algorithm

Input : Reversible decoder C

Output : An online testable reversible decoder C^T

- 1) Construct c' by replacing every reversible gate R in C by TRC. The parity input bits of TRC are set such that P_{ia} = P_{ib} in the construction of TRC. C's reversible.
- 2) Let n be the number of reversible gates in C. Construct a (2n+1) x (2n+1) test cell (TC) as shown in figure .



- 3) First 2n inputs are the outputs parity bits from each of the n testable reversible cell TRC of C' gate.
- 4) The last bit of the input, called e is either set to logic 0 or logic 1.
- 5) First 2n inputs are transferred to the output without any change.
- 6) The last output bit (T) of the test cell (TC) is $T = [(P_{0a1} \oplus P_{0b1}) + (P_{0a2} \oplus P_{0b2}) + \dots + (P_{0an} \oplus P_{0bn})]$
Where P_{oak} and P_{obk} are the output parity bits of the kth TRC of C'.
- 7) Cascade C' and TC as stated in step 2 to obtain C^T.

As errors are detected dynamically at-speed during normal working of the circuit and without affecting the functionality of the reversible circuit, the proposed C^T qualifies as an online testable reversible circuit.

3.2 Proposed Technique:

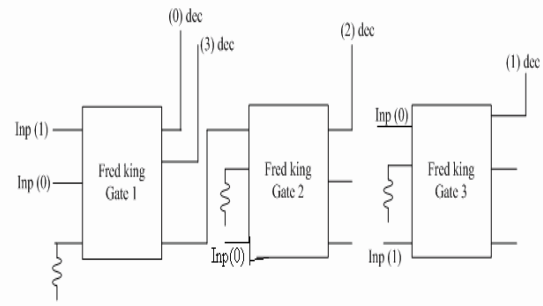
A reversible decoder circuit is converted to an online testable reversible decoder. A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2n unique output lines.

Let us consider 2 to 4 decoder. The truth table of the decoder is shown in table :

| I ₁ | I ₂ | O ₁ | O ₂ | O ₃ | O ₄ |
|----------------|----------------|----------------|----------------|----------------|----------------|
| x | x | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Truth table for 2 to 4 Decoder

The construction of reversible decoder uses three Fredkin gates (F₁, F₂ and F₃).



2 to 4 Reversible Decoder

Inp (0) and Inp (1) are the one bit inputs to the decoder and O₁, O₂, O₃ and O₄ are the output bits of the decoder. Algorithm is used to convert the decoder circuit into an online testable decoder.

Input: Reversible Decoder circuit C

Step 1: Replace Fredkin gate with its testable reversible TRC for K=1, 2, 3. Let the input vector be [a, b, c] and the output vector be [O₁, O₂, O₃]. The deduced Fredkin gate DR_a can be obtained with the following inputs and outputs as shown in figure.

Inputs: a, b, c and P_{ia}

Outputs: O₁=a; O₂=a ⊕ ab ⊕ ac

O₃=b ⊕ ab ⊕ ac; P_{oa}=O₁ ⊕ O₂ ⊕ O₃ ⊕ P_{ia}

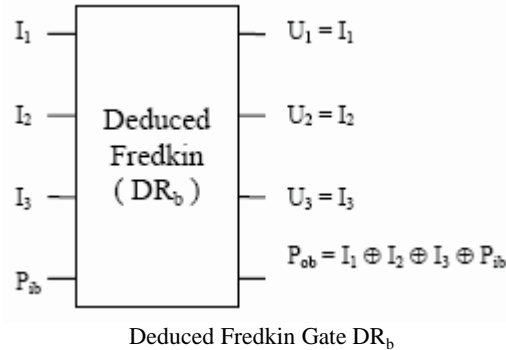
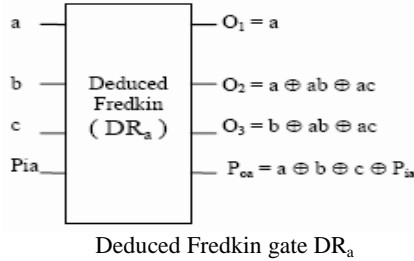
P_{oa}=a ⊕ (c ⊕ ab ⊕ ac) ⊕ (b ⊕ ab ⊕ ac) ⊕ P_{ia}=a ⊕ b ⊕ c ⊕ P_{ia}

To construct DR_b , take X to be a 3×3 gate That has inputs as $[I_1, I_2, I_3]$ and outputs as $[U_1, U_2, U_3]$, where U_i and I_i are related as $U_i = I_i$ for $i=1, 2, 3$. The deduced gate DR_b is shown in figure..

Inputs: I_1, I_2, I_3 and P_{ib}

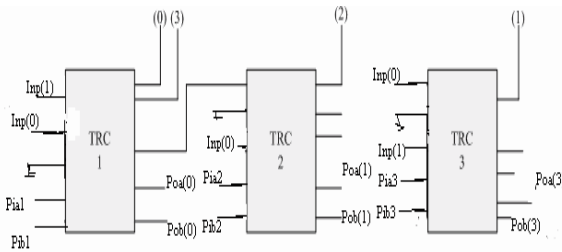
Outputs: $U_i = I_i$ where $i=1, 2, 3$.

$$P_{ob} = P_{ib} \oplus U_1 \oplus U_2 \oplus U_3 = P_{ib} \oplus I_1 \oplus I_2 \oplus I_3$$

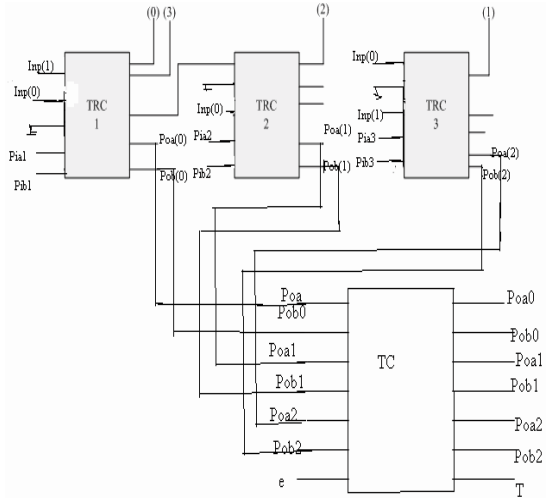


Step 2: Add test cell (TC) with $2n+1=7$ input line. As $n=3$ for the decoder circuit, TC has $2n+1$ input lines. Connect its first six input lines to the parity bits P_{oak} and P_{obk} of the Fredkin gate for $K = 1, 2, 3$. First six lines are passed to the output lines without any change. Output bit T is the error detecting bit. The value of T will determine if there is an error in the circuit.

Circuit thus obtained is the online testable reversible decoder.



Testable Reversible Decoder



Online testable reversible decoder circuit

The circuit thus obtained is online testable reversible decoder. Let us consider the case when the input vector $[I_1, I_2] = [1, 0]$ from the truth table of decoder the output vector O should be $[0, 0, 1, 0]$. In this case, the parity vector $[P_{oa1}, P_{ob1}, P_{oa2}, P_{ob2}, P_{oa3}, P_{ob3}]$ is equal to $[0, 0, 1, 1, 0, 0]$ and error = 0 which shows that the circuit is error free.

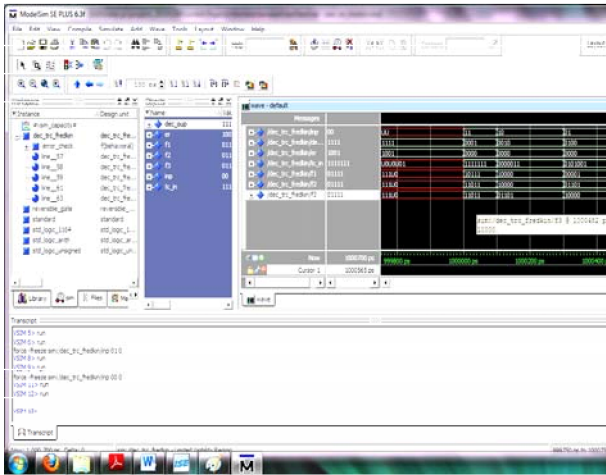
Suppose if there is some error in the circuit, say in Fredkin gate 1, for the given input vector $[I_1, I_2] = [1, 0]$ output of Fredkin gate 1 is $[1, 0, 1, 0]$ instead of $[0, 0, 1, 0]$, parity vector $[P_{ia1}, P_{ib1}, P_{ia2}, P_{ib2}, P_{ia3}, P_{ib3}]$ will be equal to $[0, 1, 1, 1, 0, 0]$ and hence error = 1, so the circuit is erroneous.

3.3 Advantage of Online Testability:

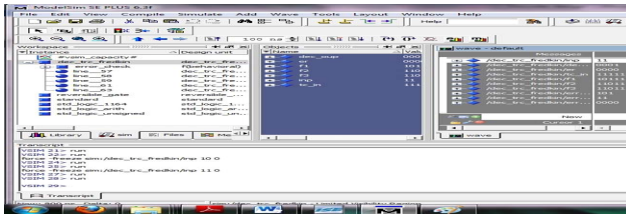
An important advantage of the technique is that the logic design of a reversible circuit remains the same and the reversible circuit need not be redesigned for adding the testability feature to it. Another advantage is that the technique ensures that the garbage generated during the process of conversion to testable reversible circuit is minimized. The resultant testable circuit can detect online any single bit errors that include single stuck faults and single event upsets.

4. RESULTS

4.1. Waveform Showing with Error in TRC:



4.2. Waveform Showing Without Error in Design:



4.3 Scope :

The work is focused on the design of online reversible circuit that can be tested. The design includes different block like implementation of algorithm for the reversible logic gate tests. All design need to be verified to ensure that no error in VHDL programming before being simulated. The second scope is to implement the design into FPGA hardware development board. This process is implemented if all designs are correctly verified and simulated using particular software. Implementation includes hardware programming on FPGA or downloading hardware design into FPGA and software programming. Creating test vector program also include in the scope of the project. Test vector is a program developed and is intended as the input interface for user as well as to control data processing performed by the hardware. These computation values should be verified and tested to ensure the correctness of the developed module. Appropriate software is used to compare the computation performed by the FPGA hardware with the software.

REFERENCES

- [1] D.P.Vasudevan, P.K.Lala and J.P.Parkerson, CMOS realization of online testable reversible logic gates.
- [2] SK Noor Mahammad, Siva Kumar Sastry Hari, Shyam Shroff and V Kamakoti, Constructing online testable circuits using reversible logic.
- [3] Dilip P. Vasudevan, Parag K.Lala, Fellow IEEE, Jia Di, and J.Patrick Parkerson, Reversible Logic Design with online testability.
- [4] R. W. Keyes and R. Landauer, Minimal Energy Dissipation in logic.
- [5] "Design of a Reversible Binary Coded Decimal Adder by Using Reversible 4-bit Parallel Adder" Hafiz Md. Hasan Babu and Ahsan Raja Chowdhury Department of Computer Science and Engineering, IEEE 2005
- [6] C.H.Bennett, Logical reversibility of computation.
- [7] Samir Palnitkar, Verilog HDL.A guide to digital design and synthesis. Second edition IEEE 1364-2001 complaint.
- [8] Charles H Roth, Jr, Digital systems design using VHDL, Thomson BROOKS/COLE
- [9] R. W. Keyes and R. Landauer, "Minimal energy dissipation in logic," IBM J. Research and Development, pp. 152–157, March 1970.
- [10] C.H. Bennett, "Notes on the history of reversible computation," IBM J. Research and Development, vol. 32, pp. 16–23, January 1988. [11]. R. Landauer, "Irreversibility and heat generation in the computing process," IBM J. Research and Development, vol. 3, pp. 183–191, July 1961.
- [11] C. H. Bennett, "Logical reversibility of computation," IBM J. Research and Development, pp. 525–532, November 1973. [13] R. Feynman, "Quantum mechanical computers," Optics News, Vol. 11, pp. 11–20, 1985.
- [14] Bennett, C., "Logical Reversibility of Computation," IBM Journal of Research and Development, 17, 1973, 525-532.
- [15] Hafiz Md. Hasan Babu, Md. Rafiqul Islam, Ahsan Raja Chowdhury and Syed Mostahed Ali Chowdhury, "Reversible Logic Synthesis for Minimization of Fulladder Circuit", IEEE Conference on Digital System Design 2003, Euro-Micro'03, Belek, Antalya, Turkey, 2003, pp. 50-54.
- [16] P. Kerntopf, "Synthesis of multipurpose reversible logic gates," Euromicro Symposium on Digital System Design (DSD'02), pp. 259–267, 2002.
- [17] N. Margolus, "Physics and computation," Ph. D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1988.
- [18] S. G. Younis and T. F. Knight, "Practical implementation of charge recovering asymptotically zero power cmos," Proceeding of the 1993 symposium on Research on integrated systems, MIT press, pp. 234–250, 1993.
- [19] S. G. Younis and T. F. Knight, "Asymptotically zero energy split-level charge recovery logic," Proc. Workshop Low Power Design, Napa Valley California, pp. 177– 182, 1994.
- [20] J. Lim, K. Kwon and S.-I. Chae, "Reversible energy recovery logic circuit.

AUTHOR'S PROFILE



Smt. K. N. Hemalatha

working as Lecturer in the department of Electronics and Communication, Dr.Ambedkar Institute of Technology., Bangalore -56.
Her specialization is in VLSI and Embedded systems.



Smt. Girija.S.

working as Senior Lecturer in the department of Electronics and Communication, Dr.Ambedkar Institute of Technology., Bangalore -56.
Her specialization is in computer networks and Embedded systems.



Smt. Manjula B. B.

working as Sr.Lecturer in the department of Electronics and Communication, East West Institute of Technology., Bangalore -10.
Her specialization is in VLSI and Embedded systems