

Convolutional Encoder using Reversible Logic Circuits

Prashik Lokhande,
Student M.E.

(Electronics & Telecommunication), KJSCE, Vidyavihar,
Mumbai, India

Prof. Jyoti Varvadekar,

Department of Electronics & Telecommunication, KJSCE,
Vidyavihar, Mumbai, India

Abstract- Low power design is the important issue in VLSI design. Reversible logic circuit is one of the techniques receiving attention of researchers for low power VLSI design. In this technique logical reversibility is achieved using reversible logic gates. Convolutional encoder is widely used in digital systems for error correction. And hence attempt is made to design a convolutional encoder using reversible logic circuit. In this paper exhaustive study and implementation of convolutional encoder and decoder using reversible logic circuit using GDI (gate diffusion input) technique is presented.

Keywords: Reversible logic circuits, GDI

I. INTRODUCTION

Reversible logic is one of the most vital issue at present time and it has different areas for its application, those are low power CMOS, quantum computing, nanotechnology, cryptography, optical computing. Reversible circuits are circuits that have the same number of inputs (k) and outputs (k) and are one-to-one mappings between vectors of inputs and outputs, thus the vector of input states can be always uniquely reconstructed from the vector of output states. Researchers are trying to develop circuits using reversible logic circuits for low power design [1, 2]. Sometimes there is loss of data when data is transferred from memory to other part of system on chip and hence while implementing low power design, researchers should also make attention on data loss, so IC manufacturer are manufacturing memories with error correction coding techniques to reduce the loss and guarantee loss proof transmission. Convolution coding is the basic technique to overcome the data loss and widely used in digital systems. The work is done to design convolutional encoder and the decoder with synchronous parallel decoding using reversible logic circuits [2, 3, 4, 5].

II. REVERSIBLE LOGIC CIRCUIT

Reversible Logic Circuit is one of the methods at circuit and logic level, the energy recovery level, which employs the reversible logic concept. A circuit is reversible if its computes are bijective i.e. there is the one to one mapping between input and the output, so that from output of the circuit it is able to determine the input to the circuits. The logical structure to realize the reversible logic circuits helps to

effectively model the faults generated in the circuits than current methods, built in self test is easy for these designs.

Reversible Logic Function: A function is reversible if it satisfies following two criterions;

- i. Number of inputs equal to the number of outputs
- ii. Every output vector has a unique pre-image

If $I_v (I_1, I_2, I_3, \dots, I_n)$ is the input vector and $O_v (O_1, O_2, O_3 \dots O_k)$ is the output vector then $I_v = O_v$.

Reversible Logic Gate: A reversible logic gate is ($n \times k$) device where n is the number of input bits and k is the number of output bits with one to one mapping i.e. $n = k$

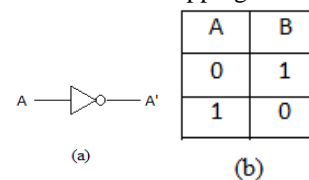


Fig.2.1 (a) NOT Gate (b) truth table of NOT gate [2]

In conventional circuits NOT gate is the only reversible gate, as from output its input can be determined. While, AND OR and XOR gates are irreversible, as we can't determine input to gate from their output.

Quantum Cost: Every reversible gate has a cost associated with it known as quantum cost. Quantum cost of a reversible gate is the number of elementary operations required to implement its functionality, hence quantum cost defines the complexity of the circuit. Fig.2.1 shows a 1x1 simple NOT gate. CNOT (Controlled NOT) is the 2x2 reversible gate. Many 3x3, 4 x 4 gates have been proposed. All the reversible gates can be optimized by the NOT gate. If V is the root of NOT gate and V^+ is its Hermitian, then the quantum cost of the gate is calculated by counting the number or of V and V^+ in the gate.

The quantum gates has the following properties

$$V * V = \text{NOT} \quad (1)$$

$$V * V^+ = V^+ * V = I \quad (2)$$

$$V^+ * V^+ = \text{NOT} \quad (3)$$

Garbage output: This is the most important and prominent feature of reversible gate. The output of the circuit which is

not is used as the input to another circuit is called garbage output.

A reversible circuit is realized by three basic components:

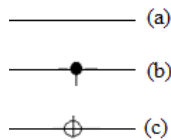


Fig 2.2 Components of reversible circuit (a) don't care line (b) control line (c) target line

The input on the don't care line is passed to output without change. Input at control line controls output on target line, if input to control line is '0' the input to target line will be passed as it is to its output and if input to control line is '1' the inverse of the input line is passed to its output.

Reversible gates: There have been many reversible gates synthesized as below, of which CNOT or Toffoli gate are used mostly to synthesize the reversible circuits. Any reversible circuit can be implemented using CNOT and TOFFOLI gate; hence these gates are called universal reversible gates.

NOT Gate: It is 1x1 reversible gate with zero quantum cost. The input gets inverted at output, and hence we can determine what was the input to the gate, so it is the reversible gate present in conventional circuits.

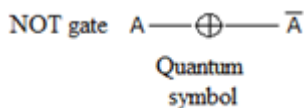


Fig 2.3 NOT Gate

Feynman/CNOT gate: It is 2x2 reversible gate with quantum cost of 1. The black dot in the circuit shown in fig 2.4 (b) is called control point. If A = 1, then inverse of B will be the output. As the NOT operation on B is controlled through control point by input A, the gate is called Controlled NOT.

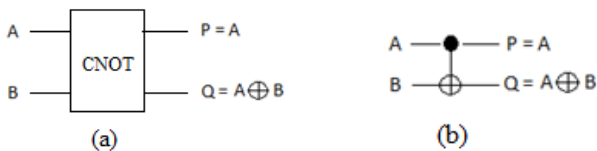


Fig 2.4 (a) CNOT Gate and (b) its Quantum realization [2]

Table I: truth table for Fynman gate

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Conventional circuits are not reversible; it can be made reversible by adding extra garbage bits to output, making number of input equal to number of output, and then making permutations of the states of output so as to have correct mapping between input and output. Using this method many reversible gates are proposed.

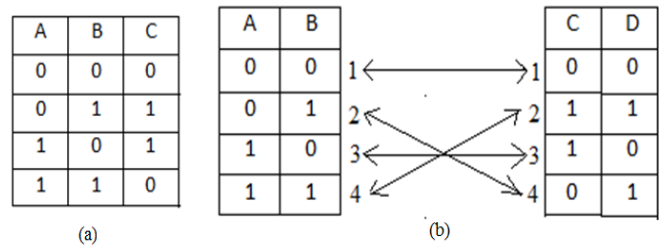


Figure 2.5: (a) truth table convention XOR gate (b) bit-'D' is added to make reversible XOR gate

III. CONVOLUTIONAL ENCODING

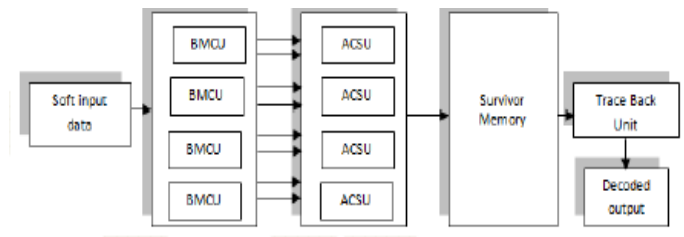


Fig.3.1 Parallel Viterbi decoder

In parallel Viterbi decoder, there are multiple BMCU (branch metric unit) and ACSU (add compare and select unit), to calculate the branch metrics and path metrics for all states of the Trellis per stage. Similarly the storage of survivor paths for all states of that particular stage of trellis is also done at the same time. The number of BMCUs and ACSUs are equal to 2, where n is the encoder's constraint length. So due to parallel nature of Viterbi decoder, the Trellis implementation becomes much faster. After Trellis calculation and survivor path storage, trace back operation is started in the same manner as used in conventional way of the trellis method. After the calculation of the path metrics, path metrics are stored in the registers, and then survival path is found out. To decode the data two methods are used trace back method and register exchange. In the RE approach, a register is assigned to each state. The register records the decoded output sequence along the path from the initial state to the final state. At the last stage, the decoded output sequence is the one that is stored in the survivor path register, the register assigned to the state with the minimum path metric.

In these conventional methods all the blocks are working sequentially, and so require extra clocking circuitry to allow synchronisation between them. The proposed design does have blocks that do the same in parallel way, without need of extra clocking circuit for synchronisation between them. The

decoding is done using trace back unit as it is less complex than register exchange method.

Proposed design:

The encoder has constraint length 3 with code rate of 1/2. Initially all bits of the shift register is set to zero. For the input 10100 the encoder output is (11 10 00 10 11).

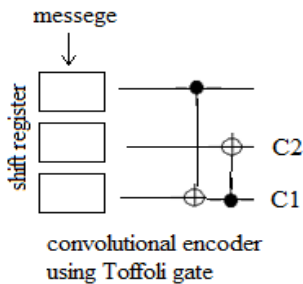


Fig . 3.2 convolutional encoder

Decoder: In the proposed decoder there is no survivor path storage trace back unit decodes the survival path in parallel. All blocks are implemented using reversible gates.



Fig. 3.6 Block diagram for proposed decoder

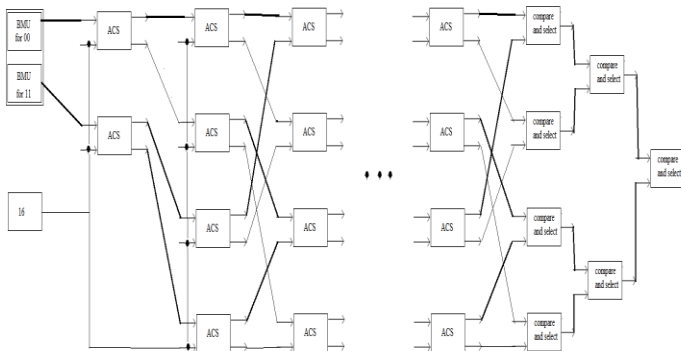


Fig.3.7 Trellis for generating path metric

Implementation of Convolutional decoder

Branch metric unit:

The first step in the Viterbi decoding algorithm requires calculation of the branch-metric. The branch metric is the distance from the received code word to all the possible branch words. An implementation of the block is shown in Figure 3.8. The architecture comprises of a XOR gate and a counter. The branch word depends on the constraint length, the generator matrix, and the code rate. One input to the XOR gate is the received code symbol which is the encoder output and the other input is the generated sequence in the polynomial. XOR gate determines the difference in the number of transitions in the inputs and counter counts the

total number of differing bits. In this case, the possible branch words are (X0, X1) 00, 01, 10 and 11.

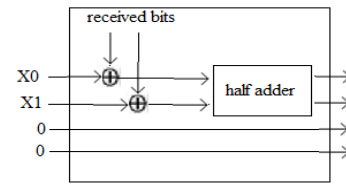
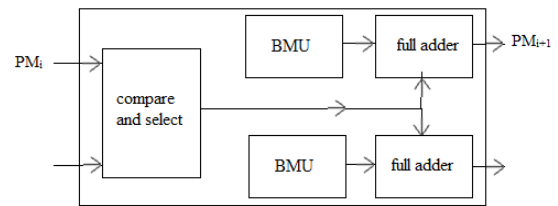


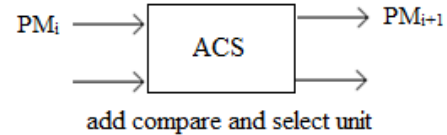
Fig.3.8 Branch metric unit

ACS (Add compare and select) unit:

Realization of ACS unit in figure 3.9 has adder, comparator and selector. Two inputs are the path metrics from previous state are given to the compare and select unit to select lowest one and added to the branch metric of the current state using 4 bit full adder to generate new path metric. Initial value of the other input of the adder is taken as zero. PM_i is the current state path metric and PM_{i+1} is path metric for next state.



(a)



(b)

Fig.3.9 (a) blocks in ACS unit (b) representation of ACS unit

Compare and select unit:

if C=0 then $PM_{i+1} = PM_{ij}$
if C=1 then $PM_{i+1} = PM_{ijn}$

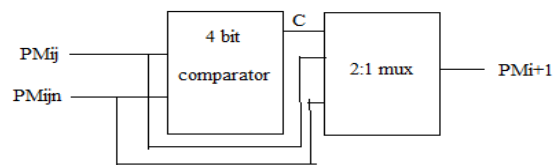
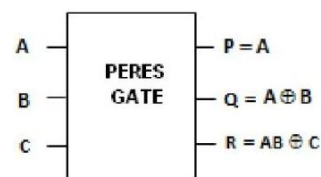


Fig.3.10 compare and select unit

Full adder: The reversible full adder is composed from reversible Peres gate as shown in fig.3.11



(a)

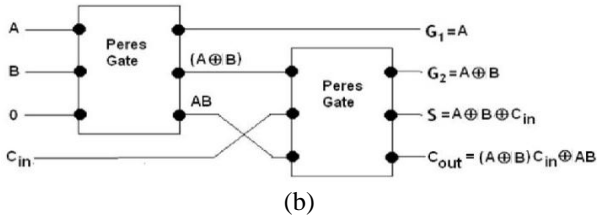


Fig.3.11 (a) Peres gate (b) full adder using Peres gate

Multiplexer: Fredkin gate is used to design 2:1 multiplexer to select the small path metric as shown in figure.

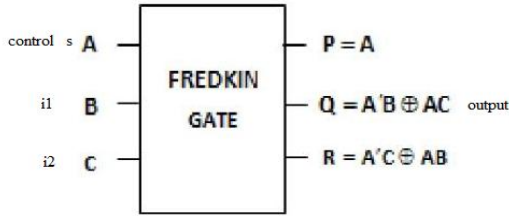


Fig.3.12 Multiplexer using reversible Fredkin gate

Trace back circuit:

In the proposed design trace back unit is used to detect the survival path with minimum metric. In this circuit a logical flag is generated to indicate the ACS with minimum path metric. This flag is used to detect the lowest path metric in the previous state. If flag is set to 1 then the ACS unit has lowest metric in that state or set to zero otherwise. In this way a survival path with minimum metric is detected, and so here the storage for the path metric and branch metrics is not required.

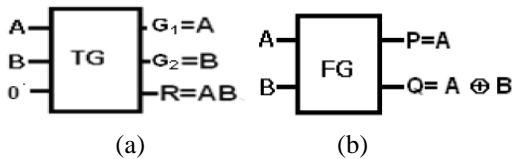


Fig. 3.13 (a) TG gate as AND (b) gate Fynman gate

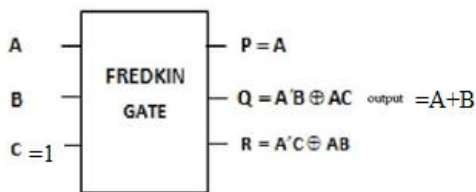


Fig 3.14 Fredkin gate

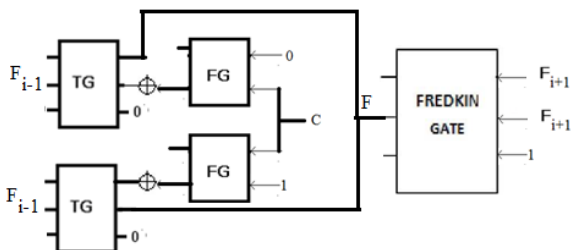


Fig 3.15 Trace back unit

Decoder circuit: The decoder circuit is formed using XOR gate. Suppose that the noise introduces the error in the output of the encoder, and the sequence becomes (01 10 10 10 11), the path metric generation in trellis is shown in fig 5.9, at the same time flags generated by the trace back circuit is shown in figure 5.10. To implement the circuits at transistor level GDI technique is used.

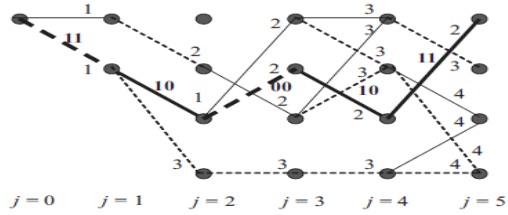


Fig.3.16 Trellis diagram showing survival path while decoding

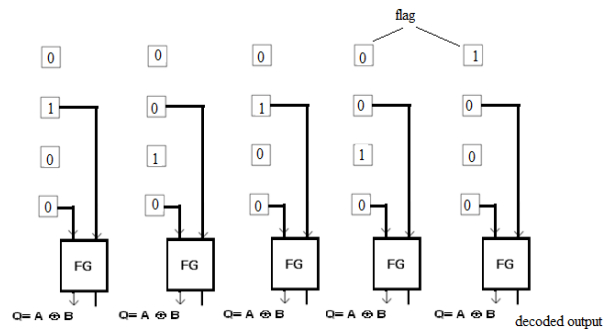


Fig. 3.17 decoding circuit for trellis

IV. IMPLEMENTATION

All the blocks are implemented using GDI technique.

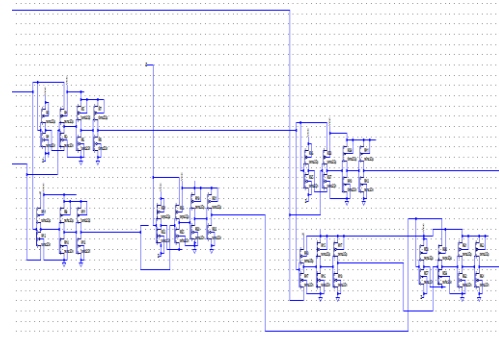


Fig 4.1 Full adder

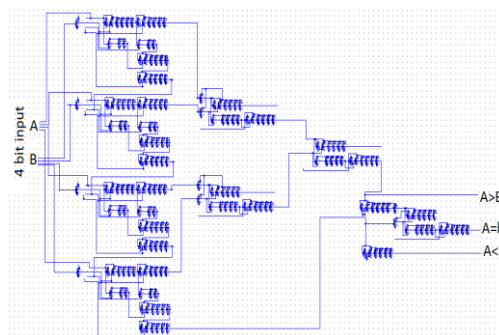


Fig 4.2 Comparator

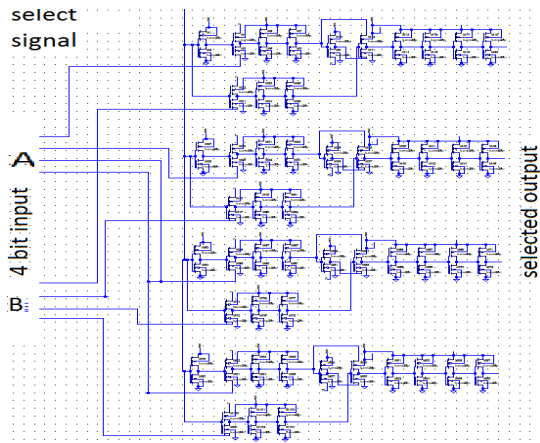


Fig 4.3 2:1 mux

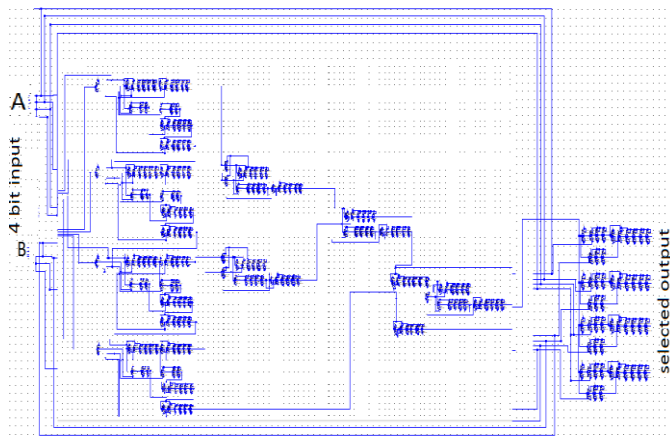


Fig 4.4 Compare and select unit

Table II: comparison between irreversible ACS and reversible ACS

	Irreversible		Reversible (mw)	
	Power (mw)	Delay	Power (mw)	Delay
Full adder	11.14	645ps	7.009	525ps
comparator	15.12	991ps	12.04	439ps
2:1 mux	6.91	800ps	3.192	687ps
Compare and select unit	21.22	925ps	15.232	700ps
ACS unit	32.36	980ps	22.241	837ps

V. CONCLUSION AND FUTURE WORK

The proposed design do not requires extra clocking circuitry for synchronisation between the BMU, PMU, trace back unit and decoder and survivor path memory. Circuit blocks were implemented in LTspice with 180nm TSMC process, the comparison between the power dissipation of the irreversible and reversible ACS unit is given in table II. Future work is to design the logic circuit to collect the decoded output and store in the memory stack.

REFERENCES:

- [1] Bennett, C.H., "Logical reversibility of Computation", IBM J. Research and Development 17: pp. 525-532, 1973.
- [2] Design of Arithmetic Circuits Using Reversible Logic Gates and Power Dissipation Calculation, Madhusmita Mahapatro, Sisira Kanta Panda, Jagannath, 2010 International Symposium on Electronic System Design, 978-0-7695-4294-2/10 \$26.00 © 2010 IEEE DOI 10.1109/ISED.2010.25.
- [3] DESIGN OF LOW POWER VITERBI DECODER USING ASYNCHRONOUS TECHNIQUES, International Journal of Advances in Engineering & Technology, July 2012.
- [4] Hardware Implementation of Viterbi Decoder for Wireless Applications, International Journal of Computer Communication and Information System (IJCCIS)- Vol2. No1. ISSN: 0976-1349 July – Dec 2010.
- [5] A CMOS Hard-Decision Analog Convolutional Decoder Employing the MFDA for Low-Power Applications, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 55, NO. 9, OCTOBER 2008.
- [6] Design of Arithmetic Circuits Using Reversible Logic Gates and Power Dissipation Calculation, Madhusmita Mahapatro, Sisira Kanta Panda, 2010 International Symposium on Electronic System Design, 978-0-7695-4294-2/10 \$26.00 © 2010 IEEE
- [7] Gate Diffusion Input (GDI) Logic in Standard CMOS Nanoscale Process, Arkadiy Morgenshtein, Idan Shwartz and Alexander Fish, 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, 978-1-4244-8682-3/10/\$26.00 c2010 IEEE.