Power-Efficient Add-Compare-Select Unit Implementation Based On Gate Diffusion Input (GDI) Logic

Rehana Farheen[1], Dr. Kaleem Fatima[2]

Assistant Professor[1], HOD[2]

Department of ECE

MJCET Engineering College, Hyd, Telangana, India

Corresponding Authors' email id: *rehana.farheen@gmail.com[1], kaleemfatima@gmail.com[2]*

Abstract

This paper presents an area efficient, low-power androbust Add Compare Select(ACS)unit for Viterbi Decoder using GDI (Gate Diffusion Input) logic. Viterbi decoder is used for decoding data encoded using Convolution Forward Error Correction codes or data that suffers from inter-symbol interference. They occur in a large proportion of digital transmission. Viterbi decoders employed in digital wireless communications are complex and dissipate large power. In present scenario the minimization of power consumption have emerged as a key design constraint over the last few years. The proposed method focuses on GDI, which is a low power technique of digital circuit design. Dynamic component of power is reduced in GDI technique as source of PMOS is not permanently connected to Vdd. It also reduces the latency of the circuit. The GDI cell is similar to the standard CMOS inverter, but there are some important differences. The ACS has been implemented using GDI cell with 0.18µm technology. The proposed design is obtained with reduced power dissipation in comparison with CMOS.

Keywords: *Viterbi decoder, GDI circuit, CMOS, low-power, high-speed*

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

I. INTRODUCTION

The increasing demand of complex mobile systems, which has been observed during the last years in the worldwide market, led the designers to take into account a new objective in the design of complex digital circuits: the minimization of power consumption [1]. The systems like laptop and palmtop computers, cellular phones, wireless modems and portable multimedia applications is one of the most important reasons that fuel the need for a low-power design. The need for minimization of power dissipation of a system is also enforced by some thermal considerations; a big portion of the energy requested by a device from the power supply is converted into heat [2]. In this way heat dissipation system and cooling mechanisms become indispensable for the correct and safe operation of the device and also for its reliability.

The commonly used encoding schemes for minimizing the probability of error are: Block codes, cyclic codes, convolution codes. Convolution codes are widely used because of simple encoding and efficient decoding techniques. Convolution codes were first mentioned by Elias [3] in 1955. Andrew J. Viterbi Algorithm belongs to a large class of FEC (Forward Error Correction) codes known as convolution codes which are used in a wide range of applications such as wireless communications, digital mobile telephony, digital TV broadcast, CD-ROMs, and magnetic disks. The basic idea behind the Viterbi decoder is to maximize the correlation between received vector and the table of possible code words while sequentially performing the opposite operation of the encoder. The Viterbi decoder operates by finding the maximum likelihood of the decoding sequence. Several attempts for increasing the throughput and reducing the power dissipation and the area in the Viterbi decoder have been reported in the literature [4-10].

This paper has been organised as follows: section I presents introduction, Viterbi encoder/decoder has been described in context of ACS in section II, followed by the GDI techniques design of the ACS unit.

II. ADD COMPARE SELECT UNIT

A. Convolution Encoder

The operations of the convolution encoder and decoder for constraint length of 3 are

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

presented in this section. The convolution encoder has a rate of ½ (k/n) with a constraint length of 3. With an encoder, the 3 bit shift register provides the memory and two modulo-2 adders provide convolution operations. For each bit in the input sequence, two bits are output, one from each of the two modulo-2 adders.

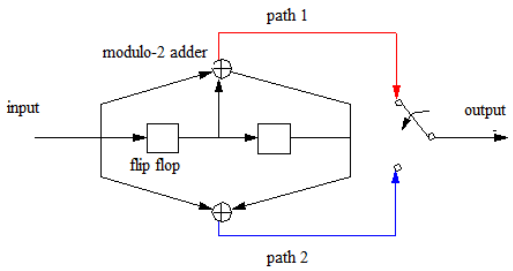


Figure 1: Convolution Encoder with rate of ½ (k/n) with a constraint length of 3

Figure 1 represents a convolution encoder that takes one bit in the input and returns two bits in the output. We can describe it by impulse response (or the generator polynomial) of each of the two paths. The impulse response is defined as the response of a path to a symbol 1 applied to its input, with each flip-flop initially set to zero.

B. Convolution decoder

As figure 2 encoding and decoding for in a communication. Although convolution encoding is a simple procedure, decoding of a convolution code is much more complex task. Several classes of algorithms exist for this purpose: Threshold decoding (use topological to reduce complex), Sequential decoding (use Tree diagram), and Viterbi decoding (use Trellis diagram).

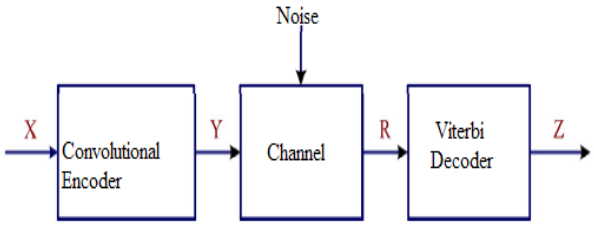


Figure 2: Convolution encoder and Viterbi decoder.

The basic units of Viterbi decoder are branch metric unit, add compare and select unit and survivor memory management unit [11].

- Branch metrics unit (Calculates the distances between the received signal and the ideal signals).
- Add-Compare-Select unit (Accumulates path metrics).
- Survivor path decode (Keeps track of the path through the trellis).

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

C. Add Compare Select Unit

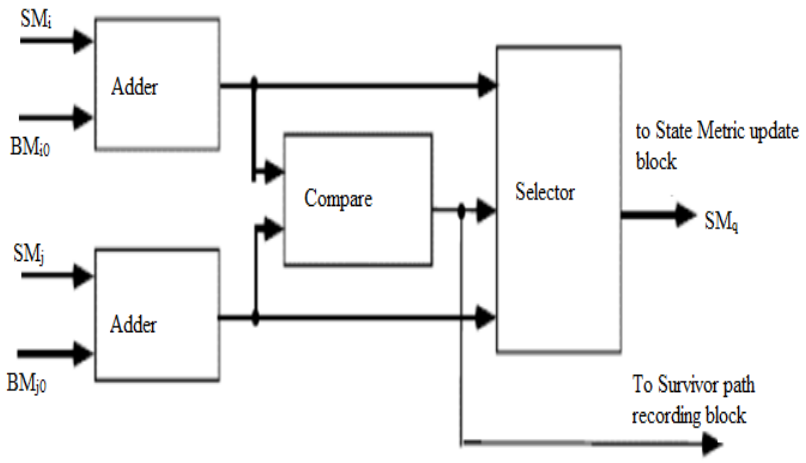


Figure 3: Block diagram of ACS

The Add Compare Select Unit (ACSU) which adds the Branch Metrics (BM) to the corresponding Path Metrics (PM), compares the new PMs, and then stores the selected PMs in the Path Metric Memory (PMM); at the same time, the ACSU stores the associated survivor path decisions in the Survivor Memory Unit (SMU).

The Block diagram of the Add Compare and Select unit is shown in the Figure 3. State metric (SMi,j) and Branch Metric (BMi,j) are the two inputs to the Adder unit. Each butterfly wing is usually implemented by a module called ACS module. The two adders compute the partial path metric of each branch, the comparator compares the two partial metrics, and the selector selects an appropriate branch. The new partial path metric updates the state metric of state p, and the survivor path-recording block records the survivor path. The adder unit which is proposed in the design consists of two full adders and one half-adder. Output of the Branch metric unit (BM) is added with the previous path metric and the obtained output is the new path metric for the next branch. The input sequence from a0 (1100), b0 (1001), a1 (1011), b1 (1101), a2 (0111) and b2 (0011) are added to the adder unit. The 1st bit of a0 and b0, i.e. 11, is the input of half-adder and produces the output of the half adder operation sum1 as ’0’ and carry as ‘1’. The half adder carry is given to the next full-adder which takes the 3-bit input as a1, b1 and carry, i.e. 111 and produces the output as full-adder operation sum2 as ’1’ and carry as ‘1’.

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

III. GATE DIFFUSION INPUT

In this paper, a new low-power design technique that allows solving most of the problems has been proposed. The GDI approach allows implementation of a wide range of complex logic functions using only two transistors. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to CMOS and existing PTL techniques), while improving logic level swing and static power characteristics and allowing simple top-down design by using small cell library.

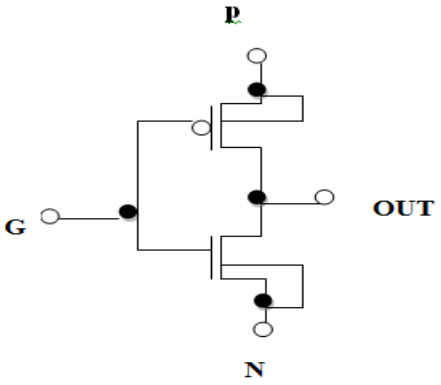


Figure 4 : GDI basic cell.

The GDI method is based on the use of a simple cell as shown in Fig. 4.1. At first glance, the basic cell reminds one of the standard CMOS inverter, but there are some important differences. 1) The GDI cell contains three inputs:G(common gate input of nMOS and pMOS), P (input to the source/drain of pMOS), and N (input to the source/drain of nMOS). 2) Bulks of both nMOS and pMOS are connected to N or P (respectively), so it can be arbitrarily biased at contrast with a CMOS inverter.

Table 1 gives the various functions that can be implemented using basic GDI cell and the number transistors required for standard CMOS process.

TABLE 1: Functions implemented using GDI cell and the transistors required using standard CMOS process

N	P	G	Out	Function	No. Transistors required using GDI	No. Transistors required using CMOS
0	1	A	A`|	Inverter	2	2
0	B	A	A`B	Function1	2	6
B	1	A	A`+B	Function2	2	6
1	B	A	A+B	OR	2	6
B	0	A	AB	AND	2	6
C	B	A	A`B+AC`	MUX	2	12
B`	B	A	A`B+B`A	XOR	4	16
B	B`	A	AB+A`B`	XNOR	4	16

The basic GDI cell consists of four terminals. Consider Figure 5, When ‘P’ of the transistor is given ‘0’ it will cut-off from its operation, hence the logic either ‘1’ or ‘0’ at the input ‘a’ will be reflected at the output ‘z’. Thus the output will be z = a*b.

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

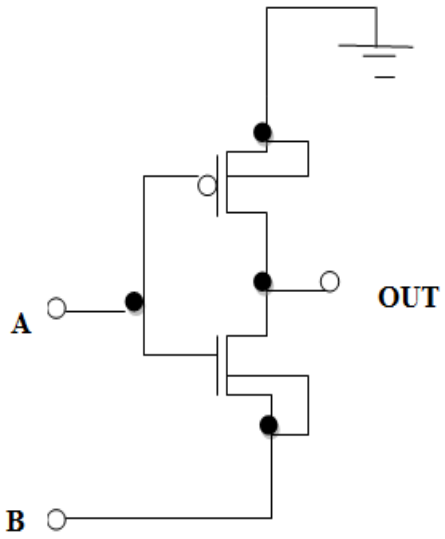


Figure 5: GDI AND gate

Among all the design techniques, GDI proves to have the minimal number of transistors. Each GDI gate was implemented using only two transistors. As shown in Table 1&2, a large variety of functions can be implemented by using a basic GDI cell, which consists of only two transistors.

The Multiplexer (MUX) is the most complex function that can be implemented with a basic GDI cell. Compared to other existing CMOS and PTL MUX implementation alternatives which usually require 8-12 transistors, it is obvious that 2-transistor GDI MUX implementation is more efficient by means of transistor count.

However, it is also important to examine the efficiency of other functions that can be implemented with a single GDI cell. Table 2 shows a comparison of the number of transistors that is required to implement various functions in GDI and CMOS. As expected, the GDI approach is more efficient for the AND, OR, F1 and F2 functions.

The same number of four transistors is required to implement the NAND and NOR functions, which are the most popular gates in today's Static CMOS designs: (1) implementation of these functions are very efficient in CMOS (only four transistors);

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

(2) each one of the functions is a universal set, i.e. any Boolean function can be synthesized using only NAND or NOR gates. Similar to Static CMOS, GDI provides two basic functions, F1 and F2. Each one of these functions is a universal set and consists of only two transistors, compared to four transistors in NAND and NOR.

Therefore, in general, every digital circuit can be implemented using only F1 or F2 gates or a combination of both. Moreover, the F1 and F2 functions can be used to synthesize other functions more efficiently than the NAND and NOR gates. In addition, the Static CMOS gates always provide the same function at their output, the GDI F1 and F2 gates can be modified to provide other functions, using the same basic 2-input gates with a simple change of inputs. This can enable the composition of a complex Boolean statement more efficiently than when constrained to only a certain function.

Table 2: Logic gate comparisons (gdi, cmos, transmission gate, and nmos pass gate) using the circuit topologies from table1.[12]

Gate type in series	Logic expression	GDI			CMOS			TG			N-PG		
		Power (μW)	Delay (nsec)	# tr.	Power (μW)	Delay (nsec)	# tr.	Power (μW)	Delay (nsec)	# tr.	Power (μW)	Delay (nsec)	# tr.
MUX	$\bar{A}B+AC$	35.7	1.1	8	49.7	2.1	24	44.9	1.0	16	47.5	3.1	16
OR	$A+B$	26.3	1.2	8	32.9	1.7	12	36.2	1.3	16	32.6	2.7	16
AND	AB	25.7	0.9	8	34.1	1.4	12	30.8	0.8	16	30.1	2.8	16
F1	$\bar{A}B$	31.2	0.8	8	45.2	1.5	12	31.8	1.1	16	31.8	2.5	16
F2	$\bar{A}+B$	32.0	1.3	8	43.1	1.9	12	33.2	1.4	16	29.6	3.5	16

IV. DESIGN AND SIMULATION

The half adder carry is given to the next full-adder which takes the 3-bit input as a1, b1 and carry, i.e. 111 and produces the output as full-adder operation sum2 as ’1’ and carry as ‘1’.

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

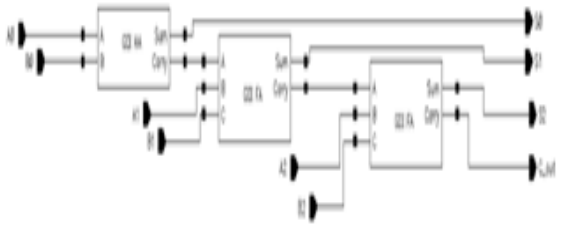


Figure 6: 3-bit Adder unit of ACSU

The "carry" is given to the next full-adder input as 3-bit input a2, b2 and carry, i.e. 100 which produces the output as full-adder operation sum as '1' and carry as '0'. All other input bits are processed similarly. The input value of 4-bit comparator a0, a1, a2, a3 and b0, b1, b2, b3 is 0011 and 1011.

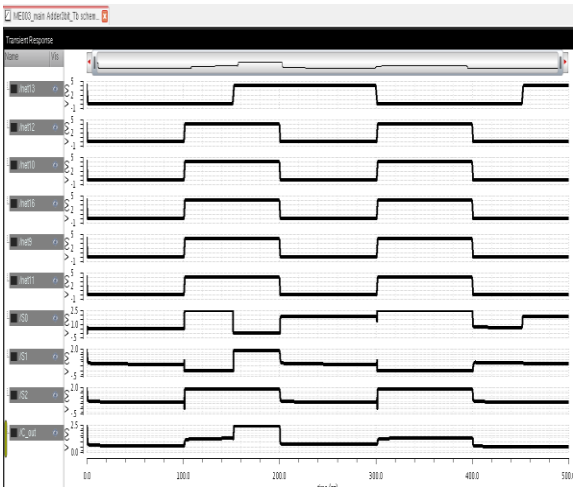


Figure 7: Simulation Result of Adder.

When the two inputs to the comparator are A=0011 and B=1011, then the output line A<B will go to high state i.e. less than state. If A and B are given as A=0111 and B=0100, then the output line A>B will go to the high state i.e. greater than state. Likewise all the input values are worked in the comparator.

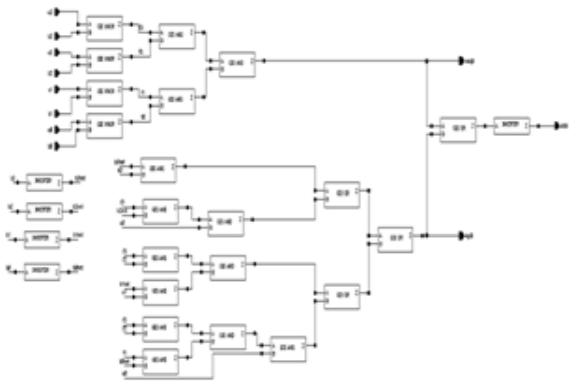


Figure 8: 4-Bit Comparator

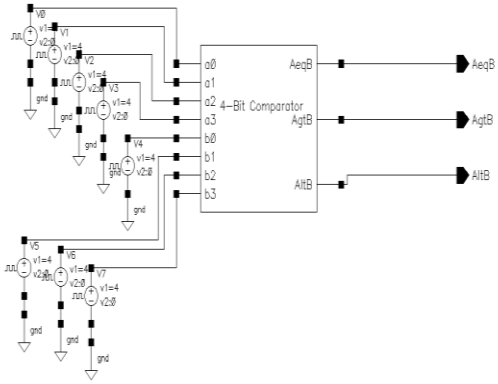


Figure 9: Test Bench of 4 Bit Comparator

Here the A<B value is taken because it is the smallest value and hence the A<B state output is given to the next SMU unit.

A. Selector Unit Design Using GDI

The output of the comparator is given as the select signal for the multiplexer which is used to select the minimum path metric of the decoded message bit in the Viterbi

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

decoder. The selector unit consist of four 2:1 MUX and the select signal for all the multiplexers are from the A<B output of the comparator. Hence the selector selects the minimum path metric value. The 4-bit input is a0=1101, b0=1111, a1=1101, b1=1000, a2=1111, b2=1110, a3=1001, b3=1110 and select line value is 1010. When the select line value is high i.e. '1', the output value z0 will be b0, i.e. '1'. When the select line value is '0', the output value z0 will be a0. Likewise all other input values z1, z2 and z3 are taken and the outputs are obtained.

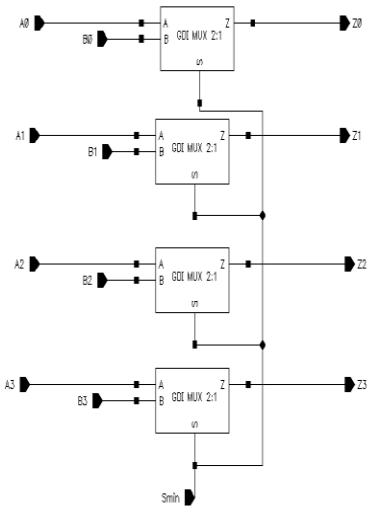


Figure 10: 4-Bit Selector

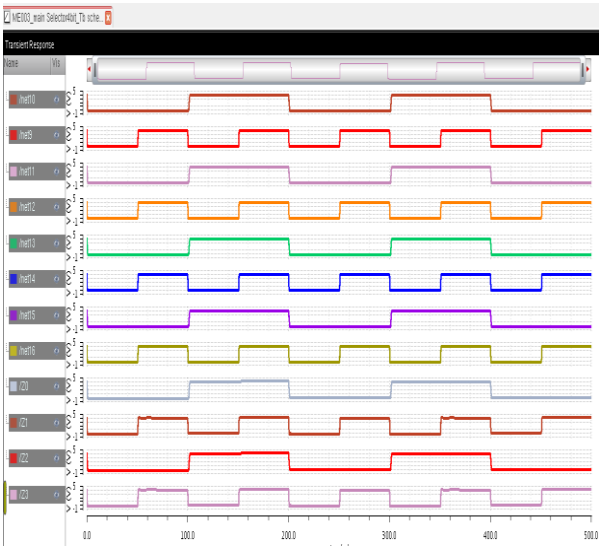


Figure 11: Simulation result of 4-Bit selector

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

Figure 11 shows the output waveform of the 4-Bit selector unit which selects the minimum value of the input and passes it to the output line z0,z1,z2,z3.

B. Performance Analysis

Table 3 shows the comparison of GDI technique designing the viterbi decoder with that of CMOS design[13]. For the constraint length of 3 of viterbi decoder, the gate count of new technique GDI takes half of that of CMOS design, therefore the area and power gets lesser giving this technique more attraction for emerging designers aiming at reliable and efficient designs with lower power and gate counts.

TABLE 3: Comparison table of GDI and CMOS

	GDI	CMOS
Constraint Length	3	3
Gate Count	568	1682
Area mm2	2.369	4.99
Power mW	10.395	49.5

V. CONCLUSION

This paper illustrates an efficient scheme to reduce power dissipation in ACS using GDI technique. This technique has been implemented on basic logic gates initially followed by MSI combinational circuits like adders and mux. The analysis shows that the GDI technique is novel and an effective technique for reducing power consumption and the transistor count effectively reduces the size of the chip. With respect to chip area, power consumption and transistor count, GDI technique is significantly advantageous over other techniques.

The convolution encoder and ACS unit in decoder for the constraint length K=3 and code rate ½ are designed using both the GDI and CMOS technique. The power dissipation of proposed design was found less by 80% as compared to CMOS. Depending on the application and the system requirements, this approach can be used to reduce the power dissipation of the digital systems. Hence GDI technique can be used to design low power circuits such as digital wrist watches, radio frequency identification (RFID), sensor nodes, Laptops, pacemakers and battery operated devices such as, cellular phones etc.

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

REFERENCES

1) J. M. Rabaey, A. Chandrakasan, B. Nikolic, "Digital Integrated Circuits", 2nd edition, Prentice Hall, 2002.

2) Sung-Mo Kang, Yusuf Lablebici, "C MOS Digital Integrated Circuits: Analysis and Design", 3rd edition Tata Mc-Graw Hill, 2003.

3) G. Davis and Jr. Forney, "The Viterbi algorithm", Proceedings of IEEE, vol. 61, no. 3, pp. 268-278, Mar. 1973.

4) Xun Liu and M. C. Papaefthymiou, "Design of a High-Throughput Low-Power IS95 Viterbi Decoder", in Proc. of IEEE DAC 2002, pp. 263-268, June 2002.

5) Chi-Ying Tsui; Cheng, R.S.-K.; Ling, C. "Low power ACS unit design for the Viterbi decoder". in Proc. Of the IEEE Symp. on Circuits and Systems, pp. 137-140, 1999.

6) \ I. Bogdan, M. Mumunteanu, P.A. Ivey, N.L. Seed, and N. Powell, "Power Reduction Techniques for a Viterbi Decoder Implementation", ESPLD 2000 (European Low Power Initiative for Electronic System Design) Third International Workshop, Rapallo, Italy, ISBN 90-5326-036-6, pp 28-48, July 2000.

7) P.A. Riocereux, E.M. Brackenbury, M. Cumpstey,and S.B. Fruber, "A Low-Power Self-Timed Viterbi Decoder", in Proc. 7th International Symp. On Asynchronous Circuits and Systems, pp. 15-24, 2001. [8] B. Javadi, M. Naderi, H. Pedram, A. Afzali-Kusha, and M. K. Akbari, "An Asynchronous Veterbi Decoder for Low Power Applications", PATMOS 2003, Torino, Italy, pp. 471-480, Sep., 2003.

8) G. Jung, J. J. Jong, G. E. Sobelman, and K. K. Parhi,"High-Speed Add-Compare-Select Units Using Locally Self-Resetting CMOS", in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS02), May 2002, Arizona, USA.

9) T. Gemmeke, V. S. Gierenz, and T. G. Noll, "Scalable, Power and Area

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved

Efficient High Throughput Viterbi Decoder", in Proceedings of 27th European Solid-State Circuits Conference (ESSCIRC01), Villach, Austria, Sep., 2001.

10)]Sundeepkumar Agarwal, Pavankumar V K, Yokesh R., "Energy –Efficient, High Performance Circuits for Arithmetic Units". 21st International Conference on VLSI Design, pp. 371-376, 2008.

11) 'W. S. Ju, M. D. Shieh, and M. H. Sheu, "A low-power VLSI architecture for the Viterbi decoder," in IEEE Proc. of the 40th Midwest Symposium on Circuits and Systems, vol. 2, pp. 1201-1204,Aug. 1997.

12) T. Kalavathidevi, C. Venkatesh, "Gate Diffusion Input (GDI) Circuits Based Low Power VLSI Architecture for a Viterbi Decoder", IJECE , VOL. 10, NO. 2, SUMMER-FALL 2011

Page 12-23 © MANTECH PUBLICATIONS 2017. All Rights Reserved