International Journal of Modern Trends in Engineering and Research
www.ijmter.com

e-ISSN No.:2349-9745, Date: 2-4 July, 2015

Fast and Efficient Encoder with concurrent Bubble error correction for Flash ADC

Sarfraz Hussain[1], Ashok K. Ray[2]
[1]*Department of Electronics and Communication Engineering, NERIST, Nirjuli, sarfrazenator@gmail.com*
[2]*Department of Electronics and Communication Engineering, NERIST, Nirjuli, ashokrayec@gmail.com*

Abstract- Encoder used in Flash ADC converts Thermometer code to Binary code. It is the bottleneck in terms of speed in an ADC. The proposed encoder provides high speed with concurrent bubble error correction. It consists of two encoding blocks. First is a 1 - out of - N coder circuit and second is encoder block consisting of NAND and NOR gates. It can be used in Flash ADCs for very high speed applications. There are many encoders available like ROM encoder, Wallace tree encoder, Multiplexer based decoder, Priority encoder but in terms of performance and speed Fat tree encoder is better. The proposed encoder is the modified Fat tree encoder. The encoder doesn't require any clock like in case of ROM and has fewer components as compared to other encoders. The 1 - out of - N coder circuit selects one out of the N number of inputs and the other part of the encoder block converts the outputs to binary code by using combinational circuits.

Keywords-Fat tree encoder; Flash ADC; Bubble Error; ROM encoder; Priority encoder.

I. INTRODUCTION

Flash ADC has two main blocks: Comparator and Encoder. Comparators in a Flash ADC are connected in parallel which makes it to work at a faster rate. There are total $2^N - 1$ number of comparators. The only limiting factor in speed is encoder. Encoder is the bottleneck of Flash ADCs. Fig. 1 shows the Flash ADC blocks.

The role of an encoder is to encode 2^N number of codes into N number of codes. The encoder used in a Flash ADC converts Thermometer code to equivalent Binary code. There are many types of encoders: ROM encoder [1], Wallace tree based encoder [2], Multiplexer based decoder [3], Priority encoder [4], and Fat tree encoder [5]. Among these, ROM encoder was used mostly in all the Flash ADCs. Other encoders were designed with different technologies like GaAs technology and SiGe Bipolar technology but the speed still remains the bottleneck while operating at higher frequencies.

Fat tree encoder designed with CMOS technology provides better performance in terms of speed and power dissipation. The proposed encoder is modified Fat tree encoder. It has two blocks: 1 - out of - N coder block and an encoding block. The 1 - out of - N coder block converts Thermometer code into equivalent N codes and it selects only one value concurrently correcting the bubble error. The second block converts the N codes to Binary code.

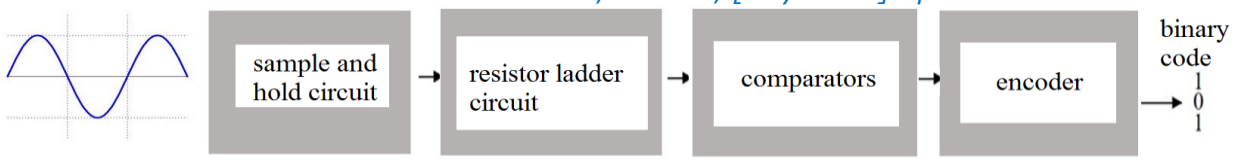


Figure 1. Blocks of flash ADC design

II. DIFFERENT TYPES OF ENCODER

2.1 ROM Type Encoder

ROM type encoder is the most common encoder used in flash ADCs. The design is simple and easy to formulate. But on the other hand it requires a clock that limits its processing speed. It consists of

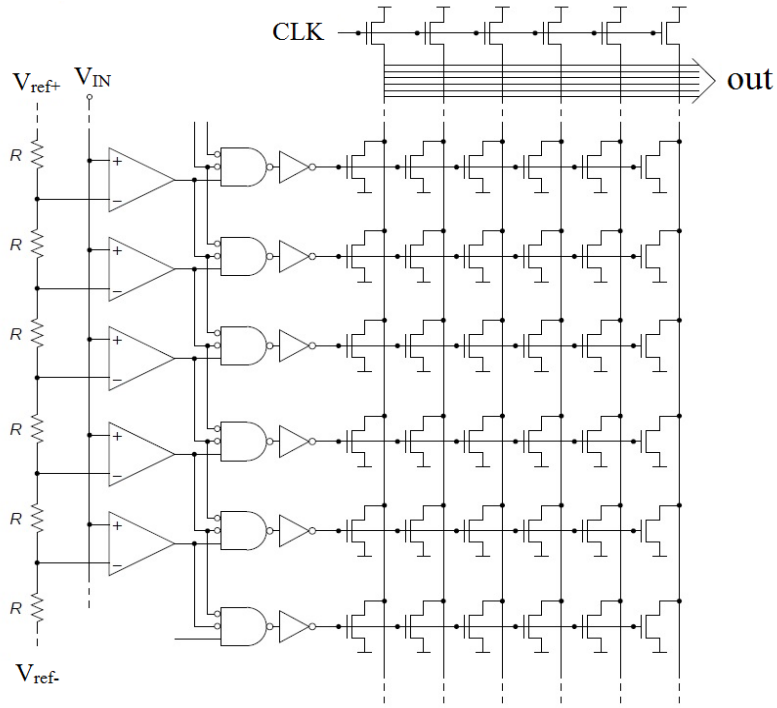


Figure 2. Flash ADC with ROM encoder

NMOS and resistors. INVERTERS are added to the NOR ROM encoder circuit to get the required output. Fig. 2 shows the ADC in which ROM encoder converts the Thermometer code (TC) - to - Binary code (BC).

2.2 Wallace Tree Encoder

Wallace tree encoder comprises of full adders. For every increase in 1 bit adds one more stage to the design. It can be used for correcting higher order bubble errors. Power dissipation is also less in case of Wallace tree encoder compared to ROM encoder. It is also called as ones counter. It counts the number of ones from the input and gives the output in binary code. It counts the number of ones in the input therefore the bubble errors of any order can be corrected.

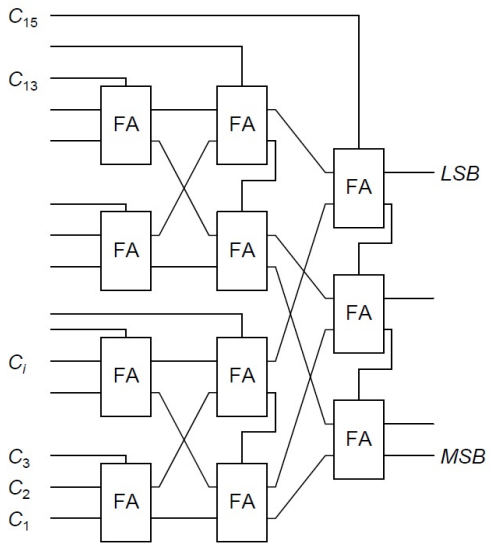


Figure 3. Four bit Wallace Tree Encoder

In low power applications this encoder can be used. It provides better results for any resolution but still speed is compromised. Fig. 3 shows the Wallace tree encoder. This encoder is also a fast multiplier.

2.3 Multiplexer based Decoder

Multiplexer based decoder consists of multiplexers (MUX).

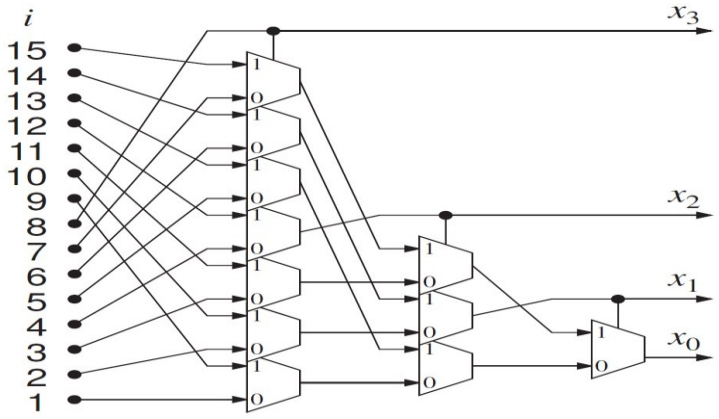


Figure 4. Four bit Multiplexer based decoder

It requires less hardware and its critical path is smaller than the Wallace tree decoder. Thus it is faster than Wallace tree and ROM type encoder. In Flash ADC, 2x1 Multiplexers is used in which a select line is connected to choose one of the inputs. Multiplexer (MUX) can be designed by using transmission gates or complementary pass transistors also. Fig. 4 shows a 4 bit Multiplexer based decoder.

2.4 Priority Encoder

Priority encoder is another type of encoder that encodes Thermometer Code - to - Binary Code. It is mainly used where interrupts are used. When used in Flash ADC it gives the priority to one of the comparator outputs and neglects rest outputs of the comparators. It reduces the calculations thus increasing the speed of the encoder as well as the Flash ADC. Fig. 5 shows a priority encoder and Table 1 shows its truth table.

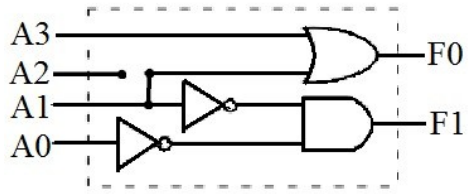


Figure 5. Four to two line Priority encoder

Table 1. Truth Table for 4 to 2 line Priority Encoder

Inputs				Outputs	
A3	A2	A1	A0	F1	F0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

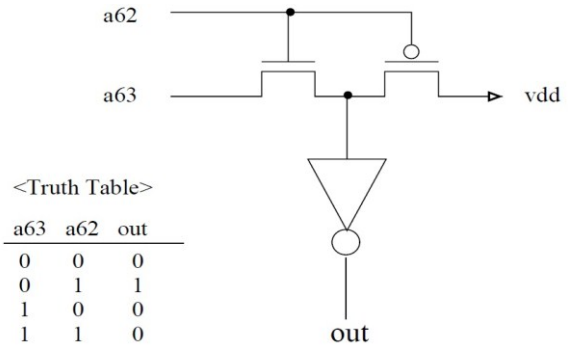
2.5 Fat Tree Encoder

Fat tree encoder is the fastest among these encoders and it has the advantage over the power dissipation. It not only provides high speed but also low power dissipation. It consists of two blocks:

Table 2. Truth Table for Binary Code Generator

Inputs							Outputs		
X6	X5	X4	X3	X2	X1	X0	B2	B1	B0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	1
0	1	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	1	1	1

1 - out of - N code generator block and binary code encoder block. The 1 - out of - N code generator block selects only one out of the rest inputs. And the second block coverts those outputs to binary code using combinational circuits. Table 2 shows the inputs and outputs of Binary code encoding block for a 3 - bit Flash ADC. Fig. 6 shows the Fat tree implementation logic [5].

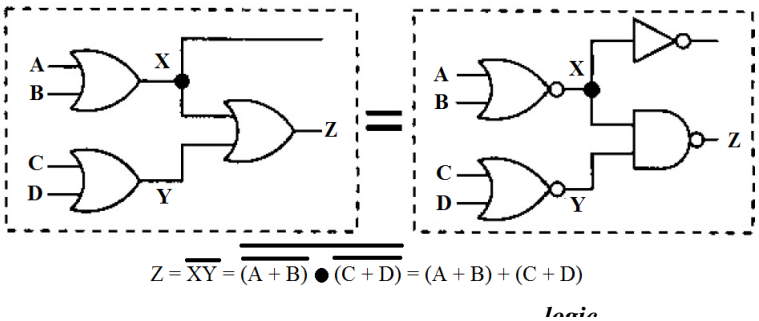


<Truth Table>

a63	a62	out
0	0	0
0	1	1
1	0	0
1	1	0

***Figure* 7. 1- out of - N code generator**

The signal delay for Fat tree encoder is $O(log_2N)$, for ROM is $O(N)$ and for Wallace tree encoder is $O(log_{1.5}N)$ [6]. Thus, concluding Fat tree to be the fastest.
Table 2 can be summarized as follows:



$Z = \overline{XY} = \overline{(A + B)} \bullet \overline{(C + D)} = (A + B) + (C + D)$

Figure 6. Fat tree implementation logic

$$B0 = X0 + X2 + X4 + X6 \quad (1)$$
$$B1 = X1 + X2 + X5 + X6 \quad (2)$$
$$B2 = X3 + X4 + X5 + X6 \quad (3)$$

Here X0, X1, X2, X3, X4, X5, X6 are inputs and B1, B2, B3 are outputs of encoder block.
Fig. 7 shows the logic diagram of 1 - out of - N code generator circuit. One of the inputs is connected to both NMOS and PMOS gate. Whereas the other input is connected to the source of the NMOS.

III. PROPOSED FAST ENCODER

The proposed encoder is a modified Fat tree encoder. One additional input to the bulk terminal reduces the addition of other transistors. Moreover an additional input to bulk terminal does the bubble error correction of order 1. There are three inputs from which one of the inputs is connected to NMOS and PMOS. One from the remaining two inputs is connected to the source and the other is connected to the bulk of the NMOS transistor. Fig. 8 shows the new 1- out of - N code generator circuit and Table 3 shows its truth table. The other block *i.e.* encoder block comprises of: NAND gates followed by NOR gates instead of NOR gates followed by NAND gates, thus reducing the number of NOR gates while increasing the number of NAND gates. Since the driving capacity of NAND is more than NOR and it is faster as well therefore this design would result in a faster encoder circuit.

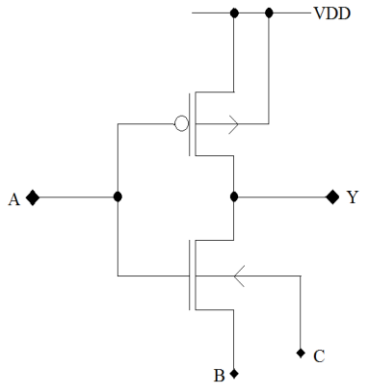


Figure 8. New 1 – out of – N code generator with bubble error correction

Table 3. Truth Table for New 1-out of-N Code Generator

Inputs			Outputs
C	B	A	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Table 4. Truth Table for Binary Code Generator

Inputs							Outputs		
X6	X5	X4	X3	X2	X1	X0	B2	B1	B0
1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	0	0	0	1
1	1	1	1	1	0	1	0	1	0
1	1	1	1	0	1	1	0	1	1
1	1	1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1

Table 4 shows the Binary code generator truth table and it can be summarized as follows:

$$B0 = X0 \bullet X2 \bullet X4 \bullet X6 \quad (4)$$

$$B1 = X1 \bullet X2 \bullet (X5 \bullet X6) \quad (5)$$

$$B2 = X3 \bullet X4 \bullet (X5 \bullet X6) \quad (6)$$

Here X0, X1, X2, X3, X4, X5, X6 are inputs and B0, B1, B2

IV. CONCLUSIONS

The ROM encoder is used mostly because it is simple to design but the requirement for clock limits the speed. On the other hand Wallace tree encoder, Multiplexer based decoder and priority encoders are fast but fat tree encoder is not only faster than these but also consumes less power.
The proposed encoder is an improved version of Fat tree encoder concurrently correcting the bubble error. Using just one NMOS and one PMOS, one can utilize all the terminals of an NMOS to provide required output. It consumes less power and it is faster. Fat tree encoder uses OR representation while the new encoder uses AND representation. This improvement helps increasing the speed and the driving capacity. With further transistor sizing the outputs can be better. Such a fast encoder can be used in Flash ADCs where speed and power is of concern. Also there are some common terms that can be used once instead of using it twice which reduces the encoder block terms and therefore consume less chip area.

REFERENCES

[1] K. Uyttenhove and M. Steyaert, "A 6-bit 1 GHz Acquisition Speed CMOS Flash ADC with Digital Error Correction," *IEEE Proc. Custom Integrated Circuits Conference*, pp. 249-252, 2000.
[2] F. Kaess, R. Kanan, B. Hochet and M. Declercq, "New Encoding Scheme For High-Speed Flash ADC's," *IEEE Proc. Circuits and Syst.*, vol. 1, pp. 5-8, June 1997.
[3] E. Säll and M. Vesterbacka, "A multiplexer based decoder for flash analog-to-digital converters," *Proc. TENCON 2004*, Nov. 21-24, 2004.
[4] M. Morris Mano, Michael D. Ciletti, "Digital Design", 4th Edition, Prentice Hall, 2006, ISBN 978-0-13-198924-5.
[5] I. Yoo, D. Lee, K. Choi, and A. Tangel, "Future-Ready Ultrafast 8Bit CMOS ADC for System-on-Chip Applications," *IEEE International ASIC/SOC Conference,* pp. 455-459, 2001.
[6] D. Lee, J. Yoo, K. Choi, and J. Ghaznavi, "Fat-tree encoder design for ultrahigh speed flash analog-to-digital converters," in Proc. IEEE Midwest Symp. Circuits Syst, pp. 233–236, Aug 2002.
[7] Sudakar S. Chauhan, S. Manabala, S.C. Bose, and R. Chandel, "A New Approach To Design Low Power CMOS Flash A/D Converter", *International Journal of VLSI design & Communication Systems (VLSICS)*, Vol.2, No.2, June 2011, p.100.
[8] Samad Sheikhaei, Shahriar Mirabbasi, and Andre Ivanov, "A 0.35µm CMOS Comparator Circuit for High-Speed ADC Applications" *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on,* Vol. 6, p. 6134-6137.
[9] Pradeep Kumar, Amit Kolhe, "Design & Implementation of Low Power 3-bit Flash ADC in 0.18µm CMOS", *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN: 2231-2307, Volume-1, Issue-5. November 2011.
[10] Phillip E. Allen and Douglas R. Holberg, CMOS Analog Circuit Design, Oxford University Press, Inc. 2002.
[11] R. J. Baker, H. W. Li and D.E Boyce, CMOS Circuit Design, Layout And Simulation, New York, *IEEE Press*, 2008.

