A High Speed Low Power Encoder for a 5 Bit Flash ADC

George Tom Varghese and K. K. Mahapatra
Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela, India
E-mail: georgetomvarghese@gmail.com

***ABSTRACT*- The present investigation proposes an efficient low power encoding scheme intended for a 5GS/s 5 bit flash analog to digital converter. The designing of a thermometer code to binary code is one of the challenging issues in the design of a high speed low power flash ADC. An encoder circuit in this paper translates the thermometer code into the intermediate gray code to reduce the effects of bubble errors. To maintain the high speed with low power dissipation, the implementation of the encoder through pseudo NMOS logic is presented. The proposed encoder is designed using 90nm technology in 1.2 V power supply using CADENCE tool. The simulation results shown for a sampling frequency of 5GHz and the average power dissipation of the encoder is 0.3149 mW which is very less in comparison with current mode logic encoder implementation.**

Keywords – Analog to digital converter, Flash ADC, Pseudo NMOS logic, Pseudo Dynamic CMOS logic.

I. INTRODUCTION

The flash ADC is known for its fastest speed compared to other ADC architectures. Therefore, it is used for high-speed and very large bandwidth applications such as radar processing, digital oscilloscopes, high-density disk drives, and so on. The flash ADC is also known as the parallel ADC because of its parallel architecture.

Figure 1 illustrates a typical flash ADC block diagram. As shown in Fig. 1, this architecture needs $2^n - 1$ comparators for a n-bit ADC; for example, a set of 31 comparators is used for 5-bit flash ADC. Each comparator has a reference voltage that is provided by an external reference source. These reference voltages are equally spaced by V_{LSB} from the largest reference voltage to the smallest reference voltage V1. An analog input is connected to all comparators so that each comparator output is produced in one cycle. The digital output of the set of comparators is called the thermometer code and is being converted to gray code initially (for minimizing the bubble errors) and further changed into a binary code through the encoder [1]. However, the flash ADC needs a large number of comparators as the resolution increases. For instance, a 6-bit flash ADC needs 63 comparators, but 1023 comparators are needed for a 10-bit flash ADC. This exponentially increasing number of comparators requires a large die size and a large amount of power consumption [3].

The proposed encoder is designed using pseudo NMOS logic style for achieving highest sampling frequency of 5GS/s and low power dissipation with respect to current mode logic. The design of the encoder with detailed description is presented in section II. The implementation of the encoder using pseudo NMOS logic style is presented in section III. Simulation results and conclusion are provided in the subsequent sections.

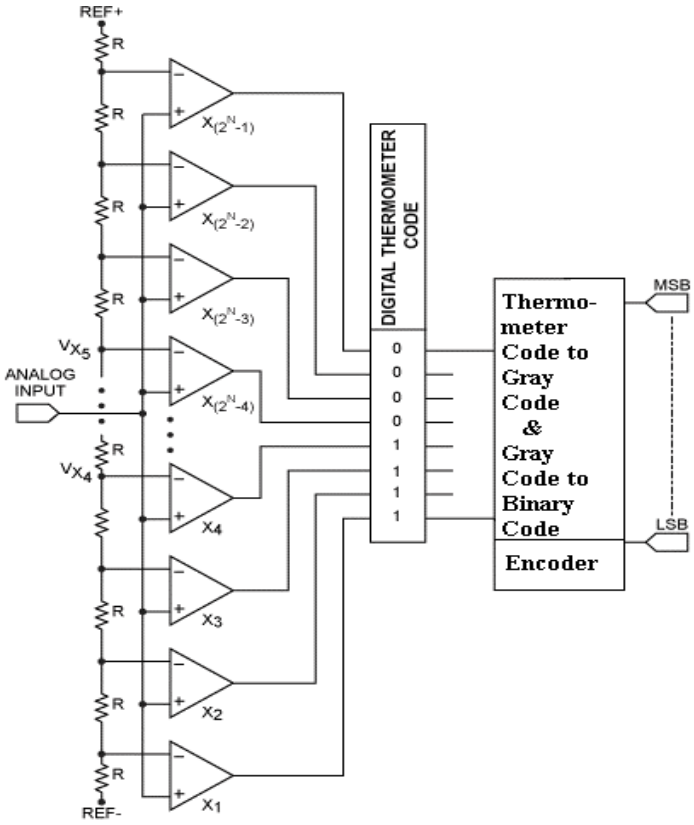


Fig. 1 Flash ADC Block Diagram

II. DESIGN OF THE PROPOSED ENCODER

Conversion of the thermometer code output to binary code is one of the bottlenecks in high speed flash ADC design [2]. The bubble error usually results from timing differences between clock and signal lines and it is a situation where a '1' is found above zero in thermometer code. For very fast input signals, small timing difference can cause bubbles in the output code. Depending on the number of successive zeroes, the bubbles are characterized as of first, second and higher orders. To reduce the effect of bubbles in thermometer code, one of the widely used methods is to convert the thermometer code to gray code [5, 6]. The truth table corresponding to 5 bit gray code is presented in Table1. The relationship between thermometer code, gray code and binary code is given below.

$G4 = T_{16}$

$G3 = T_8.\overline{T_{24}}$

$G2 = T_4.\overline{T_{12}} + T_{20}.\overline{T_{28}}$

$G1 = T_2.\overline{T_6} + T_{10}.\overline{T_{14}} + T_{18}.\overline{T_{22}} + T_{26}.\overline{T_{30}}$ (A)

$G0 = T_1.\overline{T_3} + T_5.\overline{T_7} + T_9.\overline{T_{11}} + T_{13}.\overline{T_{15}} + T_{17}.\overline{T_{19}} + T_{21}.\overline{T_{23}} + T_{25}.\overline{T_{27}} + T_{29}.\overline{T_{31}}$

$B4 = G4$

$B3 = G3 \text{ XOR } B4$

$B2 = G2 \text{ XOR } B3$

$B1 = G1 \text{ XOR } B2$ (B)

$B0 = G0 \text{ XOR } B1$

The equations for this encoder are derived from the truth table provided in Table 1.

G4	G3	G2	G1	G0	Thermometer Code
0	0	0	0	0	00000000000000000000000000000000
0	0	0	0	1	00000000000000000000000000000001
0	0	0	1	1	00000000000000000000000000000011
0	0	0	1	0	00000000000000000000000000000111
0	0	1	1	0	00000000000000000000000000001111
0	0	1	1	1	00000000000000000000000000011111
0	0	1	0	1	00000000000000000000000000111111
0	0	1	0	0	00000000000000000000000001111111
0	1	1	0	0	00000000000000000000000011111111
0	1	1	0	1	00000000000000000000000111111111
0	1	1	1	1	00000000000000000000001111111111
0	1	1	1	0	00000000000000000000011111111111
0	1	0	1	0	00000000000000000000111111111111
0	1	0	1	1	00000000000000000001111111111111
0	1	0	0	1	00000000000000000011111111111111
0	1	0	0	0	00000000000000000111111111111111
1	1	0	0	0	00000000000000001111111111111111
1	1	0	0	1	00000000000000011111111111111111
1	1	0	1	1	00000000000000111111111111111111
1	1	0	1	0	00000000000001111111111111111111
1	1	1	1	0	00000000000011111111111111111111
1	1	1	1	1	00000000001111111111111111111111
1	1	1	0	1	00000000011111111111111111111111
1	1	1	0	0	00000000111111111111111111111111
1	0	1	0	0	00000001111111111111111111111111
1	0	1	0	1	00000011111111111111111111111111
1	0	1	1	1	00000111111111111111111111111111
1	0	1	1	0	00001111111111111111111111111111
1	0	0	1	0	00011111111111111111111111111111
1	0	0	1	1	00111111111111111111111111111111
1	0	0	0	1	01111111111111111111111111111111
1	0	0	0	0	11111111111111111111111111111111

Table1. Gray Code Encoder Truth Table

III. IMPLEMENTATION OF PROPOSED ENCODER

There are different logic styles to implement the encoder design. Generally the implementation will be done using static CMOS logic style. The advantage of static CMOS logic style is that it is having the lowest power consumption with a lower speed. So for achieving a low power with high speed, other logic styles are preferred. Here the design is implemented using logic style called pseudo NMOS logic[8].

The pseudo NMOS logic circuit consists of a PMOS transistor with gate connected to ground, a bunch of NMOS transistors for the implementation of the logic style in the pull down network and an inverter. For the implementation of a specific logic circuit with N inputs, pseudo NMOS logic requires N+1 transistors instead of 2N transistors in comparison with static CMOS logic. Pseudo NMOS logic is an attempt to reduce the number of transistors with extra power dissipation and reduced robustness.

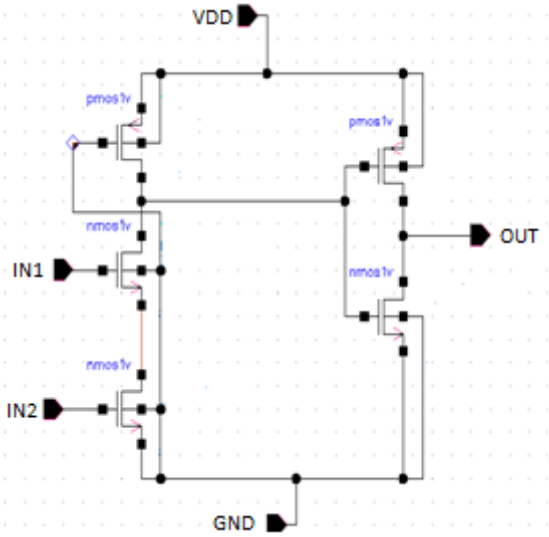


Fig. 2 Schematic of two input AND Gate Using Pseudo NMOS Logic

The basic structure of two inputs and gate using pseudo NMOS logic style is shown in Fig. 2. The PMOS transistor in the pull up network is connected to ground that will make the pull up network to be pulled on all the time. The output will be evaluated conditionally depending upon the value of the inputs in the pull down network. The inverter on the output transforms the inverted gate to non inverted gate. Since the voltage swing on the output and the overall functionality of the gate depend on the ratio of the NMOS and PMOS sizes, the transistor sizing is crucial in the implementation design.

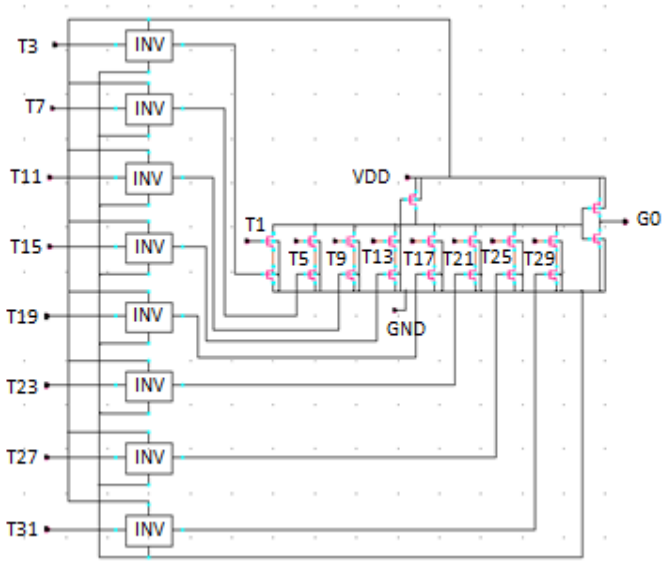
The disadvantage with pseudo NMOS logic is that it has static power consumption. (The power dissipation occurs when a direct current flows between VDD and ground. That is when both pull up and pull down networks are switched on simultaneously). The nominal high output voltage of (V_{OH}) of pseudo NMOS logic is VDD (Assuming that the pull down network is switched off) and the nominal low output voltage (VOL) is not zero. This will results in reduced noise margins. For the implementation of the positive logics (eg: AND, OR gate) a static CMOS inverter is added at the output side. This will improve the noise margin of the circuit.In spite of static power dissipation, the pseudo NMOS logic consumes less

amount of power because of the reduced number of transistors and the absence of other components (resistors) used for the implementation in comparison with current mode logic.

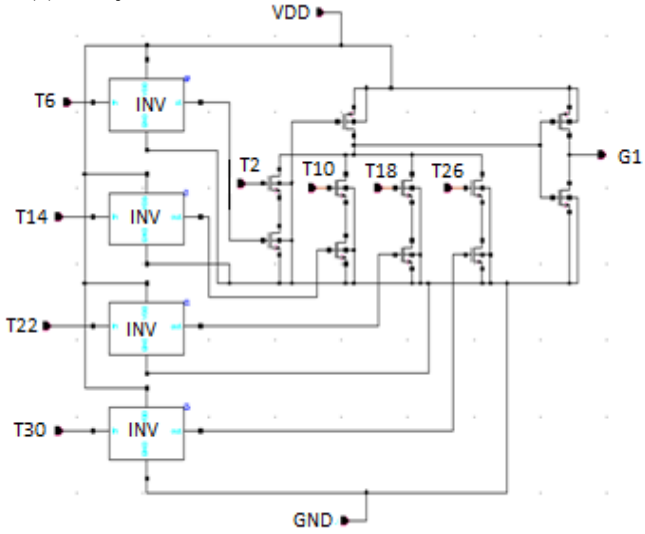
The schematic of the gray code encoder for each bit is designed using the proposed logic is shown in Fig. 3. With the help of XOR gate, the gray code will be converted to binary code. The schematic of the two input XOR gate is shown in Fig. 4. The XOR gate is implemented with a pseudo dynamic CMOS logic [7] to maintain the synchronization with the clock. Pseudo dynamic CMOS logic circuit consists of a PMOS transistor with gate connected to clock, a bunch of NMOS transistors for the implementation of the logic style in the pull down network and an inverter.The transistor sizes are given in Table 2.

(W/L) PMOS	300nm/100nm
(W/L) NMOS	120nm/100nm

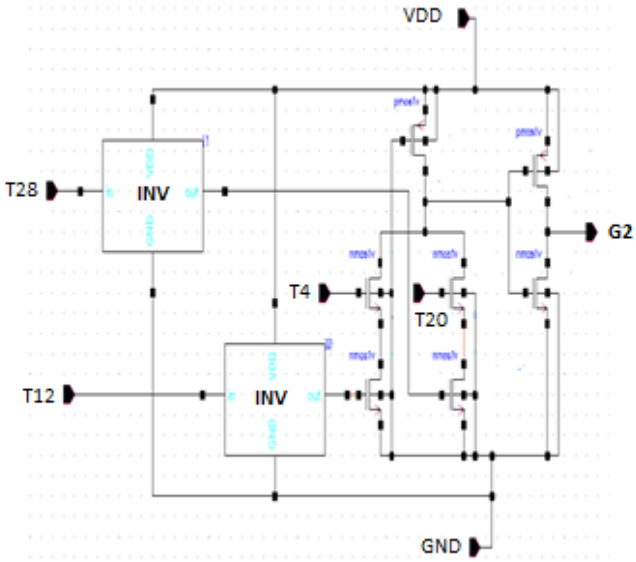
Table 2. Transistor Sizes



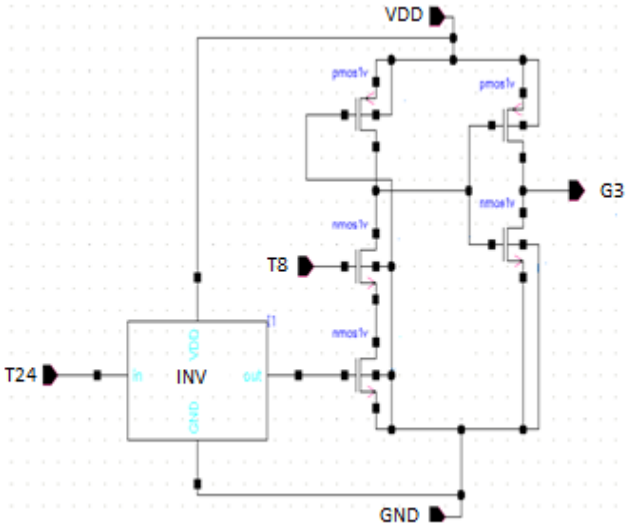
(a) Gray Code Bit0 Generation Circuit



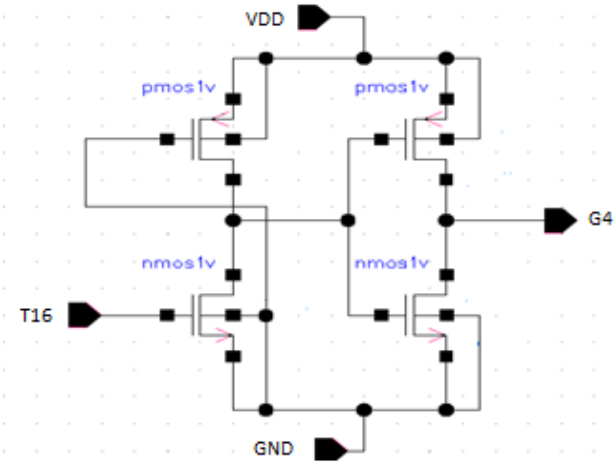
(b) Gray Code Bit1 Generation Circuit



(c) Gray Code Bit2 Generation Circuit



(d) Gray Code Bit3 Generation Circuit



(e) Gray Code Bit4 Generation Circuit

Fig. 3 Schematic of Gray Code Encoder using Pseudo NMOS Logic

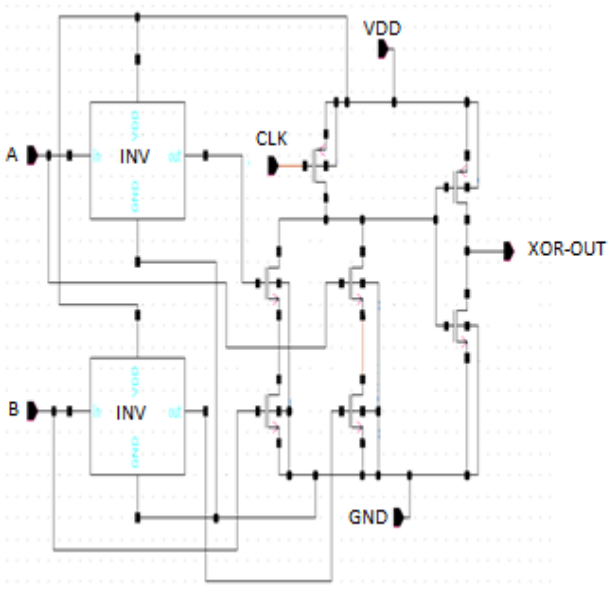


Fig. 4 Schematic of 2 Input XOR Gate using pseudo dynamic logic

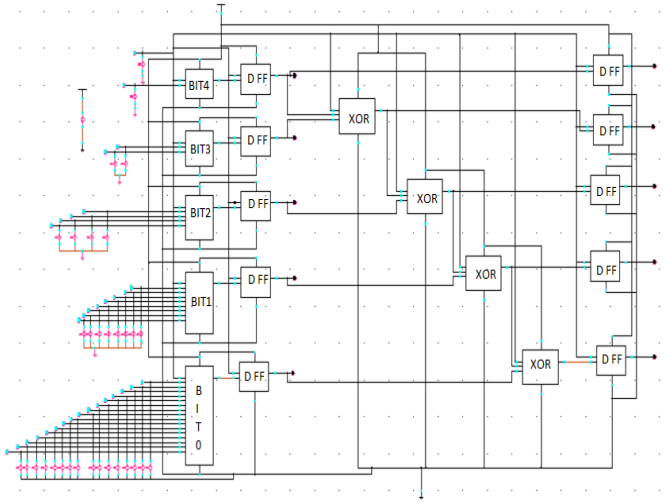


Fig5. Schematic of Total Encoder

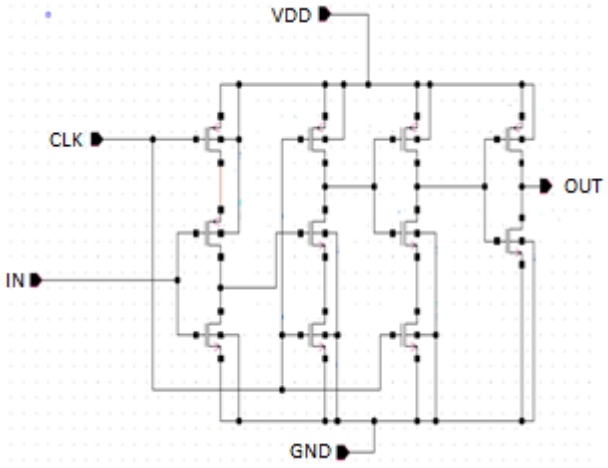


Fig. 6 Schematic of D Flip-Flop

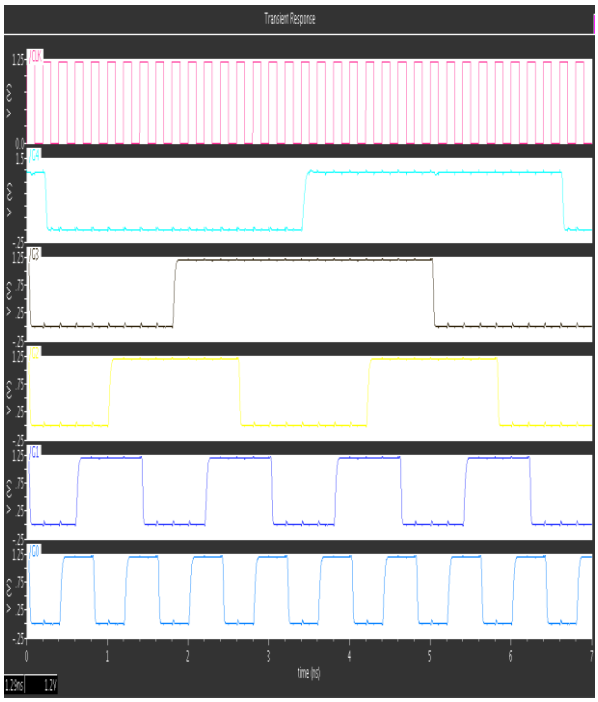


Fig7. Simulation of Gray Code Encoder

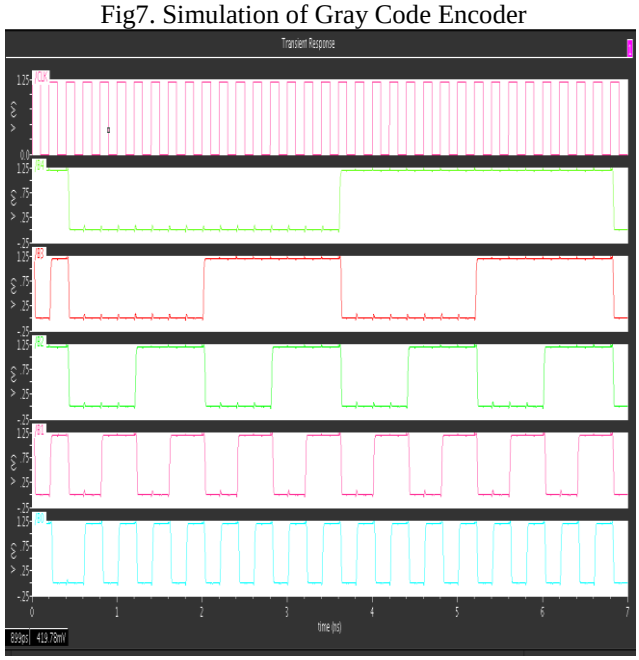


Fig8. Simulation of Binary Code Encoder

IV. SIMULATION RESULTS AND DISCUSSION

The encoder is designed as shown in Fig. 5 and tested using all the input combinations from the truth table and verified. At the output of the gray code a D flip-flop [Fig. 6] is added to get the undistorted waveform. The gray code output is shown in the Fig. 7. As derived from the equation (B), the gray code will be converted to binary code and the simulation results are shown in Fig. 8. The summary of the encoder simulation results is shown in the table. In most of the 5 bit flash

ADC's, the maximum sampling frequency achieved can be up to 3.5GS/s [4]. With the use of proposed encoder maximum sampling frequency of 5GS/s can be achieved. The average power dissipation of the proposed encoder is 0.3149 mW. The results are presented and compared with current mode logic encoder [2] in the Table 3.

Results	Current Mode Logic Encoder	Proposed Encoder
Architecture	Flash	Flash
Resolution	4 bits	5 bits
Technology	180 nm	90 nm
Sampling Frequency	5GHz	5GHz
Vdd	1.8 V	1.2 V
Current	2.22 mA	0.2624 mA
Power Dissipation	4 mW	0.3149 mW

Table3. Summary of Proposed Encoder

The results show that the new design consumes less power than the current mode logic encoder[2]. The power dissipation is reduced because of the usage of the reduced number of transistors for the implementation of the logic.In comparison with the static CMOS logic encoder, the proposed encoder contains a reduced number of transistors, thereby reducing the cost of the encoder also.

V. CONCLUSION

The speed and power of an encoder play an important role in the design of flash ADC. The proposed encoder uses a pseudo NMOS logic to reduce the power of the encoder. The encoder is designed and simulated using 90 nm technology using CADENCE tool. The encoder which is operating at 5 GHz, consumes 0.3149 mW from 1.2 V supply. The performance of the encoder makes it suitable for the design of high speed low power flash ADC.

REFERENCES

[1]. D.Lee, J.Yoo, K.Choi and J. Ghaznavi, "Fat-tree encoder design for ultrahigh speed flash analog to digital converters" I proc. IEEE Midwest Symp. Circuits Syst, pp 233-236, Aug 2002

[2]. S. Sheikhaei, S. Mirabbasi, A. Ivanov, "An Encoder for a 5GS/s 4bit flash A/D converter in 0.18um CMOS", Canadian Conference on Electrical and Computer Engineering, pp 698-701, May 2005.

[3]. R.Baker, H.W.Li, and D.E. Boyce, CMOS Circuit Design, Layout and Simulation. Prentice Hall 2000.

[4]. Sunghyun Park, Yorgos Palaskas, Ashoke Ravi, Ralph.E.Bishop, and Michael P. Flynn, " A 3.5 GS/s 5-b Flash ADC in 90nm CMOS", IEEE Custom Integrated Circuits Conference 2006.

[5].Niket Agrawal, Roy Paily, "An Improved ROM Architecture for Bubble eroor Suppression in High Speed Flash ADCs", Annual IEEE Conference, pp 1-5,2005.

[6]. Mustafijur Rahman, K.L. Baishnab, F.A. Talukdar, "A Novel ROM Architecture for Reducing Bubble and Meta-stability Errors in High Speed Flash ADCs", 20th International Conference on Electronics, Communications and Computer, pp 15-19, 2010.

[7].Vinayashree Hiremath, " Design of High Speed ADC" , M.S. Thesis, Wright State University, 2010.

[8]. Jan M Rabaey, Anantha Chandrakasan, Borivoje Nikolic, "Digital Integrated Circuits, a design perspective", second edition, Prentice Hall 2011.