# A High Speed Encoder for a 5GS/s 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 high speed 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 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 increase the speed of the encoder, the implementation of the encoder through pseudo dynamic CMOS logic is presented. The proposed encoder is designed using 90nm technology at 1.2V power supply using CADENCE tool. The simulation results shown for a sampling frequency of 5GHz and the average power dissipation of the encoder is 1.919mW.**

**Keywords – Analog to digital converter, Flash ADC, 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 Figure 1, this architecture needs $2^n - 1$ comparators for an n-bit ADC; for example, a set of 31 comparators are 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 speed of the encoder is crucial in the design of flash ADC. The proposed encoder is designed using a new logic style called pseudo dynamic logic style for achieving highest sampling frequency of 5GS/s. The design of the encoder with detailed description is presented in section II. The implementation of the encoder using pseudo dynamic logic style is presented in section III. Simulation results and conclusion are provided in the subsequent sections.

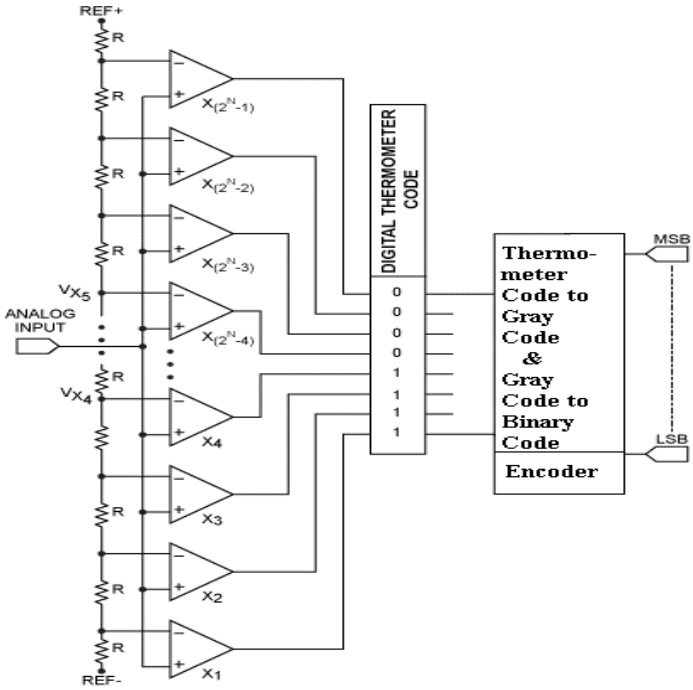



Fig1. Flash ADC Block Diagram

## 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 difference 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 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.

$$G4 = T_{16}$$

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

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

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

$G0 = T_1.\overline{T_3} + T_5.\overline{T_7} + T_9.\overline{T_{11}} + T_{13}.\overline{T_{15}} + T_{17}.\overline{T_{19}} +$

$T_{21}.\overline{T_{23}} + T_{25}.\overline{T_{27}} + T_{29}.\overline{T_{31}}$

$B4 = G4$

$B3 = G3 \text{ XOR } B4$

$B2 = G2 \text{ XOR } B3$

$B1 = G1 \text{ XOR } B2$ (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 | 00000000000111111111111111111111 |
| 1 | 1 | 1 | 0 | 1 | 00000000001111111111111111111111 |
| 1 | 1 | 1 | 0 | 0 | 00000000011111111111111111111111 |
| 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 higher speed, other logic styles are preferred. Here the design is implemented using a new logic style called pseudo dynamic CMOS logic [7].

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. In comparison with dynamic CMOS logic, pseudo dynamic CMOS logic does not require an NMOS evaluation transistor in series with the NMOS block, since the inputs to this circuit are properly synchronized.

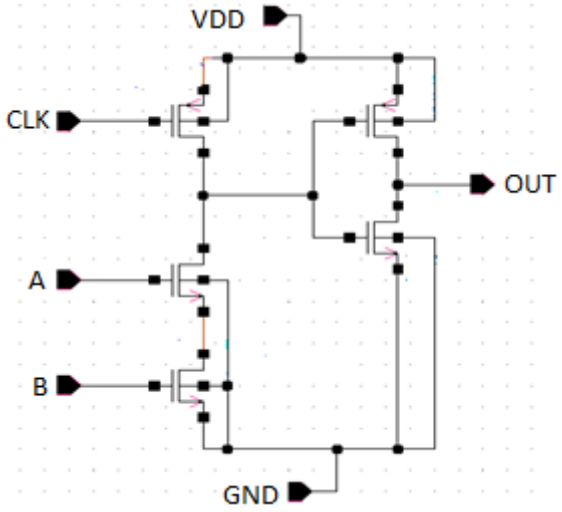
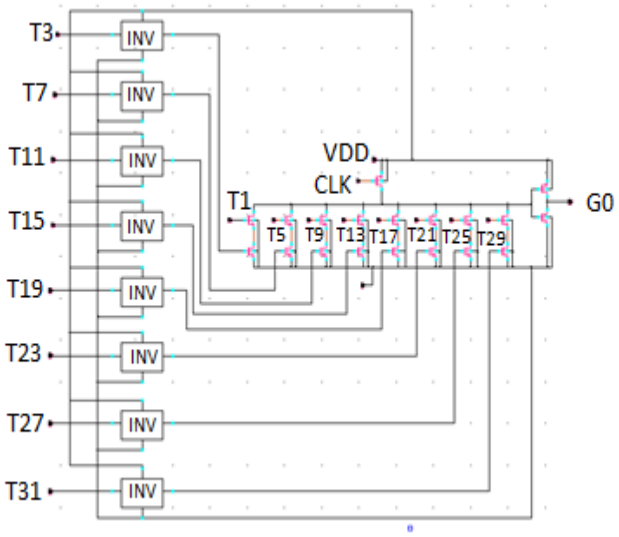
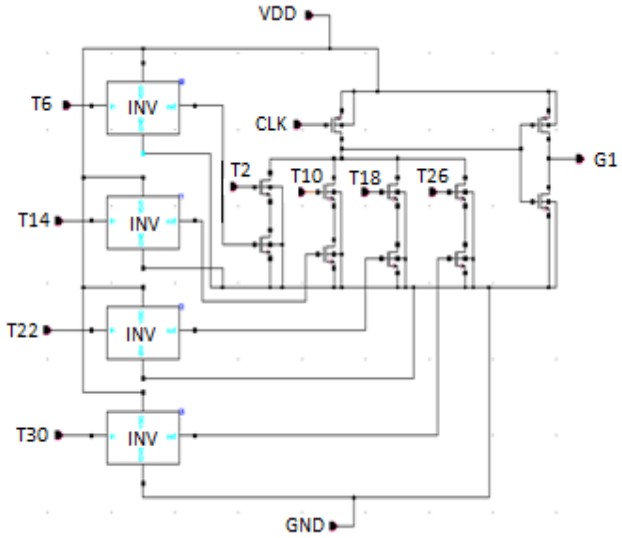



Fig2. Schematic of 2 input AND Gate Using Pseudo Dynamic CMOS Logic

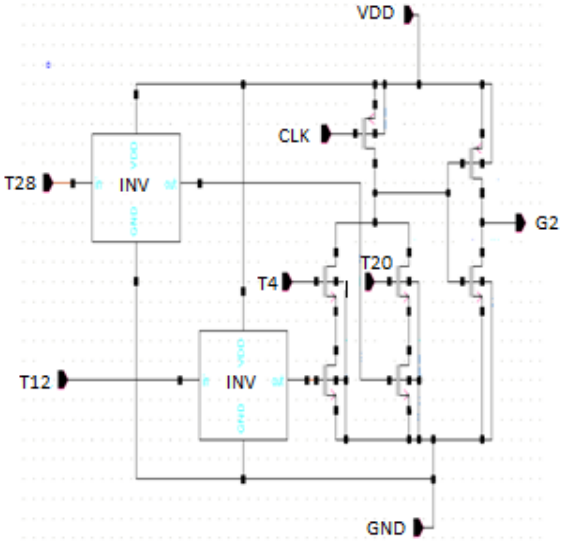
During the pre-charging phase of the dynamic CMOS, the PMOS transistor is on and the NMOS network will be switched off and in the evaluation mode the PMOS transistor is switched off and NMOS network will be conditionally switched on based on the inputs. Whereas in pseudo dynamic CMOS logic, the NMOS evaluation block can be enabled during pre-charging phase, since it has no influence on the output voltage due to the presence of inverter. The basic structure of the pseudo dynamic logic (2 input and gate) is shown in figure 2. During pre-charging phase, if NMOS logic block is enabled, the output comes to settle down to a value decided by a resistive divider of the PMOS pull-up and NMOS logic block. It must be ensured that the voltage at the inverter input does not below $V_{IH}$ (Minimum input voltage applied at the input which will be taken as a logic one). The size of PMOS and NMOS transistors must be chosen in such a way that the proper functioning of the circuit will be maintained. Consider the case with NMOS logic is enabled and clock is low. With proper size of the transistor, the voltage at the input of the inverter will be less than $V_{IH,}$ thus maintaining logic at the output of the inverter. If the PMOS transistor size is increased beyond a specific limit, the intermediate voltage exceeds $V_{IH}$ thereby pulling down the inverter output. Finally it will result in the malfunctioning of the circuit. So the transistor sizing is crucial in the implementation design. The disadvantage with the pseudo dynamic CMOS logic is that it has static power dissipation (The power dissipation occurs when a direct current flows between VDD and ground. That is when both pull up and pull own networks are switched on simultaneously). The schematic of the gray code encoder for each bit is designed using proposed circuit is shown in figure 3. With the help of xor gate, the gray code will be converted to binary code. The schematic of 2 input xor is shown in fig 4.



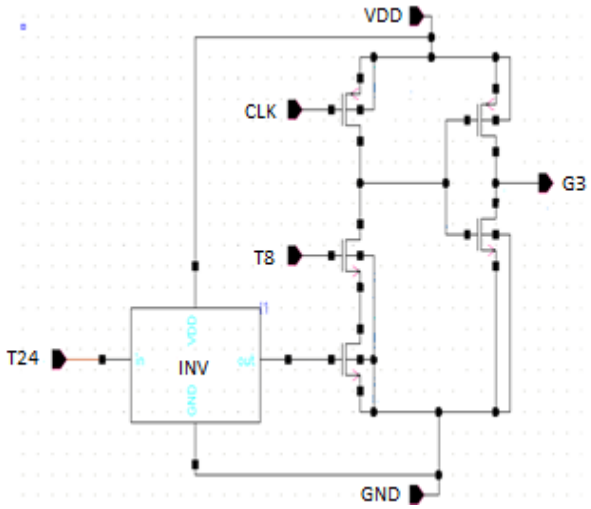

(a) Gray Code Bit0 Generation Circuit



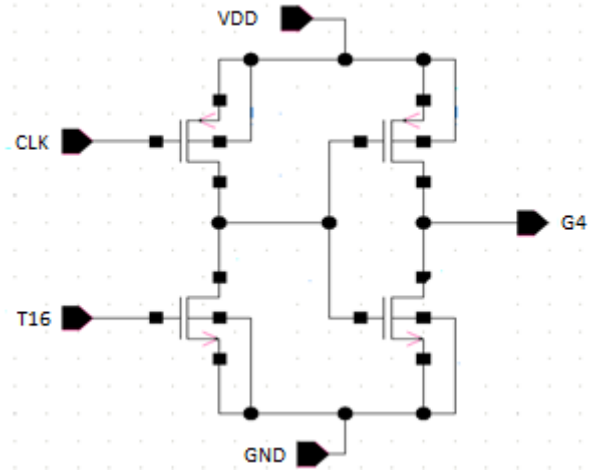

(b) Gray Code Bit1 Generation Circuit




(c) Gray Code Bit2 Generation Circuit




(d) Gray Code Bit3 Generation Circuit




(e) Gray Code Bit4 Generation Circuit

Fig3. Schematic of Gray Code Encoder using Pseudo Dynamic CMOS Logic

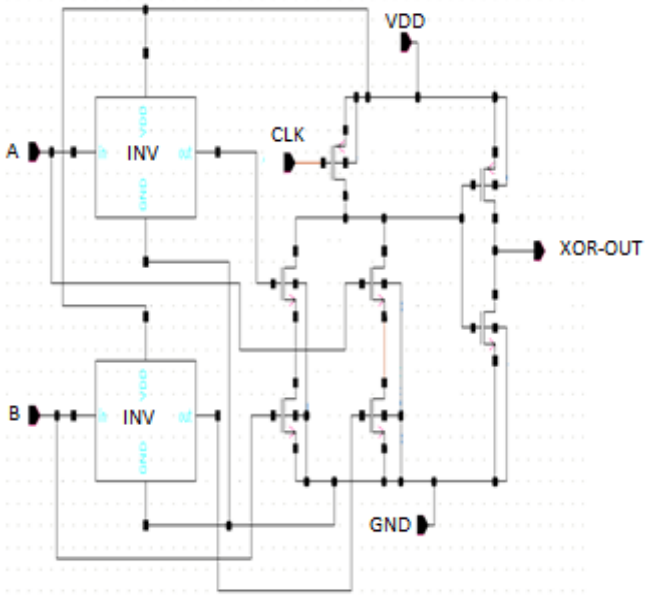



Fig4. Schematic of 2 Input XOR Gate

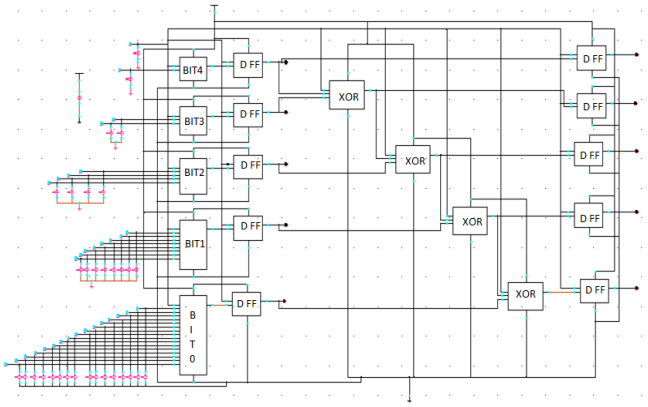



Fig5. Schematic of Total Encoder

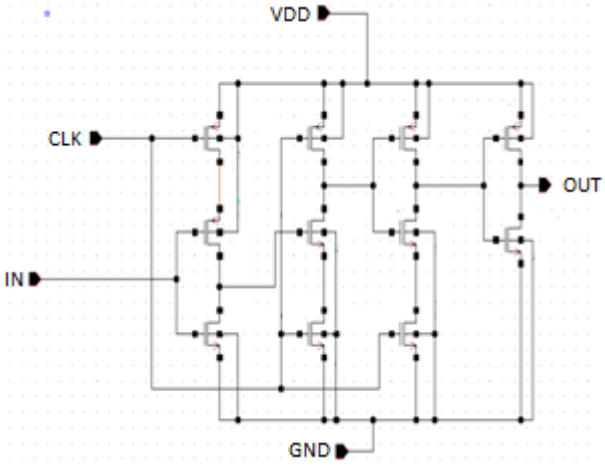



Fig6. Schematic of D Flip-Flop

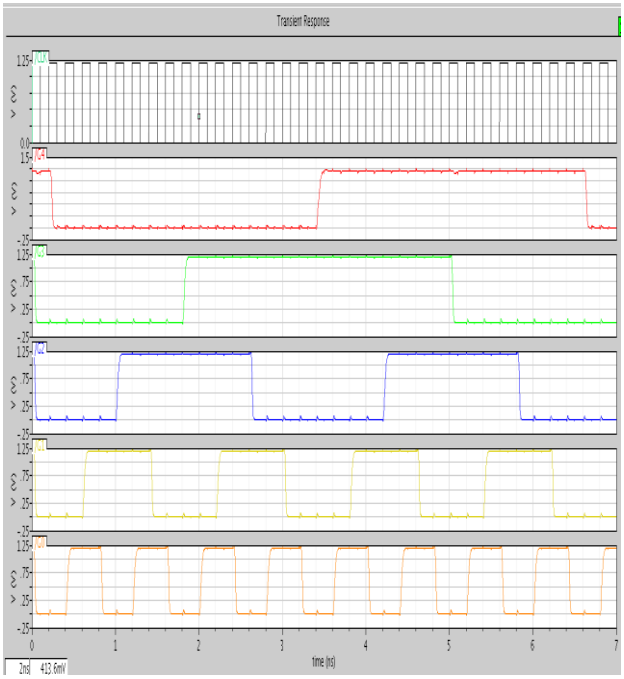



Fig7. Simulation of Gray Code Encoder

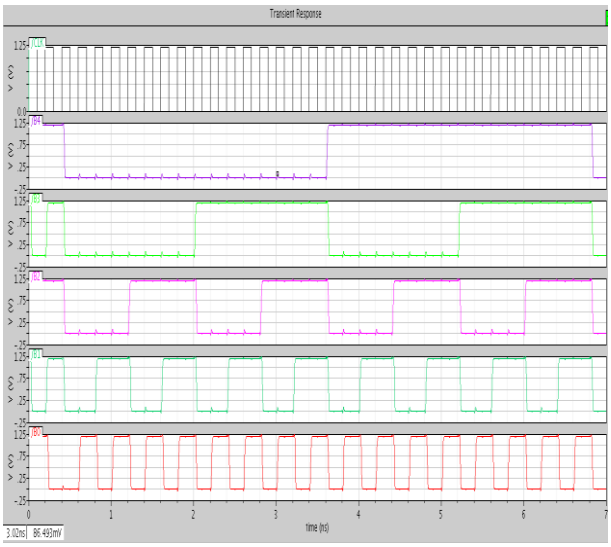



Fig8. Simulation of Binary Code Encoder

## IV. SIMULATION RESULTS AND DISCUSSION

The encoder is designed as shown in fig5 and tested using all the input combinations from the truth table and verified. At the output of the gray code a D flip-flop [fig6] is added to get the undistorted waveform. The gray code output is shown in the figure7. As derived from the equation (B), the gray code will be converted to binary code and the simulation results are shown in figure8. 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 1.919 mW. The results are presented in the table2.

| Results | Proposed Encoder |
|---|---|
| Architecture | Flash |
| Resolution | 5 bits |
| Technology | 90 nm |
| Sampling Frequency | 5GHz |
| Vdd | 1.2V |
| Current | 1.599 mA |
| Power Dissipation | 1.919 mW |

Table2. Summary of Proposed Encoder

The results show that the new design is much faster than the other encoders. 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 of an encoder plays an important role in the design of flash ADC. The proposed encoder uses a new logic style called pseudo dynamic CMOS logic to improve the speed 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 1.919 mW from 1.2 V supply. The performance of the encoder makes it suitable for the design of high speed 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", Cnadian 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.