A Novel Design of LPHS Encoder for 5-bit Flash ADC

Gangavarapu Amarendra[1], P. Anjaneyulu[2], D.V.SuryaChandraBabu[3]
[1,2,3]*Assistant professor, Department of ECE, MLRIT, Hyderabad, India*

***Abstract*— This paper describes the designing of a 5GS/s 5-bit flash analog to digital converter with Low Power, High Speed (LPHS) efficient encoding scheme. The Conversion of thermometer code to binary code is one of the challenging task in the design of Flash ADC. The proposed encoder circuit translates thermometer code into intermediate gray code to reduce the effects of bubble errors. Pseudo dynamic CMOS logic style is used to achieve high speed in the proposed encoder. The proposed encoder is designed using 45nm technology at 0.9 V supply voltage using CADENCE tool. The simulation results were shown for a sampling frequency of 5GHz and average power dissipation of the encoder is 388µW.**

***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 drivers, and so on. The speed of the encoder is crucial in the design flash ADC. The proposed encoder is designed using pseudo dynamic logic style [1]. Obviously the usage of clocking in the circuit achieves highest sampling frequency 5GS/s.

Flash ADC resolution and power should have trade off. However, the flash ADC needs a large number of comparators as the resolution increases. For instance, a 6-bit flash ADC require 63 comparators and 64 resistors, a 10-bit flash ADC require 1023 comparators and 1024 resistors. This exponential increase in both comparators and resistors results in large die size and a large amount of power consumption [2]. The flash ADC is also known as the parallel ADC because of its parallel architecture.

Figure 1 depicts a typical flash ADC block diagram. It requires 2^n-1 comparators and 2^n resistors; for example 31 comparators and 32 resistors are required for 5-bit flash ADC. Each comparator has a reference voltage provided by an external reference source. All these reference voltages are equally spaced by V_{LSB} from highest to smallest reference voltage. The other input of the comparator is used to apply the analog voltage. Comparator outputs the compared result for every single cycle. The digital output from the set of comparators is called the thermometer code and is being converted into gray code and further changed into binary code through the encoder [3].

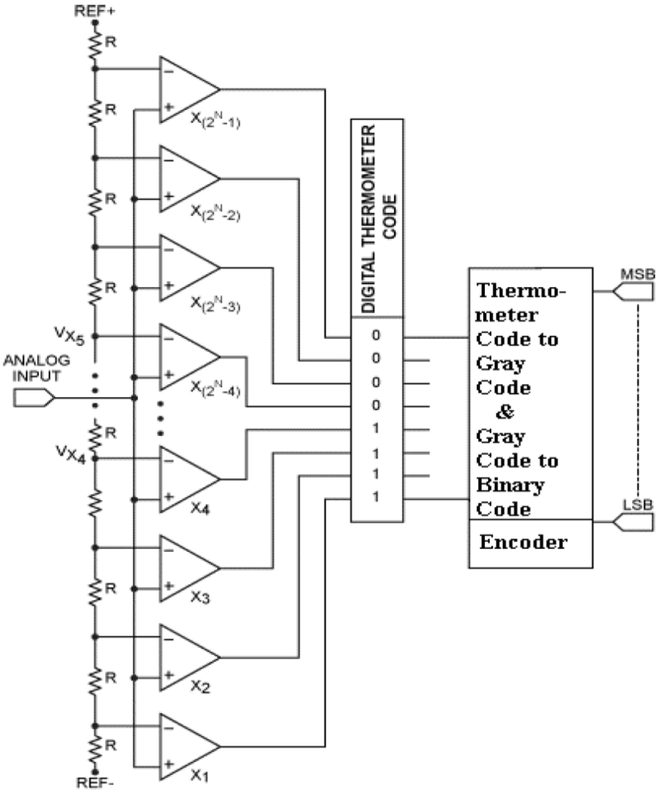

Fig. 1. Flash ADC Block Diagram.

Section II describes the architecture of LPHS encoder with detailed description, Section III deals with the implementation of the encoder using pseudo dynamic logic style, Section IV presents the simulation results followed by conclusion at the end.

II. DESIGN OF LPHS ENCODER

Achieving high encoding speed and conversion of thermometer code output from comparators to binary code without any bubble errors are the challenging issues in designing high speed flash ADC [1, 3].

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. Depending on the number of successive zeroes, the bubbles are characterized as of first, second and higher orders. The effect of bubble errors are greatly reduced by converting thermometer code to intermediate gray code and then to binary code instead of thermometer to binary code [6, 7].

The truth table corresponding to 5-bit gray code is presented in table I and truth table corresponding to binary code is presented in table II. The relationship between thermometer code to gray code and gray code to binary code is given below. These equations are derived from the table I and II.

$$G_0 = T_1.\overline{T_3} + T_5.\overline{T_7} + T_9.\overline{T_{11}} + T_{13}.\overline{T_{15}} + T_{17}.\overline{T_{19}} + T_{21}.\overline{T_{23}} + T_{25}.\overline{T_{27}} + T_{29}.\overline{T_{31}}$$

$$G_1 = T_2.\overline{T_6} + T_{10}.\overline{T_{14}} + T_{18}.\overline{T_{22}} + T_{26}.\overline{T_{30}}$$

$$G_2 = T_4.\overline{T_{12}} + T_{20}.\overline{T_{28}}$$

$$G_3 = T_8.\overline{T_{24}}$$

$$G_4 = T_{16}$$

$$B_4 = G_4$$

$$B_3 = G_3 \text{ XOR } B_4$$

$$B_2 = G_2 \text{ XOR } B_3$$

$$B_1 = G_1 \text{ XOR } B_2$$

$$B_0 = G_0 \text{ XOR } B_1$$

G_4	G_3	G_2	G_1	G_0	T_{30}..............................T_0
0	0	0	0	0	0000000000000000000000000000000
0	0	0	0	1	0000000000000000000000000000001
0	0	0	1	1	0000000000000000000000000000011
0	0	0	1	0	0000000000000000000000000000111
0	0	1	1	0	0000000000000000000000000001111
0	0	1	1	1	0000000000000000000000000011111
0	0	1	0	1	0000000000000000000000000111111
0	0	1	0	0	0000000000000000000000001111111
0	1	1	0	0	0000000000000000000000011111111
0	1	1	0	1	0000000000000000000000111111111
0	1	1	1	1	0000000000000000000001111111111
0	1	1	1	0	0000000000000000000011111111111
0	1	0	1	0	0000000000000000000111111111111
0	1	0	1	1	0000000000000000001111111111111
0	1	0	0	1	0000000000000000011111111111111
0	1	0	0	0	0000000000000000111111111111111
1	1	0	0	0	0000000000000001111111111111111
1	1	0	0	1	0000000000000011111111111111111
1	1	0	1	1	0000000000000111111111111111111
1	1	0	1	0	0000000000001111111111111111111
1	1	1	1	0	0000000000011111111111111111111
1	1	1	1	1	0000000000111111111111111111111
1	1	1	0	1	0000000001111111111111111111111
1	1	1	0	0	0000000011111111111111111111111
1	0	1	0	0	0000000111111111111111111111111
1	0	1	0	1	0000001111111111111111111111111
1	0	1	1	1	0000011111111111111111111111111
1	0	1	1	0	0000111111111111111111111111111
1	0	0	1	0	0001111111111111111111111111111
1	0	0	1	1	0011111111111111111111111111111
1	0	0	0	1	0111111111111111111111111111111
1	0	0	0	0	1111111111111111111111111111111

Table I. Gray code encoder truth table

B_4	B_3	B_2	B_1	B_0	G_4	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1

0	0	0	1	0	0	0	0	1	1
0	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1	0
0	0	1	0	1	0	0	1	1	1
0	0	1	1	0	0	0	1	0	1
0	0	1	1	1	0	0	1	0	0
0	1	0	0	0	0	1	1	0	0
0	1	0	0	1	0	1	1	0	1
0	1	0	1	0	0	1	1	1	1
0	1	0	1	1	0	1	1	1	0
0	1	1	0	0	0	1	0	1	0
0	1	1	0	1	0	1	0	1	1
0	1	1	1	0	0	1	0	0	1
0	1	1	1	1	0	1	0	0	0
1	0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	1	0	0	1
1	0	0	1	0	1	1	0	1	1
1	0	0	1	1	1	1	0	1	0
1	0	1	0	0	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	0	1
1	0	1	1	1	1	1	1	0	0
1	1	0	0	0	1	0	1	0	0
1	1	0	0	1	1	0	1	0	1
1	1	0	1	0	1	0	1	1	1
1	1	0	1	1	1	0	1	1	0
1	1	1	0	0	1	0	0	1	0
1	1	1	0	1	1	0	0	1	1
1	1	1	1	0	1	0	0	0	1
1	1	1	1	1	1	0	0	0	0

Table II. Binary code encoder truth table.

III. IMPLEMENTATION OF LPHS ENCODER

There are different logic styles to implement the encoder design. Generally many of the implementations will be done using static CMOS logic style. The advantage of it is lower power consumption but it is limited to lower speeds. So for achieving higher speeds logic style has to be changed. The new logic style used for encoder implementation is pseudo dynamic CMOS logic [1].

Pseudo dynamic logic consists of PMOS transistor gate connected to clock in the pull up, logic is implemented using a group of NMOS transistors in the pull down network and for appropriate output logic an inverter will be placed at the output. In comparison with dynamic CMOS logic, pseudo dynamic CMOS logic does not require a NMOS evaluation transistor in series with the NMOS logic block, since all the inputs are properly synchronized.

In dynamic CMOS logic style, during pre-charge phase the NMOS transistor is on and the NMOS network is switched off. NMOS network will be on conditionally based on inputs. Whereas in pseudo dynamic CMOS logic style, the NMOS evaluation block can be enabled during pre-charge phase, since it has no influence on the output due to presence of inverter at the output. Obviously in evaluation phase the NMOS logic block output dominates but during pre-charge phase, if NMOS logic block is enabled, the output comes to settle down to a value decided by a resistive divide network formed by PMOS pull up and NMOS logic pull down block.

The transistor sizing is the crucial part in the design of LPHS encoder. It must be ensured that the voltage at the inverter input is appropriate. Consider the case when NMOS logic is enabled, if clock is low obviously PMOS will be on. Here the input to inverter should be less than V_{IH} (minimum input voltage applied at the input which will be taken as a logic one), with proper sizing of the PMOS transistor the input of the inverter will be less than V_{IH}. Similarly the other case need to have inverter input greater than V_{IH}. The disadvantage with pseudo dynamic logic style is that it has static power dissipation.

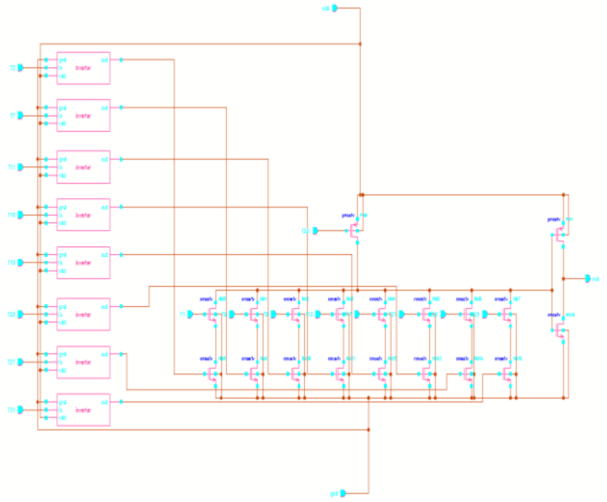

Fig. 2. Gray code bit G_0 generation circuit

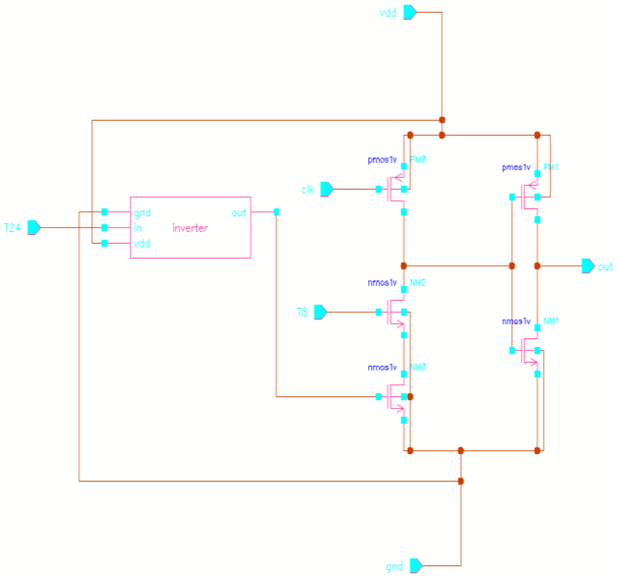

Fig. 5. Gray code bit G_3 generation circuit

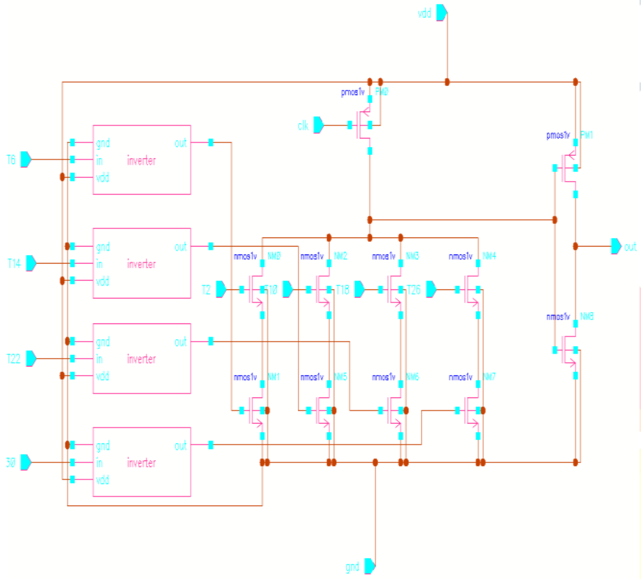

Fig. 3. Gray code bit G_1 generation circuit

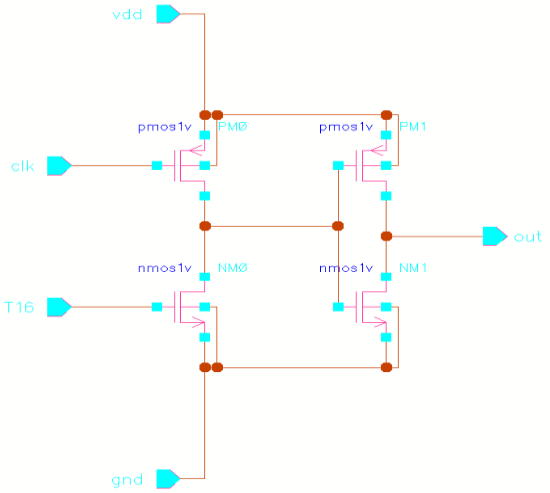

Fig. 6. Gray code bit G_4 generation circuit

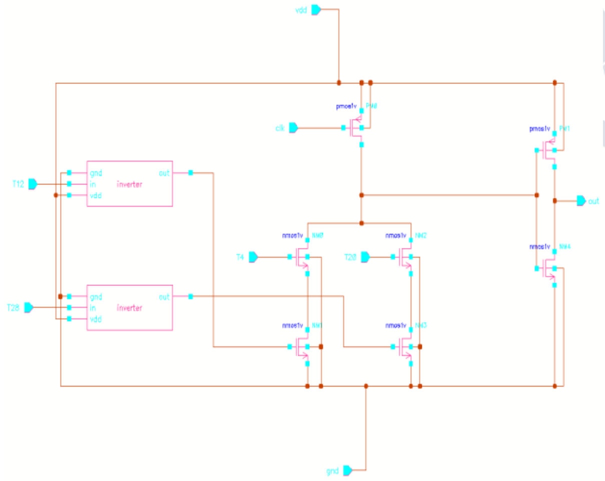

Fig. 4. Gray code bit G_2 generation circuit

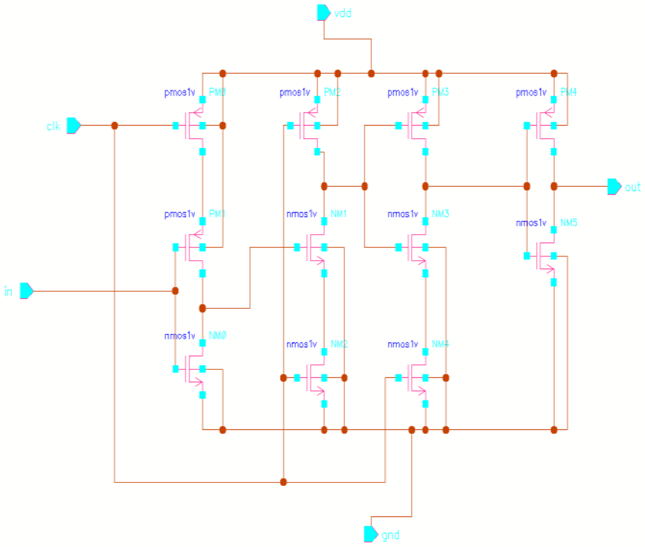

Fig. 7. Schematic of D Flip-Flop

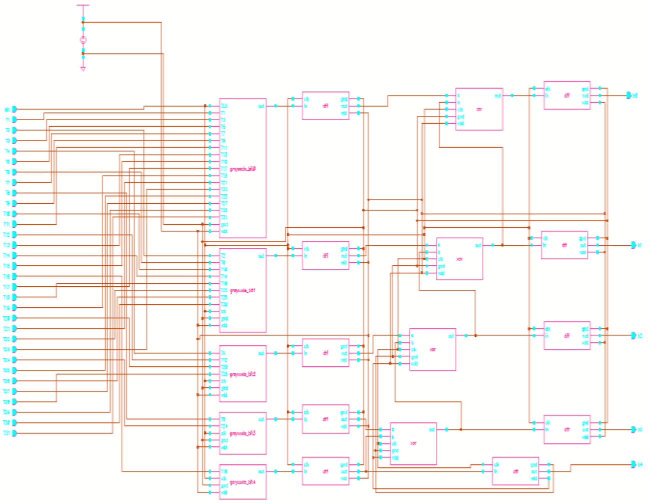

Fig. 8. Schematic of LPHS encoder

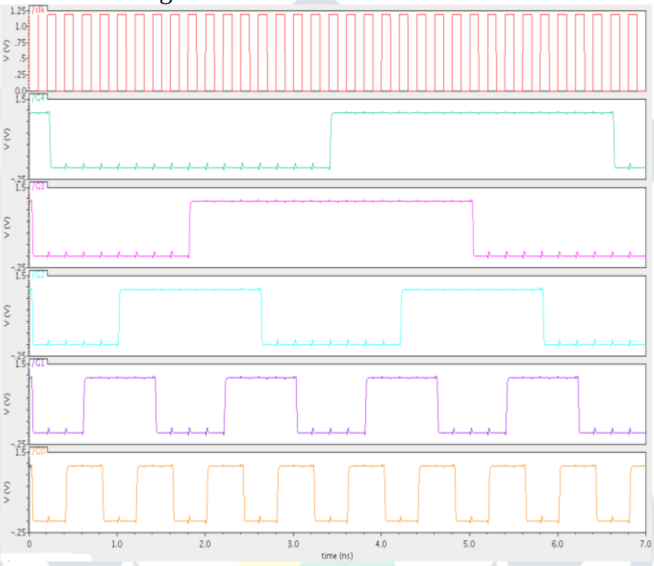

Fig. 9. Simulated intermediate gray code

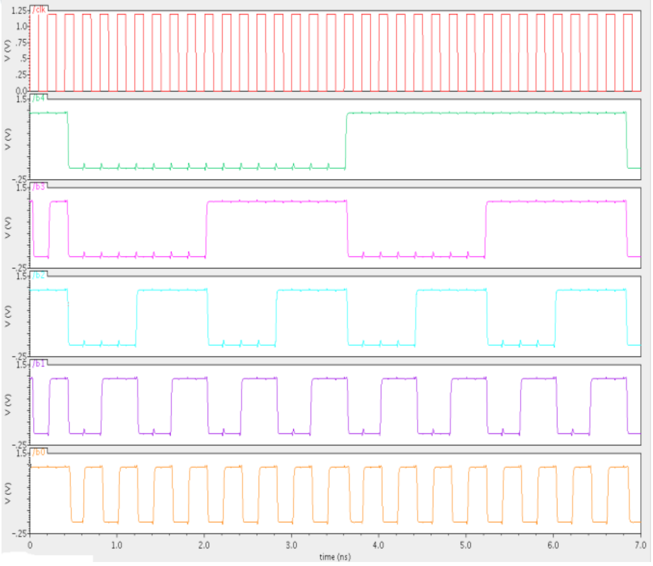

Fig. 10. Simulated output binary code

IV. SIMULATION RESULTS

The encoder is designed as shown in fig 8 and tested using all the input combinations from the truth table I and II. A D Flip-Flop shown in fig 7 is added after the gray code generation circuits and at the output of XOR gate to get the undistorted waveform. The converted intermediate gray code is shown in fig 9. The final output binary code is shown in fig 10. The summary of the LPHS encoder simulation results is shown in table III. In most of the 5-bit flash ADC's, the maximum sampling frequency achieved can be up to 3.5GS/s [4]. The proposed new encoding scheme achieves maximum sampling frequency of 5GS/s. The average power dissipation of the LPHS encoder is 0.388mW. The new encoding scheme uses less number of transistors than other static styles tends to reduction in cost and also achieves highest speed. The results are presented in Table III.

RESULTS	LPHS ENCODER
Architecture	Flash type
Resolution	5-bit
Technology	45 nm
Sampling Frequency	5GS/s
V_{dd}	0.9 V
Current	0.323mA
Power Dissipation	0.388mW

Table III. summary of LPHS Encoder.

V. CONCLUSION

The speed of an encoder plays an important role in the design of flash ADC. In addition to speed one has to take care about power dissipation. The proposed LPHS 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 45 nm technology using CADENCE tool. The encoder which is operating at 5GHz, consumes 0.388 mW from 0.9 V supply. The advantages of the encoder made it suitable for the design of high speed flash ADC.

REFERENCES

[1]. George Tom Varghese and K. K. Mahapatra, “ A High Speed Encode for a 5GS/s 5 Bit Flash ADC”.

[2]. D. Lee, J.Yoo, K. Choi and J. Ghazanavi, “Fat-free encoder for ultrahigh speed flash ADCs” I proc. IEEE Mid-West Symp. Corcuits Syst, pp 233-236, Aug 2002.

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

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

[5]. Sunghyun Park, Yorgos Palakas, Ashoke Ravi, Ralph. E. Bishop, and Michael P.Flynn, “A 3.5GS/s 5-bit Flash ADC in 90nm CMOS”, IEEE Custom Integrated Circuits Comference 2006.

[6]. Niket Agarwal, Roy Paily, “An Improved ROM Architecture for Bubble error Suression in High Speed Flash ADCs”, Annual IEEE Conference , pp 1-5,2005.

[7]. Mustafijur Rahman, k. L. Baishnab, F. A. Talukdar,”A Novel ROM Architecture for Reducing Bubble and Met-Stability Errors in High Speed Flash ADCs”,20th International Conference on Electronics, Communications and Computer, pp 15-19,2010.

[8]. Vinayashree Hiremath, “Design of High Speed ADC”, M.S. Thesis, Wright State University, 2010.