FAT TREE ENCODER DESIGN FOR ULTRA-HIGH SPEED FLASH A/D CONVERTERS

Daegyu Lee, Jincheol Yoo, Kyusun Choi, and Jahan Ghaznavi

The Pennsylvania State University
Department of Computer Science & Engineering
University Park, PA 16802
{dalee, jyoo, kyusun, ghaznavi}@cse.psu.edu

ABSTRACT

The thermometer code-to-binary code encoder has become the bottleneck of the ultra-high speed flash ADCs. In this paper, the authors presented the fat tree thermometer code-to-binary code encoder that is highly suitable for the ultra-high speed flash ADCs. The simulation and the implementation results show that the fat tree encoder outperforms the commonly used ROM encoder in terms of speed and power for the 6 bit CMOS flash ADC case. The speed is improved by almost a factor of 2 when using the fat tree encoder, which in fact demonstrates the fat tree encoder is an effective solution for the bottleneck problem in ultra-high speed ADCs.

1. INTRODUCTION

A flash analog-to-digital converter (ADC) is known for its high speed operation. An n bit ADC's front-end consists of $N-1$ (where $N=2^n$) voltage comparators, comparing fully parallel the incoming analog signal with $N-1$ reference voltages. The comparators produce the digital thermometer-code (TC), and the remaining back-end of a flash ADC consists of a thermometer code-to-binary code encoder, as shown in Figure 1.

The most common implementation of the TC-to-BC (Binary Code) encoder has been the ROM/PLA circuits[1, 2, 3]; however, the TC-to-BC encoder becomes the bottleneck of the ultra-high speed above 1 GHz sampling rate flash ADCs. The alternate encoder designs such as Wallace tree encoder[4] at 1 GHz sampling rate flash ADC implemented with the GaAs technology, XOR encoder[5] at 8 GHz rate ADC implemented with the SiGe bipolar technology, and pipeline encoder[6] at 5 GHz rate ADC implemented with the Josephson-junction super-conduction technology have been used. More recently, flash ADCs implemented with the CMOS technology perform the sampling rate in excess of 1 GHz[7, 8, 9]. The authors noticed that indeed the TC-to-BC encoder was the bottleneck of the ultra-high speed CMOS flash ADC[8]. In this paper, the authors present the fat tree TC-to-BC encoder that is highly suitable for the ultra-high speed flash ADCs.

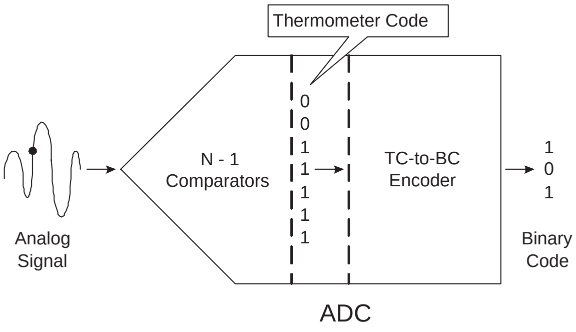


Figure 1: Code conversion in n bit flash ADC

The main advantage of the fat tree encoder over the other encoders is the high encoding speed. Also the fat tree encoder consumes less power compared with the ROM encoder. However, the fat tree layout design is more difficult and time consuming than the ROM layout design.

2. FAT TREE TC-TO-BC ENCODER

The TC-to-BC encoding is carried out in two stages in the fat tree encoder: the first stage converts the thermometer code to one-out-of-N code. The one-out-of-N code is same as an address decoder output. This code conversion is done in N bit parallel using N gates. Figure 2 shows the two-stage fat tree TC-to-BC encoder.

The second stage converts the one-out-of-N code to binary code using the multiple trees of OR gates. Figure 3 shows an example of a 4 bit ADC case. A 16 bit one-out-of-N code is presented to the leaf nodes of the tree and 4 bit binary code output is produced at the root nodes of the trees. One may superimpose one tree over the other tree and imagine a 3-D visualization of the trees in Figure 3. An

This work was supported in part by Pittsburgh Digital Greenhouse (EDTD00-2) through a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development.

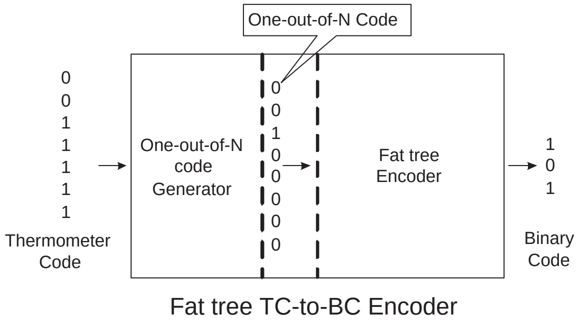


Figure 2: The two-stage fat tree TC-to-BC encoder

edge count of a node increases as the tree height increases - so it is named a fat tree.

Algorithmically, the fat tree circuit signal delay is $O(\log_2 N)$ whereas the ROM circuit signal delay is $O(N)$. Moreover, the Wallace tree encoder[4] signal delay is $O(\log_{1.5} N)$. Therefore, the fat tree circuit is the fastest speed circuit.

One may also notice in Figure 3 the fact that there are three OR gate delays from any leaf node to any root node. The signal delay from the $N-1$ inputs to the n outputs is uniform in the fat tree encoder. This is an important property allowing high speed wave pipelining without the pipeline registers.

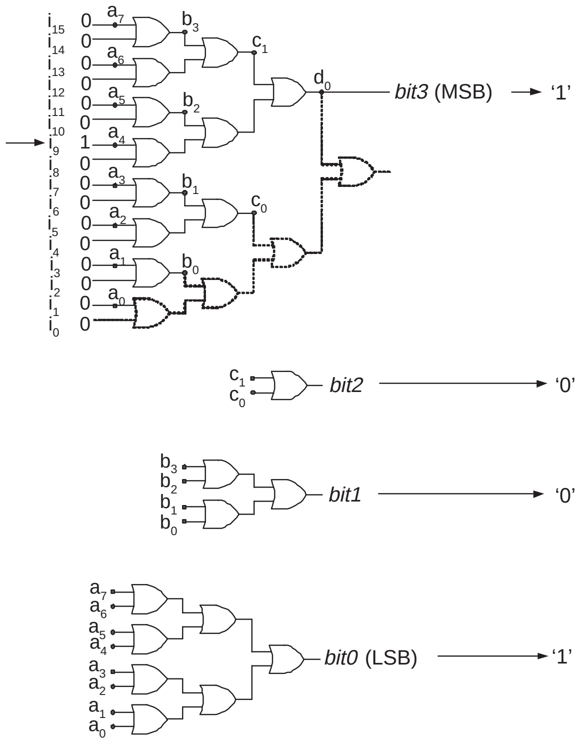
Apart from the speed, the fat tree of OR gates is an all-digital circuit and it does not require a clock signal, sense amps, or pull-up resistors. The fat tree circuit is more noise tolerant than the ROM circuit. Full static CMOS implementation of the OR gates eliminates any static power consumption otherwise necessary in circuits with pull-up resistors. Therefore, the fat tree circuit is less power consuming than the ROM circuit.

On the other hand, the layout of the fat tree is more difficult to design than the ROM. A tree is considered as a regular structure but it is not as regular as the ROM. Moreover, the fat tree is a 3-D structure, challenging to layout on a two dimensional chip.

3. FAT TREE ENCODER IMPLEMENTATION

For a more efficient implementation in CMOS, one may replace the OR gates with NOR and NAND gates using the DeMorgan's theorem, shown in Figure 4. The authors designed the 6 bit fat tree encoder, attempting to make the layout as much regular as possible.

To evaluate the performance of the fat tree encoder, authors designed ultra-high performance thresholding inverter ADCs[8] in $0.18\mu m$ CMOS technology. Figure 5 shows two complete 6 bit ADC layouts: Figure 5(a) shows the ADC with fat tree encoder and Figure 5(b) shows the ADC with ROM encoder. The encoder section is the right most column of each ADC layouts shown. A total of 485 transistors are used in the fat tree encoder implementation and a total of 222 transistors are used in the ROM encoder implementation. Both the fat tree encoder and the ROM encoder layout designs were done in full custom; the fat tree encoder layout turned out to be 20% less area than the ROM encoder layout! And one should note the fact that the transistor size of the fat tree encoder and the transistor size of the ROM encoder are different.



bit0 = a0 + a1 + a2 + a3 + a4 + a5 + a6 + a7 = 1

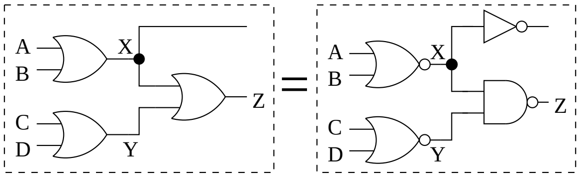
bit1 = b0 + b1 + b2 + b3 = 0

bit2 = c0 + c1 = 0

bit3 = d0 = 1

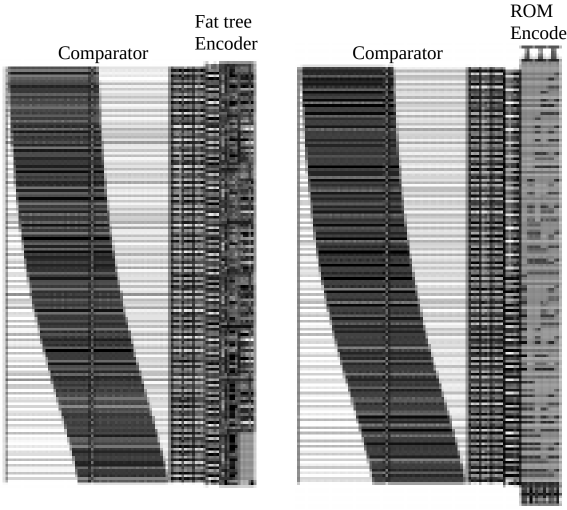
Figure 3: An example of a 4 bit ADC case

A chip containing the two 6 bit ADC layouts has been fabricated through MOSIS service, using $0.18\mu m$ digital CMOS process (T1AX-LO-EPI). The die photo does not show Figure 5 due to six layers of metals over the ADC layout and the two 6 bit ADCs are a part of a larger project placed on the chip. Figure 6 shows the oscilloscope displays of the two ADC operations, the ADC with the fat tree encoder in Figure 6(a) and the ADC with the ROM encoder in Figure 6(b). The full testing of the both ADCs are in progress.



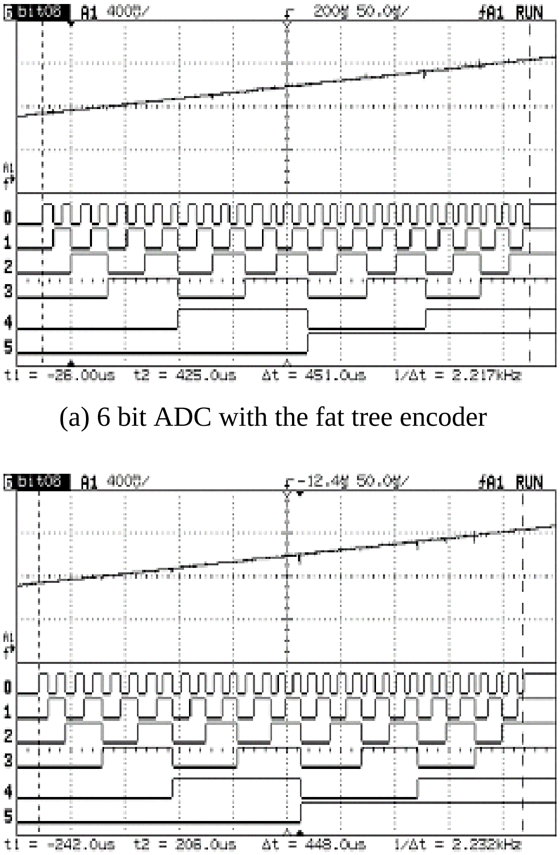
$$Z = \overline{XY} = \overline{\overline{(A + B)}\bullet\overline{(C + D)}} = (A + B) + (C + D)$$

Figure 4: Fat tree logic implementation



(a) Fat tree Encoder ADC (b) ROM Encoder ADC

Figure 5: Two complete 6 bit ADC layouts



(a) 6 bit ADC with the fat tree encoder

(b) 6 bit ADC with the ROM encoder

Figure 6: The oscilloscope displays of the 6 bit flash ADCs

4. SIMULATION RESULTS

In SPICE simulation, the fat tree encoder ADC output shows 2.00 $GSPS$ (Giga Samples Per Second) speed and the ROM encoder ADC output shows 1.11 $GSPS$. The speed improved almost double using the fat tree encoder.

The fat tree encoder ADC consumed average 22.70 mW and the ROM encoder ADC consumed average 32.64 mW over the simulation interval. Table 1 summarizes the two ADC implementations and simulation results. The results clearly show the fat tree encoder implementation is the better choice over the ROM encoder in terms of speed and power for the 6 bit CMOS flash ADC case. Table 2 shows the summary of the 8 bit and 10 bit fat tree encoder simulation results in 0.25 μm CMOS technology.

Table 1: Summary of the two ADC implementations and simulation results

	Fat tree encoder ADC	ROM encoder ADC
Technology	CMOS 0.18 μm	CMOS 0.18μm
Max. Speed	2.00 $GSPS$	1.11 $GSPS$
Power Consumption	Avg. 22.70 mW Max. 38.65 mW	Avg. 32.64 mW Max. 54.41 mW
Transistor Count	485	222
Area	ADC: 0.0546 mm^2 Encoder: 0.0074 mm^2	ADC: 0.0613 mm^2 Encoder: 0.0094 mm^2

Table 2: The 8 bit and 10 bit fat tree encoder simulation results

	8 bit	10 bit
Technology	CMOS 0.25 μm	CMOS 0.25μm
Worst case delay	1.14 $nSec.$	2.17 $nSec.$
Power Consumption	Avg. 1.79 mW Max. 13.83 mW	Avg. 8.91 mW Max. 17.91 mW
Transistor Count	2803	11419
Area	0.048 mm^2	0.191 mm^2

5. CONCLUSION

Alternate TC-to-BC encoder designs are needed for the ultra-high speed flash ADCs because the TC-to-BC encoder has become the bottleneck of the flash ADCs. The authors proposed and presented the fat tree TC-to-BC encoder that is highly suitable for the ultra-high speed flash ADCs. The simulation and the implementation results show that the fat tree encoder outperforms the commonly used ROM encoder in terms of speed, power, and area for the 6 bit CMOS flash ADC case. The speed is improved by almost a factor of 2 when using the fat tree encoder, which in fact demonstrates the fat tree encoder is an effective solution for the bottleneck problem in ultra-high speed ADCs. Although the fat tree layout design is more difficult and time consuming than the ROM layout design, these problems can be overcome by using a layout generation tool for the fat tree encoder. The authors has applied for a patent for the fat tree encoder circuit.

6. REFERENCES

[1] S. Padoan, A. Boni, C. Morandi, and F. Venturi, "A novel coding schemes for the ROM of parallel ADCs, featuring reduced conversion noise in the case of single bubbles in the thermometer code," *IEEE International Conference on Circuits and Systems*, 2:271–274, 1998.

[2] M. Sugawara, H. Yoshida, M. Mitsuishi, S. Nakamura, S. Nakaigawa, Y. Kunisaki, and H. Suzuki, "A 2.5V 100MS/s 8bit ADC using Pre-Linearization Input Buffer and Level Up DAC/Subtractor," *Symposium on VLSI Circuits Digest of Technical Papers*, pp. 170–173, 1998.

[3] M. Ito, T. Miki, S. Hosotani, T. Kumamoto, Y. Yamashita, M. Kijima, T. Okuda, and K. Okada, "A 10bit 20MS/s 3V Supply CMOS A/D Converter," *IEEE J. of Solid-State Circuits*, 29(12):1531–1536, December 1994.

[4] R. Kanan, F. Kaess, and M. Declercq, "A 640mW High Accuracy 8-bit 1GHz Flash ADC Encoder," *IEEE International Symposium on Circuits and Systems*, 2:420–423, 1999.

[5] P. Xiao, K. Jenkins, M. Soyuer, H. Ainspan, J. Burghartz, H. Shin, M. Dolan, D. Harame, "A 4b 8GSample/s A/D Converter in SiGe Bipolar Technology," *IEEE International Solid-State Circuits Conference*, pp. 124–125, 1997.

[6] H. Luong, D. Hebert, and T. Duzer, "Fully Parallel Superconducting Analog-to-Digital Converter," *IEEE Transactions on Applied Superconductivity*, 3(1):2633–2636, March 1993.

[7] M. Choi, A. Abidi, "A 6b 1.3GSample/s A/D Converter in 0.35μm CMOS," *IEEE International Solid-State Circuits Conference*, pp. 126–1278, 2001.

[8] J. 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.

[9] B. Yu, W. Black, "A 900MS/s 6b Interleaved CMOS Flash ADC," *IEEE Custom Integrated Circuits Conference*, pp. 149–152, 2001.