

An 8-Bit Current-Mode Folding ADC with Optimized Active Averaging Network

M. Azin* and M. Sharif Bakhtiar¹

In this paper, an 8-bit CMOS current-mode folding-interpolating ADC is presented. A new active averaging-interpolating network is described, which results in a better error correction factor compared to its resistive counterpart. Using novel circuits for fast settling and careful transistor sizing, a fast (>160 Msps) and low power (70 mW in 2.5 V supply voltage) 8-bit ADC, with a total chip area of 1×1.4 mm in a 0.25 micron CMOS process, is demonstrated.

INTRODUCTION

The increasing demand for digital processing systems makes high-speed and low-voltage analog to digital converters key elements in many applications, such as wireless digital communication and video processing. CMOS flash ADCs can satisfy the high speed requirement. However, the complexity and, consequently, the power consumption of flash ADCs grow exponentially as resolution increases. This is why flash converters are only used for low-resolution (6-7 bits) applications [1,2]. Pipeline ADCs are used for many high-speed applications, which require a higher resolution. Pipeline ADCs, nonetheless, introduce an inherent latency, which makes them unsuitable for applications such as control and data storage. The folding-interpolating technique is an alternative that avoids the complexity of flash ADCs. Folding ADCs can also achieve high-speed, high-resolution and low power characteristics, as well as a low latency [3-5].

Cross-connected differential pairs are used to fold the input signal of the ADC [6]. The number of differential pairs is referred to as the folding factor. To achieve a higher resolution, a larger folding factor is needed, which, in turn, results in a larger capacitive load at the output nodes of the folding blocks. Moreover, the low offset requirement for the differential input pairs directly translates to large input

devices, which, in turn, slow down the circuit. On the other hand, the signal frequency at the folder's output is higher than the input frequency by an amount proportional to the folding factor. This is why the output nodes of the folding blocks dramatically limit the speed of a folding-interpolating ADC at higher resolutions. The application of an OTA at the output of the folders can improve the ADCs bandwidth effectively [4]. However, this approach is not well suited for low voltage applications.

In this work, a novel low voltage method to improve the speed and the settling time of the folding blocks is presented. A new structure for offset averaging is also presented. Unlike resistive offset averaging structures [3], the proposed structure can be used in current mode circuits and results in a better Error Correction Factor (ECF).

System level considerations of the converter and the designed architecture are given in the following section. Then, a new active averaging network is presented, which is compared to conventional averaging networks and, circuit implementation of the ADC is described. Post layout simulation results and the conclusion are presented, respectively, in the final sections.

ARCHITECTURE

Because of the longer delay time of multistage architectures, a single stage architecture is selected for the ADC implementation. As shown in Figure 1, four folding blocks are used to make 32 zero crossing points and the folding factor is selected to be 8.

In order to make additional zero crossing points,

*. Corresponding Author, Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, Ohio, USA.

1. Department of Electrical Engineering, Sharif University of Technology, Tehran, I.R. Iran.

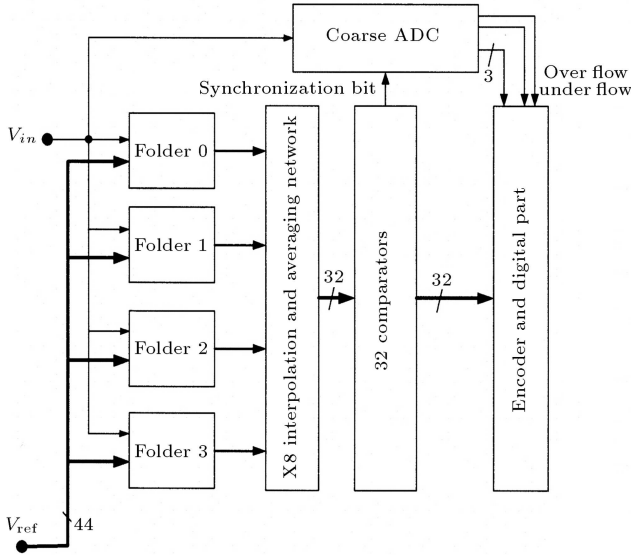


Figure 1. Proposed Single stage ADC.

the 4 differential outputs of the folding blocks are fed into an interpolating network, resulting in 32 folded signals. As described in the next section, interpolation and averaging networks are implemented using current mirrors. To detect the zero crossing points of 32 folded signals, 32 current comparators are used. While bit 0 (LSB) to bit 5 are derived in the main path, ten additional comparators are also used in the coarse ADC path to produce the two most significant bits (B7 and B6), as well as overflow and underflow bits. As shown

in Figure 2, bit 5 (B5) is also used to synchronize the main ADC with the coarse ADC. A change in the output code is detected by two coarse comparators and is synchronized with the transitions in the main ADC output by the synchronization bit, which is the 32nd output of the main comparators. This synchronization method [7] desensitizes the output with respect to the offsets, as high as 1/16 of the full input range of the coarse comparators. As depicted in Figure 2, six of the ten coarse comparators are used to determine most significant bits and the others are used to provide underflow and overflow signals.

OFFSET AVERAGING

Averaging is an effective mean to reduce offset induced errors. First order resistive networks are typically used for offset averaging, where outputs of the folding blocks are interconnected through a resistive network (Figure 3). The averaging network may also be used for interpolation of the folded signals. In Figure 3, the load resistance of the folding blocks, represented by resistors R_0 s and R_1 s, are interpolating-averaging resistors.

In an offset averaging network, each output is a linear combination of input signals. Therefore, if Sin_i and $Sout_i$ represent the input and output signals of the network, respectively, it can be shown that:

$$Sout_0 = \sum_{i=0}^{M-1} h_i \cdot Sin_i, \quad (1)$$

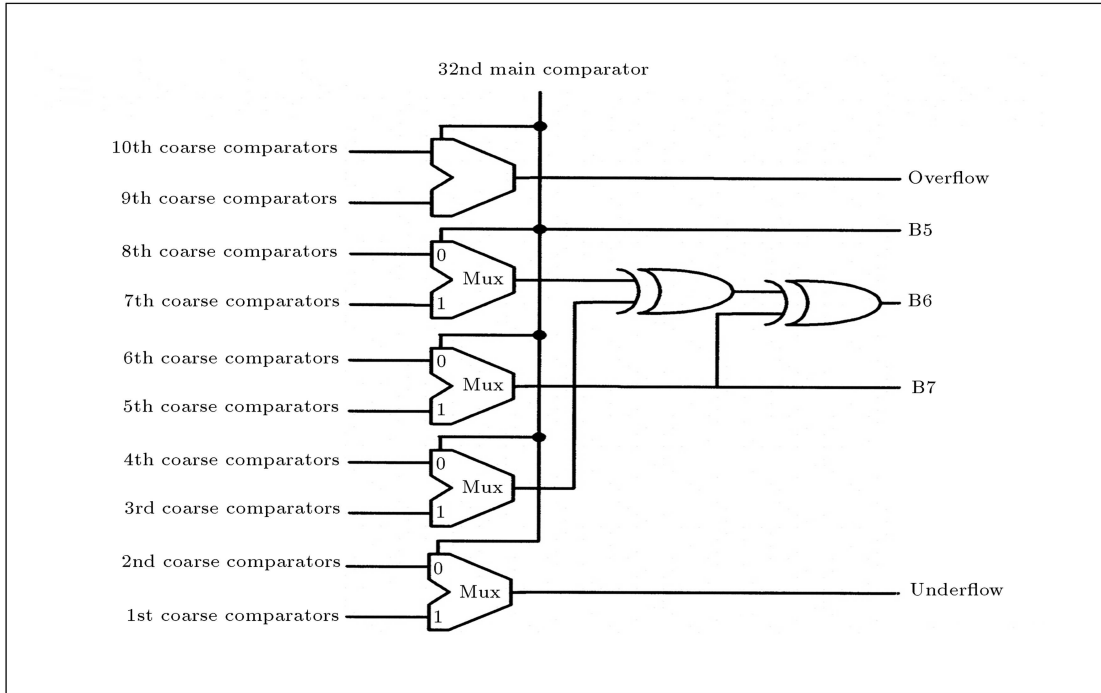


Figure 2. Coarse ADC is synchronized with main ADC.

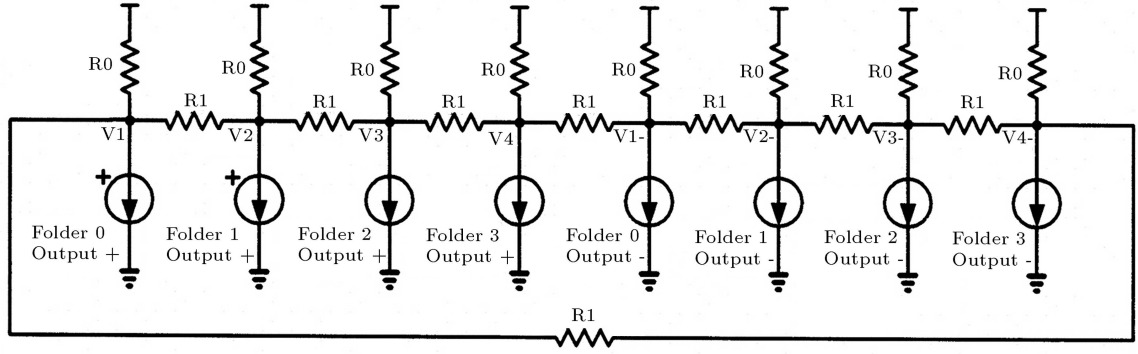


Figure 3. First order resistive averaging network.

where M is the number of total input signals of the averaging network and h'_i s are offset averaging factors. Suppose that M input signals of an averaging network are sorted such that zero crossing points of Sin_k are between the zero crossing points of Sin_{k+1} and Sin_{k-1} , then it can be shown that:

$$\text{Sin}_i = -\text{Sin}_{\frac{M}{2}+i} \quad \text{for } 0 \leq i \leq \frac{M}{2} - 1. \quad (2)$$

Equation 2 means that the second half of each input signal has the opposite polarity of the corresponding first half. If Sin_0 has a zero value, then it can be written that:

$$\text{Sin}_i = -\text{Sin}_{M-i} \quad \text{for } i \neq 0. \quad (3)$$

Using Equation 3 leads to:

$$h_i = h_{M-i} \quad \text{for } i \neq 0. \quad (4)$$

Errors in the input blocks of a folding-interpolating ADC can be divided into vertical and horizontal errors [3]. An averaging network improves these errors in the following ways:

1. Vertical error: This is shown in Figure 4a, where V_{in} is the input voltage to the ADC and V_o is the output signal of a folding block. Vertical error is caused by deviation of the folding characteristic curve away from the ideal characteristic in the V_o direction. A mismatch of the tail current sources

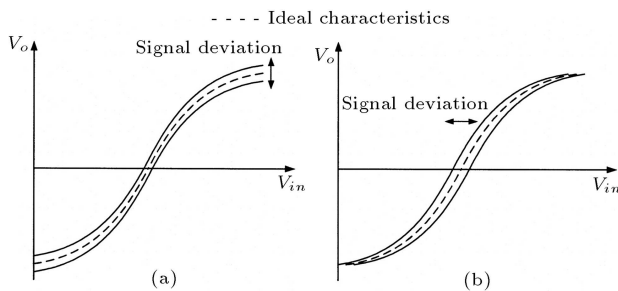


Figure 4. (a) Vertical error; (b) Horizontal error.

in a folding block is a cause of vertical error, which remains constant when the input signal of the ADC is changing;

2. Horizontal error: This is the deviation of the characteristic curve in the V_{in} direction from the ideal characteristic (Figure 4b). One major source of horizontal error is the mismatch in the folder pairs. In contrast to vertical errors, horizontal errors depend on the input value. Since the maximum slope occurs at the zero crossing points of the folding characteristic, horizontal error has its maximums at these points. Conversely, the lowest value of the horizontal error occurs when the output signal is at its extreme. This is shown in Figure 5, where e is the horizontal error normalized to its maximum.

ECF (Error Correction Factor) is defined as the effective error at the input of an averaging network to the output error, i.e.:

$$\text{ECF} = \frac{\delta_{\text{Sin}} \cdot \text{Goa}}{\delta_{\text{Sout}}}, \quad (5)$$

where δ_{Sin} and δ_{Sout} are the errors at the input and output of the averaging network, respectively, and Goa is the averaging gain. With respect to Equation 1, it

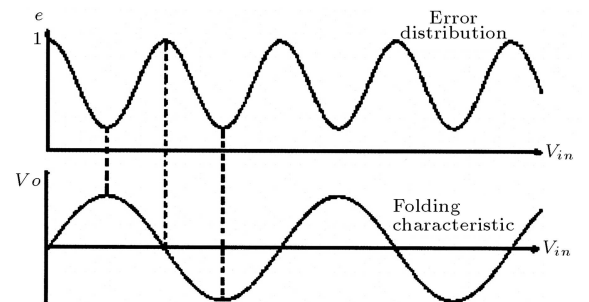


Figure 5. Horizontal error has its largest value at zero crossing points and its lowest at maximums and minimums.

can be shown that:

$$\delta_{\text{Sout}0}^2 = \sum_{i=0}^{\frac{M}{2}-1} \left(h_i - h_{\frac{M}{2}+i} \right)^2 \cdot \delta_{\text{Sin}i}^2. \quad (6)$$

The absolute values of the errors for Sin_K and $\text{Sin}_{M/2+K}$ are equal. If $\delta_{\text{Sin}i}$ is the vertical error, then the output error at zero crossing points is:

$$\delta_{\text{Sout}0}^2 = \left[\sum_{i=0}^{\frac{M}{2}-1} \left(h_i - h_{\frac{M}{2}+i} \right)^2 \right] \cdot \delta_{\text{Sin}0}^2. \quad (7)$$

And, for the horizontal error, one can write:

$$\delta_{\text{Sout}0}^2 = \left[\sum_{i=0}^{\frac{M}{2}-1} \left(h_i - h_{\frac{M}{2}+i} \right)^2 \cdot e_i^2 \right] \cdot \delta_{\text{Sin}0}^2, \quad (8)$$

where e_i is the value of e for Sin_i , when Sin_0 is zero. It can be shown that e_0 is equal to 1 and other e_i 's are less than 1. Goa can be obtained by differentiating Equation 1. Substituting Equations 7 and 8 in Equation 5 determines ECFs for vertical and horizontal errors as follows:

$$\text{ECF}_{\text{VE}} = \text{Goa} \cdot \left[\sum_{i=0}^{\frac{M}{2}-1} \left(h_i - h_{\frac{M}{2}+i} \right)^2 \right]^{-1/2}, \quad (9)$$

for vertical errors;

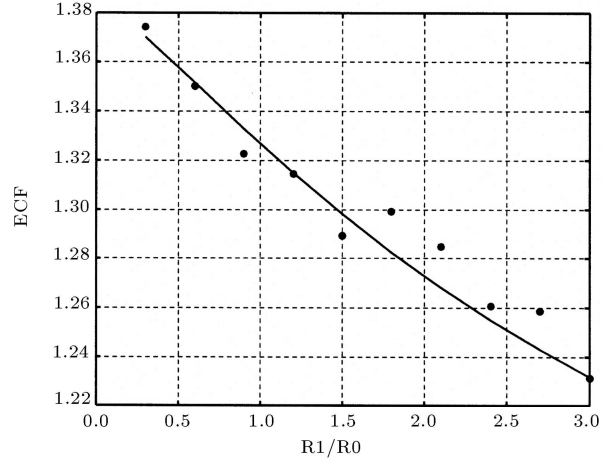
$$\text{ECF}_{\text{HE}} = \text{Goa} \cdot \left[\sum_{i=0}^{\frac{M}{2}-1} \left(\left(h_i - h_{\frac{M}{2}+i} \right)^2 \cdot e_i^2 \right) \right]^{-1/2}, \quad (10)$$

for horizontal errors.

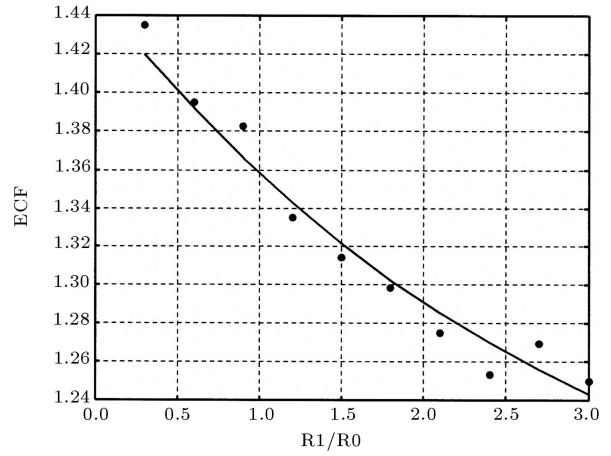
A resistive averaging network was first designed for the ADC. ECF was then calculated for different values of R_1/R_0 using Equations 9 and 10. The results are compared with those of Monte Carlo simulations in Figure 6.

Equation 1 can be implemented by current mirrors, as shown in Figure 7. For component and area saving, this network can also be employed as an active interpolating network. Averaging factors are determined by optimizing the ECF equations; this is why the circuit is denoted as an optimum averaging network. The optimization of Equation 10 results in the addition of $\frac{1}{3}$ of each input signal to its two neighboring signals, in order to make an output signal. Using a Monte Carlo simulation, it is shown that the optimum averaging network improves vertical errors by a factor of 1.3 and horizontal errors by a factor of 1.5.

For the first order resistive averaging network, the above improvement with the same Goa would be limited to 1.2 for vertical errors and 1.3 for horizontal errors.



(a)



(b)

Figure 6. ECF versus (a) Vertical errors and (b) Horizontal errors. Circles are Monte Carlo simulation results.

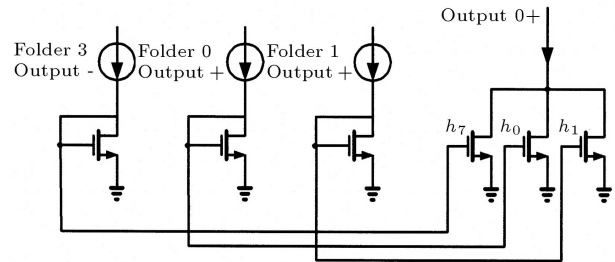


Figure 7. Current mode averaging network.

CIRCUIT IMPLEMENTATION

Folding Blocks

Figure 8 shows the circuit of a folding block. Among eleven used differential pairs, eight (3rd to 10th) are the main folders, two (2nd and 11th) are to enhance the range for interpolation and one (1st) is added for the DC balance. The output currents of the

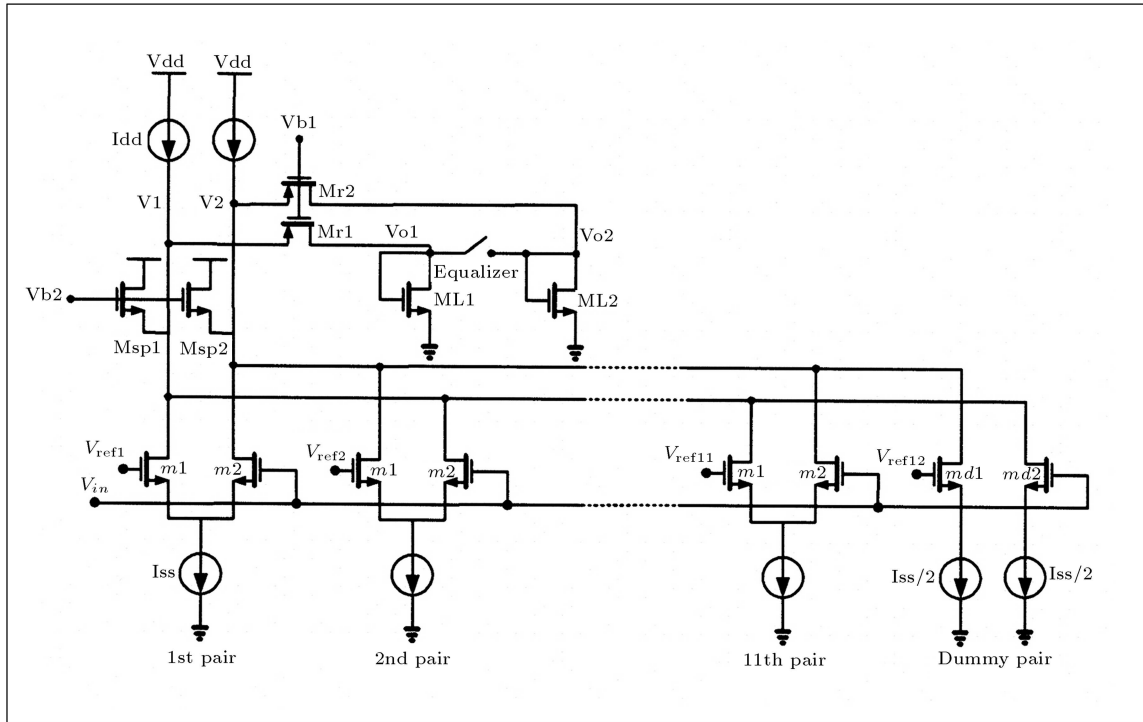


Figure 8. Folding block circuit.

11 differential pairs and a dummy pair are added together to make two folded currents. These currents are subtracted from two constant currents and, then, flow through the cascoded transistors, Mr1 and Mr2. The two output currents are fed into the current mirror transistors (ML1 and ML2), which drive the interpolating/averaging network. In order to increase the speed of the output nodes, an equalizing switch shorts these nodes to each other for about 1/3 of the clock period.

Two novel methods are used to speed up the folder response to large input signals:

1. Large positive input enhancements: For large positive input pulses, the common mode voltage of V1 and V2 falls (Figure 8), turning Mr1 and Mr2 off. In order to improve the common mode voltage settling, two NMOS transistors (Msp1 and Msp2) are added to the circuit. Vb2 is set at one threshold voltage above the common mode voltage of V1 and V2. When the voltage at these nodes decreases, due to a positive input, Msp1 and Msp2 turn on, in order to speed up the folding block settling.
2. Large negative input enhancements: A differential pair, with single ended input, shows a slower response to falling input than to rising input. This is because the tail node, where the source of the two input devices are connected together, has a slew rate limited behavior for a falling input. Tail settling, for rising and falling inputs, is compared

in Figure 9. To improve ADC speed with a falling input pulse, additional current sources are paralleled with each tail current source (Figure 10a). These additional current sources turn on when a falling input pulse is detected. The detector circuit, which consists of a differentiator and an inverter, is shown in Figure 10b. As shown in Figure 11, a considerable improvement is achieved, due to the injection of the current pulses through the additional current sources.

The finite output resistance of the tail current sources results in an input-dependent current offset in the output signals of a folding block. The input dependency of this offset is due to the fact that for a differential pair with $V_{in} < V_{ref}$, the current source voltage is constant and for $V_{in} > V_{ref}$, the source voltage follows the input signal. The variation

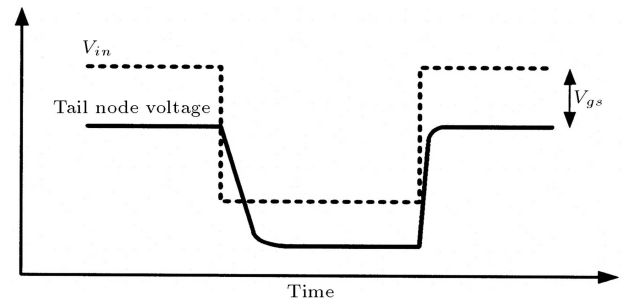


Figure 9. Tail node settling behavior.

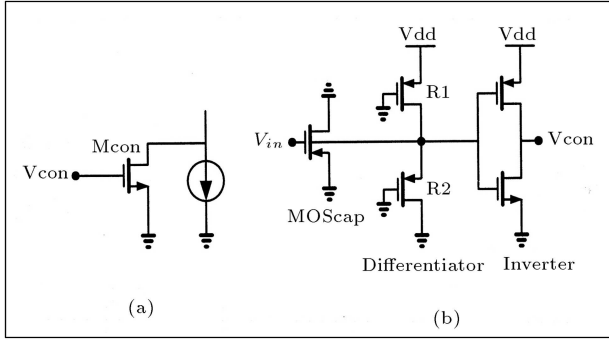


Figure 10. (a) Additional current source; (b) Falling input detector.

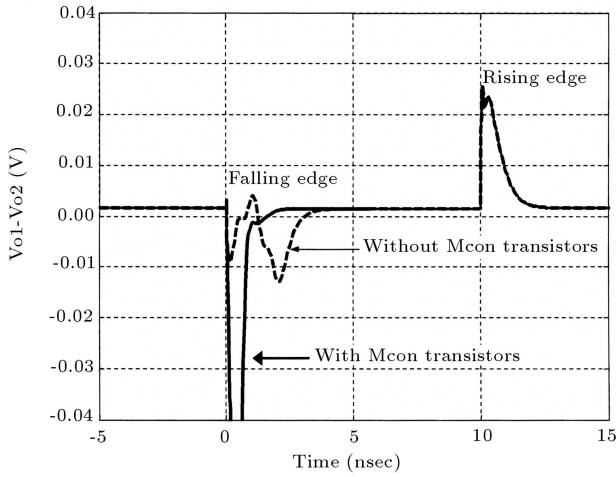


Figure 11. The effect of additional current sources and falling input detector.

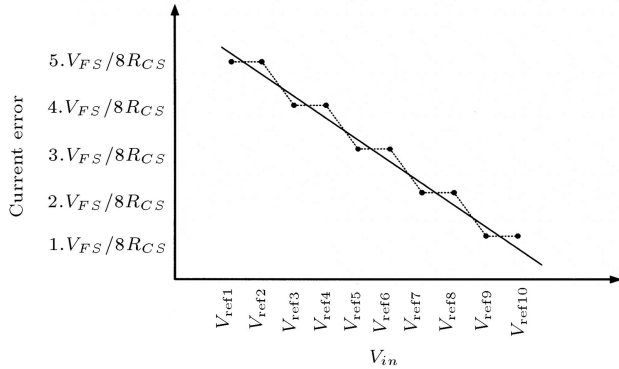


Figure 12. Error induced by current source finite resistance.

of output offset with the input voltage is shown in Figure 12 (dotted line), where V_{FS} is the full input range voltage and R_{CS} is the current source output resistance.

To compensate for the current error, the input voltage is applied to a dummy pair, with:

$$R_{cs_dummy} = 2.R_{cs},$$

and:

$$V_{ref12} = V_{ref11} + \frac{3}{16}V_{FS}.$$

The output current of this pair has the same input dependency as shown in Figure 12 (solid line). If the output current of this dummy pair is subtracted from the output current of the folding block, the current offset will be limited to the difference of the two curves in Figure 12. Using this method, the current error is reduced by a factor of 16. In practice, because of the nonlinear behavior of the current source resistance, this improvement is limited to a factor of 5.

Comparators

Current comparators are used to detect the zero crossing point of 32 folded signals. Because of the pre-amplification of signals in the folding blocks, the requirements for comparators in the folding-interpolating ADCs are fairly relaxed. The designed current comparator is shown in Figure 13.

The two input signals are subtracted from two constant currents and then flow into the bistable circuit ($M1 \sim M4$). When V_{eq} goes low, the bistable circuit

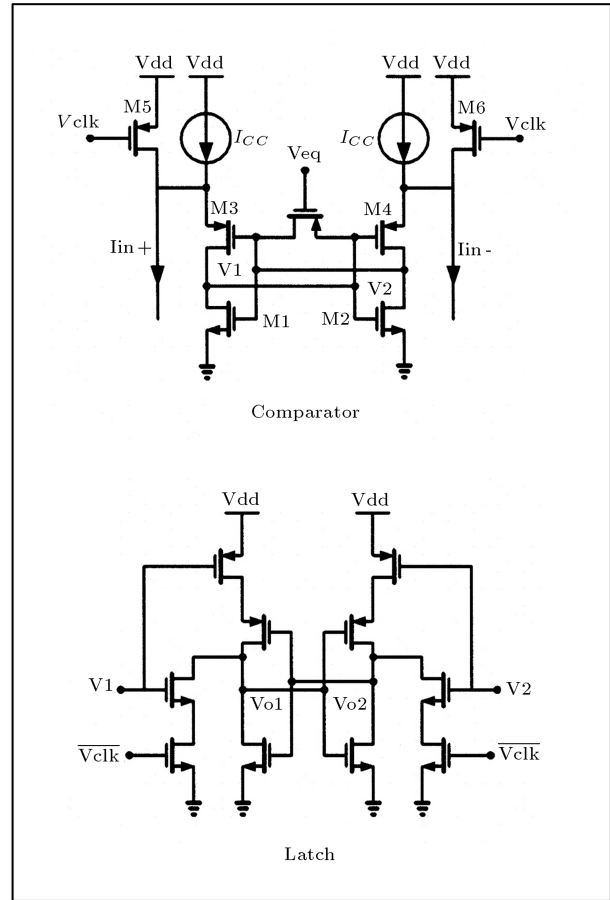


Figure 13. Current comparator and latch.

switches to one of the two stable states, according to the difference of the input currents. It can be shown that, if I_{CC} has a value slightly larger than the common mode value of the input currents, the mismatch of M1 and M2 has a negligible effect on the comparator offset. As I_{CC} increases, the effect of the M1 and M2 mismatch becomes more dominant. In order to have a precise comparator, I_{CC} should be as close to the common mode value of the inputs as possible. On the other hand, a small value of difference between I_{CC} and the common mode value results in a large settling time for the bistable circuit. In order to have precision and speed simultaneously, two transistors (M5 and M6) turn on shortly (1 nsec) after V_{eq} goes low. This will cause a higher current flow into the bistable circuit and push the bistable faster towards its final state. A careful transistor sizing leads to a 1 μ A current offset and 1 nsec comparison time for the comparators.

ADC IMPLEMENTATION

The ADC was implemented in a 0.25 μ m digital CMOS process with one poly and five metal interconnection levels. The designed layout measured 1×1.4 mm², excluding the reference ladder. The chip layout is shown in Figure 14. Because of the matching and output resistance requirements, the tail current sources are responsible for the large area occupied by the folding blocks.

Monte Carlo simulations show the maximum INL and DNL values of 0.7 LSB and 0.5 LSB, respectively (Figure 15). Taking the extracted parasitic into account, the dynamic performance of the ADC was simulated, based on the assumption of a 10-bit resolution and a 2.5 nsec settling time for the front-end track and hold circuit.

The output spectrum for a full-scale sinusoidal input signal is shown in Figure 16. Table 1 shows the overall performance and specification of the ADC,

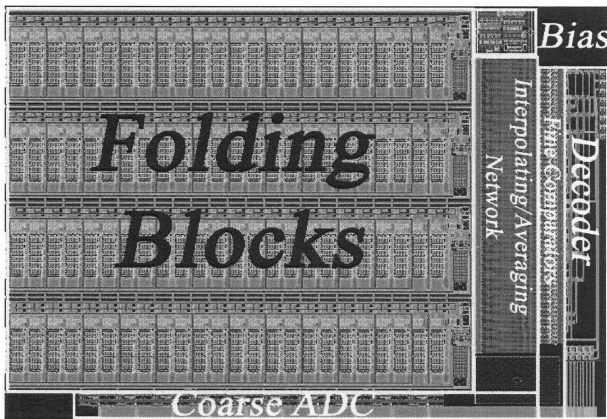


Figure 14. Chip layout.

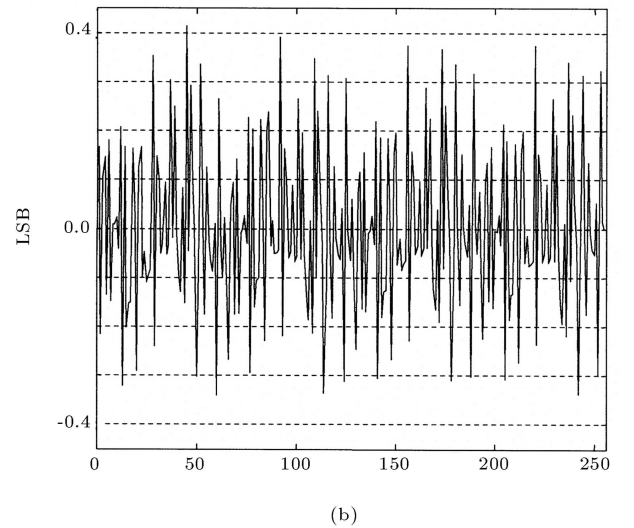
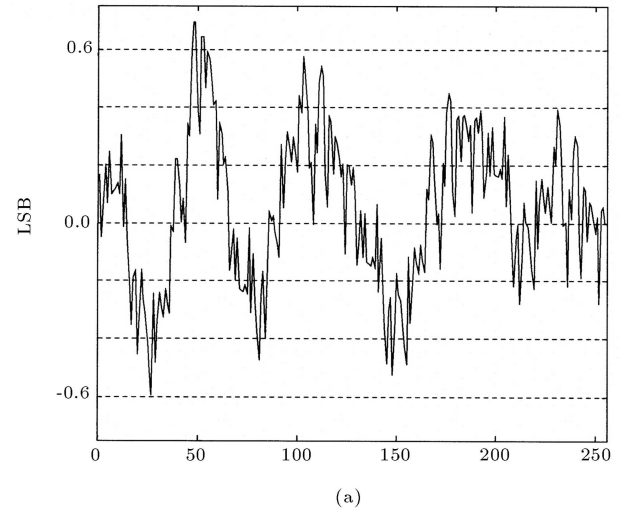


Figure 15. (a) INL error; (b) DNL error.

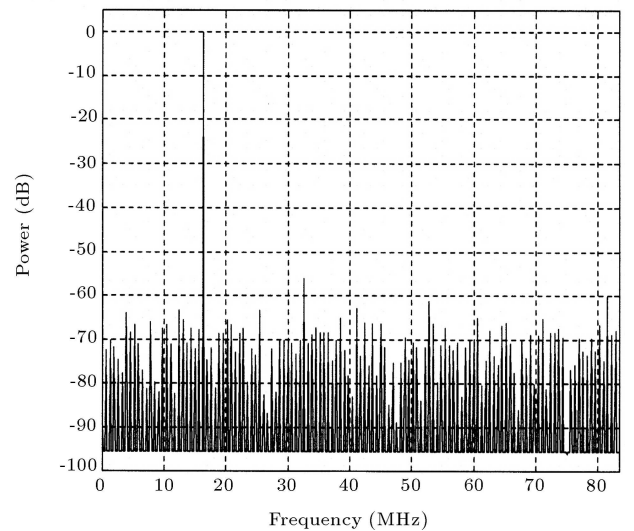


Figure 16. Output spectrum, input frequency = 16.27 MHz, clock frequency = 166.6 MHz.

Table 1. Simulation results summary.

Technology	CMOS 0.25 μm
Supply Voltage	2.5 volt
Full Scale Input Voltage	1.3 volt
INL, DNL	$< 0.7, < 0.5$
Maximum Clock Frequency	166 Msps
ENOB (Low Frequency)	7.5
ENOB (Nyquist Frequency)	7.3
SFDR	> 55 dB
Power Dissipation (Reference Ladder Included)	70 mW

Table 2. Performance comparison of recent works.

Author	Date	Bits	Speed (Msps)	Power (mW)
[8]	2001	8b	100	165
[9]	2003	8b	300	200
[10]	2001	8b	30	18
[11]	2004	8b	600	200
[12]	1997	12b	60	300
[13]	1996	8b	80	80
This work	-	8b	166	55

based on the post layout simulations. The performance parameters of this work are compared with other similar ADCs in Table 2.

CONCLUSION

The design of a high-speed, 8-bit folding-interpolating ADC was presented. A new optimized current mode interpolating and averaging method was used. It was shown that the presented current mode interpolating and averaging technique results in a higher degree of error correction compared to the conventional resistive method. The use of novel speed enhancement techniques provided the possibility of increasing the sampling frequency to more than 160 MHz with an SFDR better than 55 dB.

REFERENCES

1. Tsukamoto, S., Endo, T. and Schofield, W. "A CMOS 6-b 400-MSample/s ADC with error correction", *IEEE J. Solid-State Circuits*, **33**, pp 1939-1947 (Dec. 1998).
2. Tamba, Y. and Yamakido, K. "A CMOS 6-b 500-MSample/s ADC for a hard disk driver read channel", in *ISSCC Dig. Tech. Papers*, pp 324-325 (Feb. 1999).
3. Pan, H., Segami, M., Choi, M., Cao, J., Hatori, F. and Abidi, A. "A 3.3V, 12b, 50MS/s A/D converter in 0.6- μm CMOS with over 80dB SFDR", in *ISSCC Dig. Tech. Papers*, pp 40-41 (Feb. 2000).
4. Nauta, B. and Venes, A.G.W. "A 70-MS/s 100-mW 8-b CMOS folding and interpolating A/D converter", *IEEE J. Solid-State Circuits*, **30**, pp 1302-1308 (Dec. 1995).
5. Pan, H. "A 3.3-Volt 50-MS/s A/D converter in 0.6 μm CMOS with 80-dB SFDR", PhD Thesis, University of California, Los Angeles (1999).
6. Razavi, B., *Principles of Data Conversion System Design*, New York, IEEE Press (1995).
7. van de Plassche, R.J. and Baltus, P. "An 8-bit 100-MHz full-Nyquist analog-to-digital converter", *IEEE J. Solid-State Circuits*, **23**, pp 1334-1344 (Dec. 1988).
8. Choe, M.J., Song, B.S. and Bacrania, K. "An 8-bit 100MSample/s CMOS pipelined folding ADC", *IEEE J. Solid-State Circuits*, **36**(2), pp 184-194 (Feb. 2001).
9. Li, Y. and Sanchez-Sinencio, E. "A wide input bandwidth 7-bit 300MSample/s folding and current-mode

- interpolating ADC", *IEEE J. Solid-State Circuits*, **38**(8), pp 1405-1410 (Aug. 2003).
10. Sigenobu, T. et al. "A 8-b 30MS/s 18mW ADC with 1.8-V single power supply", *Proc. Symp. VLSI Circuits*, pp 209-210 (2001).
 11. Geelen, G. and Paulus, E. "An 8b 600MS/s 200mW A/D converter using an amplifier preset technique" *IEEE International Solid-State Conference* (Feb. 2004).
 12. Vorenkamp, P. and Roovers, R. "A 12-b, 60-M Sample/s cascaded folding and interpolating ADC", *IEEE J. Solid-State Circuits*, **32**, pp 1876-1886 (Dec. 1997).
 13. Venes, A.G.W. and van de Plassche, R.J. "An 80-MHz, 8-b CMOS folding A/D converter with distributed track and hold preprocessing", *IEEE J. Solid-State Circuits*, **31**, pp 1846-1853 (Dec. 1996).