A 150 Msamples/s Folding and Current Mode Interpolating ADC in 0.35μm CMOS

Robert M. Senger, Paul M. Walsh, Jérôme Le Ny

Abstract— **An ADC using folding and interpolating techniques has been realised in $0.35\,\mu m$ CMOS. A current-mode approach has been adopted. Fully differential current mode interpolating within the folder allows fast operation with low supply voltages. The folding ADC architecture reduces the number of comparators. The ADC has a dynamic range of $1.9\,V$ and can digitize a $75\,Mhz$ full-scale input signal. The 8 bit converter occupies $2\,mm^2$ and dissipates $150\,mW$ from a single 3.3V supply.**

Keywords— **Analog-to-digital conversion, CMOS, folding, interpolating**

I. Introduction

THE proliferation of digital signal processing across a wide range of applications has made analog-to-digital converters a key block in most mixed signal interface systems. High speed 8 bit ADC's have applications in communications, local area networks and flat-panel displays. In CMOS, an easy approach is to use a flash ADC [1], [2], [3]. CMOS flash ADC's with array averaging are reported to achieve a $1.3\,Gsamples/s$ conversion rate. However the complexity of the flash ADC grows exponentially as resolution increases because the number of comparators increases by 2^n (where n is the resolution). This makes flash ADC's unsuitable for resolutions $> 8\,bits$ because they consume large area and power. Many high speed ADC architectures have been reported to try and overcome the problems of flash ADC's. Pipeline and Subranging architectures are examples of such efforts and have been successful in many video/image processing and comunications applications [4], [5]. Both subranging and pipeline have advantages of high resolution for lower power and high speed. However these ADC's are often more complex to design. In most Pipeline ADC's, opamps generate a residue that drives the next stage. The settling time of the residue amplifier can limit the speed of the pipeline ADC. Similary the speed of the subranging ADC is limited by the settling time of the reference voltages and the preamplifier speed. The folding ADC implemented in this paper is an alternative approach that combines the simplicity of a flash with the residue type operation of a pipeline ADC [6]-[8]. Many innovative ideas have been published to improve the architecture such as resistive interpolating [8], averaging and cascading [10]. This paper describes current-mode interpolation to reduce the number of preampifiers and increase the conversion speed over resistive interpolation.

In section II, we begin with a general description of folding and interpolation and then describe the proposed architecture. Details of the system implementation are covered in section III. Section IV contains a yield analysis and section V summarizes the experimental results.

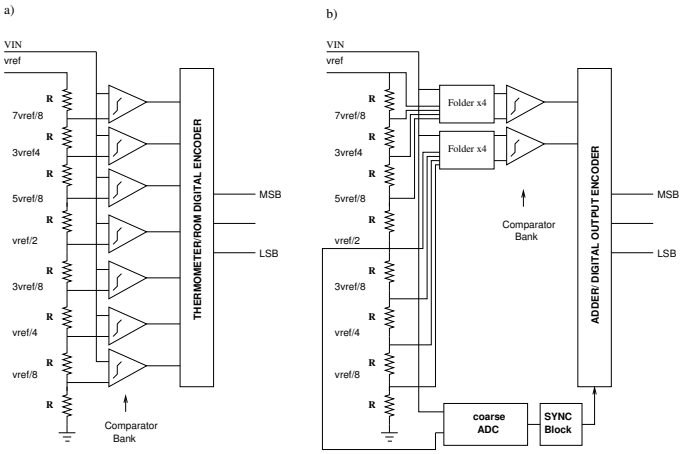

Fig. 1. Principle of folding.

II. Principles of Folding and Interpolation

In essence , folding is a technique to reduce the number of comparators used in the flash architecture. The folding concept dates back to the earliest data converters [9]. The folding ADC inherits the parallel nature of the flash ADC and can be described in the context of simple flash A/D converter. Fig. 1a) shows a 3 bit flash ADC. It compares the input to 7 reference voltages. A decoder decodes the thermomenter code, generating a 1 in n code which is passed to a ROM to produce a binary representation. The flash ADC has a comparator per code which gives the technique speed. At any given instance only a few comparators with trip voltages around the input provide useful information. The unused comparators are redundant. We can exploit this to reduce the number of comparators. This is done by modifying the flash into a folding ADC (see Fig 1b)). The motivation for folding is to achieve higher resolution than flash ADC's by reducing the number of comparators. The number of comparators required is reduced by the degree of folding. In Fig. 1b) we have added a folding block which does some analog pre-processing of the input and also a 1 bit coarse ADC. Fig. 2a) shows the output characteristic of a flash ADC and a folding ADC with a fold factor of 4. The folder block and fine ADC generate a sawtooth waveform and the coarse ADC determines if the output is above or below midscale. If the input is between $\frac{V_{ref}}{4}$ and $\frac{V_{ref}}{2}$ the coarse ADC outputs a zero (indicating the input is below mid-scale ($\frac{V_{ref}}{2}$)). From the negative slope in the triangle characteristic of the fine ADC, the encoder adds the number of zero's output from the comparators to get its 2 bit result. Similarly if the input is

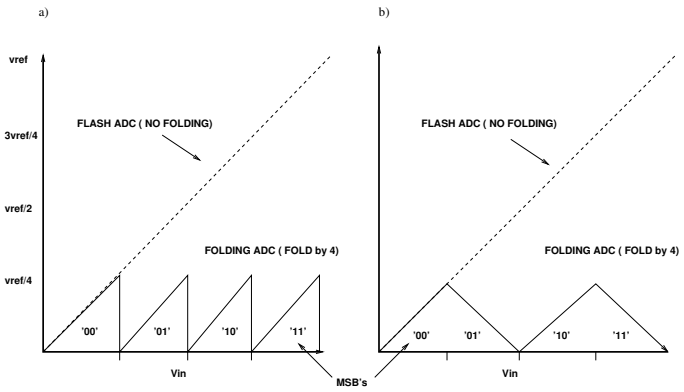

Fig. 2. a) Sawtooth characteristic b) Triangular characteristic

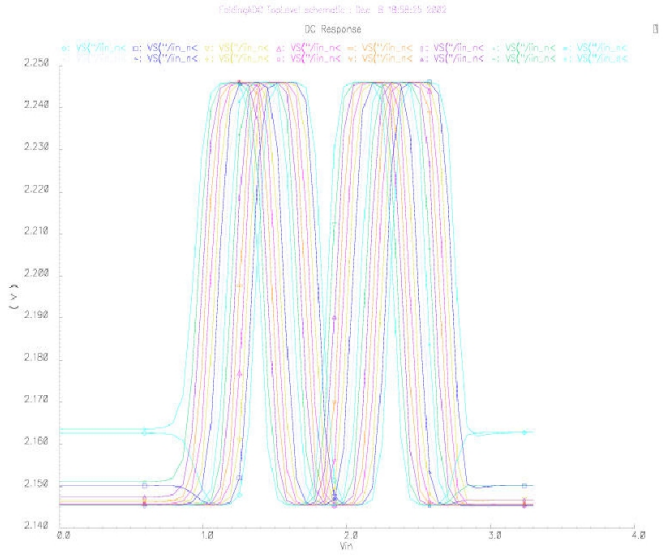

Fig. 3. DC Characteristic of the 16 folders

between zero and $\frac{V_{ref}}{4}$, the triangle characteristic is positive and the encoder adds the number of one's output from the comparators to get the 2 bit result.

In practice the sawtooth waveform is difficult to generate [11]. A triangular waveform is easier to produce (Fig. 2b)). In practice its corners tend to become rounded as the input changes with time. Fig. 3 shows the output of each folding block for a fold factor of four with a ramp input. Each folder output is shifted by an LSB and is input to a comparator. It can be difficult to control the thresholds of the folders over process and temperature. A more robust approach is to compare each folder output with its compliment, where the differential outputs of the offset folder blocks are input to two differential comparators. Our design uses a fold factor of 4. This requires 64 comparators and 64 folders to generate an 8 bit result (256 codes). The complexity, area and power consumption may not be significantly reduced compared to a flash implementation. We can reduce the number of folding blocks by using interpolation to generate the intermediate folded signals. The reduction in folding stages reduces the input capacitance of the converter and folder offset errors are averaged among the interpolated signals. The final design contains 16 Folders, each folding by 4, and an interpolation factor of 4. The 64 comparators produce a 7 bit cyclic thermometer code.

III. Circuit Analysis

The overall system architecture is shown in Fig. 6. The folders/interpolators and encoder realise the fine ADC. Several offset parallel folders generate differential folding signals, and interpolation increases the number of folding signals available for comparison. The comparator outputs form a cyclic code which is passed through a bubble encoder to remove bubbles and sparkles. The fold factor in this design is four. This means the input is partitioned into quadrants whereby the 7 bit fine adc operates on each quadrant. A coarse encoder is required to determine in which quadrant the input lies.

A. Folder

The folder block is implemented as several differential pairs connected in an alternating fashion to a differential load, Fig. 4b). This approach works well in bipolar, but CMOS implementation is slow because of large drain capacitances. The speed can be increased by using low load impedance circuit techniques whereby the folder outputs are differential currents. This is the approach we adopted. These currents are fed into the interpolating block to generate intermediate folding signals. We further improved our speed by implementing the folder and interpolator in a single stage (Fig. III-Ab)). Performance can be improved by adding a preamplifier before each differential pair. This offers many advantages including small differential pair transistors with small drain capacitance. Transistor matching demands are reduced by the gain of the preamp. The preamp devices themselves can be large to produce good matching.

B. Interpolation

An 8 bit ADC with a fold factor of four requires 64 comparators and 64 folders. The folder overhead hardly justifies itself. To make the folding architecture efficient we reduce the number of folders through interpolation.

The principle of interpolation is illustrated in Fig. 4c). The superscripts enumerate the folds produced by folding amplifiers. By taking the average of two folds, a third can be created that will have a zero crossing midway between the zero crossings of the two primary folds. Interpolation factors higher than two can be realised. Resistive [7], current [6] or active [8] interpolation can be used to produce additional folds. Resistive interpolation uses resistor dividers to interpolate and has the disadvantage of requiring small resistors for high speed operation and needing fast drive circuits. In this work, a current division interpolation technique is used. Two folder output currents are divided into sixteenths by identical common source NMOS transistors. In Fig. 4b) we show interpolation by two which only

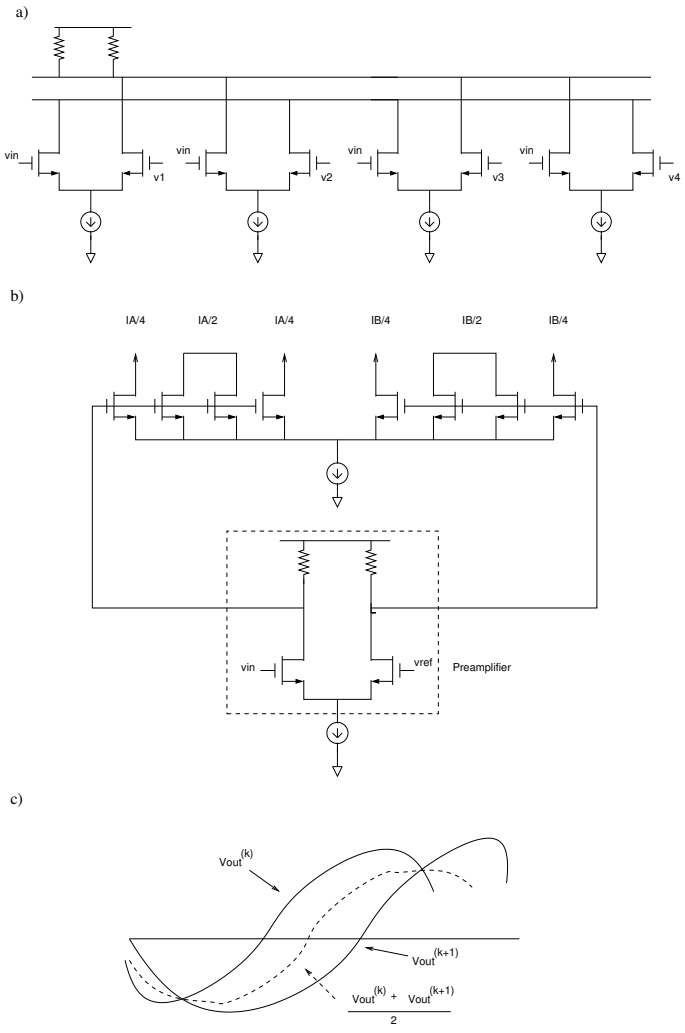

Fig. 4. a) Classical implementation of the folder. b) Inclusion of the current division c) Voltage interpolation

involves dividing the folder current by 4. $\frac{I_A}{4}$ is added to $\frac{I_A}{4}$ of the next folder to form the interpolated signal, except for the last folder which is connected to the first one in an inverse configuration (i.e. $\frac{I_A}{4}$ added to $\frac{I_B}{4}$). Higher orders require 2^n transistors. This is a significant disadvantage over voltage/resistive interpolation which only requires n resistors. On the other hand current interpolation is faster and can afford us a larger interpolation dynamic range. We've chosen to interpolate by 4 as a trade off between area, speed and accuracy. Errors in the interpolator adversly affect the bias and input signals. Reducing the bias current to the interpolators can remedy this by reducing feedthrough in the current sources and using large well-matched devices.

C. Comparator

A high speed differential current mode comparator was designed for this work, based upon [12]. The circuit is depicted in Fig. 5. The comparator consists of a differential input stage, two regenerative flip flops and an S-R latch.

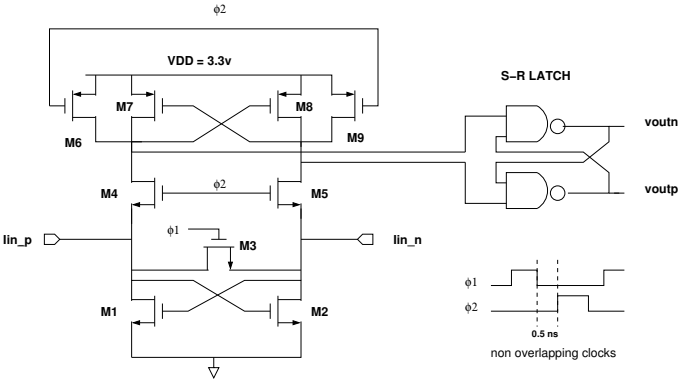

Fig. 5. Current comparator.

No offset cancellation is exploited, which reduces power consumption as well as die area and increases speed. A few modifications were made to the circuit. Kickback introduces distortion in the interpolator and can affect the zero crossings. It is reduced by passing the input currents through a low voltage cascode current mirror. In [12], regeneration takes place in two stages, when ϕ_1 closes , and when ϕ_2 opens. In Fig. 5 normal mode regeneration takes place in one clock phase, when ϕ_2 goes high. Prior to this no current flows through M_1 and M_2 therefore no ΔV exists between nodes I_{in_p} and I_{in_n}. The second latching stage M_7, M_8 serves to assist the regnerative latch M_1 and M_2 providing more gain and a faster decision. We found this alteration to [12] provided better performance as a current mode input or transimpedance comparator.

D. Encoder

The encoder required in a folding ADC is more complicated than encoders that might be used in a flash ADC. This difference is due to the fact that the comparators' outputs form a cyclic code instead of the standard thermometer code. The cyclic code contains one more bit of information than a thermometer code, thus the 64 comparator outputs from our fine ADC are actually encoded into 7 bits. This was accomplished by first XOR'ing comparator 64 (the top-most comparator connected to the folding block that hooks up to the highest reference voltage on the resistive ladder) with the remaining 63 comparator outputs. This encoding 63 bit result is in the form of a thermometer code that is converted to a 1-in-N code using an array of 63 3-input AND gates which provide the necessary bubble suppression [13]. The 63 bit 1-in-N code is fed into a ROM which decodes bits[5:0] from the 1-in-N code. A 63 bit cascaded adder could be used instead of a ROM to provide improved bubble suppression but at the cost of slower encoding speed and much higher power consumption. Bit[6] can be taken directly from comparator 64 and combined with bits[5:0] to form the lower 7 bits of the ADC output. It should be noted that there was no bubble suppression for comparator 64, and because this comparator is used to XOR with the lower 63 comparators (as well as for the sync block, described below), large ADC errors would occur if

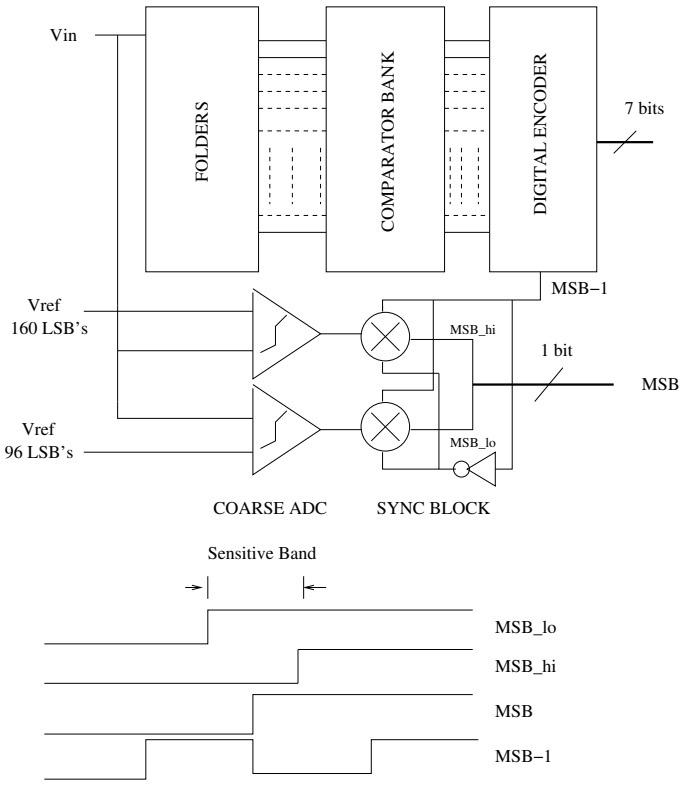

Fig. 6. System architecture.

there was an offset on this comparator. If there had been more time, digital logic could have been implemented to also include bubble suppression for this comparator's output.

E. Coarse ADC and Synchronization Circuit

The folding factor in this design was four. This means the input is partitioned into four quadrants whereby the 7 bit fine ADC operates on each quadrant (see Fig. 2b)). The triangle nature of the folded input means an additional bit (bit 7) can be obtained from the folder for the transition from a rising to a falling characteristic. This is possible because the encoder alters from counting one's output from comparators to counting zero's. We can detect this trip point and decode it as MSB-1. Because we lose 1 bit when we fold by four, a 1-bit coarse ADC is required to determine in which quadrant the input lies to form the MSB. A small voltage offset or timing discrepancy between the coarse and fine adc can cause large errors in the output code. Fig.7 shows a misalignment at the MSB transition codes for a 8-bit folding ADC with a 2-bit coarse ADC (our current design uses a 1-bit coarse ADC). One technique to avoid this synchronization problem is to build the coarse ADC with the same folder and interpolation blocks as the fine ADC and then use a sync block to align the digital outputs. The disadvantage of this technique is the hardware overhead. Instead we've used two comparators in a typical flash ADC configuration to determine the MSB transition point. These comparators are tied to the resitive ladder 32 LSB's above and below the reference voltage for

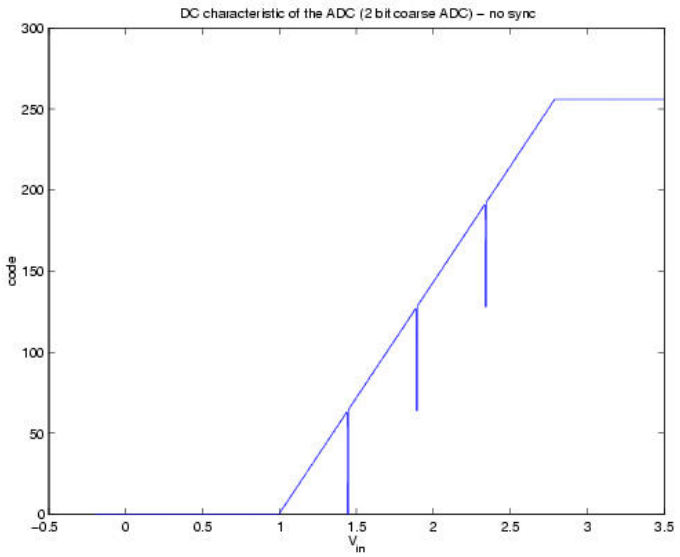

Fig. 7. 8-bit ADC with unsynchronized 2-bit coarse ADC.

the MSB and their outputs are MSB_hi and MSB_lo (see Fig. 6). Together they define a senstive band of 1/4th full scale around the MSB transition. If the input is within this band, the MSB should transition on the same edge as MSB-1 from the folding block ADC. This is achieved by selecting MSB_hi as the MSB if MSB-1 is 1, and MSB_lo if MSB-1 is 0.

IV. Matching, Yield and Distorsion

The previous section neglected to mention many of the effects that degrade the performance of folding and interpolating ADC's.

A. Folder

The CMOS folder DC characteristic is composed of segments that correspond to a differential pair transfer characteristic. For a given input, all but one of the differential pairs in a folder are saturated. The one active differential pair produces the shape of the fold around the reference voltage at its input. MOS transistors have a square law I-V characteristic, so the diff pair characteristic is

$$I_{out+,diffpair} = \frac{I_{tail}}{2} + \frac{\mu_n C_{ox}}{4}\frac{W}{L}(V_{in} - V_{ref}) \times\sqrt{\frac{4I_{tail}}{\mu_n C_{ox}(W/L)} - (V_{in} - V_{ref}^2)} \quad (1)$$

The voltage required to switch the diff pairs' current from one tail to another is

$$\Delta V = \sqrt{\frac{2I_{tail}}{\mu_n C_{ox}(W/L)}} \; or \; \sqrt{2}(V_{gs} - V_t) \quad (2)$$

Equation 1 shows that the diff pair introduces some nonlinearity. But as we are only interested in the zero crossings, i.e. the points where $V_{in} = V_{ref}$, this does not play an important role.

B. Input Mismatch

Mismatches in the input source coupled pairs are of significant importance as they influence the folder crossover points. The predicted offset of a source coupled pair is:

$$\sigma(V_{scp}) = \sqrt{2}\,\sigma(V_t) = \sqrt{2}\frac{A_{V_t}}{\sqrt{WL}} \tag{3}$$

Usng a preamplifier means the effective offset is divided by the gain of the preamp, $A_p = g_m R_l$. Therefore the input offset becomes $\frac{\sigma(V_{scp})}{A_p}$. Another advantage of the preamplifier is to provide more transconducance and make the switching of the differential pair sharper in Eq. 2. This is needed in order to avoid intersymbol interference and becomes critical as the supply voltage decreases when we want to keep the number of folds. The offset voltage in the preamplifier itself can be kept small because we use large transistors to provide enough transconductance.

C. Folder Current Source

Mismatch in the folder current source can cause an offset in the left and right current biases. Errors in these complimentary biases result in changes in the crossover points and DNL errors. The DNL error is reduced by interpolation.

D. Current Division Mismatches and Interpolation

The main source of error comes from the current division stage, i.e. from mismatches between the 16 transistors forming each side of one differential pair (same principle as on Fig. 4 b) with $2^4 = 16$ transistors needed to interpolate by 4). These mismatches translate into mismatches in the current division. Through one transistor on one side of the differential pair will flow a current:

$$I_i = I_{mean}(1 + \frac{\sigma(I_{mean})}{I_{mean}}) \tag{4}$$

$$\frac{\sigma(I_{mean})}{I_{mean}} = \frac{2\sigma(V_t)}{V_{gs} - V_t} = \frac{A_{V_t}\sqrt{2\mu_n C_{ox}}}{L\sqrt{I_{mean}}} \tag{5}$$

Figure 8 shows the Monte-Carlo simulation considering only the mismatches in the current division, and plots the yield of the ADC for increasing $\frac{\sigma(I_{mean})}{I_{mean}}$.

In the process of our design, we first tried to increase the width of the transistors to improve the yield. But Eq. 5 shows that the standard deviation does in fact depend only on the length of the transistors. Transistor length appeared to be critical in the design of the folder, and increasing it was a major problem, even for a folding factor of 4 where the speed needed at the output of the folder was not as high as for a larger folding factor (recall that this frequency will be more than n times the frequency of the input for a folding factor of n). Therefore our design still needs to be optimized to produce a better yield.

V. Test Results

Several tests were performed to verify that our 8-bit Folding ADC was working properly. First, a slow ramp

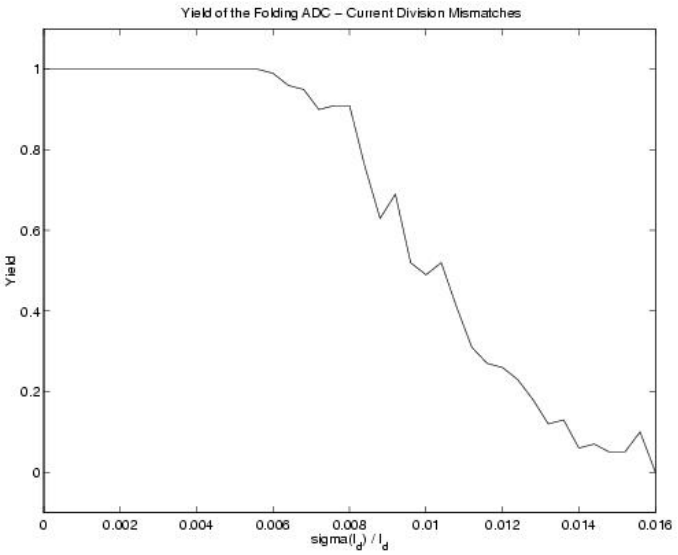

Fig. 8. Monte-Carlo simulation of the ADC for the yield, function of current division mismatches.

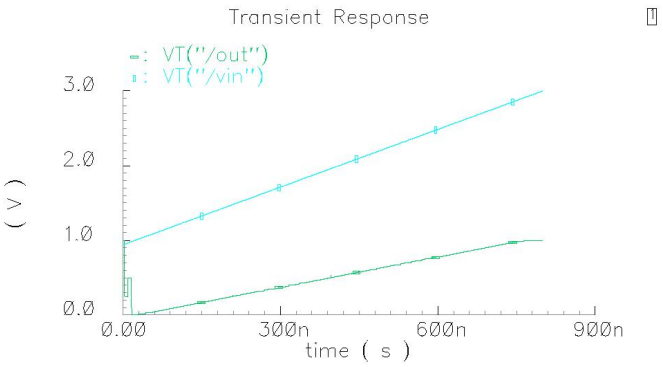

Fig. 9. ADC output with ramp input.

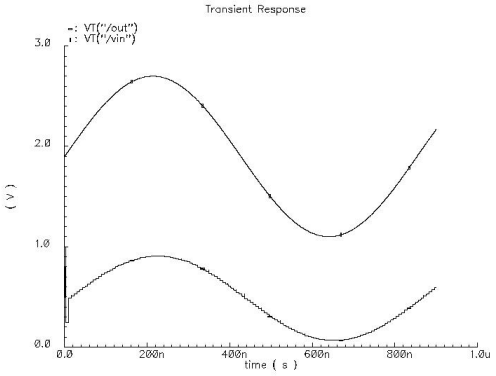

Fig. 10. ADC output with sine wave input.

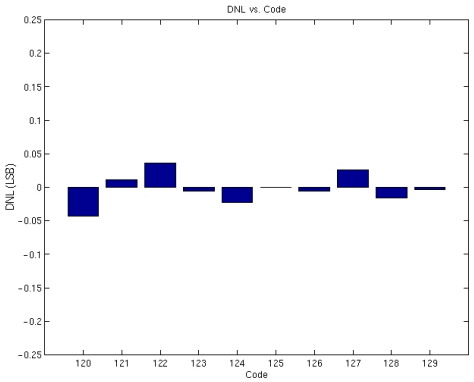

Fig. 11. DNL for 10 codes around the MSB transition

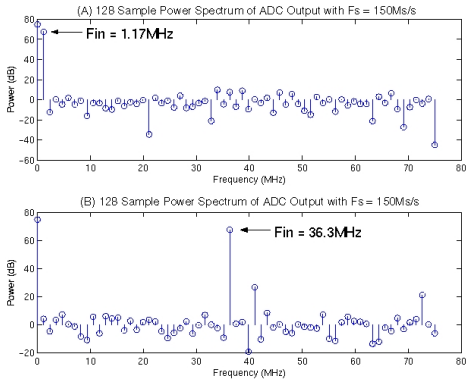

Fig. 12. FFT for (A) $f_{in} = 1.17MHz$ and (B) $f_{in} = 36.3MHz$

input was applied to the ADC from slightly below the input range (940 mV) to slightly above the input range (3.00 V) and the results are shown in Fig. 9. An ideal DAC was used on the output of the of the ADC, and its LSB size was less than that of the ADC which is why the DAC output ramp has a different slope from the input ramp. Similarly, a slow sine wave input was applied to the input to give a sine wave output of the same frequency (Fig. 10). The slight glitch at the beginning of each of these output plots is because the comparators have not yet sampled the input. The output is slightly shifted to the right due to a finite delay through the ADC.

A DNL plot (Fig. 11) of the ADC was obtained by applying a very slow input ramp that stretched from code 120-129. The slope of the ramp was such that each code was sampled approximately 20 times before the transition to the next code occured. This allowed us to calculate the input voltage at each transition and from this we calculated the DNL. The DNL simulation for just 10 codes took 8 hours and so it was impractical to simulate all 256 codes. We chose the range of the DNL simulation so that the MSB transitions from 0 to 1 (MSB=0 for code 128, MSB=1 for code 129). This tests our synchronization block and as can be seen from the DNL plot, it seems to work properly because the DNL for these codes is small. The maximum DNL observed was -0.043LSB which is very good, however, it is likely that the DNL just happens to be low for the codes of the ADC that were simulated and that other codes have a higher DNL.

The SNDR, ENOB, THD, and SFDR were tested using coherent sine wave inputs of 1.17MHz and 36.3MHz with amplitudes slightly less than full scale. The power spectrum of each frequency is plotted in Fig. 12. A well-defined peak at the fundamental frequency is clearly visible in these plots. Table I shows various parameters of the ADC tested across process corners and with the two input frequencies mentioned above. The slow process corner was tested using typical models because the slow models caused the folders to malfunction, however, this could be fixed with some tweaking of the transistors in the folder circuits. As can be seen, the ENOB is very good for low input frequencies and gradually rolls off as the input frequency increases. At high input frequencies and at the slow corner, the ENOB is very poor probably due to malfunction of the folders. The THD includes the 5 harmonics after the fundamental.

Originally, a folding factor of 8 was attempted in our design, but after extensive simulation it was determined that this was impractical because the voltage range between two successive crossings of one folder was too small for a $V_{dd} = 3.3V$. Thus, we redesigned the ADC to fold 4 times at the expense of doubling the number of folders and comparators from 32 to 64. The output encoder was originally a cascaded adder, but was replaced with a ROM to reduce the total power consumed. The total power dissipated by our 8-bit folding ADC is 150mW.

VI. Conclusion

We implemented and simulated an 8 bit CMOS folding and current interpolating ADC architecture. Our design specifications (table II) were easily achieved in the typical and fast process corners, and additional tweaking of the folders should fix the problems with the slow corner and with the yield. A distributed sample and hold architecture [8] could improve our ENOB for higher input frequencies. The fully differential analog blocks in this implementation are less sensitive to switching noise. Current mode circuits also generate less noise [14] than voltage techniques. No ideal components were used in our ADC design other than two current sources, two voltage soures, and a single clock source.

References

[1] S. Tsukamamoto, T.Endo, and W. Schofield, *A CMOS 6 bit 400-Msamples/s ADC with error correction*, IEEE J.Solid State Circuits, vol. 33, pp. 1939-1947, Dec. 1998

[2] Y. Tamba and K Yamakido, *A 6 bit 500-Msamples/s ADC for a hard disk drive read channel*, ISSCC Dig. Tech. Papers, Feb. 1999, pp. 324-325.

[3] K. Yoon, S.Park, and W. Kim, *A 6 bit 500-Msamples/s CMOS flash ADC with a background interolating auto zero technique*, ISSCC Dig. Tech. Papers, Feb. 1999, pp. 326-327.

[4] A. Abo and P. Gray, *A 1.5-V 10-bit 14.3 Ms/s CMOS pipeline analog-to-digital converter*, IEEE J.Solid State Circuits, vol.34, pp. 599-605, May. 1999.

[5] B. Brandt and J. Lutsky, *A 75 mW 10-b 20- MSamples/s CMOS subranging ADC with 59 dB SNDR,* , in ISSCC Dig. Tech. Papers, Feb. 1999, pp. 322-323.

[6] M. P. Flynn and D. J. Allstot,*CMOS Folding A/D Converters with Current-Mode interpolation* in IEEE J. Solid State Circuits , vol. 31, no. 9, Sept. 1996

TABLE I
PERFORMANCE OF ADC ACROSS PROCESS CORNERS.

	Slow Corner $T = 80^oC, V_{dd} = 2.97V$		Typical Corner $T = 27^oC, V_{dd} = 3.30V$		Fast Corner $T = 27^oC, V_{dd} = 3.63V$	
	$f_{in} = 1.17MHz$	$f_{in} = 36.3MHz$	$f_{in} = 1.17MHz$	$f_{in} = 36.3MHz$	$f_{in} = 1.17MHz$	$f_{in} = 36.3MHz$
SNDR	$49.01dB$	$11.88dB$	$49.20dB$	$39.16dB$	$49.10dB$	$43.33dB$
ENOB	$7.85bits$	$1.68bits$	$7.88bits$	$6.21bits$	$7.86bits$	$6.91bits$
THD	$61.91dB$	$25.70dB$	$62.30dB$	$39.66dB$	$62.15dB$	$44.54dB$
SFDR	$60.10dB$	$25.29dB$	$57.67dB$	$40.82dB$	$57.56dB$	$44.92dB$

TABLE II
PERFORMANCE OF FOLDING ADC'S.

Parameter	This ADC	Flynn et al. [6]	Venes et al. [8]
Technology	$0.35\,\mu m$ CMOS	$1\,\mu m$ CMOS	$0.5\mu m$ CMOS
Supply Voltage	3.3 V	3.3 V	3.3 V
ENOB	7.88 bits	7.11 bits	7.5 bits
Sampling Frequency	150 Ms/s	125 Ms/s	80 Ms/s
SNR	49.2 dB	44.6 dB	44 dB ($f_{in} = 75\,MHz$)
DNL	.0043 LSB (over 10 codes)	-	-
SFDR	57.67 dB	-	-
THD	62.30 dB	-	-
Analog Input Range	1.9 V	-	1.6 V
Input Capacitance	-	5 pF	2 pF
Area	$2\,mm^2$	$4\,mm^2$	$0.3\,mm^2$
Power	150 mW	150 mW	80 mW

[7] M. P. Flynn and B. Sheahan, *A 400-Msamples/s, 6-b CMOS Folding and Interpolating ADC* in IEEE J. Solid State Circuits , vol. 33, no. 12, Sept. 1998

[8] A. G. W. Venes and R. J. van de Plassche, *An 80 Mhz, 80-mW , 8-b CMOS Folding A/D Converter with Sistributed Track-and-Hold Preprocessing*, IEEE J.Solid States Circuits , vol.31, no. 12, Dec. 1996.

[9] B.D. Smith, *An unusual electronis analog-digital conversion method*, IRE Trans. Instrum., vol. 5, pp. 155-160, June 1956.

[10] K. Bult and A. Buchwald, *An embedded 240 mW 10-b 50 Ms/s CMOS ADC in 1-mm2* IEEE J.Solid State Circuits, vol 32, pp 1876-1886, Dec. 1997

[11] R. van de Grift and R. J. van de Plassche, *A monolithic 8 bit video A/D converter* IEEE J.Solid State Circuits, vol SC-19, pp 374-378, Dec. 1994

[12] G. M. Yin and W. Sansen, *A High-Speed CMOS Comparator with 8-b Resolution*, IEEE J.Solid State Circuits, vol 27, no. 2, Feb. 1992

[13] C. W. Mangelsdorf, *A 400MHz input flash converter with error correction*, IEEE J.Solid State Circuits, vol SC-25,pp. 184-191, Feb. 1990

[14] U. Fiedler and D. Seitzer, *A high-speed 8 bit A/D converter based on multiple folding circuit*, IEEE J.Solid State Circuits, vol SC-14,pp. 547-551, June 1979