An 8-b 650-MHz Folding ADC

Johan van Valburg and Rudy J. van de Plassche, *Fellow, IEEE*

***Abstract*—An 8-b 650-MHz folding analog-to-digital converter (ADC) with analog error correction in the comparators is presented. With an input frequency of 150 MHz, 7.8 effective bits are obtained. The ADC is implemented in a 1-μm 13-GHz triple-level interconnect bipolar process, requiring 850 mW from a single −4.5-V supply. The die size is 4.2 mm^2.**

I. Introduction

The continuously increasing sampling frequencies of ADC's, together with the demand for low-power devices, ask for converter structures with a reduced number of power consuming comparators. The most straightforward technique to convert an analog signal into an 8-b digital output code is used in the so-called full parallel or flash ADC. This structure requires 255 comparators, together with a set of 255 reference voltages to define all the quantization levels of the ADC. If the input signal V_{in} exceeds a reference voltage of a comparator, the output of that comparator will be high. At the output of the comparator block, a linear, or thermometer, code is present, where the place of transition from high outputs to low outputs is related to the value of the input signal V_{in}. The binary encoder converts the 255-b code into eight binary signals. Power consumption and chip area required for the implementation of a flash ADC are practical limits at higher sampling rates.

The two-step structure reduces the number of comparators drastically. First, a coarse decision is made using a 3-b flash ADC. This decision is converted back into an analog value using a digital-to-analog converter (DAC) and substracted from the original applied input signal. The residual signal is then converted into 32 levels using a 5-b flash ADC. The analog signal to be converted is held at a constant value by a sample-and-hold amplifier (S/H). The number of comparators is reduced from 255 to 38, which reduces power consumption as well as chip area. The two extra blocks, the DAC and S/H, are hard to design for the intended sampling frequencies. The DAC must have a linearity of at least 8 b to avoid limiting the performance of the device and the S/H should have very small settling and aperture times.

The folding structure (Fig. 1) realizes the reduction of comparators using analog preprocessing on the input signal V_{in}. Fine and coarse information are generated independently of each other, so this structure does not require a S/H and a DAC.

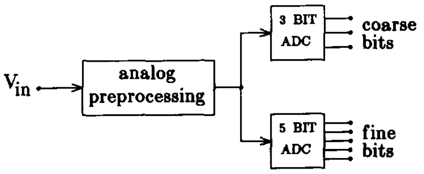


Fig. 1. Folding structure.

II. Analog Preprocessing

In a flash ADC, a differential pair is a basic element for a comparator, where in total 255 of those pairs are required to define all the code transitions. In the analog preprocessor of a folding ADC, a coupled differential pair (CDP) is a key element. Fig. 2 shows the dc behavior of a CDP, where the output V_{out} is high if $V_{low} < V_{in} < V_{high}$.

Two CDP's can be used to implement a differential output 1-b ADC (Fig. 3). The lower CDP generates a high output if $0 < V_{in} < 0.5$. At an input range of 0 to 1 V, this output can be defined as the inverted version of the most significant bit (MSB). The upper CDP is responsible for the generation of the MSB.

Four CDP's can be configurated in such a way that they generate signals to implement a 2-b ADC. Fig. 4 shows that each of the outputs has a unique active region of 250 mV. Combining the outputs C and D defines the MSB and the combination of B and D generates the MSB-1. Similar combinations can be made to generate the inverted versions to implement differential versions for MSB and MSB-1.

A 3-b ADC can be realized using eight CDP's with reference steps of 125 mV. Combination blocks are used to generate the MSB, MSB-1, and MSB-2. Only the differential version of MSB-2 is shown in Fig. 5. To further simplify the explanation, a set of eight CDP's, together with the combination block to define MSB-2, is called a folding block (FB). The designer of a folding ADC has the freedom to increase the resolution by using more than eight CDP's in a folding block. There is, however, a practical limitation using that technique. The MSB-2 pattern has approximately an eight times higher frequency than the frequency of the input signal V_{in}. Converting, for instance, a 250-MHz sine wave gives a 2-GHz signal after analog preprocessing. This factor of 8 is called the folding rate of the ADC.

Parallel use of folding blocks increases the resolution of the ADC without increasing the folding rate of the system. In Fig. 6, the upper folding block generates a differential signal MSB-2, as explained before. The second

Manuscript received April 27, 1992; revised July 20, 1992.
The authors are with Philips Research Laboratories, 5600 JA Eindhoven, The Netherlands.
IEEE Log Number 9203622.

0018-9200/92$03.00 © 1992 IEEE

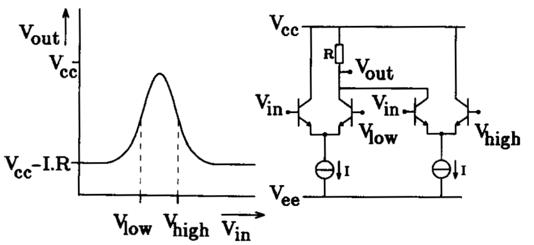


Fig. 2. Coupled differential pair.

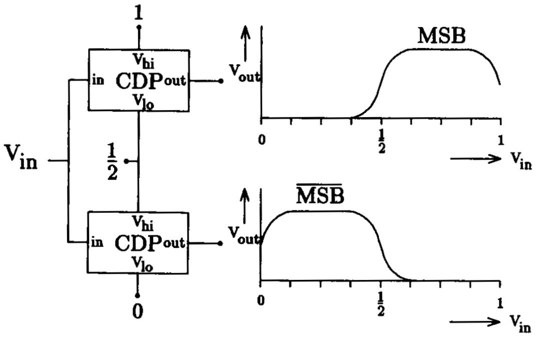


Fig. 3. 1-b ADC.

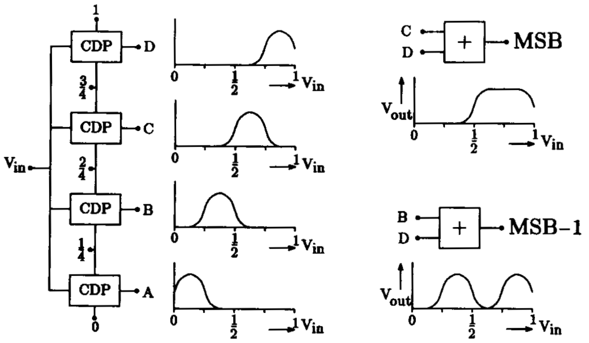


Fig. 4. 2-b ADC.

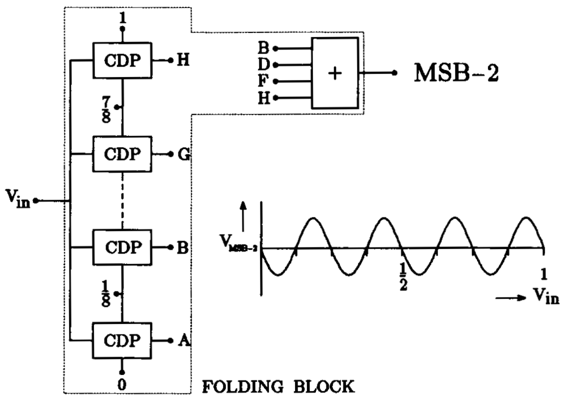


Fig. 5. 3-b ADC.

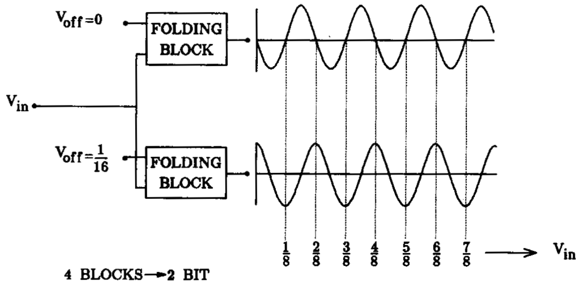

V_{off}=0
FOLDING BLOCK
V_{in}
$V_{off}=\frac{1}{16}$
FOLDING BLOCK
$\frac{1}{8}$ $\frac{2}{8}$ $\frac{3}{8}$ $\frac{4}{8}$ $\frac{5}{8}$ $\frac{6}{8}$ $\frac{7}{8}$
V_{in}
4 BLOCKS→2 BIT

Fig. 6. Parallel use of folding blocks.

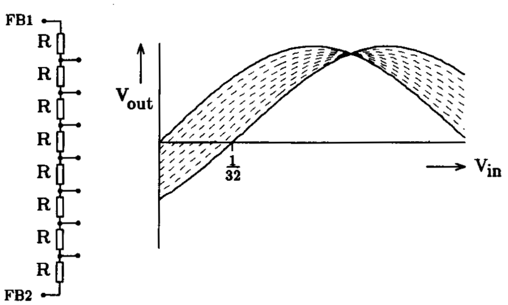


Fig. 7. Resistive interpolation.

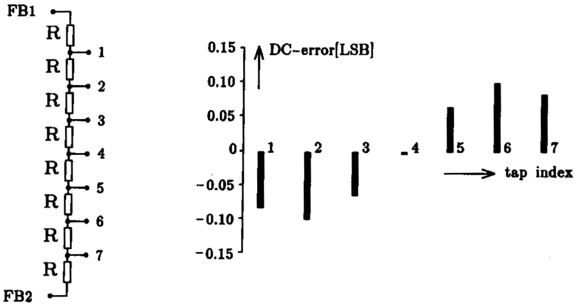


Fig. 8. DC error interpolation.

folding block has exactly the same analog behavior, but uses reference voltages with an offset of 1/16 V. It becomes clear that this second folding block defines codes between those of MSB-2. In this way the resolution is increased by 1 b. In previous designs of folding ADC's [1], [2], eight of those folding blocks were used to increase the resolution. In this device, power consumption, chip area, and input capacitance are reduced by using only four folding blocks in parallel to increase the resolution up to 5 b.

An 8-b ADC can be implemented using 32 folding blocks in parallel with offset steps of 1/256 V. The complexity of such a system would be comparable to a flash ADC. A solution to increase the resolution up to 8 b is resistive interpolation. Fig. 7 shows the concept of that technique. Two outputs of folding blocks, together with eight resistors, are used to generate the intermediate code

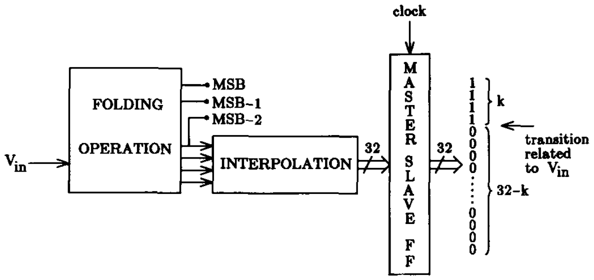


Fig. 9. Fine bit generation.

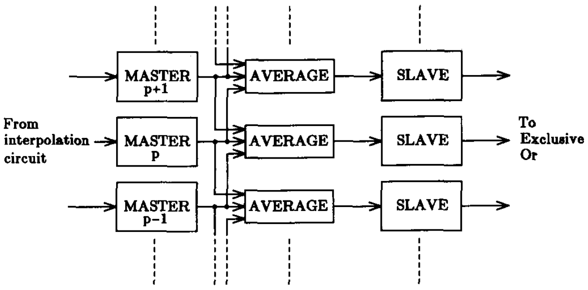


Fig. 10. Analog averaging.

transitions. The number of resistors used in one set is called the interpolation rate and equals eight in this case. In total, four sets of resistors are used to interpolate between FB1-FB2, FB2-FB3, FB3-FB4, and FB4-FB1. Because of the nonlinearity in the dc transfer curve of a CDP, there is a small dc error in the codes defined by the resistive interpolator. Fig. 8 shows this error, expressed in LSB units. The error, which never exceeds ±0.1 LSB, is acceptable for most of the applications.

III. Fine Bit Generation

After interpolation, 32 differential wave patterns are available, which contain all the information to generate the five least significant bits of the ADC. A set of 32 master-slave flip-flops (MSFF) transforms the analog information into digital data. At the output of the MSFF block a cyclic code is generated, where k MSFF's generate a digital ONE and $(32 - k)$ MSFF's generate a digital ZERO. The transition from ZEROS to ONES is, as in a flash ADC, related to the input signal V_{in} (Fig. 9). Very similar techniques are used to extract the five fine bits out of this 32-b cyclic code. In the first place, the exact transition from the ZERO block to the ONE block is located using an EXCLUSIVE-OR function (XOR) on the outputs of the MSFF block. The resulting 1-out-of-32 code is an input for a very simple binary encoder to derive a 5-b pattern. Limited bandwidth, noise, crosstalk, etc. may corrupt the operation of the binary encoder. A decision error in a comparator may cause more than one transition at the outputs of the MSFF block. A non-1-out-32 code, which appears after the XOR's, cannot be interpreted by the binary encoder. Digital error correction techniques have been published [3] and implemented to protect a binary encoder for so-called bubble errors in a linear or a cyclic code. In this device, analog error correction techniques are used.

IV. Analog Error Correction

The technique is based on averaging a set of three outputs of masters before feeding the information into the slave. Fig. 10 shows a part of the MSFF block using this averaging algorithm. If the output of master P is enclosed by outputs *both* having an inverted output compared to output P, the master decision of section P is completely overruled. The implementation of this technique is shown in Fig. 11. The master part of the flip-flop is symbolized by the master block. If the master information is latched (CLK = low), the slave samples the output of this master. The current I_{slave} flows, as a function of the master decision, through one of the two triple transistor configurations. The triple configuration is used to divide the current I_{slave} into three equal parts I_{P-1}, I_P, and I_{P+1}, all equal to $I_{slave}/3$. Partial current I_P is used in this slave, I_{P-1} is used in the lower adjoining slave, and I_{P+1} is used in the upper adjoining slave section. Using this current distribution through the whole MSFF block, the two current adding points Σ are receiving partial currents from master P, master $P - 1$, and master $P + 1$. The slave makes a decision based on the average of three master decisions. Fig. 11 shows details of flip-flop P. The currents in the resistors $R1$ and $R2$ as a function of three consecutive master decisions is shown in Table I. Looking at the shaded combinations, defined as decision errors of master P, the actual master decision of flip-flop P is overruled and the bubble error is removed from the code. The advantages of this analog averaging technique become clear looking at the desired elements used for implementation. Only four extra transistors per comparator are needed, where the power consumption is not increased at all be-

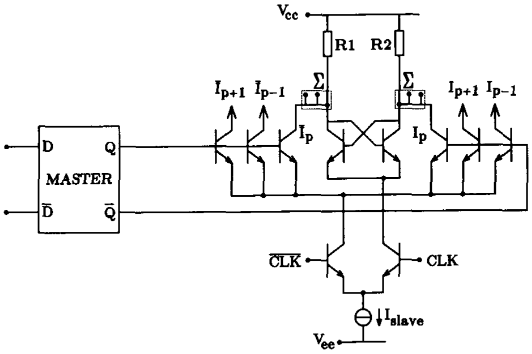


Fig. 11. MSFF with analog averaging.

TABLE I
CURRENT DISTRIBUTION

$MASTER_{p-1}$	$MASTER_p$	$MASTER_{p+1}$	I_{R1}	I_{R2}	decision slave p
0	0	0	$0 I_{slave}$	$\frac{3}{3} I_{slave}$	0
0	0	1	$\frac{1}{3} I_{slave}$	$\frac{2}{3} I_{slave}$	0
0	1	0	$\frac{1}{3} I_{slave}$	$\frac{2}{3} I_{slave}$	0
0	1	1	$\frac{2}{3} I_{slave}$	$\frac{1}{3} I_{slave}$	1
1	0	0	$\frac{1}{3} I_{slave}$	$\frac{2}{3} I_{slave}$	0
1	0	1	$\frac{2}{3} I_{slave}$	$\frac{1}{3} I_{slave}$	1
1	1	0	$\frac{2}{3} I_{slave}$	$\frac{1}{3} I_{slave}$	1
1	1	1	$\frac{3}{3} I_{slave}$	$0 I_{slave}$	1

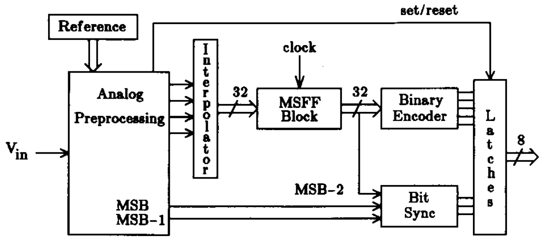


Fig. 12. System overview.

cause this technique makes use of an existing current in the slave of the comparator.

V. SYSTEM OVERVIEW

Fig. 12 shows an overview of the complete 8-b folding ADC. To summarize the trade-offs that can be made in this design, all the steps are briefly shown in this section. In the first place, the reduction rate in the number of comparators equals the folding rate and is in this device equal to eight. The folding rate fixes the number of coarse bits, which easily can be derived from the system using combinations of outputs of CDP's within folding blocks. This number equals $\log_2$ (folding rate) = 3. Parallelism increases the resolution with $\log_2$ (parallel blocks), in this case, 2 bits and the 3 missing bits in the resolution are made up by the resistive interpolator, which is responsible for $\log_2$ (interpolation rate) bits.

One of the fundamental problems using analog preprocessing is the delay difference between MSB, MSB-1 and the other signals in the ADC. This is caused by the difference in analog components used in the generation of the two most significant bits. If this information directly would be distributed to the output latches, large code errors would appear at every transition of MSB and MSB-1. Before feeding the MSB and MSB-1 information to the output, they are first synchronized with the rest of the system using the BITSYNC cell, as explained in [2].

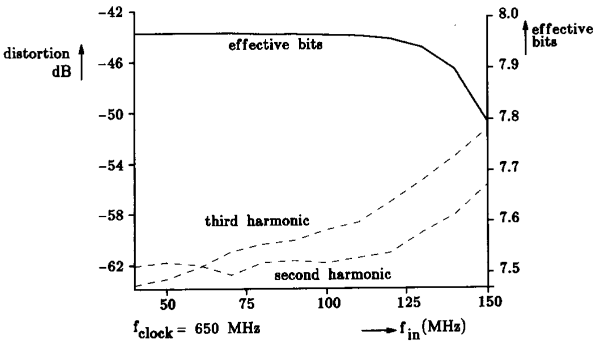


Fig. 13. Measurement results.

TABLE II
ADC SPECIFICATIONS

Resolution	8 b
Effective bits	7.8 b
Large signal analog bandwidth	150 MHz
Clock rate	650 MHz
Supply voltage	−4.5 V
Supply current	180 mA
Analog input	0 to −1 V
Linearity	< 0.5 LSB
Active die area	1.76×1.96 mm^2
Digital output	ECL 100K
Rise/fall time	750 ps
Number of elements	2500

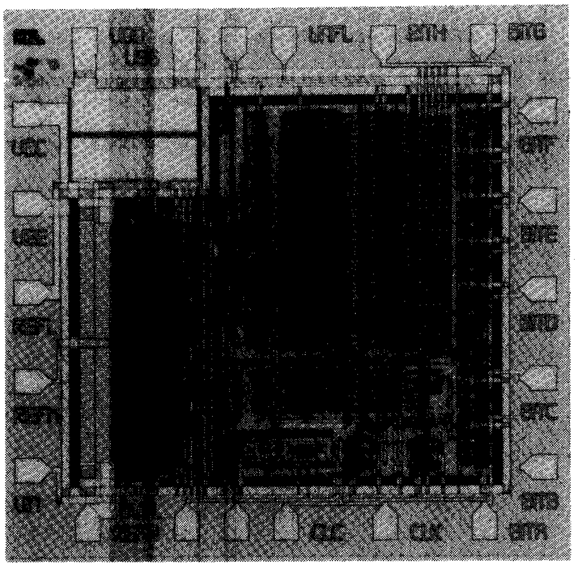


Fig. 14. Chip microphotograph.

If the input signal V_{in} exceeds the defined input range, the analog preprocessor also generates set and reset signals. The latches at the output of the system are then forced to ONE in case of overflow or to ZERO in case of an underflow.

VI. Measurements

During measurements, a low-distortion analog sine wave is applied at the ADC. The digital information is converted to an analog signal using a 1-GHz 8-b GaAs DAC. The output of the DAC is analyzed for quantization noise and distortion. The second and third harmonic distortion as functions of the analog input frequency are shown in Fig. 13. The sampling rate during this measurement was 650 MHz. An error in the BITSYNC cell limited the maximum analog input frequency to 150 MHz, where 7.8 effective bits are obtained. A complete list of specifications for the device is presented in Table II. A chip micrograph is shown in Fig. 14, where only 10% of the total chip area is used to implement the analog preprocessing.

Acknowledgment

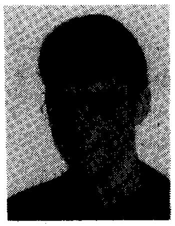
The authors would like to thank P. Baltus for his useful advice during the design of this ADC. Special thanks goes to A. van den Enden and to F. Bruekers for their educational advice during the preparation of this paper for the ISSCC 1992.

References

[1] R. J. v.d. Plassche and P. Baltus, "An 8-bit 100-MHz full Nyquist ADC," *IEEE J. Solid-State Circuits*, vol. 23, no. 6, pp. 1334-1344, Dec. 1988.

[2] R. v.d. Grift *et al.*, "An 8-bit video ADC incorporating folding and interpolation techniques," *IEEE J. Solid-State Circuits*, vol. SC-22, no. 6, pp. 944-953, Dec. 1987.

[3] Y. Gendai *et al.*, "An 8 bit 500 MHz ADC," in *ISSCC Dig. Tech. Papers*, Feb. 1991, pp. 172-173.



Johan van Valburg was born in Wadenoyen, The Netherlands, on January 3, 1963. He received the degree in computer engineering from the Hogere Technische School in Enschede, The Netherlands in 1986.

In 1987 he joined the Philips Research Laboratories in Eindhoven, The Netherlands, where he became engaged in research on adaptive filtering techniques. He is currently working on high-speed digital signal processing, with special interest in high-speed A/D conversion.

Rudy J. van de Plassche (M'83-SM'83-F'89) was born in Ijzendijke, The Netherlands, on September 24, 1941. In 1964 he graduated from Delft University of Technology. In 1989 he obtained the Ph.D. degree from the same university. The title of his thesis is "High-speed and high-resolution A/D and D/A conversion."

In 1964 he joined Philips Research Laboratories in Eindhoven, The Netherlands, where he was involved in circuit design for analog integrated circuits. His research interests are in the field of operational amplifiers, instrumentation amplifiers, analog multipliers, integrated reference sources, high-speed and high-resolution analog-to-digital and digital-to-analog converters, sample-and-hold amplifiers, and lately digital signal processing circuits and systems. In 1983 he transferred to Philips Research Laboratories, Sunnyvale, CA, where he became Group Manager of the Advanced Design group. In 1986 he returned to Philips Research Laboratories in Eindhoven, where he has since been a member of the Radio and Data Transmission Group, involved in the design of digital radio systems and digital signal processing. Starting September 1, 1989 he has been appointed as a part-time professor at the Eindhoven University of Technology in the Department of Digital Systems, responsible for telecommunication circuits. He holds 49 U.S. patents, has published more than 38 papers in IEEE Journals, and has presented papers at different international conferences.

Dr. van de Plassche was elected Fellow of the IEEE in 1989. He received the "Veder prijs" of the "Stitching Wetenschappelijk Radiofonds Veder" for the year 1988. Since 1983 he has been a member of the Technical Program Committee of the International Solid-State Circuits Conference. In this committee he served as chairman of the analog subcommittee. Today he is Secretary of the European ISSCC Program Committee. Furthermore, he is a member of the technical program committee of the European Solid-State Circuits Conference.