

A 16 bit 500ks/s 2.7V 5mW ADC/DAC in 0.8 μ m CMOS using error-correcting successive approximation

William G. Schofield b.schofield@fml.co.uk
Ian J. Dedic i.dedic@fml.co.uk
Andrew K. Kemp a.kemp@fml.co.uk

Fujitsu Microelectronics Ltd.
European Mixed Signal Design Centre
Network House, Norreys Drive
Maidenhead, Berkshire. SL6 4FJ
United Kingdom
Tel.: (01628) 504700 Fax: (01628) 504777

Abstract

An embedded 16 bit ADC/DAC for an analogue signal processor uses error-correcting successive approximation to increase speed by relaxing settling time requirements. Using a low power switched capacitor/resistor shared DAC, it can perform simultaneous A/D and D/A conversion at 500ks/s using 5mW at 2.7V and occupies 2mm² in 0.8 μ m CMOS.

Introduction

Recent work on high speed high resolution ADCs has concentrated on pipelined architectures (PADC) [1],[2]. These rely on accurate storage and processing of the analogue signal through each stage, requiring highly accurate gain stages or self/auto-calibration. They sample the input on each clock cycle (two phases, each of period T), with a conversion latency of N samples for an N bit ADC. Passing the signal from stage to stage requires settling to N bit accuracy on each phase, which for time constant t requires $tN \ln 2 = 0.69tN$. This gives a sampling period of $1.39tN$ and a latency of $1.39tN^2$, or $22t$ and $355t$ respectively for a 16 bit PADC. For N stages at least N amplifiers and comparators are required, increasing area and power compared to a successive approximation ADC (SA-ADC).

With SA-ADCs the signal must be sampled once to N bit accuracy in the input sampling phase, but this can be very fast since it only uses passive switches and capacitors if a charge-redistribution DAC is used, this also provides a sample-and-hold (S/H) function. The speed of the converter is limited by comparator switching speed and DAC settling time because these must settle and make a decision to N bit accuracy each clock cycle; converter linearity is only limited by the performance of the DAC. For a DAC time constant t, SA-ADCs have a conversion period and latency of $0.69tN^2$, or $177t$ for a 16 bit ADC, which is 8 times lower sampling rate than a PADC but half the latency (low latency is required for closed loop applications). The

simple circuits and small power and area of the SA-ADC makes it ideal for embedding into larger devices where power and area are more important than speed.

Error-correcting successive approximation (ECSA) addresses the issue of DAC/comparator settling time, giving an ADC with speed comparable to a PADC but power and area similar to a SA-ADC.

Error-Correcting Successive Approximation Algorithm

A conventional SA-ADC subtracts the sample from the DAC output, takes the sign of the result, and increments or decrements the DAC code on a bitwise basis by the current bit weight, which halves each cycle.

$$V_{in} > DAC, DAC = DAC + weight : V_{in} < DAC, DAC = DAC - weight$$

At the end of the conversion the DAC code is the conversion output, with the conversion taking N cycles to complete. Error correcting successive approximation (ECSA) adds extra cycles to the conversion to correct wrong decisions due to settling or comparator errors. The ECSA-ADC described here uses 2 clock cycles per bit (32 cycles for a 16 bit ADC), but each clock cycle can be made much shorter than the conventional SA-ADC, reducing total conversion time and latency. The ECSA-ADC changes the DAC code only if the result is larger than a threshold proportional to the current bit weight; two clock cycles are needed to make comparison with upper and lower thresholds, which is performed by adjustment of the DAC code upwards or downwards

$$V_{in} > DAC + thresh, DAC = DAC + weight : V_{in} < DAC - thresh, DAC = DAC - weight$$

The no-change state means that comparison errors are automatically corrected later in the conversion, unless they are very large. MATLAB simulations, figure 1, show that the ECSA algorithm maintains 0.5LSB DNL up to $t/T=0.79$, giving a conversion time and latency of $2tN/0.79=2.5tN$, or $40t$ for a 16 bit ADC. Allowing some additional time for sampling, the sampling rate is about 4x faster than the conventional SA-ADC, which is 2x slower than the PADC but with 4x lower latency. Because there are fewer circuits limiting the speed than in the PADC (no amplifiers or multiple stages), then much more power can be allocated to the single comparator and DAC. ECSA-ADCs can achieve similar sampling rates to PADCs with much lower latency, power and area.

Reference voltage input drive is easier because large switching transients from MSB decisions decay by the time LSB decisions are made. Lower noise than a PADC is easy to obtain because there are fewer noise sources (only kT/C from sampling and comparator input noise). Comparator noise density is low because the single comparator can have relatively high current, and noise bandwidth is low because the minimum DAC/comparator bandwidth is only $1/(0.79*2*\pi*T)=0.2/T$, or about 8x the sampling frequency for a 16 bit ADC.

Architecture

This ADC is embedded into a 0.8 μ m CMOS audio signal processor with a total power budget of 25mW at 2.7V [3]. The specification was to perform a 16 bit ADC conversion with 5mW average power consumption; the low latency and high conversion rate were required by the application which was handset active noise cancellation. Figure 2 shows the architecture of the charge redistribution DAC; the 6 MSBs use an array of 64 unit capacitors (CDAC) which also provides an input S/H function, and the 10 LSBs use a resistor DAC (RDAC) driving one of the unit capacitors; the 3 MSBs of the RDAC use 7 2R unit resistors, and the 7 LSBs use an R-2R ladder.

The overall INL depends on capacitor matching in the CDAC, so the switching sequence of the double poly unit capacitors can be chosen to minimise sensitivity to process gradients. Overall DNL depends on resistor matching in the RDAC, which uses wide common-centroid ion-implanted resistors. The settling time of the DAC for the critical LSB conversions is limited by carry glitches in the

sub-DAC, which can be reduced by adding feed-forward speed-up capacitors to the R-2R LSBs or slow-down capacitors to the segmented MSBs.

When sampling, the top plates of the capacitor array are connected to a reference voltage, and the bottom plates to the input. For conversion the bottom plates are switched to VREF, GND or the RDAC output, and the top plates are sensed by the comparator. The DAC is made differential by dividing the capacitor array into two sub-arrays C_p and C_m , each of 32 0.8pF unit capacitors with a common top plate; during the conversion the RDAC output drives one capacitor in either C_p or C_m . For a negative DAC full scale C_{m1-32} are switched to VREF, C_{p2-32} to GND and C_{p1} to the RDAC which is low. With increasing DAC code the RDAC ramps from GND to VREF-1LSB in 1LSB steps; then C_{p1} switches to VREF and C_{m1} to the RDAC output with the R-2R termination resistor is switched to VREF; then the RDAC ramps from VREF to GND. C_{m1} then switches to GND and C_{p2} to the RDAC, and this sequence repeats until positive full scale is reached.

16 bit resolution with a differential output is obtained using a single-ended RDAC, with no large transitions at the segment boundaries. The settling time of the whole DAC is limited by switch and RDAC glitch settling since this only drives a small unit capacitor, and can be a few nanoseconds. The only static power is that of the RDAC, which draws 220uA average from a 2.2V reference. The architecture of the DAC and the ECSA algorithm mean that a low-power low-noise differential comparator can be used, because very wide bandwidth is not required (theoretically 4MHz for a 20MHz clock rate and 500ks/s sample rate). This has 5 gain stages; 2 differential, 1 differential-single ended and two inverters, each with about 20dB gain and 20MHz bandwidth, and consumes 600uA. Large cascoded cross-coupled NMOS input devices are used to give low offset and noise.

The logic architecture uses a pre-empting state machine to determine the next code based upon the present cycle of the weight check and the last decision taken. The threshold register has three possible choices, including feeding the current output code back. This approach enables the algorithm to select either to keep the present output, restore the previous output or to use the present output modified to the next weight check. Fast carry look-ahead adders and transmission gate multiplexers are used to minimise propagation delay.

The CDAC switch decode and RDAC are not used while the capacitor array is sampling, so by duplicating the unit capacitors as shown in Figure 3 a simultaneous D/A function is available with little increase in power and area. Total power for the analogue circuits is $600+220=820\mu\text{A}$ (comparator + RDAC) which is 2.2mW at 2.7V. Using the ECSA algorithm, digital activity during conversion is limited to the bit weight being processed, with simple calculations. The size of the logic is approximately 1000 gates, with approximately 15% average activity at 20MHz giving 2.7mW; about 5mW total. The analogue silicon area is 1.5mm², figure 4.

Results

The converter was designed for very good small signal performance. Figure 5 shows a -72dBFS signal clearly being resolved at 500ks/s. SINAD for small signals, figure 6, shows SNR of 92dB or 0.6LSB RMS noise; which agrees closely with calculated kT/C and comparator noise. THD for larger signals is typically 0.03% (-70dB) because the embedded ADC was not laid out to be gradient insensitive, the THD is 30dB better than system specification. Measured capacitor array matching shows that -90dB THD for large signals should be achievable with layout changes; previous results [4] show that capacitor voltage coefficients will not limit THD.

References

[1] M.Mayes et. al., "Monolithic low-power 16b 1MSample/s Self Calibrating Pipeline ADC", ISSCC, pp.312-313, Feb. 1996

- [2] P.Yu et. al., "A 2.5V 12b 5MSample/s Pipelined CMOS ADC", ISSCC, pp. 314-315, Feb. 1996
- [3] I.Dedic et. al., "A 16b 100ks/s 2.7V 25mW ADC/DSP/DAC-Based Analog Signal Processor in 0.8um CMOS", ISSCC, pp. 96-97, Feb. 1997
- [4] I.Dedic, "A Sixth-Order Triple Loop Sigma-Delta CMOS ADC with 90dB SNR and 100kHz Bandwidth", ISSCC, pp. 188-189, Feb. 1994

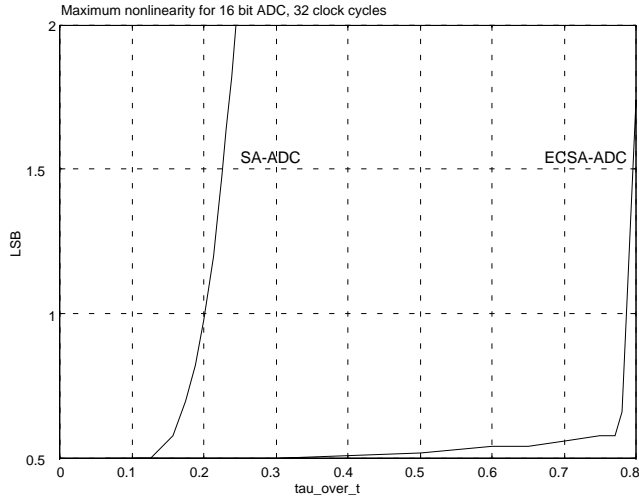


Figure 1. Matlab ECSA-ADC v SA-ADC

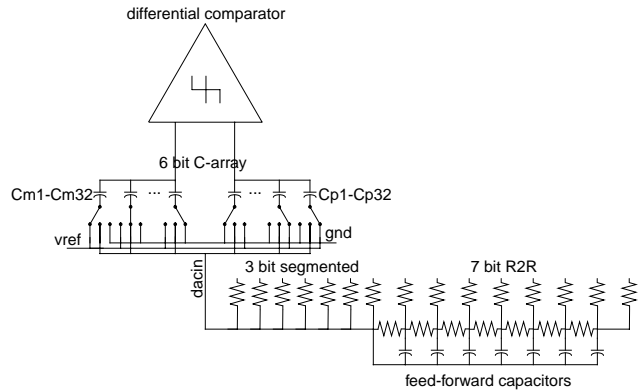


Figure 2. DAC Architecture

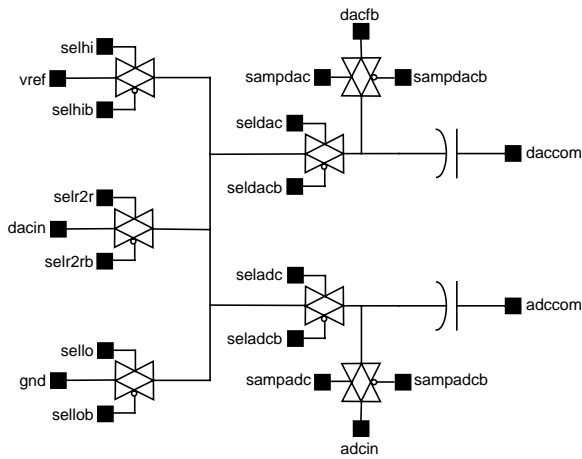


Figure 3. Capacitor Array Element

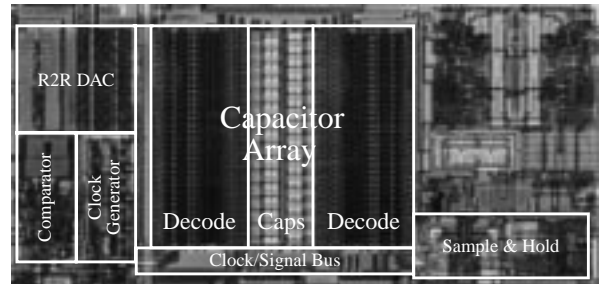


Figure 4. Converter Micro-Photograph

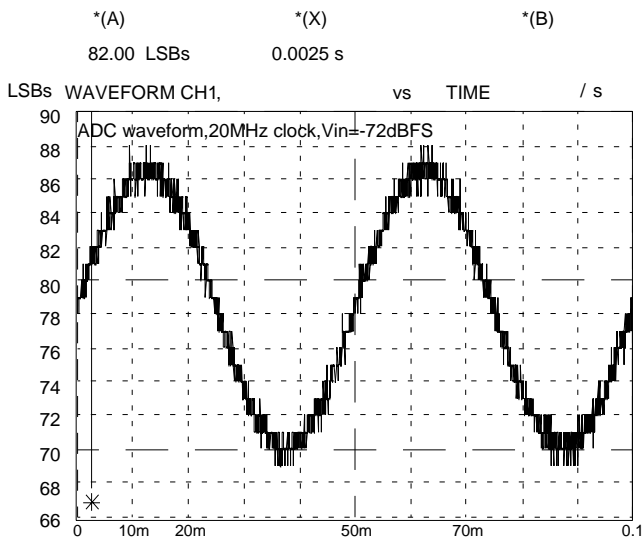


Figure 5. Small Signal Conversions

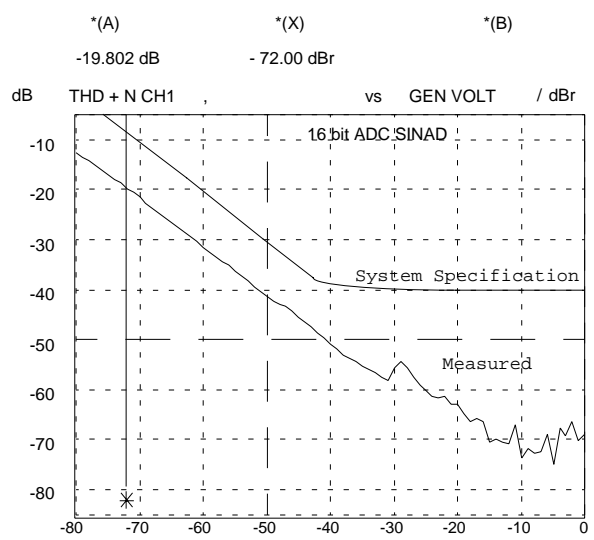


Figure 6. ADC SINAD