We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

20

Asynchronous Analogue-to-Digital Conversion Techniques

Nikos Petrellis, Michael Birbas, John Kikidis and Alex Birbas
Analogies S.A., Patras Science Park, Platani-Rio, 26504 Patras
Greece

1. Introduction

The sensor values readout is a critical issue in Factory Automation systems and control applications in general. Analogue-to-Digital Converters (ADCs) are the circuits that accept as input the analogue indication of the sensors and provide as output the corresponding digital code representation of the analogue value that can be exploited by the digital part of the controller. The most important parameters that affect the selection of an appropriate ADC are the conversion resolution, the sampling rate, the power consumption and the die area required. Several architectures have been proposed for ADCs that target different application requirements. For example, cable TV and PAL/NTSC decoders require 8-12bits resolution with 10-20MS/s sampling rate while high speed logic analyzers require 5-7bits resolution with multi-GS/s sampling rate. The industrial systems that require ADC conversion are production lines, command-control facilities, product quality measurement, communication networks, security systems etc. Industrial systems may use ADCs for the control of simple sensors like temperature, humidity, pressure, etc as well as for the interface of input devices with increased complexity like high speed and precision imagers.
The most popular ADC architectures are the Flash, Pipeline, Successive Approximation, Sigma-Delta and Folding-Interpolation ones. The Flash ADCs achieve the highest speed but occupy large silicon area and consume high power. For this reason, their resolution is practically limited to less than 8-bits. Furthermore, special analogue front-end components or technology processes need to be employed to design Flash ADCs that achieve multi-GS/s sampling rates. The Pipeline or Subrange ADCs consist of two or more ADCs with smaller resolution that operate on successive inputs in a pipelined manner and generate different groups of output bits. The throughput of pipeline ADCs is comparable to that of the Flash ADCs, the die area and power consumption are lower but the latency of a single sample is much longer. Counting and Successive Approximation ADCs consist of a digital counter that feeds a Digital-to-Analogue Converter (DAC) with increasing values until the analogue output of the DAC gets higher than the input. These ADCs require a very small number of components but need a variable large number of clock periods to reach a decision. Folding-Interpolating ADCs use a small number of comparators that compare the input successively with different sets of reference values. Finally, Sigma Delta ADCs are based on input oversampling. Parallel ADCs consist of multiple slower ADCs that are interleaved in time.

Within this context, we have developed an ADC architecture (patent pending) that extends the concept of pipeline ADCs through the employment of a binary tree structure that leads to the implementation of fast ADCs that occupy very small die area and have configurable resolution for achieving lower power consumption. The specific ADC architecture is based on the integer division of an analogue input by an appropriate power of 2. Specifically, a novel circuit has been designed that accepts as input an analogue value and generates the quotient and the residue of the integer division. Based on this principle, a Combo 4/8/12-bit ADC has been developed that operates in current mode in order to implement the functions of addition, subtraction, multiplication/division by a constant with high speed simple circuits that need low power supply. The required die area is only 0.12mm^2, the power consumption is 72mW and the average sampling rate exceeds 140MS/s for 12-bit resolution. No clock signal is required by that ADC due to its asynchronous nature. Thus, the ADC input can be connected either to a sample and hold circuit or directly to the input analogue signal. The ADC output can be latched using an independent clock.
The linearity errors are a major problem in any Analogue-to-Digital Conversion method. Several methods like trimming, real time calibration, generation of additional bits for error correction etc, have been employed for the improvement of the real time behaviour of an ADC. Beside these popular approaches, the techniques that have been used to enhance the linearity and the transistor mismatch problems in the proposed binary tree ADC architecture are furthermore discussed.
The most important quality metrics of an ADC are briefly presented in Section 2. The popular ADC architectures that were mentioned above are described in more detail in Section 3. The architecture of the proposed asynchronous ADC with binary tree structure along with implementation details are presented in Section 4. Finally, simulation results and a comparison with other ADCs are presented in Section 5.

2. ADC quality metrics

The most important parameters of an ADC that are taken into consideration in research papers as well as commercial products are the resolution, the speed, the power consumption, the various error metrics and the area required. The die area requirements of an ADC IP component are directly determined by the designed layout or the component count. Commercial ADC chips are characterized by the chip size, pin count etc, instead of just the die area. The average and maximum power consumption can be determined either in DC or preferably in AC operation/simulation.
The resolution determines the conversion step of the analogue input i.e., the input voltage change needed for triggering a corresponding change in the Least Significant Bit (LSB) of the ADC. If the input signal ranges between 0 and V_{ref}, then an ADC with 4-bit resolution has a conversion step of $V_{ref}/16$ while this step is $V_{ref}/256$ in 8-bit resolution. It is obvious that the higher the resolution is, the less distortion is introduced by the analogue to digital conversion mechanism.
The speed of an ADC can be expressed by the sampling rate, the conversion latency and the analogue input bandwidth. The maximum sampling rate (throughput) determines how fast an input signal can be sampled in order to avoid missing transitions. Again, the higher sampling frequency allowed, the less distortion of the analogue input is achieved. The sampling rate is not always proportional to the maximum frequency allowed for the

analogue input signal. Regarding the analogue input signal bandwidth this is not always clearly defined. Some authors mention that this is associated with the input signal frequency used for their measurements but do not specify whether this is an analogue signal that swings in the full input range, or an input that is appropriate for Nyquist compatible sampling. The Signal to Noise and Distortion Ratio (SNDR) and the Spurious Free Dynamic Range (SFDR) are two parameters that can provide an indication on how close the output of an ideal DAC that is connected to the ADC output is, to the original input.

The latency is the time needed for the conversion of a single input sample. The conversion time determined by the throughput is usually shorter or in the worst case equal to the latency, since multiple samples may be processed concurrently as in the case of pipelined ADCs.

Several parameters have been defined for the description of possible errors in an ADC operation. An ideal conversion is plotted with the solid line in Fig. 1, along with a conversion characterized from offset, gain and monotonic errors (dotted line). Offset errors shift left or right the ADC output while the taps' height is erroneous on the presence of gain errors. A missing code appears in Fig. 1 due to a gain error. Differential Non-Linearity (DNL) indicates how far the current code is from its previous one and can be expressed as

$$DNL_i = \frac{D_i - LSB}{LSB} \tag{1}$$

The DNL_i is the DNL error of a specific code *i*, D_i is the actual duration of this code, while LSB is the ideal duration of this code. The Incremental Non-Linearity (INL) is defined as the integral of the DNL errors of all the output codes. The INL indicates how far the ideal ADC transfer function is from the measured one. The Signal to Noise Ratio (SNR) of an ADC is defined as the ratio of the input signal energy to the noise energy from DC to the Nyquist frequency. Another error source is the temperature dependence of an ADC. Although the specifications of an ADC guarantee that no missing codes occur within a temperature range, the linearity as well as the gain and offset errors may depend on temperature variations too.

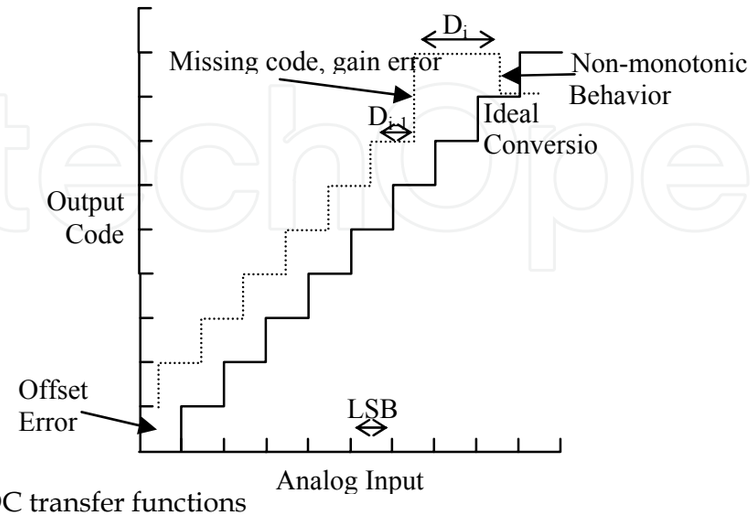


Fig. 1. Example ADC transfer functions

The ADC resolution and speed requirements for various applications are summarized in Table 1.

Application	Resolution	Speed
High Definition Data Analyzers	>16bits	>60MS/s
High Definition Imagers	12bits	80MS/s
Cable TV	8-12bits	15MS/s
High Definition TV	8-10bits	75MS/s
Magnetic Storage Read	8bits	150MS/s

Table 1. Indicative ADC resolution and speed requirements

3. Popular ADC architectures

3.1 Flash ADCs

Several Voltage-mode Flash ADCs have been proposed in the literature (Nejime et al., 1991) (Walden et al., 1990). An n-bit Flash ADC compares the input with 2^{n-1} voltage levels as shown in Fig. 2. These levels are: $V_{ref}/2^{n-1}$, 2 $V_{ref}/2^{n-1}$,…, $(2^{n-1}-1)$ $V_{ref}/2^{n-1}$ and are generated by a resistor ladder consisting of identical resistors. The input voltage (V_{in}) range is 0..V_{ref} and can be adjusted by connecting V_{ref} to an appropriate voltage. Current Mode Flash ADCs have been recently presented in the literature (Bhat et al., 2004) (Masood Ali et al., 2005). In this case, the input current is compared to 2^{n-1} current levels generated by appropriately sized current mirrors. The voltage comparators of Fig. 1 are replaced with current comparators in this case.

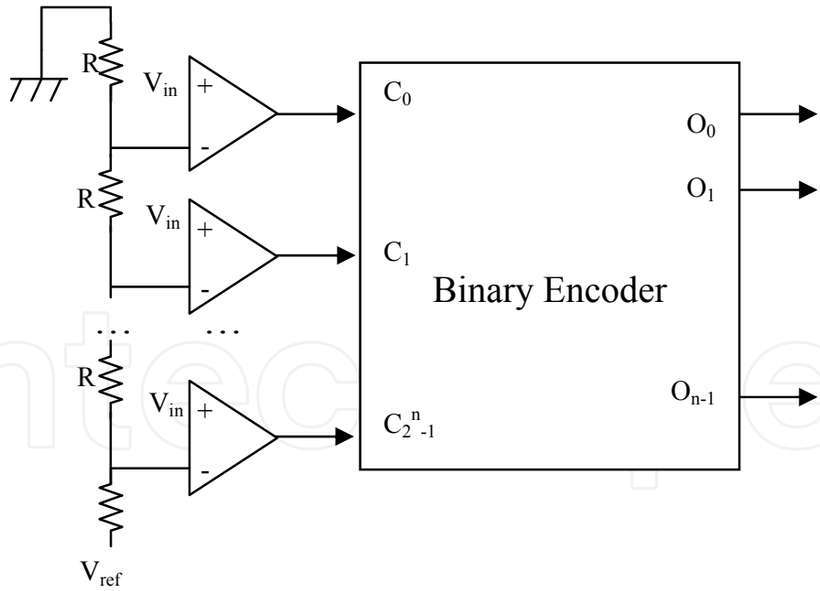


Fig. 2. Flash ADC architecture

The 2^{n-1} comparators of an n-bit Flash ADC, generate a "temperature code" that should be encoded into a binary representation. The relations that generate the binary representation $O_{n-1}O_{n-2}…O_0$ from the 2^n comparator outputs: $C_{2^n-1}C_{2^n-2}…C_0$ are the following:

$$O_n = C_{2^n-1} \tag{2}$$

$$O_{n-1} = C_{2^n-1} \oplus C_{2^{n-1}-1} \tag{3}$$

$$O_{n-2} = C_{4(2^n/4)} \oplus C_{3(2^n/4)} + C_{2(2^n/4)} \oplus C_{2^n/4} \tag{4}$$

…

$$O_0 = C_{2^n-1} \oplus C_{2^n-2} + \ldots + C_1 \oplus C_0 \tag{5}$$

The binary encoder can be implemented with domino XOR gates (Liu et al., 2003) since half of the XOR operations required by O_i are also needed in the estimation of O_{i-1}.

The source of non-linearity errors in the Flash ADC converters are the resistor mismatches in the ladder that generates the comparator voltage references. Contemporary Flash ADCs exploit special analogue techniques and process technologies to achieve multi GS/s throughput with good linearity behaviour. In (Walden et al., 1990) the authors used focused ion beam to create transistors with adjustable thresholds. In this way the resistor ladder is eliminated since the comparator thresholds are determined by the individually calibrated transistor thresholds. In (Chan et al., 2007) a submicron InP HBT technology is employed to achieve 5GS/s with 7-bit resolution. In (Park et al., 2006) inductors are used to improve speed and comparator pre-amplification bandwidth along with comparator kickback reduction techniques. The authors in (Deguchi et al., 2008) developed a 6-bit 3.5GS/s Flash ADC in 90nm CMOS technology using clamp diodes.

A Flash ADC can be implemented with asynchronous circuits. In this case, the digital outputs are settled after a period of time that depends on the comparators used and the binary encoder speed. The ADC outputs can be latched by a clock that is conformant with this settling time. The Flash architecture is the fastest one but the die area and power consumption required gets too high if the resolution is more than 7-bits due to the large number of comparators that are required.

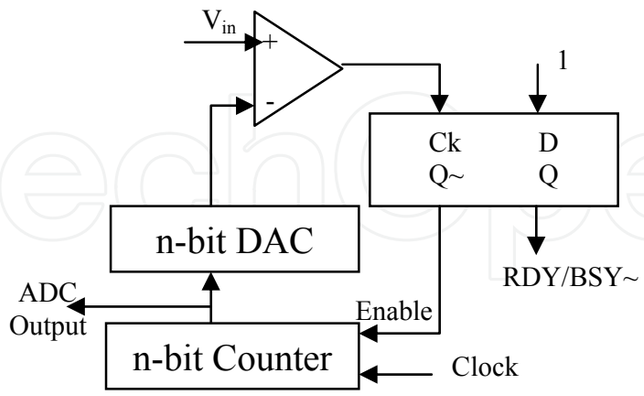


Fig. 3. Counting ADC architecture

3.2 Counting and Successive Approximation ADCs

The architecture of a Counting ADC appears in Fig. 3. A fast digital counter generates the successive digital representations of the numbers 0, 1, 2,…, 2^n-1. A Digital-to-Analogue

Converter produces an analogue value from this digital representation that is compared to the analogue input. If the digital counter output is found to be higher than the input, the ADC conversion operation is terminated. The average convergence time of this synchronous architecture is $2^{n-1}T_{ck}$, where T_{ck} is the period of the internal clock. This variable convergence time is too long but the number of components is quite small leading to slow but very low power and area ADCs. The Successive Approximation ADCs (Yuan & Svensson, 1994) (Promitzer, 2001) are an improvement to the Counting ones achieving $T_{ck}\log_2 2^n = nT_{ck}$ convergence time. Instead of a free running n-bit Counter they apply to the n-bit DAC values derived from an interpolation search. Initially the value 2^{n-1} is applied. If for example, the input value is smaller, the range $(2^{n-1}..2^n)$ is rejected and the value 2^{n-2} is applied to the DAC. Assuming that the input value is found higher, the range $[0..2^{n-2}]$ is rejected and a search in the range $(2^{n-2}..2^{n-1})$ is initiated. This procedure is repeated until the input value is approached. This algorithm is similar to a search in a binary tree. The linearity errors stem from the potential linearity errors of the DAC that is used.

3.3 Pipeline ADCs

The Pipeline or Subrange ADCs (Ahmed & Johns,2005) (Iizuka et al.,2006) achieve conversion speeds comparable to that of the Flash ADCs with significantly lower power consumption and die area. The principle of their operation is described by Fig. 4. The analogue input is used by a coarse Flash ADC to generate the most significant bits. The output of the coarse ADC is input to a DAC. The DAC's output is subtracted from the original analogue input and the difference is input to a fine Flash ADC that generates the least significant bits. Assuming that the resolution of both the coarse and the fine ADC of the n-bit Pipeline ADC are identical (n/2), then the required number of comparators is $2^{n/2}+2^{n/2}$ instead of 2^n that are required by a Flash ADC with n-bits resolution. The two constituent ADCs operate on successive samples held by the Track and Hold (T/H) circuits connected at their inputs. The Pipeline ADCs are synchronous systems since the T/H circuits of each stage require a clock signal.

The concept of the described Two-Stage Pipeline ADCs can be extended to more than two stages leading to further reduction of the required components. In the extreme case, each stage produces a single bit. In many approaches, each ADC stage generates redundant bits for error correction and calibration. For example if a 10-bit pipelined ADC has two stages, the first stage can generate 6-bits and the second 5-bits. The least significant bit of the first stage should match the most significant bit of the second stage. In the case they do not match, a calibration procedure can be initiated that modifies e.g. the references of the comparators in each ADC stage or the DAC biasing.

3.4 Other Common ADC Architectures

Folding and Interpolating ADCs (Makigawa et al, 2006) consist of a relatively small number of comparators. Different groups of reference values are applied to these comparators within a sampling period. In this way, the group that includes the closest reference to the analogue input is located using a fast interpolation algorithm.

The Algorithmic or Cyclic ADCs (Hedayati, 2004) (Chen & Wu,1998) are similar to the pipeline ADCs with 1-bit stages. Nevertheless, they consist of a single stage that is repeatedly used for the generation of each bit of the output.

Another popular architecture is the Sigma-Delta ADC (Arias et al., 2006) that is based on oversampling of the analogue input. The sampling frequency is much higher than the Nyquist limit and modulates the analogue input within the sampling period. The pulse density of the modulated input is then digitally filtered removing the noise components in the frequency domain. Sigma-Delta ADCs are implemented with integrators and 1-bit DACs. Due to the oversampling principle, the input signal frequency should be much lower (up to a few MHz) than the sampling frequency. The Sigma-Delta ADCs can easily achieve a resolution higher than 16-bits.

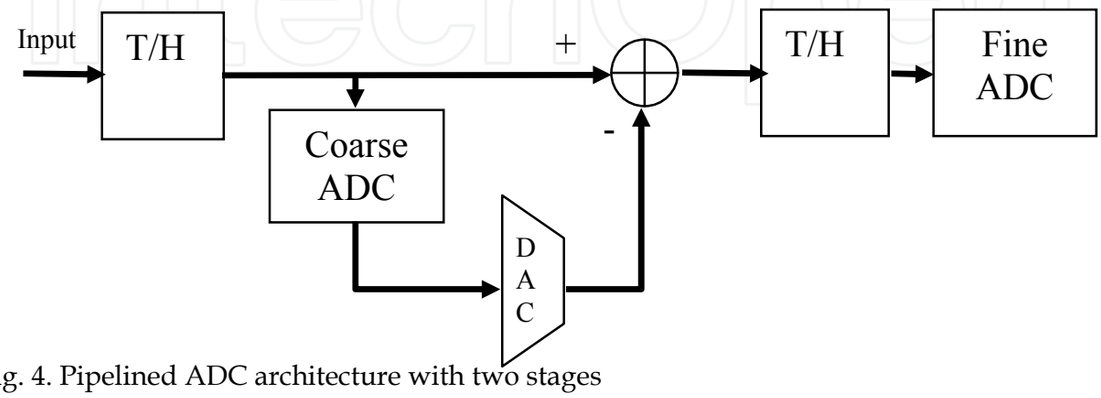


Fig. 4. Pipelined ADC architecture with two stages

Multi GS/s ADCs can be constructed with multiple parallel ADCs that are time interleaved i.e. they sample the input with a phase shift of a few psec. The individual ADCs are slower, robust circuits that dissipate low power. For example, in (Poulton et al., 2003) a parallel ADC with 80 slices is presented achieving 20GS/s sampling rate on an input analogue signal bandwidth of 6GHz. Each slice is a current mode pipelined ADC that achieves a 250MS/s throughput, occupies only 0.12mm^2 area and dissipates 57mW power. The clocking system creates 80 clocks with 250MHz frequency. The clocks used by neighbouring slices have a 50psec phase shift.

4. An Asynchronous ADC with Binary Tree Structure

4.1 General Architecture

An Asynchronous ADC architecture that requires a very small number of components and achieves high throughput is proposed in this section. The general architecture of this ADC is shown in Fig. 5. The analogue input value is divided in the root node by an appropriate power of 2. The quotient and the residue of this integer division are the inputs of two subtrees that correspond to ADCs with half resolution of the overall ADC. The integer division performed at a node of the balanced binary tree presented in Fig. 5 is 2^{2^L}, where L is the level of that specific node in the tree. The leaves are assigned to L=0. For example, the root node in an 8-bit ADC tree divides the input by 16. The residue and the quotient of this division is input to two subtrees that correspond to 4-bit ADCs. The root nodes of these 4-bit ADCs divide their input by 4. The quotient and residue of this division are input to 2-bit ADCs. Finally, these 2-bit ADCs divide their input by 2 and generate a quotient and a residue that are digitised by a comparison to a threshold.

The binary tree does not necessarily need to be balanced. For example, a 12-bit ADC has been developed with a 4-bit ADC and an 8-bit ADC that are connected to a root node. If the input of this 12-bit ADC is in the range [0..256], then the root node divides the input by 16. The quotient of this division is between 0 and 16 and is driven to the 4-bit ADC. The residue is between 0 and 15 and is driven to an 8-bit ADC with the same input range [0..256] after a multiplication by 16. Similar range adaptations may be decided during the design and simulation of the actual circuit that implements the architecture of Fig. 5, for optimisation purposes.

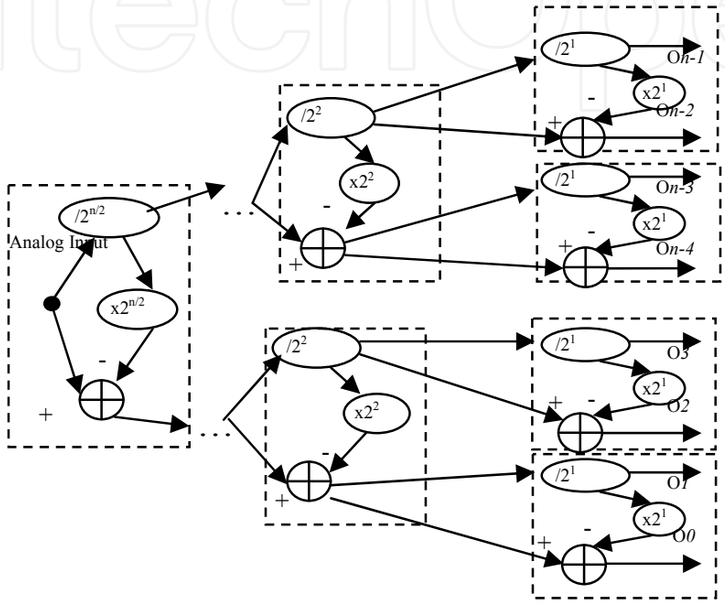


Fig. 5. Architecture of an n-bit ADC with binary tree structure

The architecture presented in Fig. 5 can be realised with current mode circuits due to the simplicity in the implementation of the various operators required, such as additions, subtractions and multiplication/division by a constant. Moreover, current mode circuits operate with low voltage supply and achieve high speed operation.
The integer division is performed by the circuit presented in Fig. 6. The input current signal I_{in} is concurrently compared against the references I_{ref}, $2I_{ref}$,…,$(N-1)I_{ref}$, where N is determined by the maximum input current allowed ($I_{max}=N \cdot I_{ref}$). If the input is between $q \cdot I_{ref}$ and $(q+1) \cdot I_{ref}$, then q comparator outputs will be high leading to the addition of q current sources at the output of the divider. The current $q \cdot I_{ref}$ represents the quotient of the division I_{in}/I_{ref}. The residue current I_r of this division is:

$$I_r = I_{in} - nI_{ref} \tag{6}$$

If N is too high then a large number of comparator and current sources are needed in the circuit of Fig. 6. In order to avoid the large area occupied by such a divider as well as its high power consumption, two simpler dividers can be connected in series as shown in Fig. 7. This is derived by the fact that a division by $N=N_2 \cdot N_1$ is equal to a division by N_1 and then, a division by N_2. The reference currents of the two dividers are selected as:

$$I_{ref1} = \frac{I_{\max}}{N_1} \tag{7}$$

$$I_{ref2} = \frac{I_{\max}}{N_1 N_2} \tag{8}$$

The area and power consumption are significantly reduced in this way, but an additional delay is introduced by the second stage of division. If

$$q = \frac{I_{in}}{N_1 N_2} \tag{9}$$

then the output of the circuit in Fig. 7 is $q \cdot I_{ref2}$.

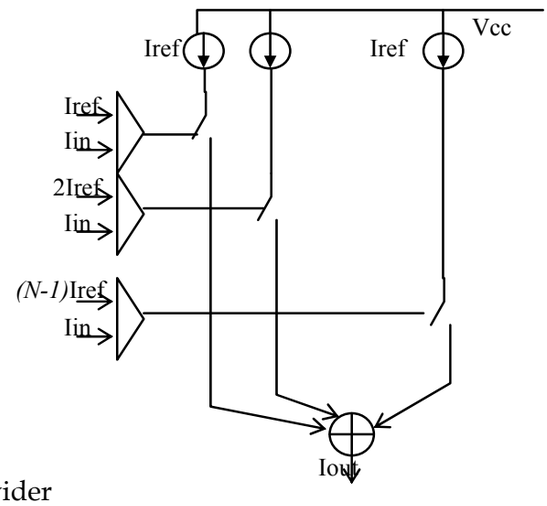


Fig. 6. The integer divider

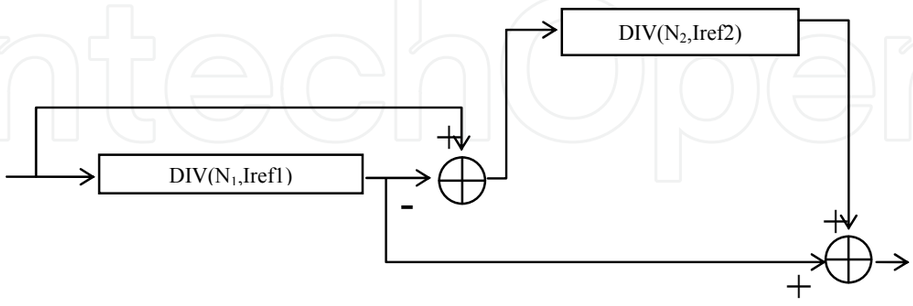


Fig. 7. Replacing a large divider by two simpler ones

4.2 Implementation Details

The integer divider of Fig. 6 can be implemented by using either synchronous or asynchronous current comparators. Synchronous comparators are assumed to be more

accurate and achieve faster convergence than the asynchronous ones. A part of the clock period is used in the synchronous comparators to reset the charged contacts of the transistors used while the comparator output convergence occurs during another part of the clock period. Nevertheless, although being the fastest and more accurate synchronous comparators are very sensitive to transistor mismatches. The mismatch immunity can be improved if cascode transistor arrangements are used, but the convergence speed is slowed down in this case. Another drawback of a synchronous comparator is the kickback effect i.e., glitches and jitter that appear to its inputs due to internal clock switching.
Asynchronous comparators offer high mismatch immunity with very low jitter. For this reason, an asynchronous comparator with positive feedback (Traff, 1992) has been employed for the implementation of the proposed divider that is presented in Fig. 6. The convergence time of this comparator ranges from 1ns to 100ns for input current difference between 10uA and 0.01uA respectively. Using asynchronous comparators, the whole ADC architecture does not need a clock signal. Consequently, if an analogue input is applied the outputs will settle in a short time period. The worst settling time observed either through simulation or experimentation can be considered as the latency period for the conversion of a single sample. Nevertheless, the outputs can be latched using a higher frequency clock if the transient output codes that are generated before the end of the ADC settling period do not break the monotonic principle. In this way, higher throughput and conversion accuracy can be achieved.

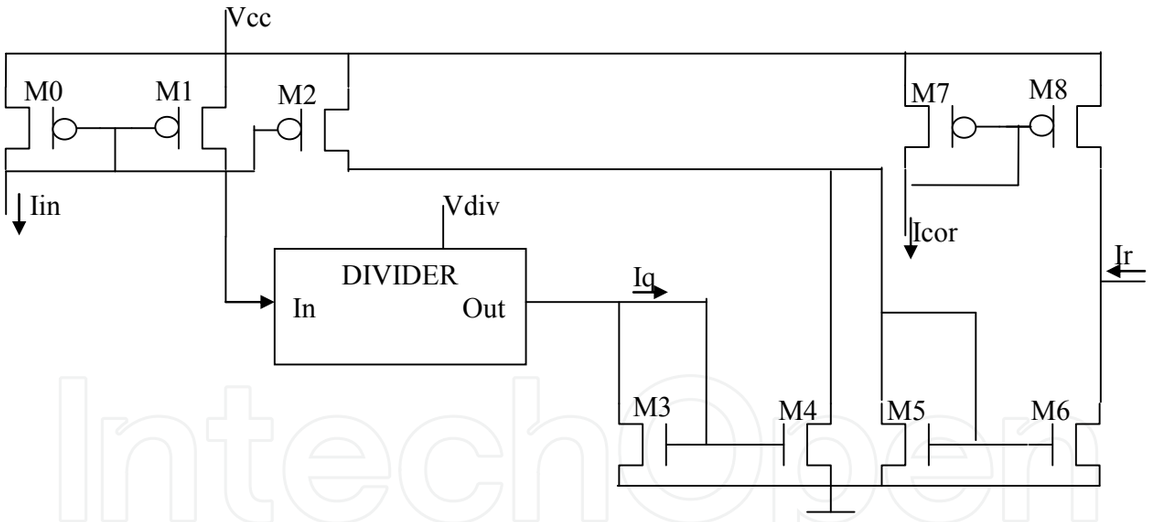


Fig. 8. Generation and offset correction of the integer division residue

The copies of the comparator input current and references as well as the current sources used in Fig. 6 are generated through cascode current mirroring. The channel length of the transistors used in mirrors that reproduce constant currents like the current sources or the comparator references is selected to have higher value (0.8um..1um) than the channel length of the transistors that are used in the mirrors of the input signal. The switches that are controlled by the comparator outputs can be implemented by NMOS or PMOS transistors or pass gates. If a switch is open, the corresponding I_{ref} current source should be preferably connected to the ground to reduce transient glitches.

The residue of the division is generated as shown in Fig. 8. In a real design environment, the simple mirrors used in this figure are replaced by cascode ones for higher mismatch immunity or gain boosted mirrors for achieving real time calibration. The PMOS mirror that consists of the transistors M0-M1-M2 generates two copies of the input current. One of them is used as input to the integer divider. The output of the integer divider (quotient) is mirrored in M4 and subtracted from the source current of M2 that generates the second copy of the input. The NMOS mirror M5-M6 accepts as input the residue of the subtraction. The potential offset I_{cor} that appears in the residue can be removed by the PMOS mirror M7-M8.

The output of the divide by 16 circuit used in the root of an 8-bit ADC is shown in Fig. 9a along with the divider input. The difference of these signals represents the residue and is shown in Fig. 9b.

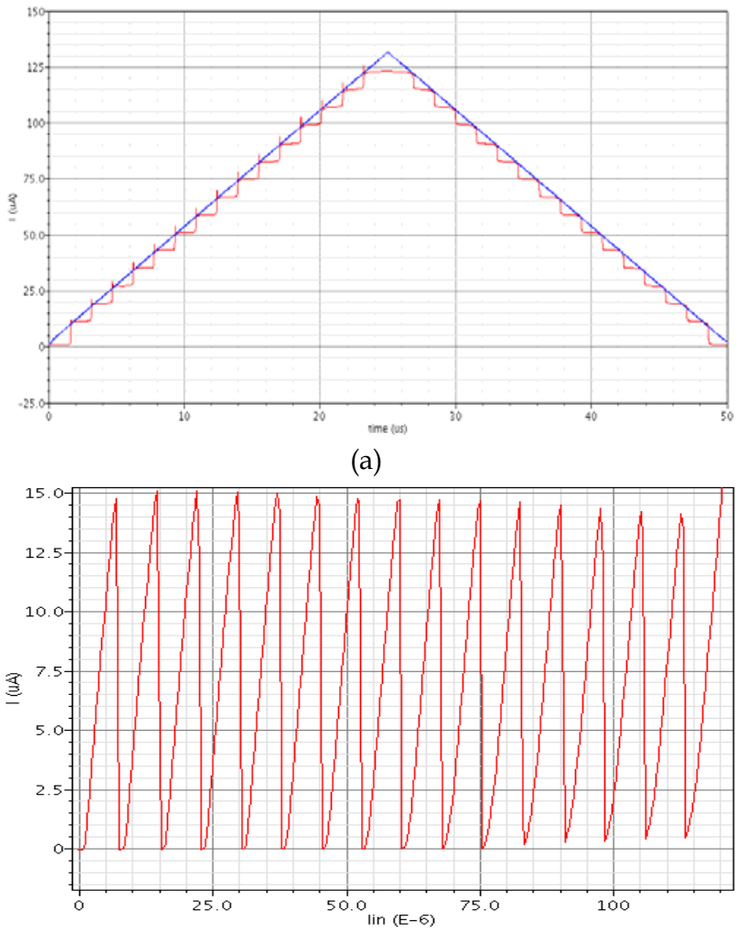


Fig. 9. The input and output of the divider (a) and their difference (b)

The top level design of an 8-bit ADC in Cadence environment is presented in Fig. 10. The displayed current mirrors and the divider DIV16xIref implement the root node of the binary

tree (see Fig. 5). The 4-bit ADC blocks are driven by the quotient and the residue of the division by 16 implemented in the root node.

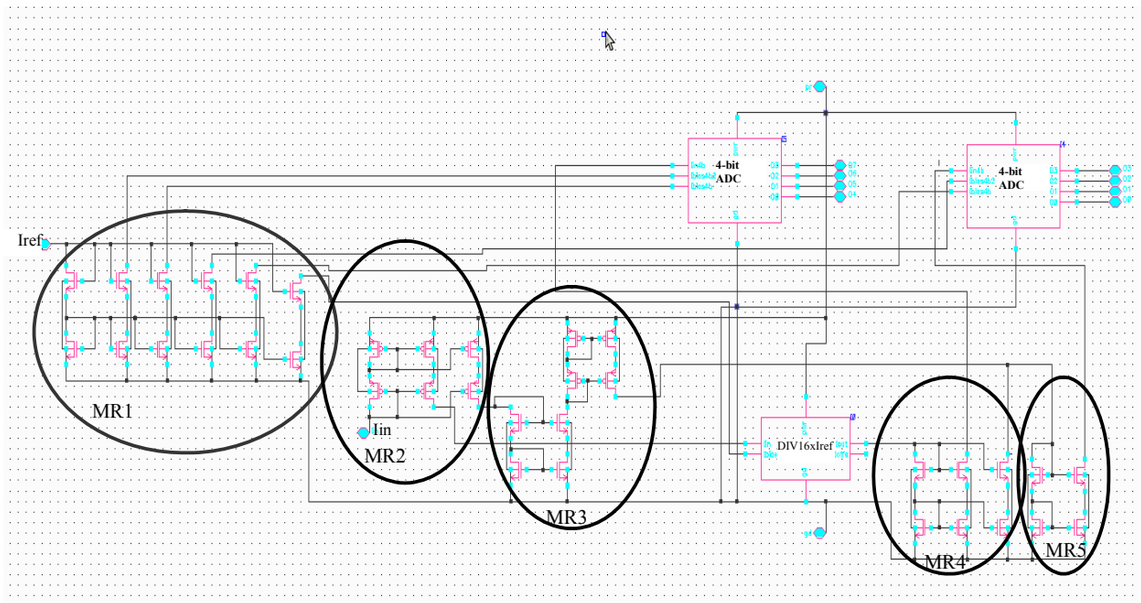


Fig. 10. Top level design of an 8-bit ADC

The mirror MR1 generates the various reference currents needed by the root divider and the 4-bit ADCs, while MR2 generates two copies of the input current. The divider output will be subtracted from one of the input copies. This specific input copy is delayed by the PMOS/NMOS mirror pair MR3 in order to compensate the delay introduced by the divider. The residue is driven into the 4-bit ADC that generates the least significant bits through MR5.

In a similar way, a 12-bit ADC has been developed by using a root node that divides the input by 16 and drives the quotient to a 4-bit ADC and the residue to an 8-bit ADC through a range adaptation circuit as already described in section 4.1. The output of such a 12-bit ADC is shown in Fig. 11.

A current mode ADC input is connected to the output of a Voltage to Current converter (V2I) since the analogue input is usually a voltage signal. A V2I circuit is characterized by its linearity. This can be expressed as the ratio of the input voltage to the output current and should remain as close to a constant as possible throughout the whole input voltage range.

A Sample and Hold (S/H) circuit may also be necessary in order to hold the ADC input stable for as long as the conversion takes place. A voltage S/H circuit should be placed between the analogue input and the V2I converter while a current S/H should be placed between the V2I and the ADC.

Half of the clock period in an S/H circuit is dedicated for the sampling of the analogue input (Sample period) while its output is kept stable during the second half clock period (Hold period). The quality of an S/H circuit depends on how stable the S/H output is during the Hold time (e.g., free of jitter, leakage etc) for the desired Hold duration. In an asynchronous ADC the sampling period is an idle time. For this reason, a pair of S/H

circuits that use complementary clocks are used. In this case, one of the two S/H circuits is always in its hold period continuously feeding the ADC with valid samples.

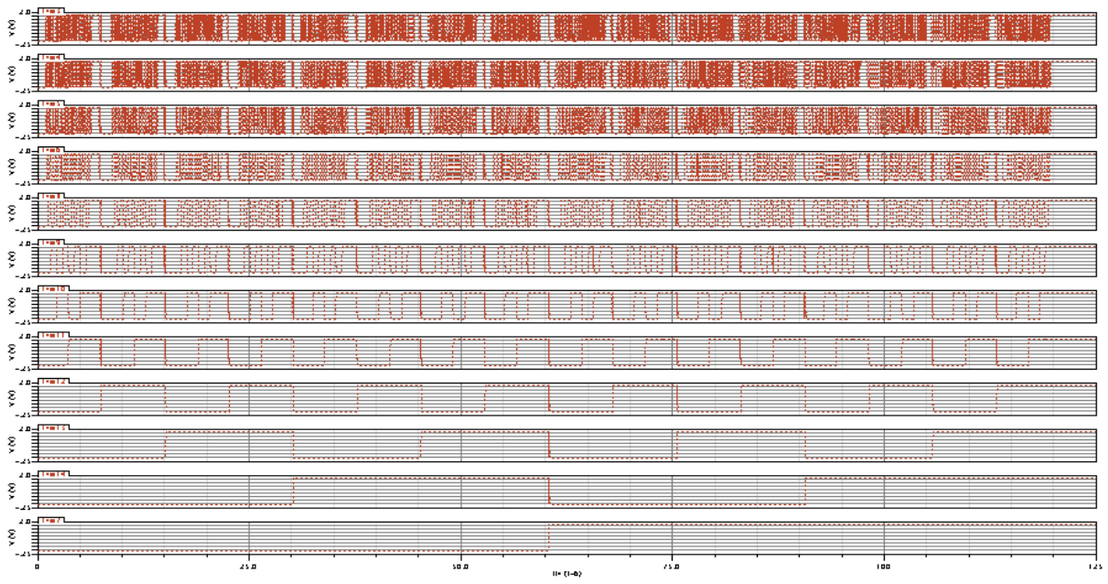


Fig. 11. The output of a 12-bit ADC

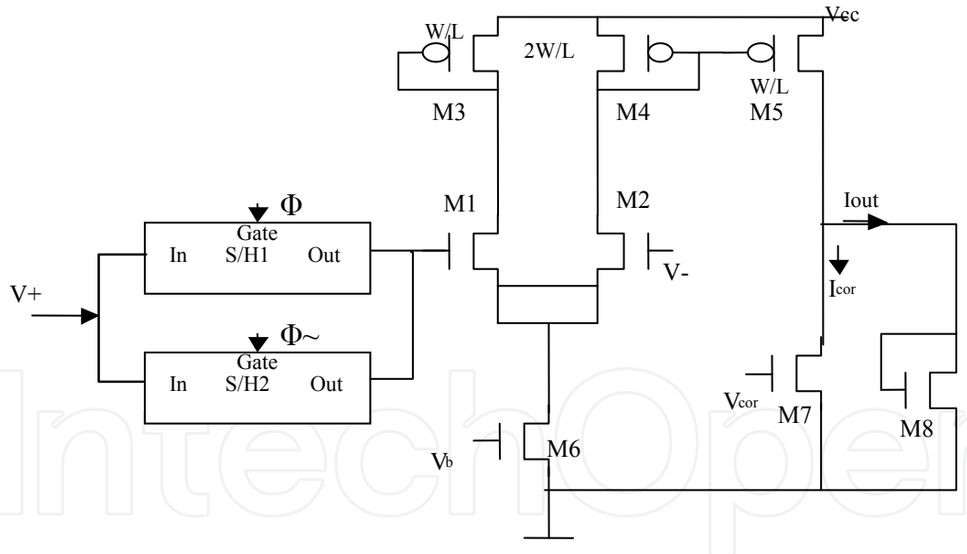


Fig. 12. S/H and V2I architecture for the developed asynchronous ADC

An appropriate V2I and S/H topology for the developed asynchronous ADC is presented in Fig. 12. The inputs of the two S/H circuits are connected to the positive pole of the analogue signal source. The outputs of the two S/H circuits are connected together to one of the differential inputs of the V2I (M1 gate). The second differential input (M2 gate) is connected to the negative pole of the analogue input source. The V2I consists of the transistors M1-M6. M1 and M2 should be identical while the size of M3-M5 should follow the designated in Fig l2 channel dimensions. The linear region of the V2I operation may not start from 0. The

undesirable offset current can be removed by biasing properly the drain current of M7 during the calibration of the ADC.

5. Simulation Results and ADC Comparison

The developed 12-bit ADC incorporates an 8-bit and a 4-bit ADC that can be isolated by powering off the rest of the circuitry in applications that favour lower power dissipation and faster operation instead of high resolution. The die area occupied by the 8-bit and the 12-bit ADC is only 0.06mm^2 and 0.12mm^2 respectively. The power dissipation is also quite low: 32mW for 8-bit and 72mW for 12-bit resolution. The average sampling rate is 140MS/s in the 12-bit case or more than 150MS/s if 8-bit resolution is required. The area, power consumption and sampling rate of the incorporated 4-bit ADC are 0.008mm^2, 11mW and more than 200MS/s respectively.

The temperature range where the developed ADC is guaranteed to operate without missing codes is -10^{o}C..+50^{o}C. This range can be extended to -30^{o}C..+90^{o}C if a calibration procedure is followed that adjusts appropriately the gain of the M5-M6 and M7-M8 mirrors of Fig. 8. This can be achieved by using gain boosted mirrors instead of the simple ones used in this figure.

The 12-bit ADC is powered by a 1.8V voltage supply if cascode mirrors are used. A 1.1V supply can be used if the ADC is designed with simple instead of cascode mirrors. For example, a 4-bit ADC developed with simple mirrors occupied only 0.0011mm^2, dissipated 3.5mW and achieved a speed of 350MS/s, with the cost of significantly higher mismatch sensitivity. Table 2 summarises the most important features of ADCs that are referenced in this chapter. In Sigma delta ADCs, the speed refers to the sampling frequency which should be significantly higher than the input signal frequency. A question mark is used if the die area occupied by an ADC is not mentioned by the authors. In the last column a composite measure is given in order to compare ADCs with different resolution developed in different technologies. The speed is multiplied by the resolution bits and then divided by the area and the power. The area used in this estimation is the actual die area divided by the technology in order to offer a fair comparison for the older ADCs although the sizes of the transistors used in mixed analogue/digital circuits do not change linearly with the technology length.

It is clear from Table 2, that the presented ADC architecture requires the lowest die area compared to other approaches with similar resolution, without sacrificing speed and with a low enough power consumption. Consequently, the proposed ADC achieves a remarkable trade off between resolution, speed, area and power consumption, that is better than most of the listed approaches.

The DNL error of the developed 8-bit ADC is plotted in Fig. 13a and the corresponding INL error in Fig. 13b. There are some significant DNL errors that appear at a few codes. These codes have a binary representation of the form xxxx0000 and xxxx1111. The DNL error at these codes can be reduced by optimising the height and offset of the teeth in Fig. 9b. More specifically, the current sources of the divider that was described in Fig. 6 can be designed with an optimised value close to I_{ref} that is decided through simulation. By modifying the value of these sources the taps' height at the output of the divider is adjusted. An appropriate scaling of the transistor sizes of the current mirrors used in Fig. 8 can also be used to adjust globally the height and offset of the teeth in Fig. 9b.

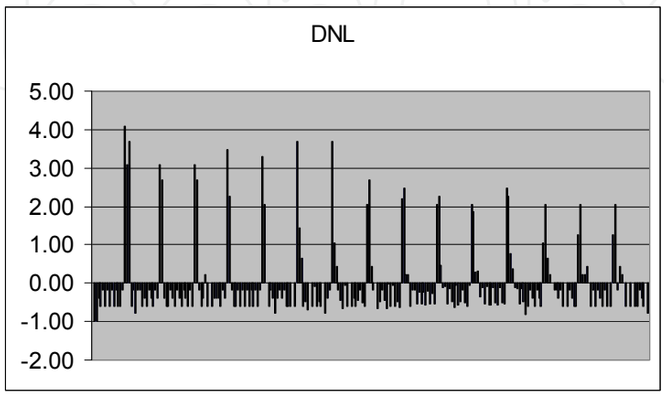
Reference	Resolution Bits	Speed	Area (mm²) @Technology	Power mW	(Speed·Bits)/ (Area·Power)
This Work	4bits-simple mirror	350 MS/s	0.0011 @90nm	3.5	32727
This Work	4bits-cascode mirror	200 MS/s	0.008 @90nm	11	818
This Work	8	150 MS/s	0.06 @90nm	32	56
This Work	12	140 MS/s	0.12 @90nm	72	17
(Ahmed&Johns, 2005)	10	50 MS/s	1.2 @0.18um	35	2.14
(Arias et al,2006) (Sigma Delta)	8.9	320MHz	0.44 @0.25um	32	-
(Bhat et al,2004)	7	80 MS/s	?	78	-
(Chan et al,2007)	7	5GS/s	?	?	-
(Chen & Wu, 1998)	10	12kS/s	4 @0.8um	2	0.012
(Deguchi et al, 2008)	6	3.5GS/s	0.1485 @90nm	98	130
(Hedayati,2004) (Sigma Delta)	11.8	20MHz	?	1.1	-
(Iizuka et al, 2006)	14	40MS/s	1.15 @0.18um	72.8	1.2
(Liu et al, 2003)	6	450MS/s	2.4 @0.5um	190	3
(Makigawa et al, 2006)	7	800MS/s	0.32 @90nm	120	13
(Masood et al, 2005)	6	2GHz	0.025 @0.18um	19	4547
(Nejime et al, 1991)	8	300MS/s	33 @2.5um	3300	0.055
(Park et al, 2006)	5	3.5GS/s	0.658 @90nm	227	11
(Poulton et al, 2003)	8	250MS/s	0.12 @90nm	57	26
(Walden, 1990)	4	400MS/s	?	100	-

Table 2. ADC comparison

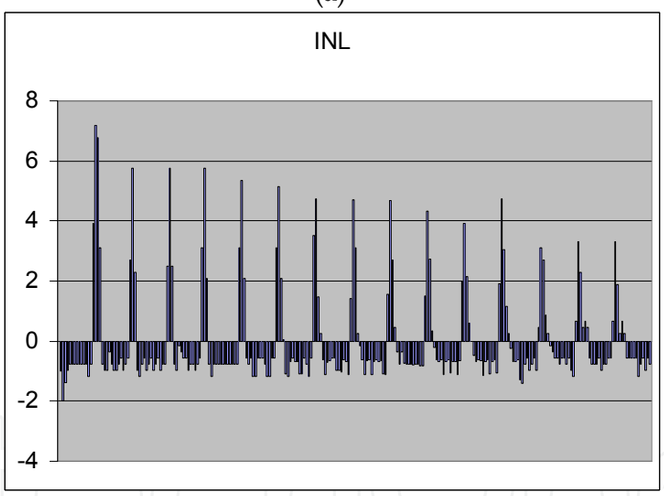
6. Conclusions

The most important Analogue to Digital Conversion techniques have been described briefly in this chapter focusing on their throughput, resolution, power consumption as well as the required area. An asynchronous Analogue to Digital Conversion technique based on a binary tree structure has been proposed. It is implemented with current mode circuits requiring a very small die area and low power consumption without sacrificing speed. More specifically, a 12-bit ADC, that has been developed based on this technique occupied only 0.12mm², dissipated 72mW and had an average throughput of 140MS/s. A thorough comparison with 16 other ADCs showed that the developed 4-, 8- and 12-bit ADCs achieve a

very good trade off between resolution, speed, area and power consumption making its use attractive in various control, sensor readout, communication and other applications.
Our future work includes the investigation of the use of several different synchronous and asynchronous comparators in order to further enhance the speed of the conversion. Higher resolution ADCs will also be developed. The use of various compression algorithms will be studied in order to achieve lower power consumption by transmitting the digitised samples with lower frequency through a smaller number of transmission lines.



(a)



(b)

Fig. 13. DNL and INL error of the developed ADC

7. Acknowledgments

This work was supported by Analogies S.A. and is patent pending (Application No. 0815802.4 / UKPTO).

8. References

Ahmed, I. & Johns, D. (2005). A 50-MS/s (35 mW) to 1-kS/s (15 uW) Power Scaleable 10-bit Pipelined ADC Using RapidPower-On Opamps and Minimal Bias Current Variation. *IEEE Journal of Solid State Circuit*, 2005, Vol. 40, No. 12, pp. 2446-2455.

Arias, J., Kiss, P., Prodanov, V., Boccuzzi, V., Banu, M., Bisbal, D., Pablo, J.S., Quintanilla, L., and Barbolla, J. 2006. A 32-mW 320-MHz Continuous-Time Complex Delta-Sigma ADC for Multi-Mode Wireless-LAN Receivers. *IEEE Journal of Solid State Circuits*, 2006, Vol. 41, No. 2, pp. 339-351.

Bhat, M.S., Rekha, S. & Jamadagni, H.S. (2004). Design of Low Power Current Mode Flash ADC. *Proceedings of the IEEE TENCON Conference.* Vol. 4, pp. 241-244, Nov. 2004.

Chan, B., Oyama, B., Monier, C. & Guitierrez, A. (2007). An Ultra-Wideband 7-bit 5Gsps ADC Implemented in Submicron InP HBT Technology. *Proceedings of the IEEE Compound Semiconductor Integrated Circuit Symposium*, Oct 2007.

Chen, C.C. & Wu, C.Y. (1998). Design Techniques for 1.5V Low Power CMOS Current Mode Cyclic ADCs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. Jan 1998. Vol. 45, No. 1, pp. 28-40.

Deguchi, K., Suwa, N., Ito, M., Kumamoto, T., and Miki, T. (2008). A 6-bit 3.5GS/s 0.9-V 98-mW Flash ADC in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 2008, Vol. 43, No. 10, pp. 2303-2310.

Hedayati, H. 2004. A Low Power Low Voltage Fully Digital Compatible ADC. *Proceedings of the IEEE International Conference on Microelectronics*, pp. 227-230, 2004.

Iizuka, K., Matsui, H., Ueda, M., and Daito, M. (2006). A 14-bit Digitally Self-Calibrated Pipelined ADC With Adaptive Bias Optimization for Arbitrary Speeds Up to 40 MS/s. *IEEE Journal of Solid State Circuits*, 2006, Vol. 41, No. 4, pp. 883-890.

Liu, F., Jia, S., Lu, Z. & Ji, L. (2003) CMOS Folding and Interpolating A/D Converter with Differential Compensative T/H Circuit. *Proceedings of the IEEE Conference on Electron Devices and Solid State Circuits,* pp. 453-456, Dec. 2003.

Makigawa, K., Ono, K., Ohkawa, K., Matsuura, K. and Segami, M. 2006. A 7bit 800Msps 120mW Folding and Interpolation ADC Using a Mixed-Averaging Scheme. *IEEE Symposium on VLSI Circuits Digest of Technical Papers*, 2006.

Masood Ali, S., Raut, R. & Sawan, M. (2005). A Power Efficient Decoder for 2.5GHz, 6-bit CMOS Flash-ADC Architecture. *Proceedings of the 9th IEEE International Database Engineering & Application Symposium*.

Nejime, Y., Hota, M. & Ueda, S. (1991). An 8-b ADC with Over-Nyquist Input at 300-MS/s Conversion Rate. *IEEE Solid State Circuits*, Vol. 26, No. 9, Sept. 1991, pp. 1302-1308.

Park, S., Palaskas, Y., Ravi, A., Bishop, R. & Flynn, M. (2006). A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS. *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 489-492. Sep. 2006.

Poulton, K., Neff, R., Setterberg, B., Wuppermann, B., Kopley, T., Jewett, R., Pernillo, J., Tan, C. & Montijo, A. 2003. A 20GS/s 8b ADC with a 1MB Memory in 0.18μm CMOS. *Proceedings of the IEEE International Solid-State Circuits Conference*. 2003.

Promitzer, G. (2001). 12-bit Low Power Fully Differential Switched Capacitor Non-Calibrating Successive Approximation ADC with 1MS/s. *IEEE Journal of Solid State Circuits,* Vol. 36, No. 7, July 2001, pp. 1138-1143.

Traff, H. (1992). Novel Approach to High Speed CMOS Current Comparators. *IEEE Electron. Letters*, Vol. 28, No. 3, 1992, pp. 310-312.

Walden, R., Schmitz, A., Kramer, A., Larson, L. & Pasiecznik, J. (1990). A Deep Submicrometer Analog-to-Digital Converter Using Focused-Ion-Beam Implants. *IEEE Solid State Circuits,* Vol. 25, No. 2, Apr. 1990, pp. 562-571.

Yuan, J. & Svensson, C. (1994). A 10-bit, 5-MS/s Successive Approximation ADC Cell Used in a 70-MS/s ADC array in 1.2um CMOS. *IEEE Journal of Solid State Circuits.* Vol. 29, No. 8, Aug. 1994, pp. 866-872.

Factory Automation

Edited by Javier Silvestre-Blanes

ISBN 978-953-307-024-7
Hard cover, 602 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Factory automation has evolved significantly in the last few decades, and is today a complex, interdisciplinary, scientific area. In this book a selection of papers on topics related to factory automation is presented, covering a broad spectrum, so that the reader may become familiar with the various fields, and also study them in more depth where required. Within various chapters in this book, special attention is given to distributed applications and their use of networks, since it is one of the most relevant subjects in the evolution of factory automation. Different Medium Access Control and networks are analyzed, while Ethernet and Wireless networks are looked at in more detail, since they are among the hottest topics in recent research. Another important subject is everything concerning the increase in the complexity of factory automation, and the need for flexibility and interoperability. Finally the use of multi-agent systems, advanced control, formal methods, or the application in this field of RFID, are additional examples of the ideas and disciplines that experts around the world have analyzed in their work.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen