Time Interleaved Counter Analog to Digital Converters

Seyed Danesh

2011

A thesis submitted for the degree of Doctor of Philosophy

The University of Edinburgh

Abstract

The work explores extending time interleaving in A/D converters, by applying a high-level of parallelism to one of the slowest and simplest types of data-converters, the counter ADC. The motivation for the work is to realise high-performance re-configurable A/D converters for use in multi-standard and multi-PHY communication receivers with signal bandwidths in the 10s to 100s of MHz. The counter ADC requires only a comparator, a ramp signal, and a digital counter, where the comparator compares the sampled input against all possible quantisation levels sequentially. This work explores arranging counter ADCs in large time-interleaved arrays, building a Time Interleaved Counter (TIC) ADC. The key to realising a TIC ADC is distributed sampling and a global multi-phase ramp generator realised with a novel figure-of-8 rotating resistor ring. Furthermore Counter ADCs allow for re-configurability between effective sampling rate and resolution due to their sequential comparison of reference levels in conversion. A prototype TIC ADC of 128-channels was fabricated and measured in 0.13μm CMOS technology, where the same block can be configured to operate as a 7-bit 1GS/s, 8-bit 500MS/s, or 9-bit 250MS/s data-converter. The ADC achieves a sub 400fJ/step FOM in all modes of configuration.

Dedicated to my father's memory

Acknowledgments

Firstly, I would like to thank Robert Henderson, my supervisor, for all his support and help. It has been a pleasure doing this project with you Robert. For sure you know how to get me working at my best. Every technical discussion we had related to this work, and your other activities, have been a great source of learning and pleasure to me. Thank you Robert.

I would like to thank Jed Hurwitz for teaching me so much, and really looking after me over the last couple of years. It's been really great working with you, and I think we've had a lot of fun. We've talked about the craziest of ideas. I'm glad we eventually implemented one of them. Who would've thought Jed, it actually works.

I would like to thank David Renshaw. David, you've looked after me ever since I set foot in this university almost 10 years ago now, through all the undergraduate labs right to today. I appreciate it a huge amount. At times it would've been impossible without your help and support. Thank you.

I would like to thank Keith Findlater. Keith, thank you for all your support, and giving me so much space to really get to working on these crazy ideas. You always looked for ways to help me out, and I hugely appreciate this. I can't imagine working for anyone better.

I would like to thank the whole team at Gigle, especially the analog team, and especially Will, Adria, Ewan and Al. You guys have been a joy to work with, and have each helped in many ways with my research.

I would like to say a special thanks to Steve Maughan, my PhD buddy. I escaped early my friend but I'll never forget those days at that prison like office at KB together, we talking absolute nonsense didn't we.

Finally, I would like to thank my close friends, and my dear family.

Table of Contents

List of Figures

Chapter 1 - Introduction

1.1 The growing requirement on communication

The number of electronic devices in the home from utilities and appliances, to luxury and pleasure, hand-held and mobile devices has increased exponentially in the last 20 years. The ever-growing digital data generated or consumed by these devices has pushed the requirements for in-home digital communication in terms of bandwidth and also coverage and reliability dramatically. Also the cloud computing model, technologies such as IPTV, and our desire to generate content on digital devices outside of our homes and workplaces has increased the requirements on networking bandwidth to these places. In general the use of content rich electronic devices has grown and with this the requirement and expectation of better digital communication between them.

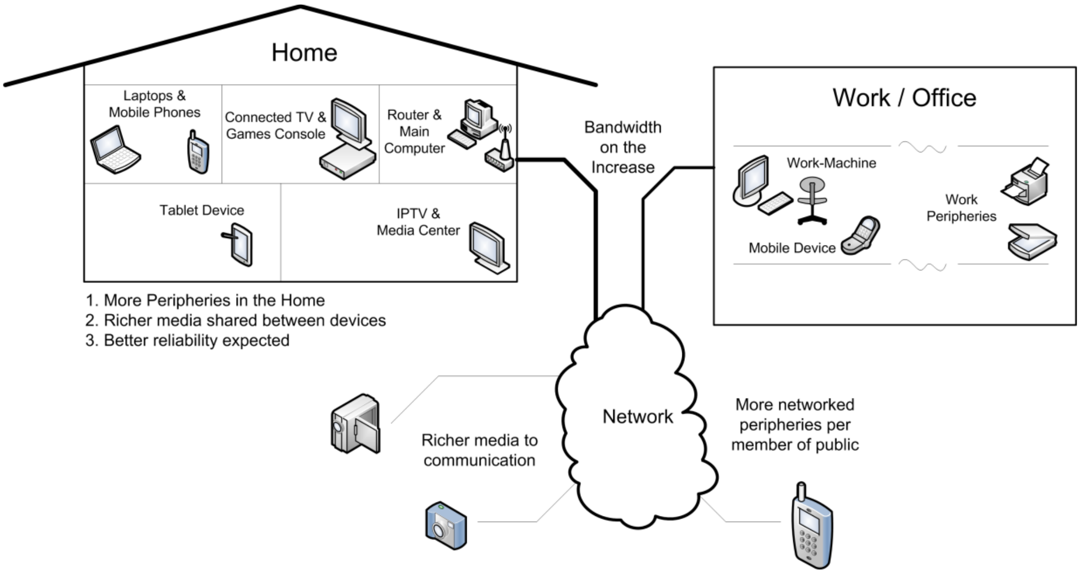

Home
Laptops & Mobile Phones
Connected TV & Games Console
Router & Main Computer
Tablet Device
IPTV & Media Center
1. More Peripheries in the Home
2. Richer media shared between devices
3. Better reliability expected
Bandwidth on the Increase
Work / Office
Work-Machine
Work Peripheries
Mobile Device
Network
Richer media to communication
More networked peripheries per member of public

Figure 1.1 – The growing demand for greater network bandwidth and converge

As the resolution of sensors, for imaging or audio, and displays and output devices have grown with Moore's law benefiting from technological advances, digital communication over a given medium within a given bandwidth is bound by Shannon's law [1] and cannot grow as easily. To extend the communication data-rates one is limited to 3 options:

1. Increase injected power levels (Improving SNR)
2. Use greater bandwidth on the channel
3. Use a new medium (new channel)

The first option of increasing the injected power levels is rarely a real option as this level is limited by emission regulations for health reasons, and also limited to prevent interference with communication on other channels due to radiation. Increasing bandwidth is a common option, apart from the huge cost associated with access to new bandwidth space, very rarely a new spectrum is available next to the current one used by a system, and potentially the injected power levels and channel characteristics are different at this new spectrum space and redesign of the digital communication apparatus is required. Exploration of new communication channels, primarily new wireline channels, has enabled growth in digital communication however these are usually difficult and time consuming and require a large investment up front to develop and understand the new channel, to define a suitable communication methodology, and to implement the communication apparatus.

Separate to the huge requirement on increase in communication bandwidth, coverage and reliability have become important factors in the digital lifestyle model. Electronic devices are no longer only used by technical specialists, with an understanding and appreciation of digital networking, but the mass-market model is now members of the public using these devices for pleasure and entertainment or non-electronic professionals in a work environment. The expectation for these rich media devices is to just work, and communicate with one another. Easy to use, simple and reliable software may be perceived to be the key to this model, but this software can only operate on top of a reliable communication system. In a communication system, to increase coverage we cannot simply increase our injected power levels, apart from regulatory reasons explained above, increasing power levels, in turn increases interference levels, defeating the original point and very rarely increasing coverage and reliability in the long term. The key to reliability is diversity. Diversity can be applied at many different levels for example,

- Diversity by use of different communication mediums (channels), used in combination, or transparent switching in case of failure in one.
- Diversity by use of different spectrum (frequency bands) within one channel, again in combination, or transparent switching in case of failure.
- Diversity by physical location, in wireless systems for example the use of two antennas,

These are some examples of how diversity in communication can be applied. Diversity comes at the cost of complexity, power consumption, and size of apparatus, however implementation of diversity is more common.

1.2 The System on Chip Communication System

In recent years the push towards reducing costs, miniaturisation and increased performance has encouraged the movement towards System on Chip (SoC) implementation of communication systems. Before looking at this implementation technique and its implication on communication bandwidth and coverage, a more detailed look at a generic communication system is required.

1.2.1 Building blocks for a communication system

It can be difficult to produce a diagrammatical representation of a communication system that covers all modern systems, since they can be very different in flow and sub-blocks. Figure 1.2 shows the main elements of a frequency and phase modulated communication system, or multi-tone or Orthogonal Frequency Domain Modulation (OFDM) system, which tend to use the frequency bandwidth more efficiently to produce systems closer to the Shannon limit of communication [2].

Not all sub-blocks shown here would be present in all systems. For example a baseband (low frequency) communication system, common in wireline systems, would not require an RF Frontend, and the Analog Front End (AFE) would connect to the hardware system, comprising of filters and signal coupling directly. The AFE in the receive path is tasked to digitise the signal on the channel with the required bandwidth to cover the entire signal of interest and the required resolution (dynamic range) to achieve the noise floor required for the digital signal processing (DSP) to read the original digital codes transmitted.

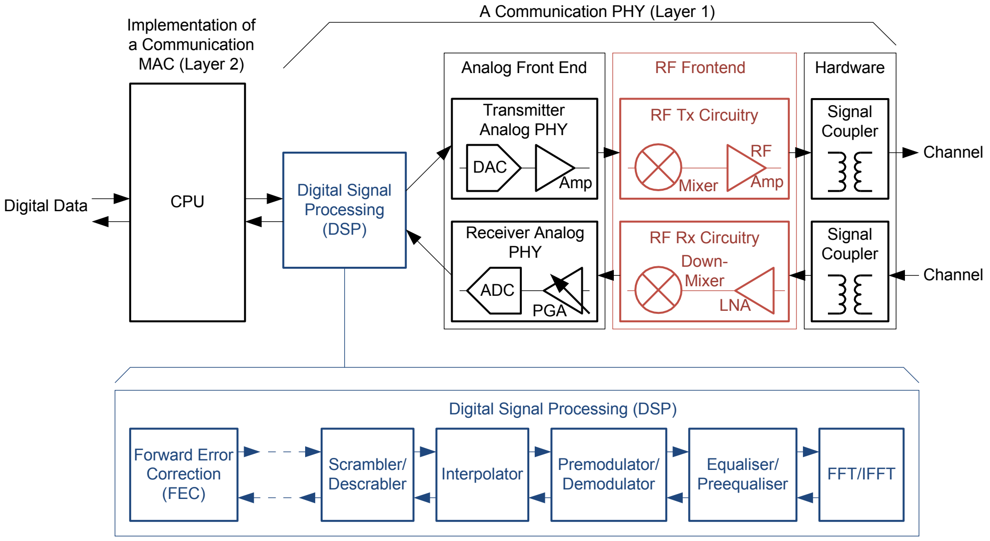


Figure 1.2 – Block diagram of a multi-carrier communication system

The sub-blocks present in the DSP sub-system can be different for different communication standards, but commonly have these main building blocks however the balance of their complexity can be widely different. The sections of the communication apparatus which deal with physical aspects of the signal, if they were to be modulating, equalising, amplifying, digitising, etc are referred to as the PHY of the communication system. All PHY sub-blocks deal with the signal in a non-abstracted way, if it were to be voltage values, or complex coefficients representing modulated values. The input and output of the PHY sub-system are digital codes, representing the communication signal at the lowest possible layer. In modern communication systems the presence of a CPU sub-system is common, running the instruction codes which manage high-level communication tasks such as synchronisation of the network, order of transmit and receive, etc. This implements the MAC subsystem, which deals with the signal at an abstracted level and controls the operation of a communication PHY. The combination of a MAC and PHY for a particular communication standard over a particular communication medium builds a communication system, which can then be used by an electronic system as a means of communicating with a network.

1.2.2 Example of an multi-tone communication PHY

A digital photo can be easily represented as a series of 1s and 0s. This data can be communicated easily over a digital backplane channel by simply driving the voltage level of the line high and low at a particular clock rate to represent 1s and 0s. With suitable synchronisation circuits, and clock recovery circuits the receiver can receive the digital stream representing the image and reproduce the photo. However to communicate this image over a busy communication channel, where only a small spectrum of frequency is available to the system, can be quite a complex problem, and a further level of abstraction is required since the information cannot be plainly placed on the line as high and low voltages representing the signal. Here we will explain how data can be communicated over a channel only using a limited bandwidth of the channel via a multi-tone communication system, which despite its greater complexity compared to a basic backplane communication system is becoming the more common and modern type of communication specially in limited spectrum systems.

To send an image the data from the image is pre-packed with specially communication information such as the target address, length, type but never the less is a stream of digital 1s and 0s which need to be communicated over a channel. The MAC manages the high-level acknowledgments, synchronisation and turn taking to allow node A to send the digital data stream to node B. For this example we will use a theoretical communication system, where we are only allowed to use the spectrum between 100MHz and 114MHz. In this example we will use 8 frequency carriers in this space, where each is 'bit-loaded' with 4 bits of information. In the example that follows, the DSP sub-system has been heavily simplified. The purpose of this example is to give an appreciation of the type of analog signal passing through a multi-tone communication AFE, and how the requirements on the AFE change for different communication standards.

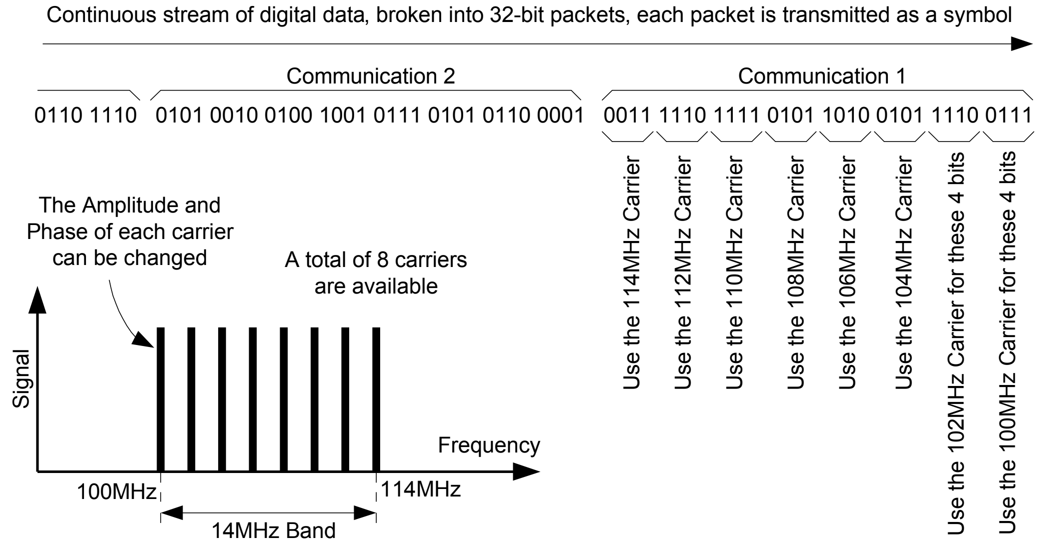


Figure 1.3 – The breaking of the digital stream into packets for transmission

The digital data stream is first passed through the Forward Error Correction (FEC) where blocks (words) of the data are coded with redundancy bits, so that in the presence of small errors due to noise and non-linearity in the communication system, the original data stream can be recovered. The output of this block is again a digital stream of 1s and 0s. We now have a stream of 1s and 0s, and we have 8 carriers to use to communicate them. This is shown in Figure 1.3. Based on the standard defined earlier, we are allowed 4-bits per carrier. Starting from the first bit of the stream, breaking the stream into 4-bit sections, each bit to be communicated on a different carrier. This division continues until 8 packets of 4 bit in length are chosen. This is all the information that can be communicated in the ‘Communication 1’ period. Now the systems requires to represent the collected bits using the 8-carriers in a unique way.

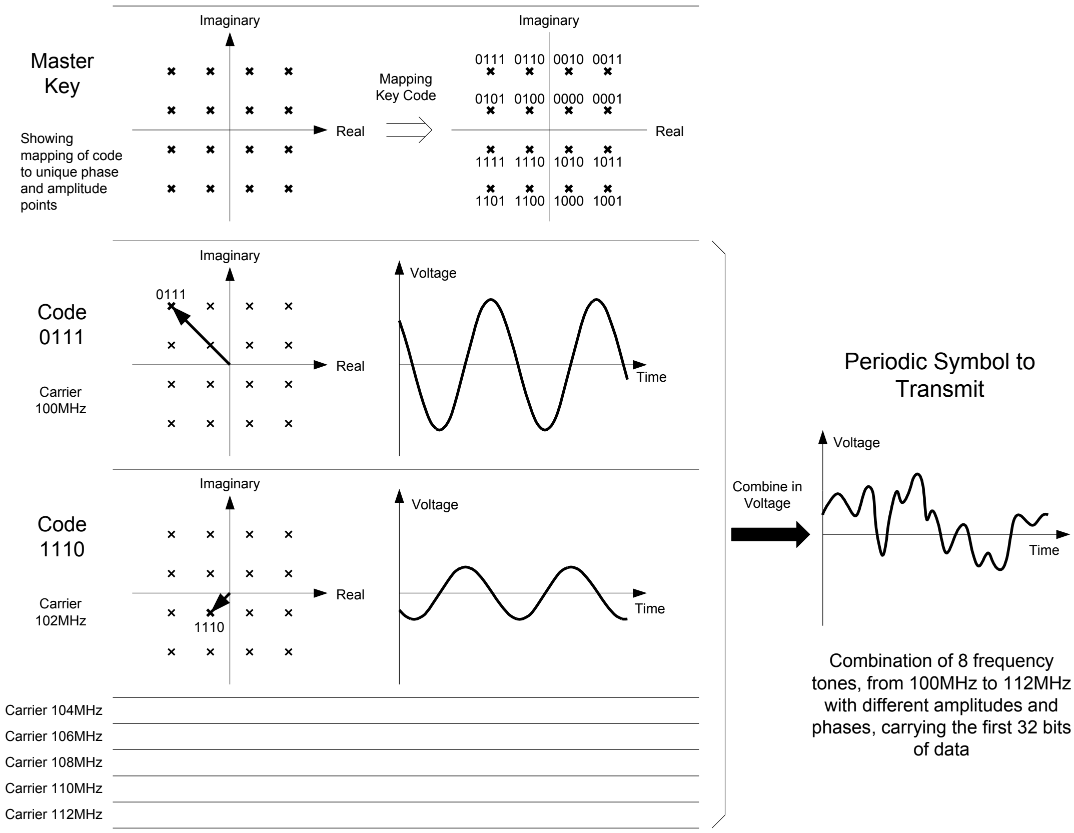


Figure 1.4 – Bit-loading of carriers and combining to build symbol.

On each carrier a Quadrature Amplitude Modulation (QAM) scheme will be used. This will be explained below. The amplitude and phase of each carrier can be adjusted to uniquely represent the code it is transmitting. For the communication system in question, a unique Master Key is defined, which is shown in Figure 1.4. This shows the mapping from a 4-bit code, to a particular amplitude and phase value. This is shown on the real and imaginary axis. For each carrier, using this master key, the amplitude and phase of that carrier can be adjusted to represent the desired 4-bit digital code which is required to be transmitted. Following this, the 8 carriers are combined (added) in the time domain to build the periodic symbol holding the 32-bits which we require to transmit. This symbol only occupies 14MHz of the channel bandwidth, and when received by the receiver hold 32-bits worth of information.

In reality, the correct QAM constellation for each carrier is constructed as a real and imaginary number in the DSP subsystem, and is passed through an IFFT to build a digital output stream in the time domain. This digital stream is then passed to the analog front-end to be transmitted through a DAC and Line-Driver. In this example no RF circuitry is required, since the band is not modulated to a higher frequency.

The signal passing through the channel will attenuate and may be subject to phase variation. At the receiver, the signal is quantised, and passed to an FFT block where amplitude and phase information for each carrier is obtained. Before the first transmission of real data, a known symbol is transmitted and the receiver uses this to learn the amplitude and phase transfer function of the channel. How the channel changes the amplitude and phase of each carrier is learned from the communication of this known symbol, and for normal operation this correction factor to amplitude and phase is applied prior to decoding of the data. This is known as equalisation. At the output of the FFT after equalisation, the amplitude and phase of each carrier is compared to the Master Key to identify what original 4-bit code was placed on this carrier. The amplitude and phase will rarely fall exactly on a QAM constellation point, due to noise, though the nearest point on the Master Key is assumed to be the original code. It is clear that if the noise in the channel, or noise of the AFE, is above a certain effective threshold for each carrier then the de-coding process will be subject to an error, since the amplitude and phases of a carrier may move to a different QAM constellation.

1.2.3 Design of a Multi-Carrier Communication Standard

Design of a multi-carrier communication system involves choosing parameters such as the frequency band used, the number of carriers, carrier spacing, injected power, maximum carrier bit-loading, symbol length, cyclic prefix length, and many others [2]. Some may be variables available to the communication standards designer and some may be pre-determined. The trade-off between these variables is far beyond the scope of this document, some text in [2] and [3] can be found to tackle these trade-offs. The purpose here is to highlight the common physical differences between different communication standards, to appreciate the implications on the design of the AFE.

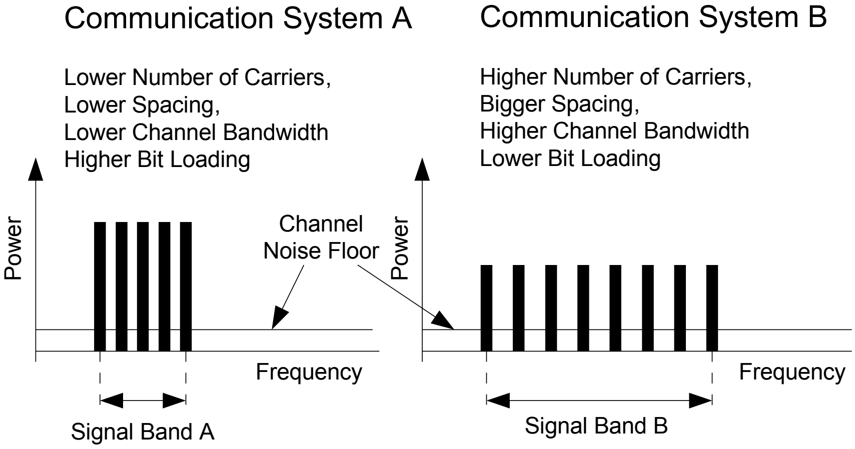


Figure 1.5 – Comparison of two theoretical communication systems

Figure 1.5 shows the spectrums of two theoretical communication systems at the input of the receiver. System A uses fewer carriers, with closer spacing, while System B uses a larger communication bandwidth with a larger number of carrier and more spacing. In this example System A uses a larger bit-loading per carrier compared to System B. In theory the two systems can be design to achieve an equal PHY communication data-rate. If the injected power per carrier were to be reduced in System B compared to A, the injected power over extended signal bandwidth could also be made similar. In this example, one could argue that System A can handle larger channel attenuation, and can hence extend the communication system reliability compared to B, while on the other hand System B can handle large nulling, variable frequency attenuation and interference better, and hence may have a better system coverage compared to A [2]. From this simple example it is clear that communication systems are configured subject to many trade-offs. Clearly System A may be more suited to a particular channel (medium) compared to B, but also even on the same channel A for example may be suited for a particular lower frequency band, while B outperforms A at higher frequency bands. Even one can think that for the same channel and same relative frequency band, system A and B can each out perform the other at different moments of time, depending on the channel circumstances. Even from this simple example, in the interest of achieving higher network converge and reliability, the importance of diversity in communication systems, and use of multi-standard, and multi-PHY communication apparatuses is clearly apparent.

The performance of communication apparatus is commonly limited by the quality of the receiver, since the receiver is required to deal with the attenuated signal, which has been subject to variable noise and variation. The receiver half modem is made from an analog front-end (AFE) section and a digital signal processing section. The dynamic range and accuracy, and signal bandwidth of the digital subsystem can be extended with greater memory depth, gate count, and parallelism respectfully. However extending the dynamic range and bandwidth of the AFE receiver subsection can be challenging and is commonly limited by the technology used for implementation. In this simple example, in the AFE, the noise floor requirement of the PGA in System A and B are equal, limited by the channel noise floor, while System B requires a larger bandwidth, but potentially a smaller gain range, assuming lower injected power and equal attenuation between System A and B. Regarding the ADC, System A requires a high-resolution converter, due to its greater bit-loading per carrier and hence greater Signal-to-Noise Ratio (SNR) per carrier, while System B requires a lower resolution but greater bandwidth and hence sampling rate. It is very clear that the A/D converter for these two example sub-systems have very different requirements.

1.2.4 SoC for communication and Multi-Standard systems

In the last decade there has been a huge drive towards implementing System on a Chip (SoC) solutions for communication apparatus. The motivations here are primarily to reduce cost and size by reducing the Bill of Material (BoM), enabling easier integration into other systems and allowing greater communication between the system's sub-blocks when implemented on a single die. There are many examples of published and commercial available SoC communication devices. These devices primarily comprise a CPU subsystem managing the MAC, the Digital Signal Processing, and an Analog Front End. While SoC in communication are beneficial for the reasons stated above, their main disadvantage is lack or limited re-configurability and versatility. For example in a 3 chip solution, where the CPU, DSP and AFE are implemented as separate entities, to re-configure the system to work to a different standard, that for example requires a different dynamic-range and bandwidth trade-off, the AFE chip could be replaced with an alternative part.

Earlier in this chapter, the importance of extending network coverage through diversity was explained. One of the powerful methods in reaching a high level of diversity is building a system that can operate to different standards of communication, potentially also on different mediums of communication. It was briefly explained, that in multi-carrier systems, standards that use high dynamic range and lower bandwidths, are designed to operate better in different environments compared to those optimised to lower bit-loading and larger bandwidths. Also in a certain environment, for example in the home, a certain channel or spectrum could be suffering from interference while a different channel or spectrum is clean. To build a communication apparatus with a high coverage rate ideally one would like a system that can communicate to many different standards over different channels with real-time re-configurability. At first this may suggest implementing many different complete communication sub-systems on one system.

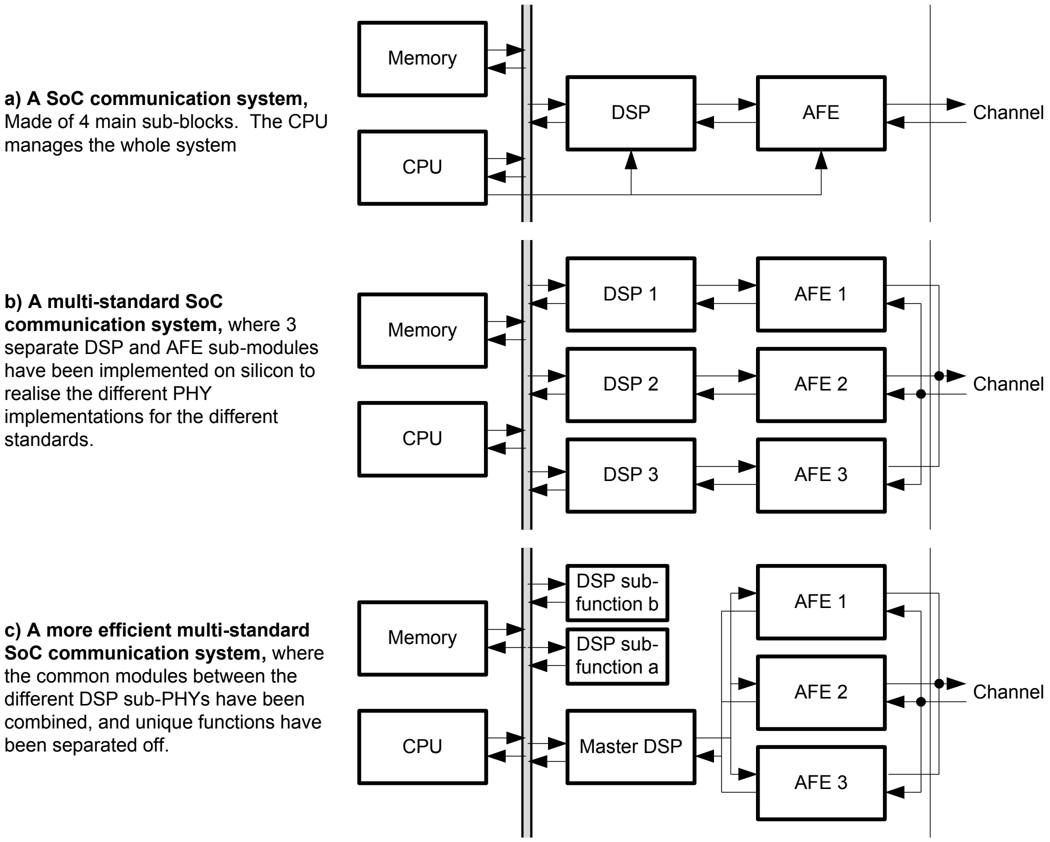

a) A SoC communication system, Made of 4 main sub-blocks. The CPU manages the whole system
Memory
CPU
DSP
AFE
Channel
b) A multi-standard SoC communication system, where 3 separate DSP and AFE sub-modules have been implemented on silicon to realise the different PHY implementations for the different standards.
Memory
CPU
DSP 1
DSP 2
DSP 3
AFE 1
AFE 2
AFE 3
Channel
c) A more efficient multi-standard SoC communication system, where the common modules between the different DSP sub-PHYs have been combined, and unique functions have been separated off.
Memory
CPU
DSP sub-function b
DSP sub-function a
Master DSP
AFE 1
AFE 2
AFE 3
Channel

Figure 1.6 – Evolution of Multi-Standard/Multi-PHY Digital Comms SoC

Figure 1.6 shows the evolution of the Multi-Standard to Multi-PHY communication SoC. The CPU and Memory between the different communication standards can be shared. Very quickly similarities between the different DSP sub-modules become clear, and different DSP modules can be merged to build a master DSP system. Some functionalities which are unique to a particular PHY can be separated, and used as part of the DSP system controlled by the CPU effectively building a re-configurable DSP sub-system. Unfortunately, re-use of the key AFE sub-blocks, particularly in the receiver and in particular the ADC has been found to be more challenging, since the requirements on these blocks can be very different for different standards, and this leads to architecturally different blocks. This is looked at in more detail in the next sections.

The important fact to take from this section is that multi-PHY communication systems hold the key to achieving higher coverage and reliability for today's communication needs, however to realistically fulfil this ambition without implications in design time, system size, and inevitably cost, one should explore the possibility of efficient re-configurability within the AFE sub-system, specially in sub-blocks where over designing to meet a super-set of the multi-PHY specifications is not practically possible.

1.3 A/D Converters, and re-configurability

Looking back at the 4 main sub-blocks of a communication AFE sub-system, the line-driver (amplifier) and Programmable Gain Amplifier (PGA), when used in closed loop, bandwidth and dynamic range or resolution, commonly measured as linearity can be fundamentally traded-off for a particular circuit, since closed-loop linearity is proportional to open-loop gain, and since the open-loop gain-bandwidth product of an amplifier is a constant, so for basic closed loop architectures the feedback factor can be adjusted to trade-off linearity (resolution) for bandwidth. Of course in practice there are many subtleties to this that makes this programmability less than trivial, and also if a less conventional architecture is used the trade-off may not be as practical.

D/A converters for communication with signal bandwidths in the 10s of MHz to 100s of MHz are commonly implemented as a current steering architecture, where their resolution is limited by matching in a particular technology, and output impedance by design [4], where the operation speed is limited by the digital backend speed, and switch driver operation, again limited by a technology. Fundamentally D/A converters are rarely the bottle-neck for speed or resolution in a communication system, and they can be over-designed to a super-set of speed and resolution for a multi-PHY communication system. The A/D converter on the other hand can be one of the hardest blocks to realise re-configurability in, especially to trade-off speed and resolution over a useful range. The reasons behind this will be explained below.

1.3.1 An Overview of Analog to Digital Converters

The task of converting an analog signal to a digital signal is an integral part of a digital communication system. Many architectures and approaches to performing this task are available and are used. Some architectures are more conventional or classical, each with their purpose and suited to a particular operation range. Figure 1.7 summarises the main 3 approaches commonly used for digitising of an analog signal. To achieve the highest sampling rate, the best approach is to compare the analog input to all digital threshold levels simultaneously. This builds a Flash ADC. Although Flash ADCs achieve the highest sampling rate, since the number of comparators used are exponentially related to the resolution, they tend to be low-resolution converters. One comparator can be used for multiple comparisons, allowing a first order trade-off of speed and resolution architecturally; this in turn builds a Folding ADC.

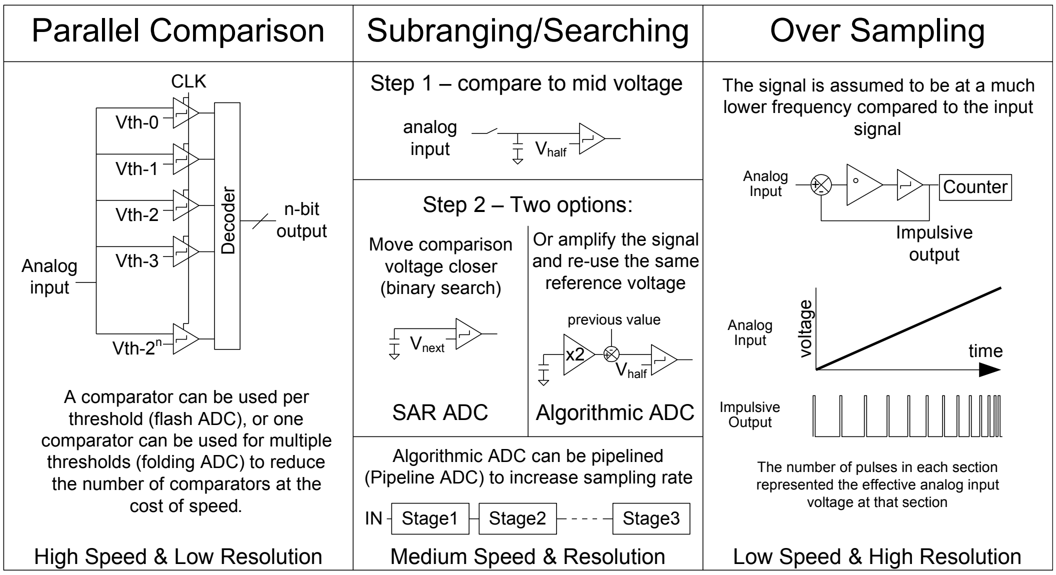

Parallel Comparison
Subranging/Searching
Over Sampling
CLK
Vth-0
Vth-1
Vth-2
Vth-3
Vth-2^n
Decoder
n-bit output
Analog input
A comparator can be used per threshold (flash ADC), or one comparator can be used for multiple thresholds (folding ADC) to reduce the number of comparators at the cost of speed.
High Speed & Low Resolution
Step 1 – compare to mid voltage
analog input
V_half
Step 2 – Two options:
Move comparison voltage closer (binary search)
V_next
SAR ADC
Or amplify the signal and re-use the same reference voltage
previous value
x2
V_half
Algorithmic ADC
Algorithmic ADC can be pipelined (Pipeline ADC) to increase sampling rate
IN
Stage1
Stage2
Stage3
Medium Speed & Resolution
The signal is assumed to be at a much lower frequency compared to the input signal
Analog Input
Counter
Impulsive output
Analog Input
voltage
time
Impulsive Output
The number of pulses in each section represented the effective analog input voltage at that section
Low Speed & High Resolution

Figure 1.7 – Overview of classical ADC architectures

If greater resolution is required, a searching or sub-ranging approach is used, which is primarily similar to a binary search algorithm. These types of converters can be broken into two categories, those that sample the input voltage, and search by moving the comparison threshold voltage, or those which use 1 threshold comparison voltage and amplify (move) the input signal around the threshold. The former are Successive Approximation Register (SAR) converter, and the latter are Algorithmic converters. Since the input is amplified and re-sampled after each comparison in an Algorithmic converter, it is possible to pipeline the comparisons to increase the overall sampling rate and hence build a Pipelined ADC.

To increase the resolution further, oversampling converters can be used. These converters continuously quantise the input signal, assuming it does not move much between quantisation cycles. The output of this quantiser is effectively subtracted from the input and the result is integrated building a Sigma-Delta quantiser loop. The integrator has a time constant and hence the output of this block is effectively a ramp with a gradient proportional to the analog input. This ramp when crossing a threshold results in a bit-stream where the occurrence of a pulse is proportional to the input signal. The pulse occurrences can be counted to find the digital representation of the input.

Further background on ADC architectures, and more detailed descriptions of different architectures can be found in common converter textbooks [5][6][7]. This text assumes a basic understanding of the common ADC architectures from the reader. The architectures relating to this work are covered with more detail in Chapter 2.

1.3.2 Comparison of A/D Architectures

The specifications for A/D converters published in the major IEEE solid-states circuit conferences in last decade have been collected in a spreadsheet by Boris Murmann, at the University of Stanford that is publicly available [8]. Although data and graphs from [8] are not directly used in this section, the concept and approach here is motivated and heavily influenced by this work and is hence referred to here. In this section a similar approach is used where papers published in the last decade at IEEE International Solid-State Circuits

Conference (ISSCC) and IEEE VLSI Symposium on Circuits on A/D converters are collected and papers following the conventional ADC architectures are identified and used to highlight trends in data converters and identify practical limitations of each architecture.

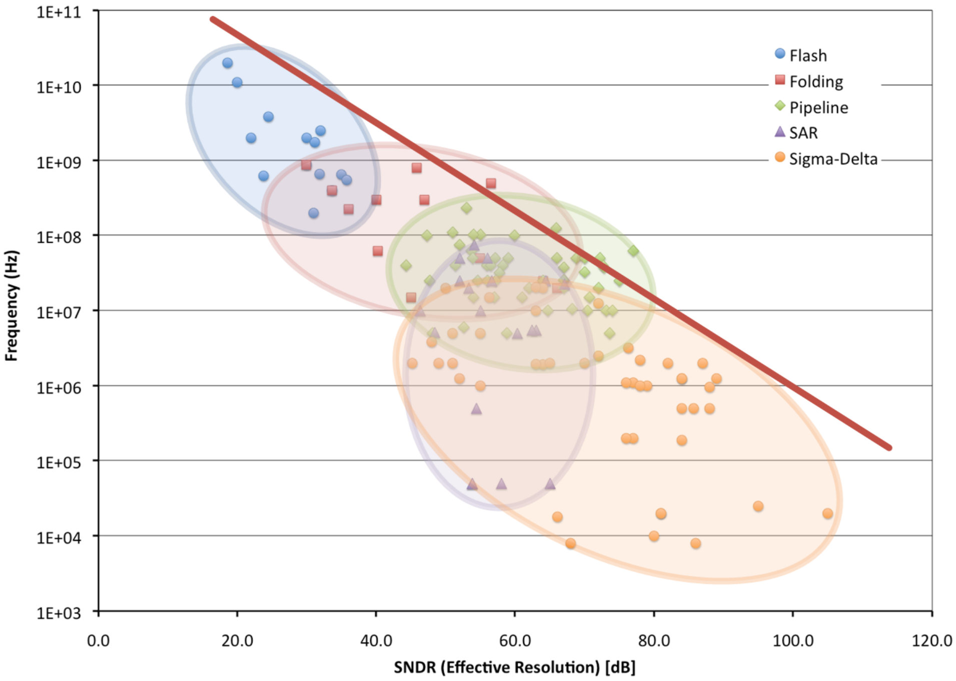


Figure 1.8 – Comparison of classical ADC architectures

Figure 1.8 shows the performance of the different ADC architectures, Flash [69-83], Folding [84-96], Pipelined [97-147], SAR [148-167] and Sigma-Delta [168-210], in the resolution and bandwidth space. For this graph the papers that follow the more classical description of each architecture published in the last 10 years, have been used. Each architecture is suited to a particular speed and resolution space. The red line drawn highlights the limit on the resolution speed product, drawn at a gradient of about 6dB per halving of frequency. This is believed to be the maximum resolution speed product possible limited to about 1ps RMS jitter. The growth in technology has always pushed this number further down, but moving along this theoretical line, to achieve different resolutions and speed ratios one must move between architectures from flash, to folding, to pipeline, to SAR, to sigma-deltas. It is also worth noting that with advances in technology and work on sigma-delta architectures, the effective bandwidth of these blocks have been increased to new limits, Looking at Figure 1.8 one may conclude that the need for SAR ADCs has been eliminated, however the problem is that this figure only shows half the story. Figure 1.9 shows the effective power-consumption per sampling rate for the different architectures.

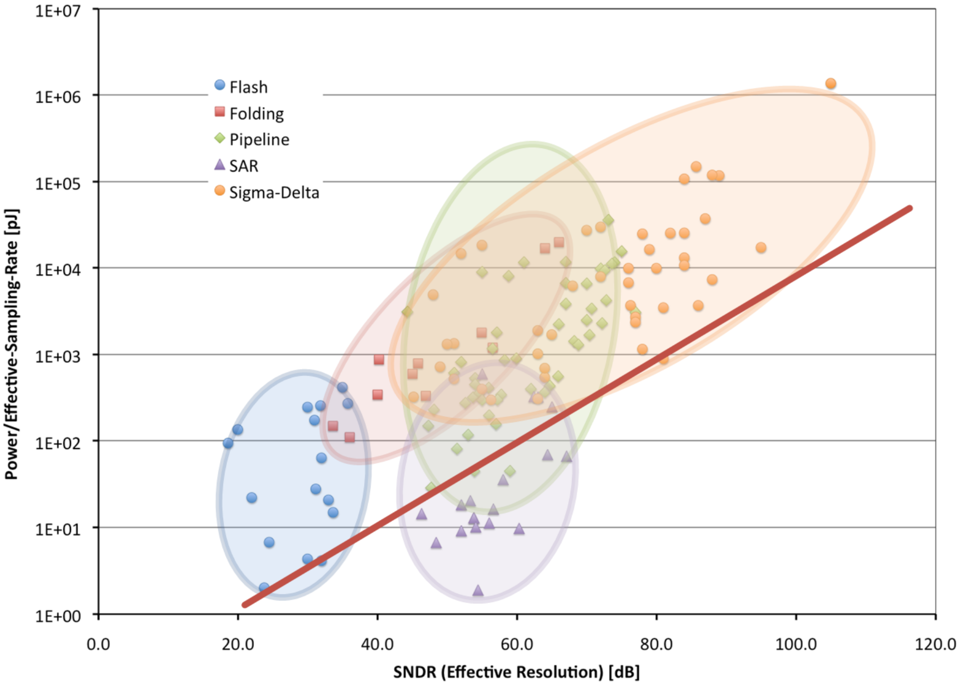


Figure 1.9 – Comparison of Power Efficiency for different architectures

This figure shows, in terms of power, how efficient each architecture operates. The red line highlighted is the 100fJ/step Figure of Merit marker placed as a reference. The closer each converter is to this line it is using the power from the supply more efficiently for its conversion. It can be clearly seen that SAR ADCs do significantly better than other architectures for efficiency, and although sigma-delta converters can be designed to have similar resolutions to SAR ADCs published, they can be up to two orders of magnitude less efficient compared to SAR ADCs at doing the conversion.

It is clear that each architecture has its strength and weaknesses. In the previous section it was explained that ideally for multi-PHY networking applications we would like to trade speed (bandwidth) for resolution for a given block to work with different standards. Referring back to Figure 1.8, it is interesting to note that no one architecture can be used to travel along the full red line. Sigma-Delta converters do cover a large section of the red-line, the bottom half, in the range of MHz to 10s of MHz, however to move along the red-line in the 100s of MHz to GHz space, moving between the architectures is required. With programmability in the sub-blocks in a sigma-delta converter, some trade-offs between resolution and speed may be possible, only at low frequency. This will be looked at in Chapter 2. At higher frequencies (which many wireline standards operate at) no re-configurability seems possible looking at the classical architectures.

1.3.3 Time Interleaving in A/D Converters

Time interleaving is a technique of using a number of identical converters in parallel with one another, but out of phase, where the input stream is captured and sampled at a higher rate compared to the operation speed of each converter, but as this stream is sampled, consecutive samples are forwarded to different ADCs to quantise, allowing each converter a longer time than the sampling rate for conversion. The digitised values from each converter is collected, and correctly ordered to pass to the output.

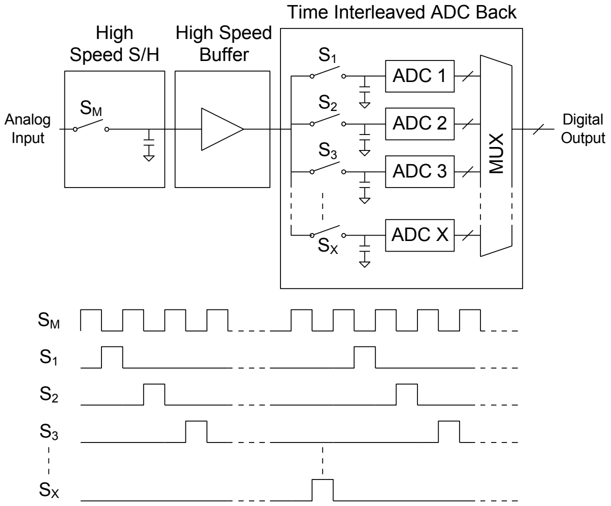


Figure 1.10 – A Time Interleaving A/D Converter block diagram

Figure 1.10 shows the block diagram of a time-interleaved ADC system. The ‘Time-Interleaved ADC Block’ appears as one high-speed ADC to the outside world, however is constructed from many lower frequency ADC blocks working out of phase from one another.

The presence of a high-speed sample and hold circuit and buffer at the front is not necessary for correct operation, and the sample and hold circuitry itself can be time-interleaved. This will be looked at in the following chapters. Time interleaving is used to push classical architectures outside of their conventional operation region shown previously in Figure 1.8.

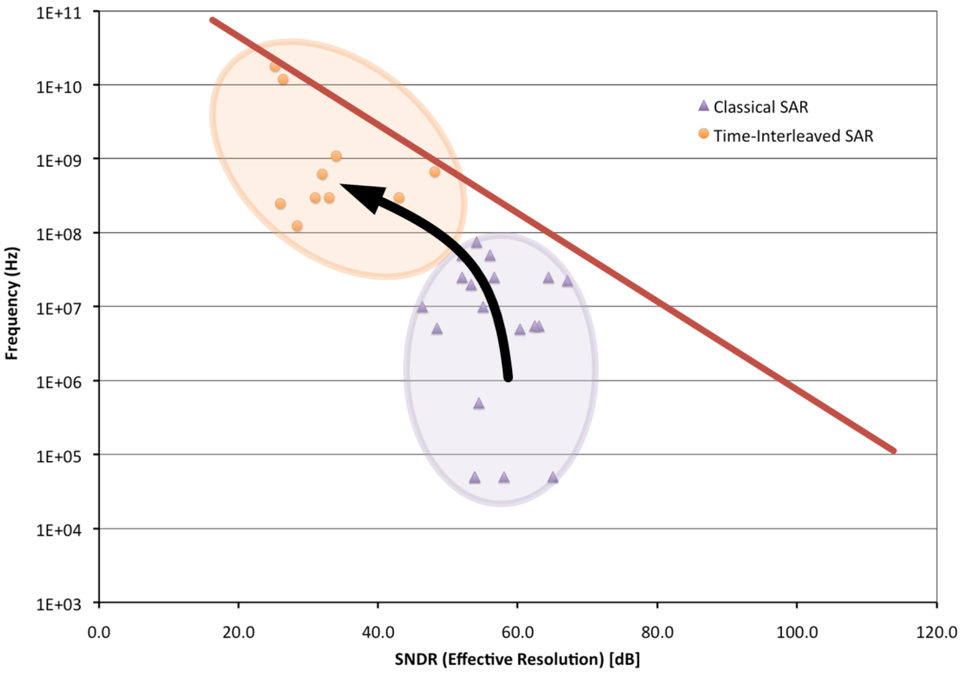


Figure 1.11 – The effect of time interleaving of SAR A/D Converters

As time interleaving can be used to extend the effective operation speed of a converter, it tends to degrade the resolution slightly due to systematic and dynamic errors induced due to mismatch in channels and matching of clock and signal distribution. Not all architectures lend themselves to time-interleaving as well as other. Figure 1.11 shows the effect of time interleaving of SAR ADCs. The devices chosen are all classical SAR ADCs [148-167], in purple, and all time-interleaved SAR ADCs [211-221], in orange, published at ISSCC and VLSI Symposium in the last 10 year. It is very clear to see how time-interleaving has been used to push the architecture into a new space of operation. SAR ADCs are the most popular choice for time-interleaving, based on publications, among the main ADC architectures, primarily due to their initial efficiency in power consumption. As will be explained in the next chapter, due to matching requirements, time interleaving can result in an increase in the overall power efficiency of the converter, despite the increase in sampling rate. Using an architecture, which is inherently more efficient, will result in a solution with power consumption acceptable for integration in SoC solutions.

It's also interesting to see how almost all time-interleaved SARs published achieve a lower resolution compared to the classical single channel SARs in the last 10 years. This is partially due to the fact that in current technologies it is very difficult for converters to cross the 1ps timing jitter line drawn above, and though time-interleaving has increased the effective sampling rate, the effective resolution will inevitably degrade. Also a time-interleaved ADC may achieve a sampling rate increase by the interleaving factor, compared to the individual channels, however commonly does not achieve the same accuracy of each individual channel at the top-level due to non-idealities at the interleaving top-level. Time interleaving is a very powerful tool to realise new specifications for different architectures, however the technique has many subtleties. The area of time-interleaving will be looked at in more detail in Chapter 2. Here only a short background was provided to help understanding of the following sections.

1.3.4 Re-configurability in A/D Converters

After a data converter is fabricated, the 3 main parameters of the system can theoretically be re-configured in the field. These are Power Consumption, Resolution, and Bandwidth. A reconfigurable ADC is one that allows the system controller to adjust the balance between these 3 factors in real time, trading Power-Consumption for Resolution or Bandwidth, or trading Resolution for Bandwidth itself. In mobile applications where the system is battery powered, the ability to reduce power-consumption in certain modes of operation is of great use. The loss of resolution at the ADC level, will at the communication level result in less available dynamic range per carrier, and hence lower potential maximum bit-loading per carrier. Making this trade-off can allow the system to adjust power consumption for maximum communication through-put (speed). Another point of re-configurability is the trade-off of power-consumption for operation speed. For time-interleaved ADC, this can be realised by powering down ADC channels, effectively reducing the number of time-interleaved devices, and hence reducing speed and power consumption. Examples of this will be shown in Chapter 2.

The other area of re-configurability is the trade-off of resolution for bandwidth, while maintaining a constant power consumption. This form of re-configurability as explained in the previous sections allowed the communication AFE to work to different standards of communication and potentially enable greater network coverage. In wireless communication, receivers which can tune to different standards of communication are often referred to as software defined radio systems. These can be realised either by using one very wideband A/D channel, or for certain types of multi-standard systems by realising a re-configurable ADC that can trade resolution and bandwidth. Re-configurable sigma-delta converters have been developed for use in software defined radio working to many different cellular standards. These will be looked at in Chapter 2. As shown in the graphs of ADC architectures, sigma deltas operate in the range of a few MHz to 10s of MHz, while many wireline standards today operate at much higher frequency bands, closer to 100s of MHz. To the author's knowledge in this frequency space no re-configurable ADC allowing trade-off of resolution and signal bandwidth has been reported to date.

1.4 Original Contribution of Thesis

This work aims to improve network coverage of digital communication systems, by exploring the implementation of an A/D converter which is able to trade resolution for bandwidth in the 100s of MHz sampling rate space. Such a data converter would play an integral part of a multi-standard and multi-PHY AFE capable of communicating over different channels to many different standards. The aim of the work is to demonstrate such re-configurability while maintain a near constant power consumption, and hence maintaining a power efficiency Figure of Merit over the whole reconfigurable space.

This converter is realised by extending the idea of time-interleaving to the extreme, applied to one of the slowest and simplest data A/D converter types, which is commonly left out of comparison graphs between data converter architectures. To achieve this, the following original work was carried out:

- Time Interleaving, and its non-idealities have been researched well in academic and industrial institutions, and much work can be found which discuss these non-idealities and apply to build time-interleaved converters. Most mathematical work focuses on the interleaving of 2 to 4 since algebraic work extending beyond this can be difficult. During the course of this work some basic modelling was done to try and understand how the non-idealities of time-interleaved systems manifest themselves when the number of channels is increased to greater than 100 channels. The work follows on from work done by [9][10][11] but further explores the significance of time-interleaving artefacts when the number of channels are increased or when such systems are used to quantise multi-tone, OFDM style signals, made of many frequency components with larger peak-to-average ratios.
- A new time-interleaved ADC architecture is proposed in this work. It builds on foundation work done in column parallel A/D converters used in CMOS image sensors. The parallel converter used in image sensors is

transformed to a time-interleaved, column serial converter, which can then be used to quantise a serial data stream. In the heart of this architecture is a newly proposed global parallel ramp generator, realised as a figure-of-8 rotating resistor string. Modelling work is carried out to understand the specifications of this block and its non-idealities. The proposed ADC architecture achieves re-configurability in field, allowing trade-off of resolution for sampling rate while maintaining a constant power consumption

- The design requirements of the sub-blocks for the new proposed ADC are analysed and a prototype implementation of the proposed ADC architecture is realised in 0.13μm Standard CMOS technology. The implemented ADC is designed to operate at 1GS/s 7-bit, 500MS/s 8-bit, and 250MS/s 9-bit configurations while maintain a near constant power consumption, and hence near constant figure-of-merit.

During the course of the work the following papers and patents were published and filed:

Danesh, S.; Hurwitz, J.; Findlater, K.; Renshaw, D.; Henderson, R.; "A Reconfigurable 1GSps to 250MSps, 7-bit to 9-bit Highly Time-Interleaved Counter ADC in 0.13μm CMOS", VLSI Symposium on Circuits, Proceedings of, 2011, 25-4

Findlater, K.; Bailey, T.; Bofill, A.; Calder, N.; Danesh, S.; Henderson, R.; Holland, W.; Hurwitz, J.; Maughan, S.; Sutherland, A.; Watt, E.; "A 90nm CMOS Dual-Channel Powerline Communication AFE for Homeplug AV with a Gb Extension", International Solid-State Circuits Conference, 2008, Digest of Technical Papers, p-p 464-628

Danesh, S.; Holland, W.; Hurwitz, J.; Findlater, K.; Henderson, R.; Renshaw, D.; "A non-uniform resolution step GHz 7-bit flash A/D converter for wideband OFDM signal conversion", International Symposium on Circuits and Systems, 2009, Proceedings of, p-p 964-967

Danesh, S.; Hurwitz, J.; "Analogue-to-Digital Conversion", United Kingdom Patent Application Number 1014418.6. Property of Gigle Networks.

1.5 Structure of Thesis

The work outlined above, and some further analysis and background material is covered in the following chapters. Chapter 2 provides further background to the area of A/D converters, their limitations and methods of quantifying and characterising their relative performances. In this chapter a closer look at time-interleaving will be presented, and results from numerical modelling of highly time interleaved system is presented accompanied by algebraic derivations from the non-idealities in time-interleaved systems. The chapter is concluded with some examples of time-interleaving, re-configurability, and calibration in different ADC architectures.

Chapter 3 will begin to explain the new ADC system, designed to realise large re-configurability in resolution and speed. The architecture and design trade-offs will be explained. The basic concept of the Global time-interleaved ramp-generator will be explained in this chapter. Chapter 4 discusses the implementation of an actual re-configurable ADC to the architecture explained in Chapter 3 in a standard 0.13µ CMOS process. The implemented ADC is design to operate at 1GS/s 7-bit, 500MS/s 8-bit, and 250MS/s 9-bit configuration while maintaining a near constant power consumption. The detailed design report is presented in Chapter 4.

Chapter 5 will look at the measurement results from silicon from the device implemented and analysis the results to identify the effect of individual non-idealises in the system. Chapter 6 draws conclusions from the work by looking back over the implementation work during the course of this project, and tries to assess if we have achieved the goals set out up front and how the architecture developed may help multi-PHY communication systems. A critical analysis of the design shows areas of improvement and in particular looks at technology scaling. Estimations are done on how this architecture would scale with technology, moving forward to 45nm and below, and looks to explore further work that can be done on the architecture to greater advance its performance and potential re-configurability options.

1.6 Conclusion and Summary

It is clear that digital communication is advancing in ways to allow better coverage and reliability, and a key part to this is the development of more diversity in the communication apparatus. One powerful method to realise diversity in the communication is the use of multi-standard and multi-PHY systems. With the goal of maintaining low cost and especially lower power consumption, ideally the sub-blocks in the system should be re-configurable to allow transformation of the PHY from operation in one standard to another. This is achieved in the DSP part of the PHY by appropriate segmentation of the system, while in the AFE is commonly achieved either by over-designing some parts of the system or implementing re-configurability in the individual AFE building blocks.

In the Analog Front End, the A/D converter can be difficult to make re-configurable when signal bandwidths in the 10s to 100s of MHz are present, and for such bandwidths simply over specifying the block to a super-set of performance can be very costly for area and in particular power consumption, if even possible. Time-interleaving has emerged as a key technique to allow ADCs of different architectures to achieve specification parameters they could not initially achieve, and to allow greater sampling rates, sometimes at the cost of linearity. In this work we exploit time-interleaving to a much higher degree compared to work previously published for main-stream communication systems, by exploring and realising a highly time-interleaved system made of units of one of the slowest type of data-converters which is commonly dismissed from data converter comparisons. To realise this a better understanding of highly time-interleaved systems are required which is obtained through numerical modelling and then combined with new circuitry to build a prototype re-configurable time-interleaved ADC implemented in silicon. The work explains the requirements and implementation work, analyses the results from silicon and draws conclusions on how this architecture will scale with technology.

Chapter 2 - Analog to Digital Conversion

2.1 Introduction

To design A/D converters that meet the requirements of the system they are used in, understanding of the specification and low-level non-idealities is critical. Meeting these specifications will increase the likelihood of the block meeting performance and functional requirements when used in a system. In this section, conventional specifications and performance metrics for A/D converters will be defined, and the common techniques to measure these at the time of design, in simulation, and on the bench for the actual fabricated ADC will be discussed. These specifications can be applied to all A/D converters, including those using time-interleaving concepts, however in time-interleaving systems the implications of low-level specifications on the converters over all performance may be different from that of the individual channel architectures. Following an outline of basic specifications and non-idealities in A/D converters, a close look at the implications of specification mismatch between the channels of a time-interleaved converter will be presented. Algebraic equations previously published, and numerical modelling and curve fitting will help to explain the effect of these differences and help to define requirements on these parameters to meet the overall ADC requirement.

When designing an A/D converter, clearly one attempts to meet all low level specifications necessary to meet the overall converter performance required by the system, however in practice this my not be possible or may come at a huge cost for area or power consumption. One of the techniques, which is becoming more common, to improve the performance of the block after fabrication is the use of calibration. Calibration now plays a key part in A/D converter design. Some calibration techniques may have implications at the system level, and may only be effective for given length of time. This chapter provideds an overview of different calibration techniques, how the correction amount is identified, how it is applied to the converter and how often correction loops are performed.

Following a high-level over-view of categorisation of different calibration techniques, the many approaches to calibration are explained using examples. The chapter is concluded by looking at the main classical ADC architectures, Flash and Folding, Pipelined converters, SAR ADCs and Sigma-Delta oversampling converters, showing examples of time-interleaving, re-configurability, and calibration in published work for each architecture.

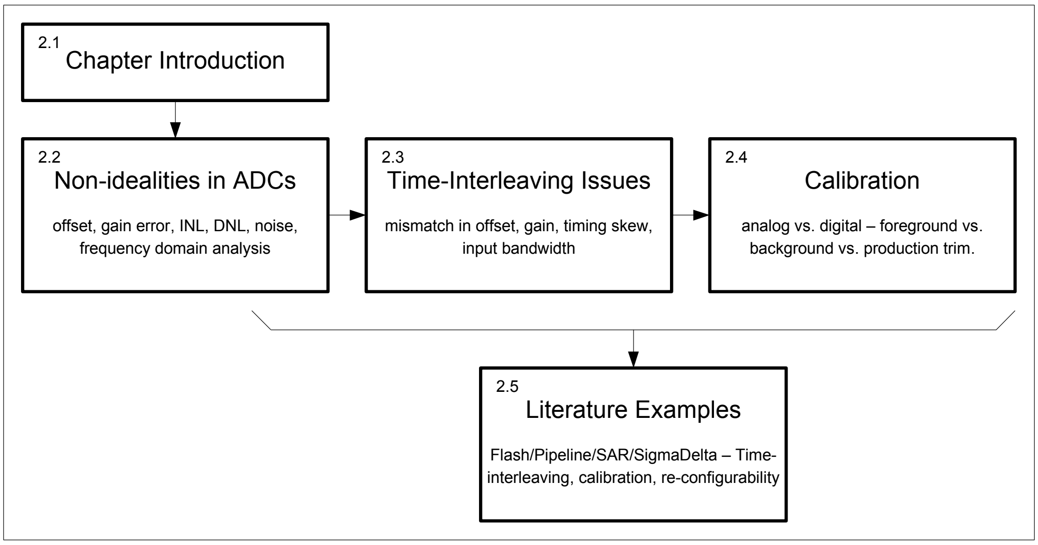

2.1
Chapter Introduction
2.2
Non-idealities in ADCs
offset, gain error, INL, DNL, noise, frequency domain analysis
2.3
Time-Interleaving Issues
mismatch in offset, gain, timing skew, input bandwidth
2.4
Calibration
analog vs. digital – foreground vs. background vs. production trim.
2.5
Literature Examples
Flash/Pipeline/SAR/SigmaDelta – Time-interleaving, calibration, re-configurability

Figure 2.1 – Structure of Chapter 2

2.2 Specifications and Non-idealities in ADCs

The non-idealities of a converter can be looked at in the time domain, or frequency domain. Some non-idealities are easier to see and quantify in the time domain, so even though the converter output may be used in a frequency and phase domain such as in communication systems, some analysis of the converter in the time domain may be useful. Also some non-idealities may have implications for performance in certain applications but not in others. It is important to understand all these issues, since when used in time-interleaving configurations, some static non-idealities which individually may not affect the performance at the application level, may turn into dynamic and more complex non-idealities which may degrade the performance in a different way. Here the main non-idealities will be reviewed and means of measuring and quantifying each at simulation and chip measurement will be presented.

2.2.1 Offset

Offset in the ADC transfer function can be clearly identified and measured. Figure 2.2 shows the transfer function of a 5-bit example ADC, its ideal transfer function and that of an ADC with 3-code error in offset.

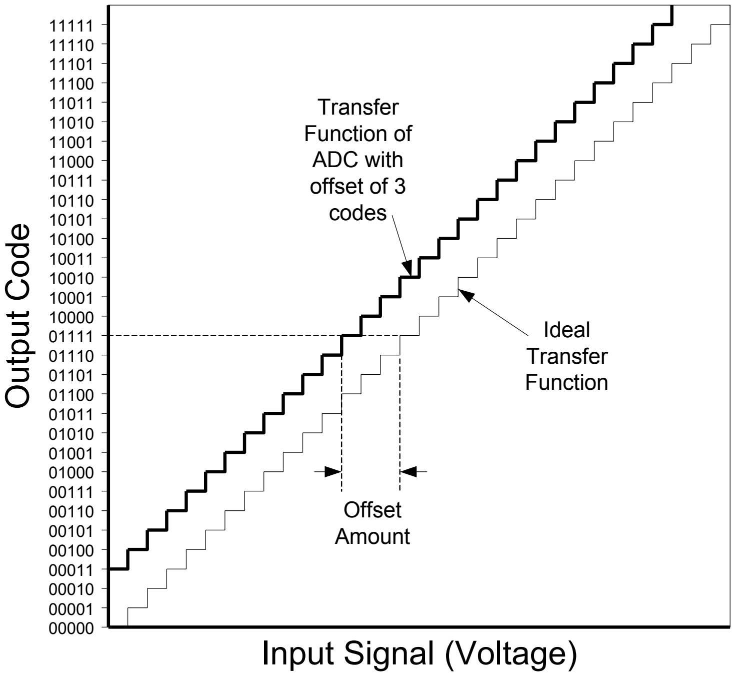


Figure 2.2 – Transfer function of an ADC with 3-code output offset

The x-axis for all ADC transfer functions is a continuous axis, representing the input analog voltage which is continuous. The y-axis has only discrete values, representing the digital code outputs. Here a 5 bit ADC is used as an example, hence only 2^5 possible values are available on the y-axis. The offset error clearly adds a fixed error to all output codes and is independent of input signal magnitude. In many applications, which use the output of the ADC in the frequency domain such as frequency and phase based communication systems, this offset error does not effect the accuracy of the converter at first order. For other applications, such as in instrumentation, where the exact analog voltage is in question, this offset directly affects the accuracy [12]. In the calibration section we will look at ways to correct this offset if it cannot be reduced to a sufficiently low values by design. For the transfer function of the converter with offset shown in the Figure 2.2, output code 00000, 00001 and 00010 will never occur. In reality due to the offset we have reduced the output range of the converter, and in fact reduced its effective resolution. This is no longer a 5-bit converter, but in reality a 4.86-bit converter at best.

2.2.2 Input Range and Gain Error

For every ADC an input range (voltage range) is defined, meaning when the input signal voltage is at the minimum-range, the output code should be zero (the lowest value), and when the input signal voltage is at the maximum-range the output code should be at maximum, and the other codes should be equally divided in between. Gain error is when the ADCs internal understanding of this signal range is different then the specifications, or that expected by the other blocks in the system.

Figure 2.3 shows the ideal transfer function, one with positive gain error and one with negative gain error. For instrumentation applications gain error can have large implications for the system accuracy [12] however for frequency domain communication applications, the exact gain error of the ADC is not so important since it will slightly reduce the signal peak power compared to the expected level, and commonly such systems require equalisation of the channel and potentially the receiver prior to frame communication. The implications of positive and negative gain error are different. In the presence of negative gain missing codes at the end for the transfer function can be seen, and in the positive gain transfer function clipping of input voltage to max code in the output can be seen. The missing code problem from negative gain is similar to the loss of effective resolution experienced due to offset. The clipping on the other hand could have much greater implications, where parts of the input signal are totally missed by the data converter. In many communication systems, a PGA is used prior to the ADC which amplifies or attenuates the input signal to use the full range of the ADC correctly regardless of what is specified as the full range of the converter [13]. Clipping of an input stream in a frequency domain communication apparatus has major implications in performance [14].

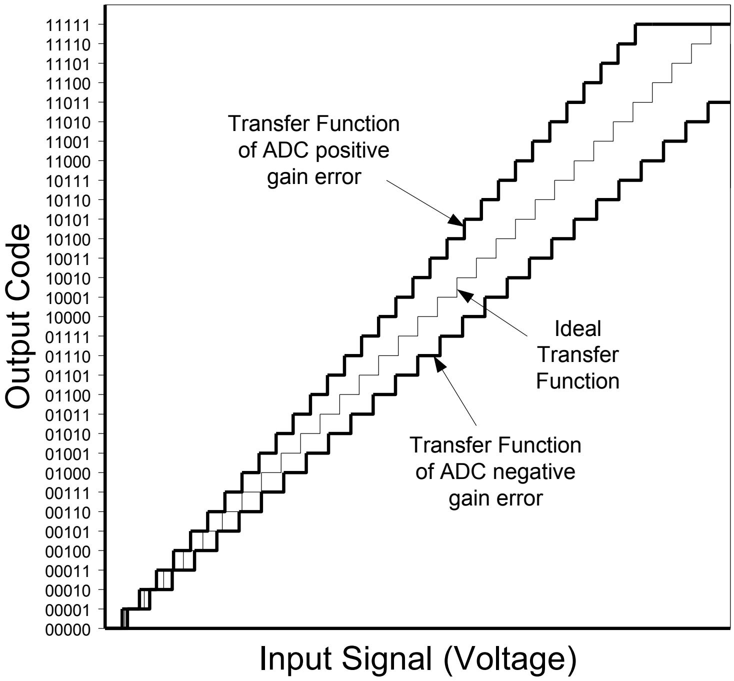


Figure 2.3 – Positive and Negative gain error in transfer function

It is worth noting that the magnitude of the error (inaccuracy) due to gain-error, both positive and negative, is signal dependent. Meaning the size of the error is proportional to the signal amplitude. Crucially, assuming the gain and transfer function of the ADC itself is linear, the inaccuracy due to gain-error, despite being signal dependent, will also be linear in relation to the signal, and hence not produce harmonic tones in the transfer function. This will be explained in greater detail later.

2.2.3 Static Non-Linearity in the Transfer Function

In the previous figures, the transfer function of the ADC, despite the step-ladder shape may be estimated as a linear slope. Non-linearity in the transfer function of the converter can be defined as curvature in the whole transfer function or non-linearity or broken segments in between. Figure 2.4 shows an example of transfer function with non-linear characteristics.

Specifying, quantifying and measuring non-linearity is an important part of ADC design. This can be done in many ways, in the voltage and time domain, or in the frequency domain. In the voltage/time domain this is done using two definitions: Differential Non-Linearity (DNL) and Integral Non-Linearity (INL). DNL is simply defined as the difference between the size of code step in the transfer function of the ADC compared to the size of an LSB. The DNL for the transfer function shown in Figure 2.4 is also shown in the graph. INL is the defined as the difference between each code value in the actual ADC transfer function compared to that of an ideal ADC. This is also shown on Figure 2.4.

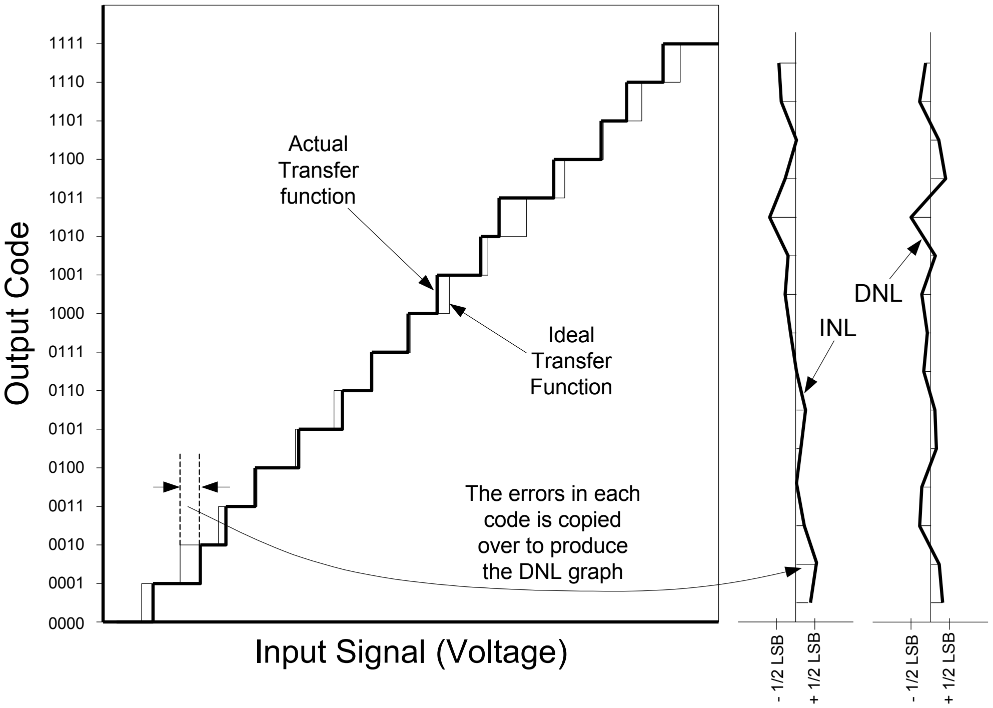


Figure 2.4 – Non-linearity in transfer function, DNL and INL

The INL values for each code are in fact the cumulative summation of all the DNL codes prior to that code in question. This is fairly obvious since DNL tracks the effective error or non-linearity between neighbouring codes, while INL tracks this error or non-linearity for the whole ADC range. INL is commonly defined in two ways, the Best Straight-Line INL (BSL INL) and End-Point INL (EP INL). The INL graph shown in Figure 2.4 is a BSL INL, where each code is compared to the ideal transfer function, as the ideal transfer function is specified. This BSL INL apart from non-linearity also reflects the range and gain error, as well as offset. As mentioned previously, in some applications offset and gain-error may not be as important. For such applications EP INL can be used which prior to the comparison, using the voltage level for the start and end code, an ideal transfer function is generated, and then compared to the ADC transfer function for INL calculation. EP INL only reflects information about non-linearity of the converter, and effectively removes the contribution of gain error and offset.

2.2.4 Noise in A/D Converters

At the system level many sources of noise may exists, at the ADC level for most applications, noise is dominated by flicker noise, and thermal noise. Flicker noise is of greater importance to instrumentation applications, working with low frequency signals, while in broadband communication the dominant source of noise in the converter is thermal noise. It is impossible to build an ADC with zero thermal noise, unless it is operated at zero Kelvin, where the ADC would be non-functional. So if the presence of noise is inevitable, the real question is to what level should the noise be reduced, in order not to limited the performance of the converter, and meet the specifications set forward by the system designer. Due to the statistical nature of noise, it is very difficult, if not impossible to remove it by calibration.

It is difficult to show noise in a basic transfer function graph, since by the very statistical nature of noise, it will appear differently for different measurements. To understand how noise affects the transfer function and its acceptable level, Figure 2.5 looks closer at a code transitional point. Here output code Y represents input voltages between B and C, and output code Z represents input voltage between C and D. Due to the nature of quantising, an error will inevitably occur since we are mapping from a continuous space to a finite space. If we choose to define Code Y to represent input value BC, then no error has occurred for the quantising of value BC, however for values bigger than BC, up to C, an error has occurred as large as half an LSB (the distance between B and C) and the same for values below BC down to B.

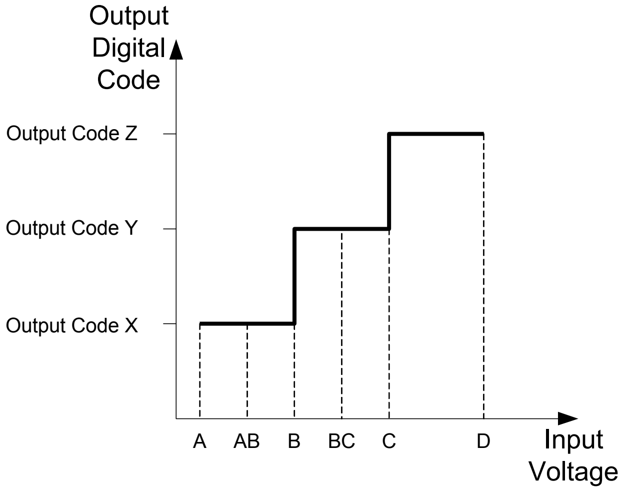


Figure 2.5 – Code transition in the presence of noise

In the presence of noise in the system, a value just below input voltage C, momentarily may be subject to noise and appear above node C and convert as output code Z. The transfer function on an ADC in the presence of noise in fact is a thick line representing the statistical probability of matching of output code to input voltage. An example 3-bit transfer function is shown in Figure 2.6. The thickness of the line represents the noise in the system. Here the noise has been drawn exactly equal to the quantisation level, and in fact not reducing the accuracy of the conversion, only dithering the mapping of input voltage to code within the quantisation error. Node A falls exactly inside the space of code 010, while Node B falls over two code ranges. If input voltage B is held at the input, the output of the ADC would toggle between code 100 and 101 due to noise, occurring 3 times more frequently than of code 100. Looking at Figure 2.6, it is clear that if the width of the transfer function is any bigger, then for example voltage A and B could move between 3 output codes statistically, which would degrade the transfer function.

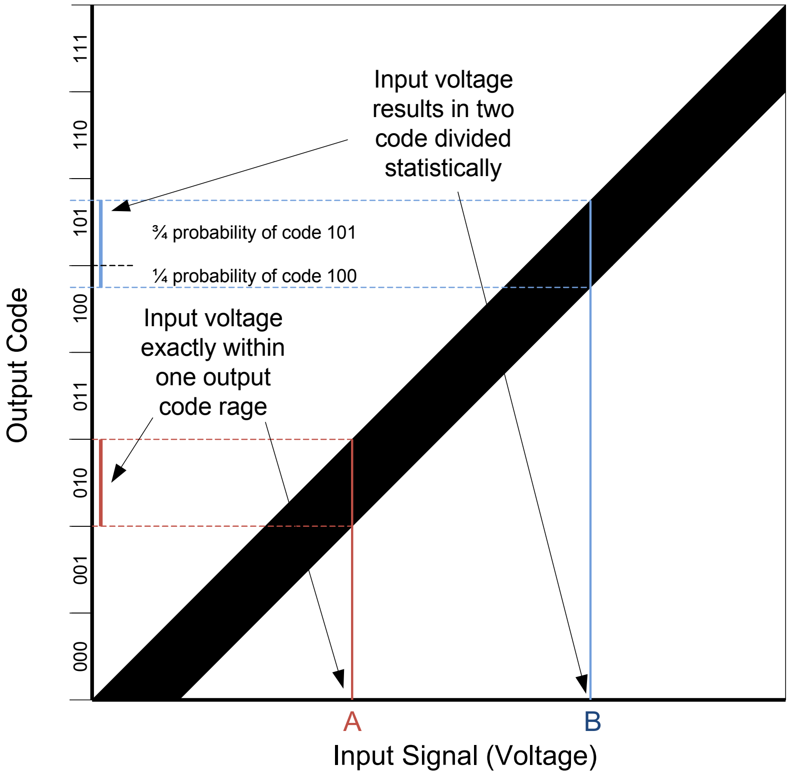

Input voltage results in two code divided statistically
¾ probability of code 101
¼ probability of code 100
Input voltage exactly within one output code rage
Output Code
111
110
101
100
011
010
001
000
A
B
Input Signal (Voltage)

Figure 2.6 – Transfer function of ADC in the presence of noise.

As mentioned above, the act of quantising itself will have some error, and this error can be looked at as noise in the system. Quantisation noise is saw-tooth wave in nature, as shown in Figure 2.7, and hence has an RMS noise value of LSB/√12 (This is the RMS of a saw-tooth wave with a peak value of LSB) [5]. Any statistical noise in the system will add in quadrature to this value and set the effective noise floor of the converter. The application will define the required signal to noise ratio.

Noise can be measured and quantified as a voltage and voltage ratio as an RMS or effective sigma value. To measure this in the time domain, the output of the ADC can be differenced from the input (the ideal signal), and then a standard deviation can be calculated from this noise series. Alternatively noise can be defined and measured in the frequency domain, as noise power per Hz. This can be defined for example for communication system, and measured using an FFT. The noise power per Hz figure can then be integrated over the band of interest to calculate the RMS noise equating back to the voltage domain standard deviation.

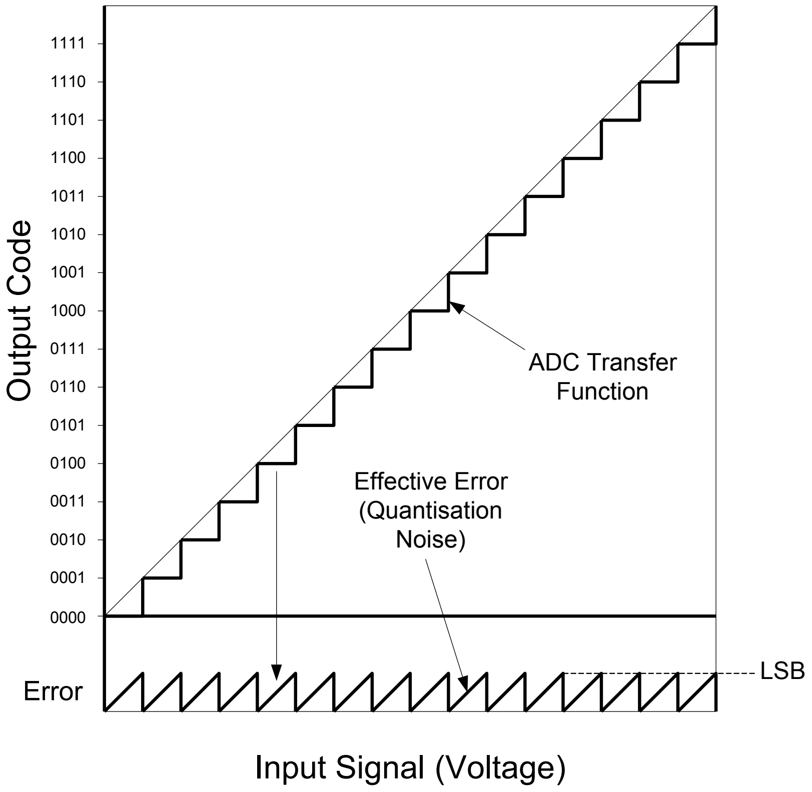


Figure 2.7 – Transfer function of ADC with magnitude of quantisation noise.

More specifically, for example in a multi-carrier communication systems, the output of the converter is fed to an FFT block (present in DSP), which looks at the output signal in the frequency domain to understand the communicated information. FFTs take a certain number of samples from the ADC as input, which represent a certain length of time. This length of time can be used to determine the lower-frequency of the band of interest. Any noise below this frequency remains constant over the length of the samples captured for the FFT, and hence will appear as an offset. The highest frequency is the sampling rate of the converter or bandwidth of the converter, regardless of the FFT frequency since any noise above the FFT frequency present in the ADC output will be folded into the band of interest. To calculate the effective RMS noise floor for an ADC used in multi-carrier communications, the noise power per Hz should be integrated from one over the FFT length up to very high frequencies.

2.2.5 ADC Non-idealities in the frequency domain

As mentioned previously, the non-idealities in an ADC transfer function can be looked at in the voltage and time domain, matching input voltages to output codes, and using metrics to determine the accuracy of this conversion. Alternatively, the performance of the converter can be assessed, specified, and measured in the frequency domain. Not all static non-idealities are as easy to observe in the frequency domain, but most are possible. More importantly the frequency domain analysis shows the relative non-idealities over the frequency band, for example showing the relative presence of noise at different frequencies. Furthermore in multi-tone digital communication the output of the ADC is used in the frequency domain, and hence it is easier to specify the block with respect to its transfer characteristics in the frequency domain.

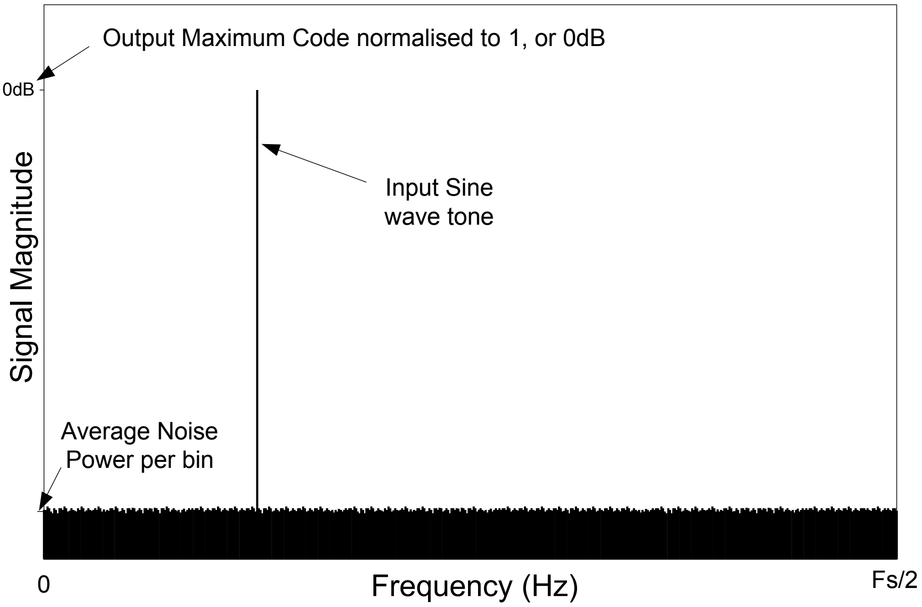


Figure 2.8 – Ideal output spectrum of an n-bit ADC with 1/8 Fs input sine-wave

Frequency domain testing is based around using a sine-wave (or collection of sine-waves) as inputs to the converter, and performing an FFT on the output code stream. Figure 2.8 shows the spectrum of an ideal n-bit ADC. For this graph an input sine-wave at 1/8 of the sampling rate has been used, and FFT of the output stream has been performed at the sampling rate of the converter. The single tone in the output represents the input sine-wave. The FFT has been normalised so that maximum code represents a 1, and hence the peak of the tone is at 0dB. Note despite the use of an ideal converter, an effective noise floor can still be observed in the spectrum. This is due to the effective noise floor due to quantising.

As previously mentioned, quantisation noise has an RMS value equal to the converter LSB/√12. This noise is uniform over the whole frequency band. When performing an FFT, the width of each bin is equal to the sampling frequency divided by the number of bins in the FFT. If the number of bins used in the FFT of Figure 2.8 was increased, the effective width (in frequency) of each bin will reduce, and the effective noise power for that bin will also decrease. This may cause the 'Average Noise Power per bin' line to move lower, however this does not mean the effective signal to noise ratio has improved. The total noise is the sum of power of all the bins apart from the signal bin. If the number of bins is increased, the power of each bin will go down, but the number of bins would have increased by the same ratio, maintaining the same overall total noise power within the band.

Figure 2.9 shows the spectrum of a non-ideal differential ADC with the same 1/8 sampling frequency input sine wave, through an FFT at the sampling rate. The gain error can be quantified as a dB number as the difference between the main 1/8 Fs tone compared to the 0dB point. The offset can also be quoted as a dB voltage number which is the power in the 0th bin (DC bin). The non-linearity of the system has resulted in spurious tones at multiples of the input frequency. In this example, the ADC is a differential system, so the power of the even harmonics is low. The largest harmonic is the 3rd.

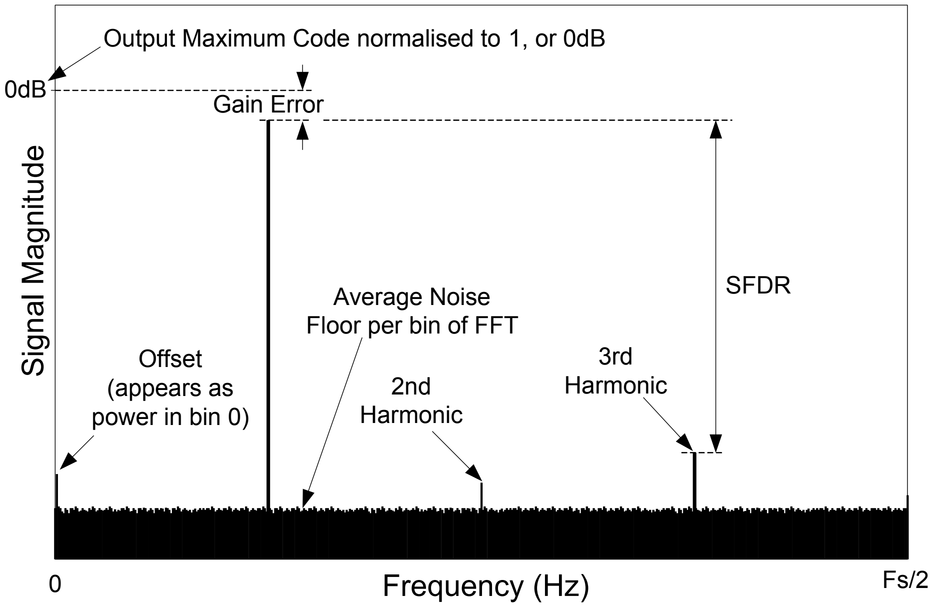


Figure 2.9 – Output spectrum of non-ideal ADC with 1/8 Fs input sine-wave

In the voltage and time-domain non-idealities, INL and DNL were explained as metrics to quantify the non-linearity of the ADC. In the frequency domain, other metrics are commonly used for measure of non-linearity.

Spurious Free Dynamic Range (SFDR) is the difference in power (in dB) between the main tone signal, and the strongest harmonic, commonly the 3rd harmonic in differential systems. SFDR can be read off the spectrum, and at first hand is used as an easy way to estimate the linearity of the system, however has limited accuracy since all other tones have been ignored.

Total Harmonic Distortion (THD) is a more accurate way to measure the linearity of a system, and is defined as the ratio of the signal tone to the sum off all harmonic (spurious) tones in the output. THD cannot be simply read off a spectrum plot, and the power in each harmonic bin of the graph should be summed offline and compared to the signal tone.

The FFT output can also be used to calculate effective noise levels in the system. The effective RMS noise of the system can be calculated by integrating the signal level in all bins excluding the signal and harmonic bins. Of course the signal and harmonic bins are made from a signal element and a noise element, but dismissing these bins all together will only introduce a very small error that can be ignored, especially if the number of bins of the FFT is high.

Signal to Noise Ratio (SNR) is the ratio of the signal tone to the sum of all other bins in the FFT excluding the harmonic bins. SNR is a means of measuring the effective noise level in the system, independent of linearity.

Signal to Noise Plus Distortion (SINAD or SNDR), is effectively the sum of SNR and THD, and is defined as the ratio of the signal tone to the sum of all other bins including noise and harmonic bins. SNDR is a real metric of the effective resolution of the converter. Since SNDR is a metric for the effective resolution of the system it can be converted to effective bits.

Effective Number of Bits (ENOB) is a mathematical conversion of SNDR for ease of comparison to original specifications of the converter and means of summarising the effective performance of the converter in units of 'bits'. SNDR can be considered the RMS power of the sine-wave, divided by the RMS noise of an equivalent ADC, which is LSB/√12. If this equation is re-arranged, to find the size of the LSB which results in the measured SNDR, the effective resolution of the system, and hence the ENOB can be found [5]. The conversion is shown in Equation 2.1.

$$ENOB = \frac{SNDR - 1.76}{6.02} \tag{2.1}$$

The frequency domain metrics for linearity such as THD and SFDR should have correlation to INL and DNL measurements from transfer functions explained earlier, since they both are means of measuring linearity of the converter. In practice it is seen that these numbers do not always correspond. In fact the effective THD and SFDR of converters is commonly different for different input frequencies. In the example above the input sine-wave was at Fs/8, while a sine-wave closer to Nyquist for most converters yields a lower THD figure. Because of this firstly it is important to measure THD, SFDR, and SNDR at different input frequencies, but also has led to new definitions for INL and DNL.

The INL and DNL described earlier, found by transfer function comparison is commonly referred to as Static INL (S-INL) and Static DNL (S-DNL), since for these measurements a very low frequency signal, effective DC, is used to build the transfer function. Further to these definitions, Dynamic INL (D-INL) and Dynamic DNL (D-DNL) are also defined, which are used to estimate the linearity of the ADC at particular frequencies.

DNL is defined as the size of code steps compared to the theoretical correct size (equal to an LSB). Rather than using transfer functions, which are difficult to construct (specially in the presence of noise), In theory one could measure DNL in a statistical way, by feeding an input signal which spends time over the whole input range equally. The output of this test would be a large sequence of codes. The number of occurrences of each code can be counted. For each code its number of occurrences at the output compared to the total number of captured codes is a representation of how long the ADC believes the input signal was within this code range, and a representation of that codes effective size on the input axis. For an ADC with equal code sizes, the occurrence of each code on the output would be equal, however if the step size of one code is bigger than the others, that code would appear more often at the output. Constructing perfectly linear ramp inputs to travel the input equally to perform this test can be difficult in practice and may not be useful. Alternatively a non-uniformly distributed wave such as a sine-wave can be used as an input. Of course on the code output stream, codes towards the edges of the range will occur much more regularly because of the shape of a sine-wave, however this irregular shape is well known and understood and at first pass the regularity plot of the output codes can be corrected for the shape of the sine-wave (Figure 2.10). Following this correction the number of occurrences of each code should be equal again, and any error from this is a representation of the DNL value for each code. The INL can be calculated from the DNL by cumulative summing. This method of DNL and INL measurement is referred to as the histogram method.

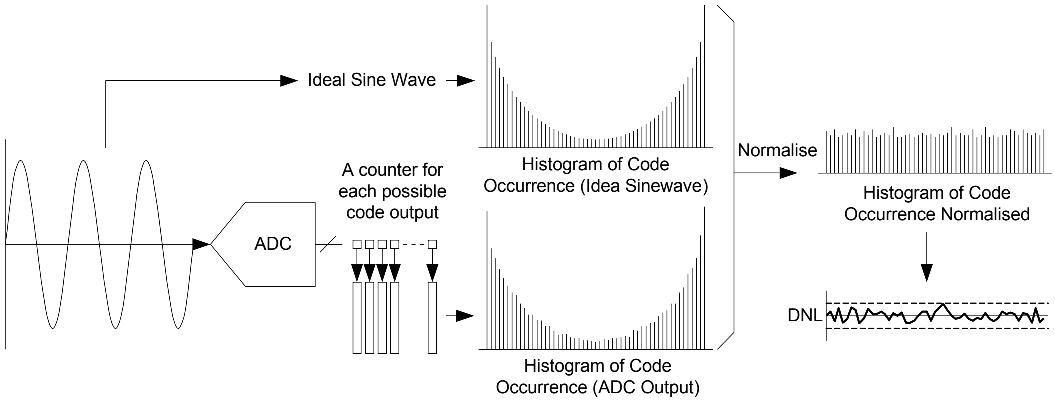


Figure 2.10 – Calculated Dynamic DNL from sine-wave using histogram

Using this method the effective DNL and INL for different input frequencies can be calculated. Dynamic DNL and INL correspond to the THD for the same input frequency. In fact the same digital output stream can be used to calculate THD though an FFT while being used for occurrence counting (binning) to estimate DNL and INL.

As explained above, due to dynamic elements within the ADC, the ADC performance is very input signal dependent. And since an ADC is not necessarily a linear-system, super-position may not apply. For this reason it is important to simulated, quantify and measure an ADC with input signals resembling the actual signal the ADC would be digitising in normal operation especially for multi-tone communication applications.

Multi-Tone Power Ratio (MTPR) is a figure representing the effective linearity of the converter when digitising a multi-tone signal. Figure 2.11 shows the output spectrum of an ADC when digitising a 32-tone input signal where all tones have equal power. These tones are equally spaced. Since the bin width of the FFT is larger than the spacing between the tones, in the output spectrum the tone, harmonic, and noise floor can be seen.

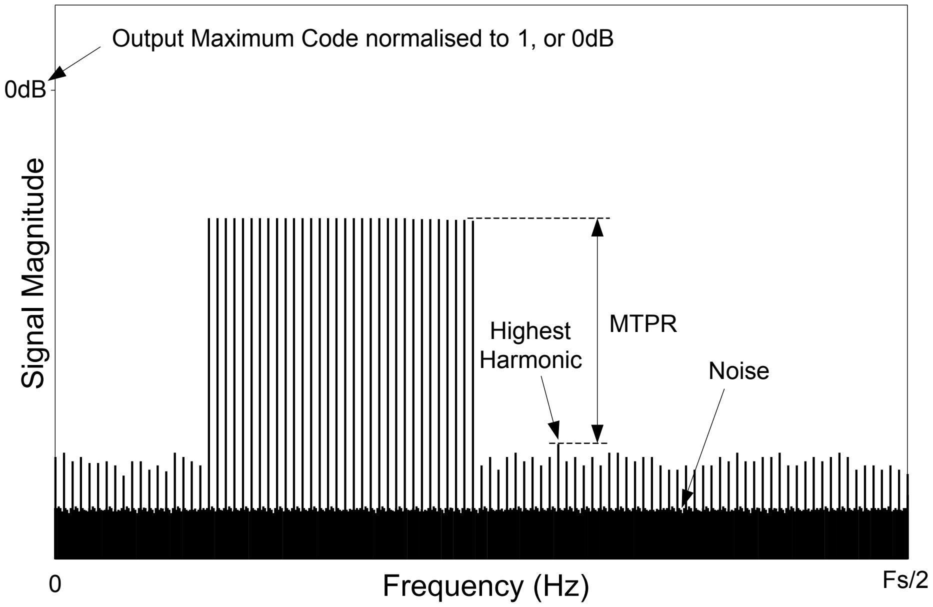
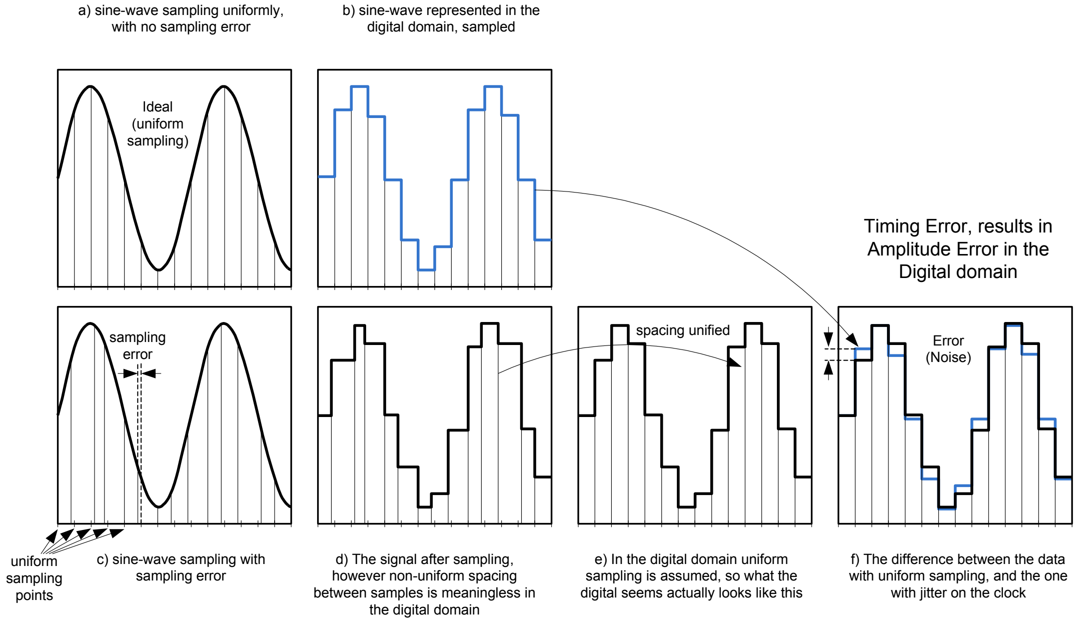


Figure 2.11 – Output spectrum for Multi-Tone MTPR test

The spurious tones visible in an MTPR test are the power combination of all the spurious harmonics of each of the input tones. In a real multi-tone communication symbol, a particular carrier may be at a low power level compared to its neighbouring carriers due to the information it is communicating. The harmonics from all the other carriers may fall inside the space of this carrier and combine causing an error in that carrier. The MTPR test is designed to identify the level of interference carriers can have on one another. In some MTPR test notches may be introduced, where one tone in the middle of the multi-tone pack is left empty, and the harmonic combining in that bin is measured as the MTPR performance. When performing MTPR tests, it is useful to use the exact number of carriers and spacing of the communication system in which the ADC will be used for.

2.2.6 Sampling Inaccuracy, Jitter and Clock Drift

All non-idealities explained up to this point look at errors in the process of quantising, however an A/D converter is the process of sampling as well as quantising. Separate to the quantising errors covered, sampling errors can also occur in A/D converters. Sampling errors cannot be shown on a transfer function, since they only manifest as errors in the system for a moving (non-DC) signal. Figure 2.12 shows sampling error occurring when digitising a sine-wave.


a) sine-wave sampling uniformly, with no sampling error

b) sine-wave represented in the digital domain, sampled

c) sine-wave sampling with sampling error

d) The signal after sampling, however non-uniform spacing between samples is meaningless in the digital domain

e) In the digital domain uniform sampling is assumed, so what the digital seems actually looks like this

f) The difference between the data with uniform sampling, and the one with jitter on the clock

Figure 2.12 – Effect of sampling error on overall amplitude error

Figure 2.12a and 2.12b show the sine-wave in the analog domain and digital domain when uniformly sampled. Figure 2.12c and 2.12d shows how the sampling points can be subject to time-domain noise (jitter), resulting in sampling at the incorrect points. In the digital domain, the arriving samples are assumed to be uniformly sampled and Figure 2.12e shows what the digital domain believes the input signal was originally. Figure 2.12f shows the signal generated with sampling jitter compared to the correctly sampled signal. Clearly errors in the amplitude axis can be seen. This error can be represented as noise in the transfer function of the converter.

Two forms of sampling error can occur with very different effects. Firstly the simple sampling noise, or jitter, also called cycle-to-cycle jitter. The second form of sampling error is long term drift of the clock, or long-term jitter. The two forms of sampling error result in very different effects.

Sampling noise, or jitter results in errors similar to those shown in the Figure 2.12. Jitter can come from many sources, including thermal noise of devices in the path of the sampling circuitry, the thermal noise of the sampling circuitry itself, or thermal noise in the VCO. Amplitude noise on the power supplies can also induce timing jitter noise.

The error, or noise, from timing jitter is much larger for fast moving signals compared to slow moving signals. This is clear since the error is the amount the signal has managed to move from the actual sampling point compared to the correct sampling point. For a long enough digital sequence of a single tone signal, the error due to sampling sometimes results in large errors and sometimes results in smaller error depending on the rate-of-change of the signal near the sampling point, assuming the jitter in the clock is non-correlated from sample to sample, the jitter can be assumed to be Gaussian in nature, and hence results in a directly Gaussian noise in the output of the ADC (i.e. transfer function of the converter) [15][16]. The amplitude of this noise, or the effective signal-to-noise ratio due to jitter is equal to [10]:

$$SNR_{jitter} = 20.\log\left(\frac{1}{\sigma_{jitter}.2\pi.f_{in}}\right) \tag{2.2}$$

Where σ_{jitter} is the standard-deviation of the sampling noise, f_{in} is the input frequency, and SNR_{jitter} is the effective Signal-to-noise ratio due to sampling error.

Long-term jitter commonly manifests itself due to temperature change or low speed power supply change, and 1/f noise in clock generation and distribution circuitry and is measured as the change in sampling clock over a long stretch of time. For example, for a high-speed clock, at every clock cycle, the period of the clock could grow by 1fs. This can be seen as a small amount, but over a long stretch of time this results in a slow drift in the effective sampling frequency. Long-term jitter, if linear drift, does not result in noise, or non-linearity in the output spectrum or for the transfer function of the system, however the effective frequency of the signal compared to the sampling clock has changed over time, this can result in problems for systems which use the output of the ADC in the frequency domain [17].

For example lets assume the following system configuration: An ADC with a 1GHz sampling rate, used in a communication system. The output of the ADC is FFTed at 1GHz, with a 1024-bin FFT. The length of the symbol will hence be 1.024µs. Assume a single tone 95.703125MHz input signal. A full cycle of this sine-wave would take 10.44898ns to complete, meaning during the length of the symbol (a FFT symbol of 1.024µs), a total of 98 cycles of the input signal will be completed, hence after FFTing the output of the ADC, this tone will sit exactly in the 98th bin of the FFT. Now if the sampling frequency drifts over the length of the frame, the sampling step size can grow or reduce, meaning over 1.024µs no longer exactly 98 cycles of the input signal will be completed, this means once FFTed some fraction of cycles of the input signal will be present in a output stream resulting in bin-leakage in the FFT. In a multi-tone system, this leakage can land in the neighbouring bins, limiting the SNR of other carriers.

In many modern multi-tone communication systems a requirement on the absolute accuracy of the sampling clock is set by the system designer. In such systems commonly fractional PLLs are used allowing the receiver to manually tune the clock to exactly the same clock as the transmitter to eliminate bin leakage problems explained above, however this adjustment is done over a very long length of time, and hence still a requirement on the circuitry exist to maintain their absolute clock accuracy over periods of time, setting a requirement on long-term-jitter. This value is calculated based on the number of carriers, the bit-loading per carrier, carrier spacing and symbol length, commonly done by DSP modelling, but once this long-term jitter number is understood it is important to confirm that the circuit meets the clock stability requirement over the length of the frame to maintain the required long-term-jitter target.

2.3 Time Interleaving and non-idealities

The concept of time interleaving A/D Converters was briefly explained in Chapter 1. Figure 2.13 shows a time-interleaving ADC with X different channels. Here the global T/H front end has been omitted for a time-interleaved T/H in each channel of the ADC. This is becoming very common trend since implementation of the one high-speed sampling front end (similar to that shown in Chapter 1) is becoming very hard for high-performance time-interleaving systems, and although this relaxes the timing matching requirements per channel (as will be explained later), the power consumption of a global front end is commonly too high, and can be a limiting factor for the overall ADC linearity [18]. In this example the track-phase is X times shorter than the hold-phase, allowing further time for the channel ADC operation. This style of sampling and timing will be explained further in Chapter 3.

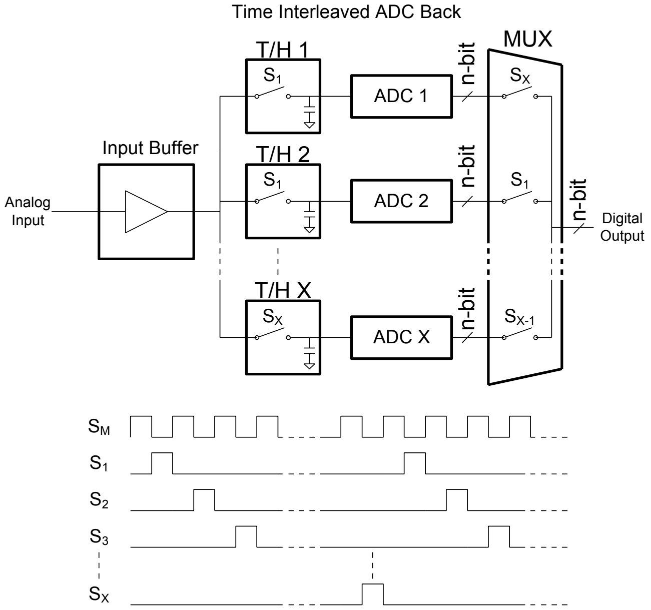


Figure 2.13 – Example of the effect of offset on time-interleaving ADCs

In time-interleaved systems, assuming all ADC channels match exactly in all specifications stated in the previous sections, the overall data-converter will have the same specifications and non-idealities as the sub-channels. Unfortunately in practice not all channels of the time-interleaved system will match in the characteristics equally, and this mismatch in performance results in artefacts in the overall converter transfer function. As will be explained below some non-idealities in the sub-channels can be transformed into totally different artefacts at the top-level when they are present at the sub-ADC levels with different amounts. The key to good performance in a time-interleaved ADC is matching between sub-channels but unfortunately perfect matching of devices and blocks is not possible on silicon. This section looks at the effect of mismatch in the offset, gain-range, sampling-timing, and input bandwidth between the different channels of the A/D converter and looks to determine what level of matching between the channels is required to meet an overall converter specification. The effect of this mismatch will be looked at for both single-tone and multi-tone inputs, and how increasing the number of time-interleaving channels affects these requirements.

2.3.1 Offset mismatch

If the offset between all channels is equal then the overall system will have the same offset as the sub-channels. As previously discussed, in multi-tone communication, offset has very little effect on the converter performance for the application. However inevitably there will be some mismatch between the offset of each channel. This error moves from a DC error to higher frequency spurious tone.

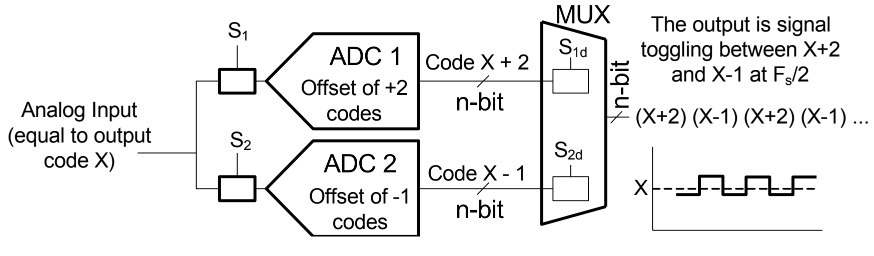

S1
ADC 1
Offset of +2 codes
Code X + 2
n-bit
MUX
S1d
n-bit
The output is signal toggling between X+2 and X-1 at Fs/2
(X+2) (X-1) (X+2) (X-1) ...
Analog Input (equal to output code X)
S2
ADC 2
Offset of -1 codes
Code X - 1
n-bit
S2d
X

Figure 2.14 – Example of the effect of offset on time-interleaving ADCs

Figure 2.14 shows how a DC signal X quantised by a time-interleaving ADCs of 2 channels results in tones at the output related to the sampling frequency. It can be shown that the output samples of a time-interleaved ADC of two channels with channel offset mismatch equals [10]:

$$out[n] = \sin(\omega nT) + V_{OS} + \frac{\Delta V_{OS}}{2}\sin\left(\frac{\omega_s}{2}nT\right) \tag{2.3}$$

Where $out[n]$ is the nth output sample, for an input signal of $\sin(\omega nT)$ where T is the signal period, V_{OS} is the average offset between the channels, ΔV_{OS} is the difference between the channel offsets, and ω_s is the sampling frequency. Clearly the input signal appears at the output, with a DC element of power V_{OS}, and a tone at $\omega_s/2$ with a power of $\Delta V_{OS}/2$.

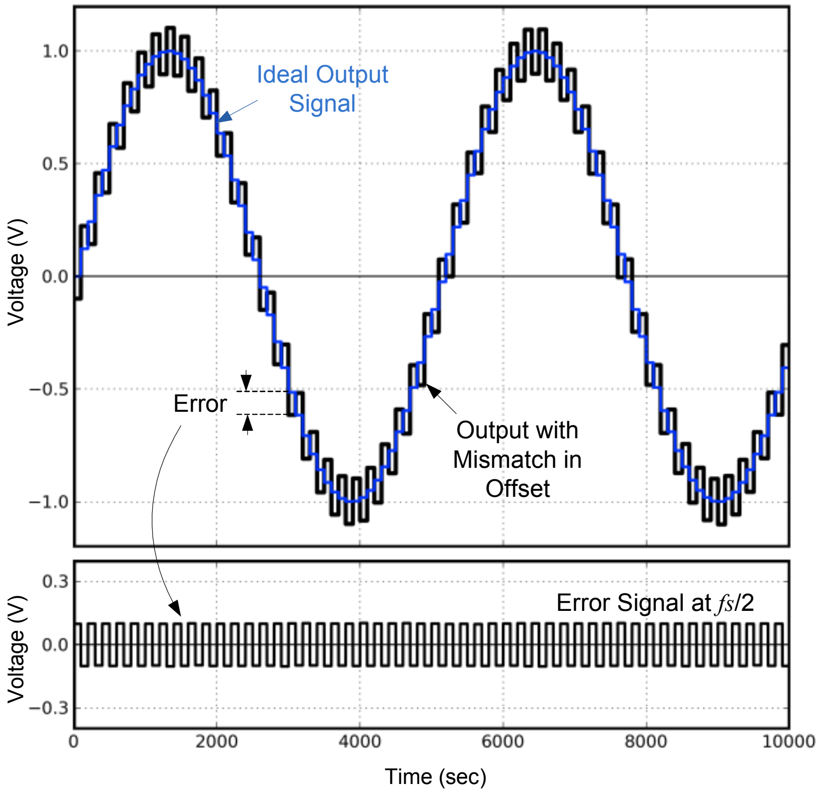


Figure 2.15 – Result of offset for sine input for a 2-channel TI ADC

Figure 2.15 shows the output signal of a 2-channel time-interleaved ADC when an input sine-wave is applied. Again it can be seen that the error signal at Fs/2, but importantly the magnitude of the error is signal independent. As the number of time-interleaving channels are increased, further tones appear at binary divisions of the sampling frequency. Figure 2.16 shows the output spectrum of a time-interleaved ADC of 2, 4, 8 and 128 channels, with a single tone inputs, with offset between channels. Here the sigma offset between channels is set to the same value for the different number of time-interleaving options. For the 2-channel ADC, apart from the signal tone, a DC tone at the 0th bin, and a tone at Fs/2 can be seen.

It can be shown that the effect of offset error remains independent from other effects such as mismatch of gain, sampling timing, and input bandwidth [11]. And since the error is independent of the input signal magnitude, the channel offset mismatch can be easily calculated, and corrected using calibration.

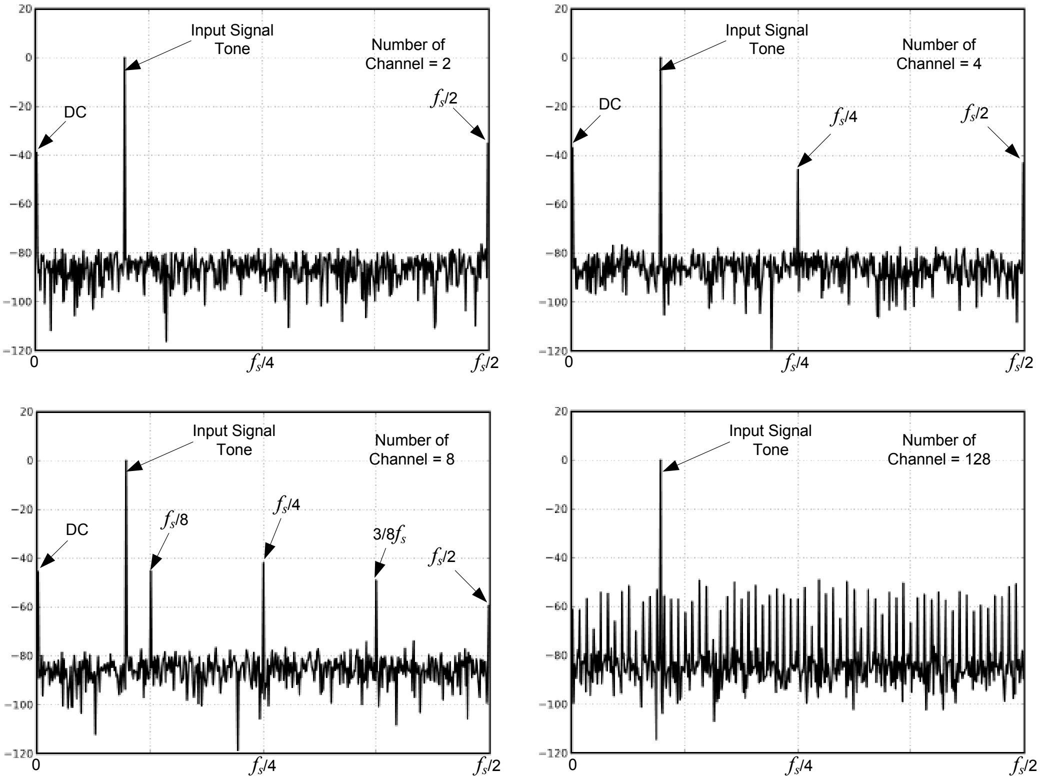


Figure 2.16 – Example of output spectrum for TI ADC with offset mismatch

Figure 2.17 shows how offset mismatch between channels limits the possible ENOB of the converter. Here the effective noise power contribution of the harmonics due to the offset mismatch is plotted against the main signal mismatch to calculate the effective SNDR. It can be seen for a doubling in sigma offset mismatch between channels, the overall ADC resolution degrades by 1-bit.

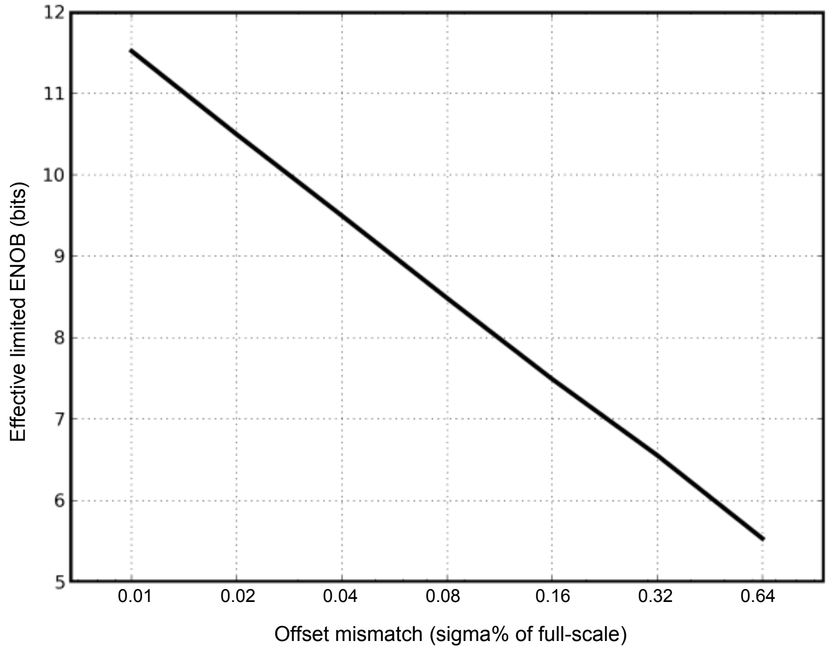


Figure 2.17 – Offset mismatch (in percentage) vs. ENOB

2.3.2 Gain mismatch

The gain between different channels of the time-interleaving ADC may differ slightly. This gain mismatch can result in a form of amplitude modulation similar to an AM signal. For a 2-channel time-interleaving ADC, in can be shown that the output samples of the ADC with gain mismatch between the channels can be simplified to [10]:

$$out[n] = G\sin(\omega nT) + \frac{\Delta G}{2}\sin\left(\left(\omega - \frac{\omega_s}{2}\right)nT\right) \tag{2.4}$$

Where $out[n]$ is the nth output sample, for an input signal of $\sin(\omega nT)$ where T is the signal period, G is the average gain between the channels, ΔG is the difference between the channel gains, and ω_s is the sampling frequency. Clearly for a 2-channel ADC, the input appears at the output with the average gain of the two channels, and an extra tone also appears folded back from Nyquist, with an amplitude dependent on the difference in gain between the channels. Most importantly although the frequency of the output tone is dependent on the input signal frequency, the amplitude of the error tone is input signal independent and is solely dependent on the gain-mismatch. Figure 2.18 shows the output signal applied to a 2-channel time-interleaved ADC with no-gain mismatch and in the presence of gain mismatch. The error signal is shown below.

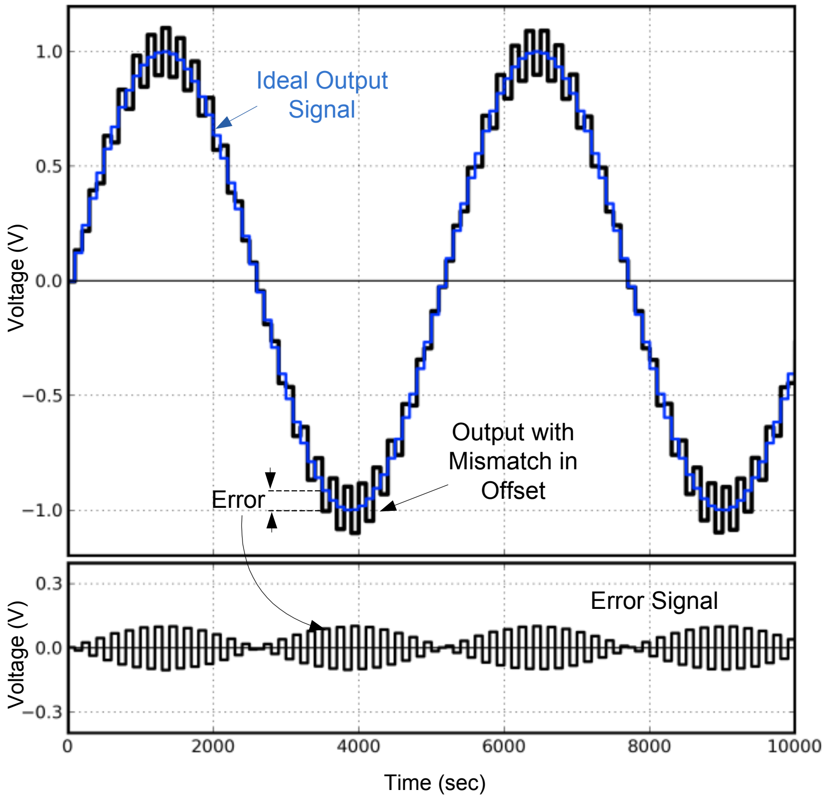


Figure 2.18 – Result of gain mismatch for sine input for a 2-channel TI ADC

Figure 2.19 shows the output spectrum for time-interleaved ADC with 2, 4, 8 and 128 channels. Clearly the frequency of the error tones is input signal dependent, however the amplitude of the signals are dependent on the size of the gain mismatch.

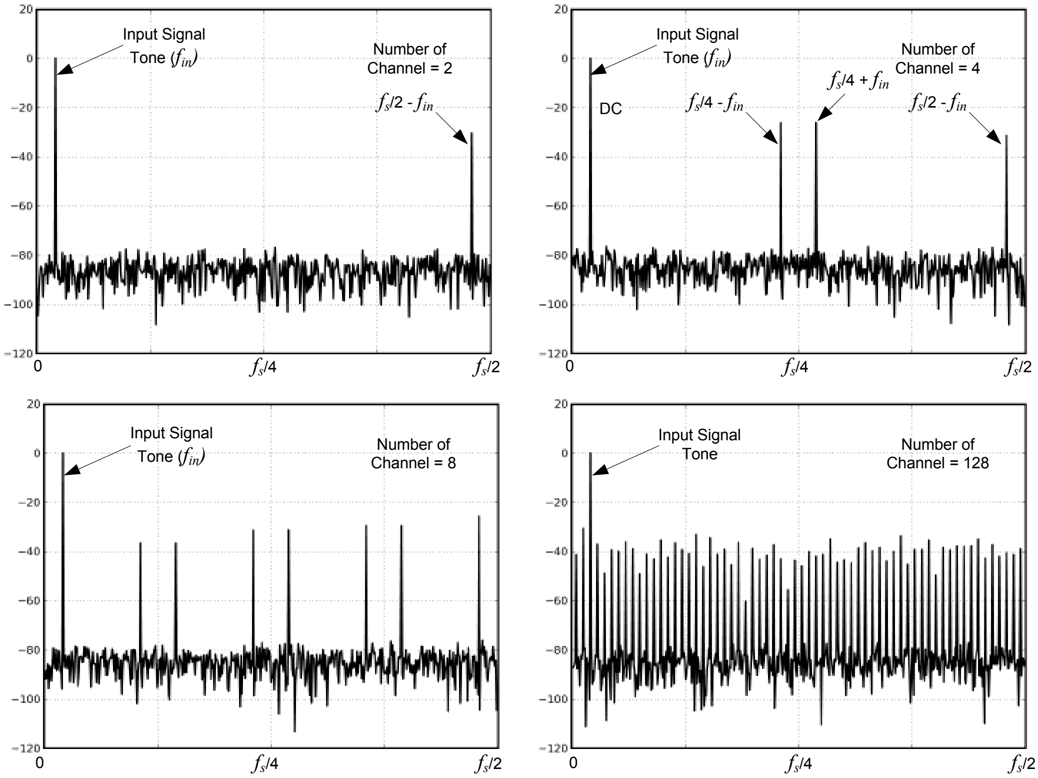


Figure 2.19 – Result of gain mismatch for sine input for a 2-channel TI ADC

Figure 2.20 shows how the absolute value of the gain-mismatch between different channels limits the possible effective ENOB. Here the power of the harmonics due to the gain mismatch is combined and compared to the signal power to calculate the effective SNDR. It is also worth noting that the noise power due to gain-mismatch has no dependence on the signal frequency.

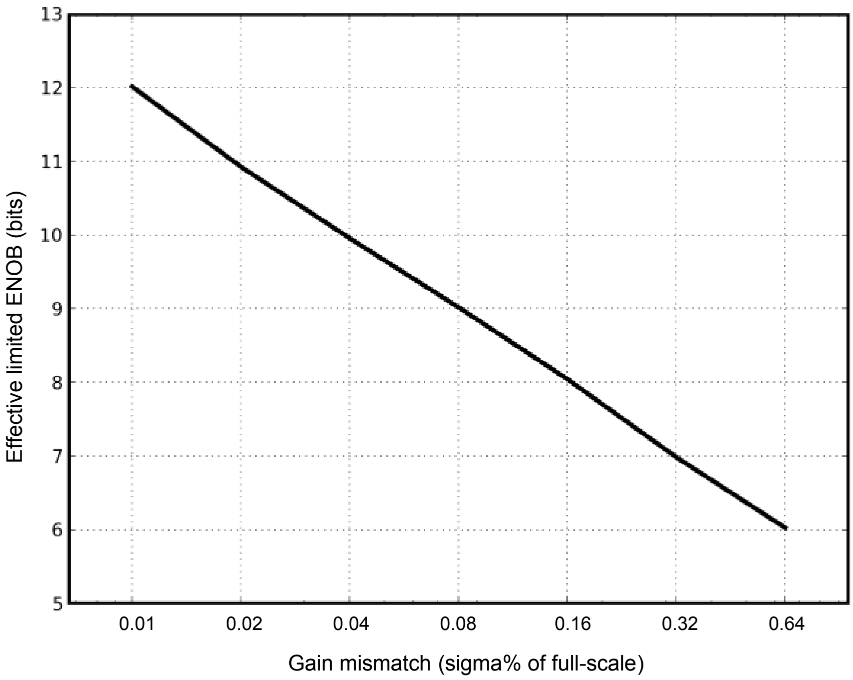


Figure 2.20 – Channel gain-mismatch (percentage) vs. ENOB

2.3.3 Input Signal Bandwidth Mismatch

As discussed previously, two options are available for the T/H or S/H front end in time-interleaved systems, either one global T/H can be used, operating at the higher level ADC sampling rate (high-frequency), which provides samples to all the sub-channels, or a T/H can be embedded in each channel. The first approach eliminates input signal routing mismatch errors, and timing skew errors, however can result in a major increase in power consumption, and can commonly degrade the linearity of the system. The second approach is becoming more common [9][18], however input signal bandwidth mismatch and timing skew require managing to meet the overall performance of the system.

Figure 2.21 shows a time-interleaved ADC with X number of channels. The T/H circuitry in each sub-ADC has been shown. The sampled voltage in each ADC passes an effective filter, through all parasitics R and Cs in the front-end routing, and the effective Ron of the switch in combination with the sampling capacitor. Inevitably some mismatch between the switches, the sampling capacitors, and parasitics in the different paths will exist. These differences result in mismatch between the different channels in two ways:

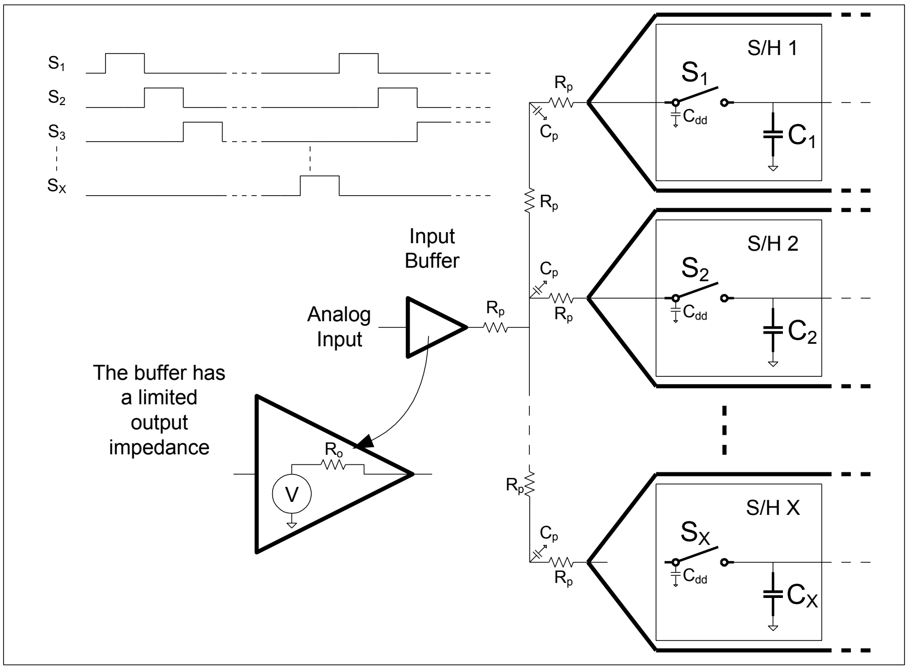


Figure 2.21 – Input signal routing in time interleaved ADC system

Settling Time Error and Differences – When the sampling switches S_1 to S_X are each closed, the input signal begins a exponential settling on the sampling capacitors in each ADC, where the settling time constant is related to the R and Cs in the input path for that particular capacitors. For complete settling systems, the settling error should be below the size of an LSB, so despite the mismatch between channels this error should remain below an LSB, however error due to settling between different channels even below an LSB will accumulate with other sources of mismatch and should not be forgotten.

Phase Response Differences – Since the input is a continuous stream, each RC present in the path of each ADC acts as an RC filter for the input signal. Since this is different for different channels, the effective filter characteristics will be input signal dependent and different frequencies may be subject to different amplitude and phase responses causing mismatch between the signals sampled onto each capacitor.

Firstly on the settling time error, as mentioned most individual channels require meeting the settling requirement independently, unless incomplete settling sampling techniques [19][20] are used, which are not common in time-interleaving systems, hence the mismatch in settling tends not be the limiting factor in performance. For the settling error, which has an exponential characteristic, to meet the requirements of an N-bit sampling system the following equation must hold:

$$1-e^{-t/RC}<1-\frac{1}{2^N} \tag{2.5}$$

where t is the available settling time and R and C are the series resistance (dominated by the Ron of the switch) and the sampling capacitance. Using the above equation, it is simple to build a table of time-constant requirements for the RC for a given resolution:

Res. (Bits)	**Time Constants (t/RC)**
6	3.6
7	3.9
8	4.2
9	4.4
10	4.6
11	4.8
12	5

As mentioned above, meeting these time-constant requirements is required for non-time-interleaving system as well. However in time-interleaving system, the difference in phase response due to mismatch in R and C in different channels is unique. This problem is one of the most complex to understand, and also the hardest to satisfy the requirements for. The complexity of the error is due to its non-linear dependence on input signal frequency, and amount of bandwidth mismatch between channels.

For simplicity, looking at a time-interleaving system of two channels 1 and 2, for input signal of sin(ωt), channel 1 has bandwidth (BW) related to R_1 (the on resistance of the sampling switch) and C_1 (the sampling capacitor), and channel 2 has a BW related to R_2 and C_2:

$$input = \sin(\omega t)$$

$$\omega_1 = \frac{1}{BW_1} = \frac{1}{2\pi R_1 C_1}$$
$$\omega_1 = \frac{1}{BW_1} = \frac{1}{2\pi R_1 C_1}$$

For a given input signal bandwidth of ω, the phase change in a single-pole system of bandwidth ω_{BW} for the input signal is:

$$\tan^{-1}\left(\frac{\omega}{\omega_{BW}}\right)$$

So for an ideal discrete time input signal of sin(ωnT) where T is the signal period, and n is the sample number, the output can be written as:

$$out[n] = \sin\left(\omega nT + \tan^{-1}\left(\frac{\omega}{\omega_1}\right)\right) \qquad n = odd$$
$$out[n] = \sin\left(\omega nT + \tan^{-1}\left(\frac{\omega}{\omega_2}\right)\right) \qquad n = even$$

Compounding these expressions into one equation can be difficult, alternatively a numerical model can be used from this data to estimate the effect of bandwidth mismatch between two channels. Figure 2.22 shows a sine-wave sampled with a time-interleaved ADC of with two channels, with bandwidth mismatch on top of the ideal output signal. The effective noise signal has been shown below.

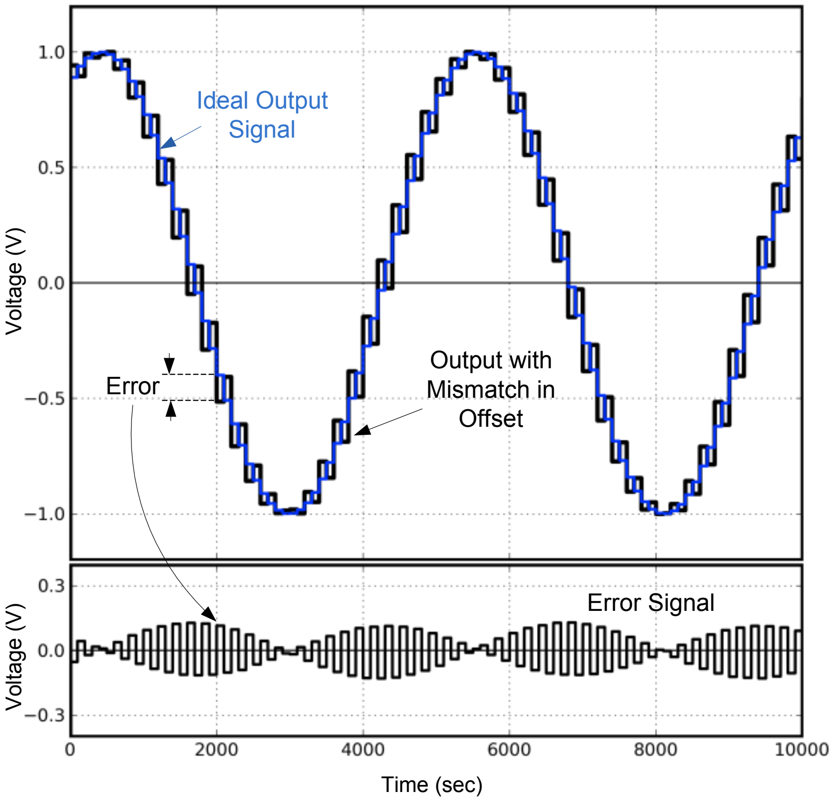


Figure 2.22 – Result of bandwidth mismatch for sine input for a 2-channel TI ADC

Bandwidth mismatch produces harmonics at frequencies similar to gain mismatch, depending on the number of channels at binary divisions of the sampling rate, plus and minus the signal frequency. However with bandwidth mismatch the amplitude of the tones as well as being dependent on bandwidth mismatch, is also dependent on signal frequency. To calculate the required bandwidth matching between channels Figure 2.23 was generated using numerical modelling, which shows the ratio of the signal tone to nominal bandwidth, sigma of bandwidth mismatch, and its effect on the overall possible resolution of the system.

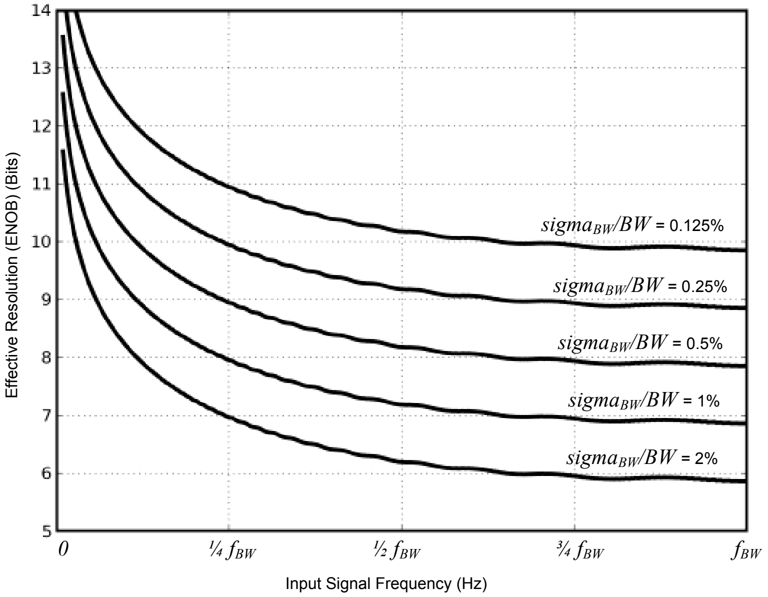


Figure 2.23 – Noise due to BW mismatch at different input frequencies

For example, to achieve a 7-bit noise floor limit due to bandwidth mismatch, one could design the front-end nominal bandwidth (f_{BW}) to be 4 times that of the highest input signal tone, while achieving 2% matching in overall RC and hence BW between the channels. Alternatively, if 1% matching is achievable, one can place the front-end nominal bandwidth at only twice that of the input signal tone and still achieve 7-bit performance.

From Figure 2.23 it can be seen, that the magnitude of mismatch maintains a linear relation with ENOB, as halving of the mismatch adds an extra bit to the effective SNR. However the relationship between signal bandwidth vs. front-end nominal bandwidth is not linear.

It is important to understand that the error due to mismatch in bandwidth has a strong relation to input signal frequency. The analysis and modelling completed above is true for a single-tone input signal, however for a multi-tone input signal the requirement on channel bandwidth matching and performance (measured in MTPR) is more complex. In first order, since the effect of an RC filter is the degrading factor, linear system theory can be used, and with superposition one could estimate the effect of bandwidth mismatch for different tones of the signal independently, and then combine the overall harmonic noise contribution. For earlier carriers in the band, the ratio of front-end bandwidth to signal frequency is high, and hence their noise contribution is low, however for later carriers in the band, this ratio is reduced, and their effective harmonic noise contribution is increased. The effective achievable ENOB for a given nominal front-end bandwidth and channel mismatch is very dependent on the characteristics of the multi-tone signal, and the best way to understand this is by using a front-end model similar to one produced to generate the graphs above, but alternatively used in conjunction with the multi-tone signal. The key point to take here is that the bandwidth mismatch requirements for time-interleaved ADCs used for multi-tone signals is more relaxed compared to the analysis done for single tone signal, and since the error is input frequency related, the relaxation on the bandwidth matching tends to be proportional to the multi-tone signal frequency width.

2.3.4 Sampling Clock Mismatch

Figure 2.24 shows a 2-channel time-interleaving ADC with distributed T/H in each channel. Sampling clock mismatch is systematic clock skew between the sampling clocks of different channels. This error produces a relative sampling point error between the channels which results in error in the main signal phase and error harmonics in the output. The mismatch or timing skew is inevitable since there will always be some unique devices (block generation circuits, clock trees, buffers, switch circuits) which will be different for the different channels, and these devices will have mismatch, and hence result in different propagation delays for the appropriate clocks resulting in systematic clock skew for the different sampling clocks.

Figure 2.25 shows the output signal from a 2-channel time-interleaving ADC subject to timing skew between the channels. Here the error signal similar to bandwidth mismatch is at sampling frequency minus the input signal frequency, and the amplitude of the error signal is proportional to the timing mismatch as well as the input signal frequency.

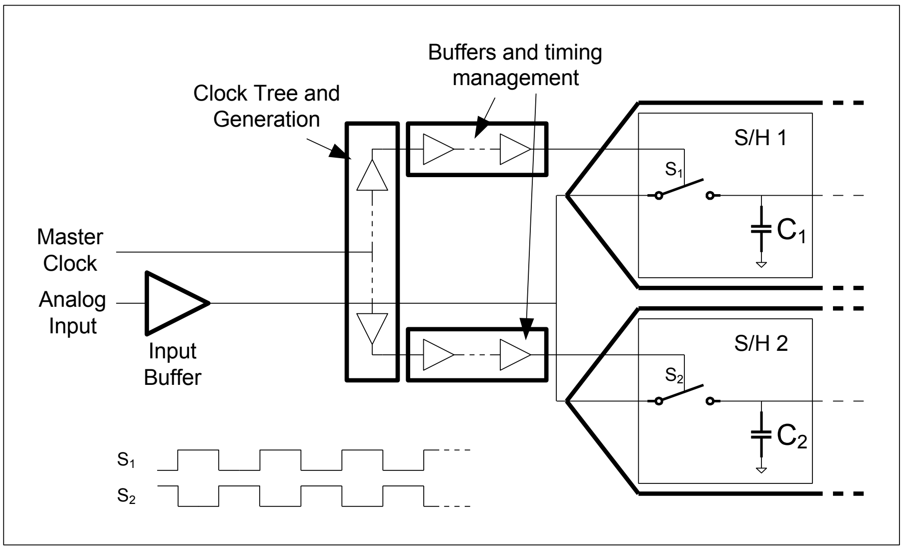


Figure 2.24 – 2-channel time-interleaving ADC with circuit element in path of clock

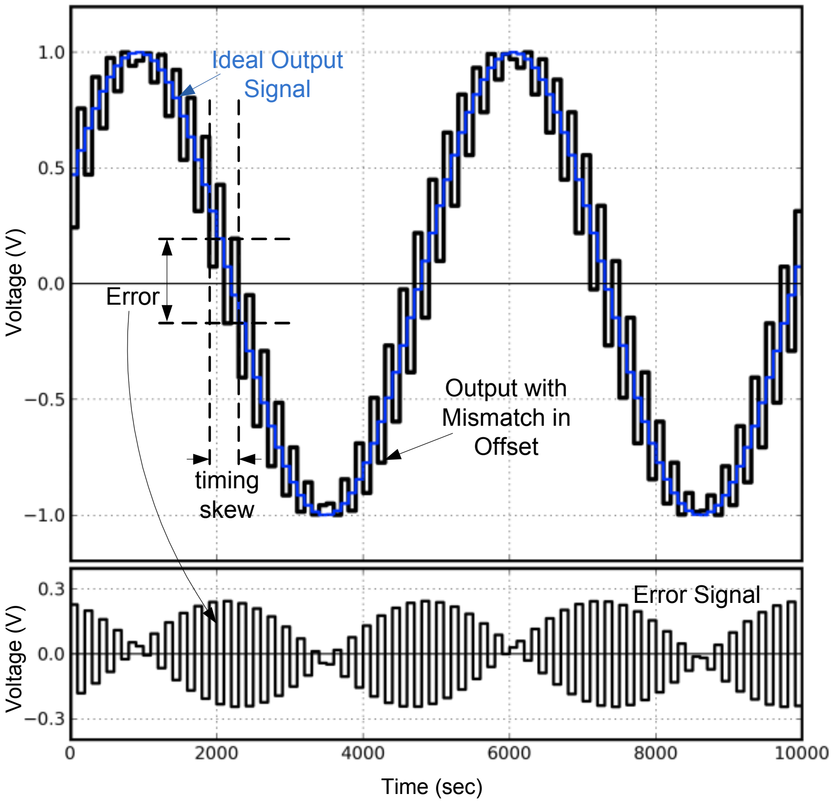


Figure 2.25 – Output of 2-channel time-interleaving ADC subject to timing skew

It can be shown [10] that the output main tone remains at the correct frequency however it is subject to phase change proportional to the timing skew and input signal bandwidth, and a spurious tone folded back from the Nyquist frequency will appear with an amplitude proportional to the input frequency and timing skew. Importantly, the amplitude of the error, unlike bandwidth mismatch, has a linear relationship to the input signal bandwidth. For greater than 2 channels, similar to gain-mismatch, the tone will appear as input frequency dependent folds around binary divisions of the sampling frequency similar to Figure 2.19.

Each channel will have an independent timing skew. Imagine for a frequency domain communication system, if the number of channels is increased proportional to the length of the samples used in a symbol, the timing skews, though systematic, have a relatively low cycle time, and will tend closer to random timing skew rather than systematic (cyclic) timing-skew for the length of the symbol. Central Limit Theorem can be used to predict an effective noise like Gaussian skew to appear. In fact as the number of channels is increased the effect of systematic timing skew becomes the same as random cycle-to-cycle timing jitter in normal A/D converters. For example if the number of channels in the ADC was equal to the number of samples in a communication symbol, since each ADC only contributes once to the output stream, systematic timing skew between the channels becomes the same as cycle-to-cycle timing jitter. Hence for larger time-interleaving factors, the effect of timing skew, similar to jitter, can be estimated as [9]:

$$SNR_{skew} = 20.\log\left(\frac{1}{\sigma_{skew}.2\pi.f_{in}}\right) \qquad (2.6)$$

Where σ_{skew} is the standard-deviation of timing skew between channels, f_{in} is the input frequency, and SNR_{skew} is the effective Signal-to-noise ratio due to timing skew. This is interesting, since it shows taking Design for Manufacturability (DFM) into account, increasing the number of time-interleaving channels relaxes the standard deviation requirement on each individual channel skew. For example for a 2-channel ADC, where a certain sigma is required to meet the resolution accuracy, the designer may choose to meet timing-matching requirements up to 3-sigma. Most manufactured parts will be well within the SNR requirement for the system, but 1 out of every 1000 parts will have a timing mismatch greater the specification and would fail test. However for a larger time-interleaving system, say 128 channels, although individual channels could have timing skews greater than sigma, since they are operating in conjunction with other channels, as long as the standard-deviation of all channels is within requirements all parts would pass test. Figure 2.26 shows this effect visually showing the effective resolution of 100 different 2-channel and 128-channel ADCs manufactured, where the design of the different time-interleaving amounts have equal sigma timing-skew.

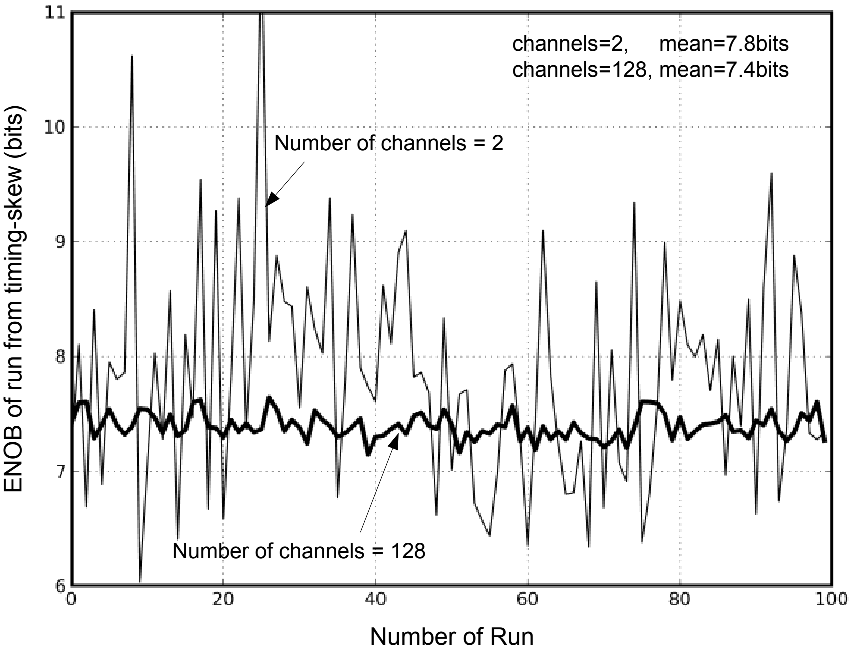


Figure 2.26 – ENOB of 100 parts with matching sigma for 2 and 128-channels

Clearly for certain samples the two-channel ADC achieves much greater ENOB compared to the 128-channel converter, in fact the effective mean ENOB for 100-different manufactured parts is better for the 2-channel devices compared to the 128-channel devices. However if the target specification for this part was

7-bit, a few parts of the 2-channel ADC fall below this specification and need to be binned at test, while the 128-channel device maintains a more consistent part-to-part ENOB value.

In this example, for all 100-parts of the 2-channel converter to meet the 7-bit requirement, the 2-channel device requires a much smaller timing-skew sigma between the two channels compared to that of the 128-channel ADC. In practice the achievable timing skew in a system apart from being related to technology, is related to the number of unique devices to a given channel the clock signal requires to travel through before the sampling switch. As the number of channels in a time-interleaving system is increased, the size of the clock tree, and complexity of clock generation circuits tends to increase, hence increasing the number of unique-to-channel devices the clock requires to travel to, hence in practice, despite the DFM point made above, increasing the number of channels can result in great challenges in meeting the timing-skew requirement.

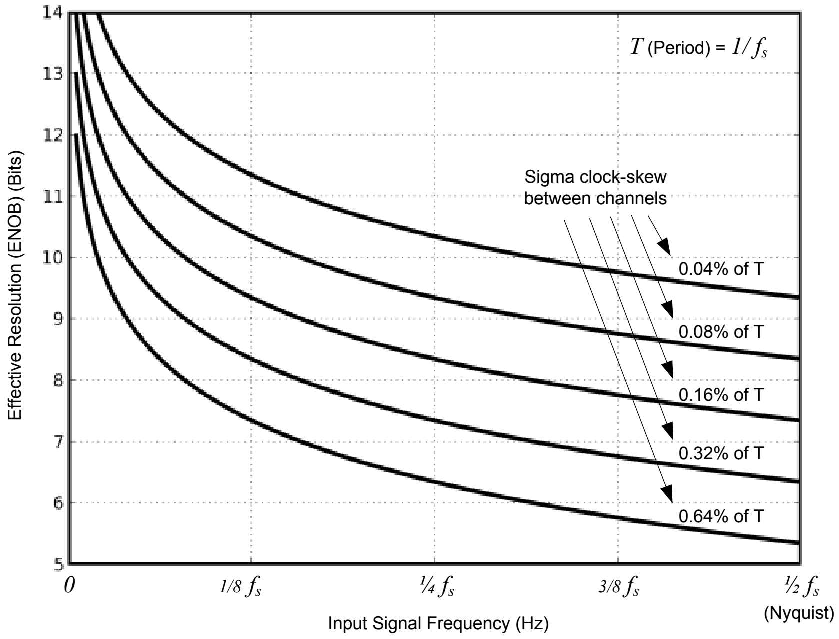


Figure 2.27 – ENOB vs. timing skew sigma for different input frequency signals

Figure 2.27 shows how the timing skew sigma between channels of a time-interleaved ADC with large number of channels affects the effective SNR for different input frequency signals. It can be seen that with halving of timing skew sigma, the effective resolution for a given input frequency increases by a bit. More importantly, in the case of timing skew the input signal frequency maintains a linear relationship with noise contribution, showing with halving of signal frequency, the effective SNR of the system due to timing skew increases by 1 bit.

For multi-tone signals, the requirement on timing skew is relaxed compared to a sine-wave at Nyquist frequency, however the reason for this is very different compared to the bandwidth mismatch problem. In the bandwidth mismatch problem, since the signal was made of different frequency components, the effective SNR degradation for different frequency components could be analysed separately and then combined. However for clock timing skew, although the error signal has a strong dependence on input signal frequency, the reason for its dependence is due to the rate of change of the signal or slew rate. For a multi-tone signal, made of summation of sinusoidal components, using the Central Limit Theorem it can be shown that the effective slew rate of the signal itself is Rayleigh Distributed [2]. Depending on the number of carriers, carrier spacing and peak to average ratio, the nominal mean of slew-rate and sigma of slew rate for a given multi-tone signal can be identified. For high number of carriers the mean slew rate can be quite low, meaning the signal is highly resilient to clock-skew between different channels of the ADC. Modelling can be used to estimate the requirement on timing-skew for different multi-tone signal characteristics, but in general this relaxes the requirement heavily.

2.4 Calibration and Error Correction

Quantifying difference aspects of an ADC's non-idealities, being offset, gain-error or non-linearity, enables the designer to improve the design until the specifications are met. CMOS process is subject to process corners and mismatch, both systematic due to poor design (layout) and statistical. Further to this in many applications the temperature, voltage supply and references may be variable.

To meet certain ADC requirements, a large amount of power or area may be required, or at time meeting all the specifications with the non-idealities of the process may not be possible at all. To enable the ADC to meet the required specifications post fabrication, calibration of the block is a popular option [21]. Calibration can be applied at many different times, from the test house, at startup, or even during operation. Separate to the timing, calibration can be applied in many different ways, by adjusting (changing) internal elements in the circuit, or by post processing of the outputs. This depends on the type of non-ideality one aims to correct. In this section, the different strategies commonly used for calibration are looked at. A calibration technique can be very dependent on the architecture and the type of error one requires to correct for. Here the general principles and techniques common for calibration will be discussed, in the following section the non-idealities of common ADC architectures will be looked at and methods developed for each will be presented.

2.4.1 Methods to apply the calibration, Digital or Analog

The key to a good calibration strategy is to go beyond a simple correction of the ADCs transfer function to meet the specification, but to understand and appreciate the reason behind the error and to correct the problem at the source. As the correction, or calibration is applied closer to the source of the error the correction will be more effective, commonly more stable and limits the number of artefacts. As the calibration is done at a higher level, the artefacts of the error may be more complex, where correcting for becomes harder and potentially less sustainable over time. For example an offset in an amplifier as part of complex ADC block may result in offsets, gain-errors and major non-linearities at the ADC transfer function, therefore it is best to correct at the amplifier rather than later in the signal chain.

Calibration techniques are often referred to as Analog or Digital. In reality even analog calibration techniques can be partially digital or even mostly digital in nature. Commonly any calibration technique that applies adjustments, tuning, trimming inside the ADC block to a circuit or top-level element is referred to as Analog Calibration or Trimming. If non-idealities within the converter are corrected outside the block, by a digital sub-system or CPU sub-system, where no parameters within the block are adjusted, this is referred to as Digital Calibration. Digital calibration is very common in SoC applications [21].

Analog calibration commonly tackles the problem at the source, however this may not be possible in all situations, or may introduce noise or a new non-linearity. Also in deep sub-micron technology the cost of digital gates is always reducing, and post processing of the digital data my be easier and smaller. Choosing the right calibration for each non-ideality is key to building an efficient solution. For example, in most converters gain error can be easily corrected in the analog domain by adjusting (trimming) the top and bottom reference voltages, while fixing gain error in the digital domain requires a complex high-resolution multiplier block. On the other hand in some converters offset error may be due to a mismatch input pair. Trimming of an input pair can be difficult and can have implications for noise and linearity, while in the digital domain offset can be corrected at low cost with an adder block.

2.4.2 Regularity: Trim, Foreground or Background

Calibration can be performed at three different times, at test, at power up or at dead-time in the application where calibrating is the main task the converter is undertaking, or during normal operation in the background. These three approaches will be explained below:

Factory Calibration (Trimming)

Calibrating the ADC at the test-house after fabrication is commonly referred to as trimming. Trimming can be done for non-idealities which are due to process corner, or mismatch of elements which do not have temperature coefficients. Basically factory trimming is used for errors which will not change, and are due to fabrication. Trimming can be both analog or digital. Analog trimming is done by changing reference levels, device sizes, or other tuneable elements inside the block. Analog trimming can be controlled by the digital subsystem, and commonly is, and is referred to as Analog trimming since it is done inside the block. Digital trimming is done by learning of correction coefficients or generation of look-up-tables in the post converter digital domain at the time of test, which can be later used at normal operation for correction.

Foreground Calibration

Foreground calibration is all forms of calibration in the field which are performed when the ADC is not in normal operation converting the analog input for the purposes of the application. Basically in Foreground calibration, the act of calibration is a foreground task. This is commonly at start-up time, or in communication applications, in the dead-time between reception intervals or when the node is transmitting and there is no need for the ADC to function.

Background Calibration

In background calibration, the calibration of the block is performed at the same time as the block is operational. This may be done in two ways: With statistical methods, or with redundancy. In statistical background calibration, knowing the statistical profile of the input signal a certain profile is expected of the output codes. Digital or Analog calibration parameters can be adjusted until the expected statistical profile is seen in the output stream. In redundancy, parts of the converter are replicated in design, and calibration on the redundant parts are done in the background. Once the calibration is complete, the redundant parts are swapped with the part in normal operation, and the swapped out parts can then be subject to calibration.

2.4.3 Quantifying the cost of calibration

It is important to quantify the cost of calibration, regardless of the method, in comparable metrics such as power, or area [21]. This is important to understand, to confirm the choice to not fix the error by design due to implications in size or power, and choosing to perform calibration was a beneficial decision. Factory trimming increases the test time, and hence increases the cost per device. This extract cost can be equated to silicon area cost. Factory trimming can be quantified as equivalent die size. This can then be compared to the cost of fixing the block by design.

For foreground calibration, the converter needs to be on and operational for longer periods of time than it would have normally needed to be, to allow foreground calibration to take place. The regularity of the calibration compared to the normal operation (in terms of length of time) should be identified, and the power consumption of the system during calibration is effectively an increase in total power consumption. For example if for a 100µs ADC operation, a 5µs calibration period is required, the effective power consumption of the block should be increased by 5%. The circuitry required to identify the magnitude of the non-ideality, and the circuit used to correct the non-ideality, increase the overall area of the block. In background calibration, the area of the block can be increased due to redundancy. This area should be compared to the size of better design which does not require calibration. For all calibration methods, the actual size and power consumption of the digital sub-blocks required should be calculated. In many instances, improving the ADC by design, despite the power consumption and size implication can be more efficient compared to calibration. The important fact to take from this section is that calibration is not free and its cost should be correctly calculated.

2.5 TI, Calibration & Re-configurability by Example

2.5.1 Introduction

The methods of quantifying non-idealities in converters, and their effect on time-interleaving systems have been explained. Calibration as a method to improve the specification of the converter has also been explained, and the different methods of performing which have been categorised. The best way to understand time interleaving, and in particular calibration, is by studying examples applied to different ADC architectures, since calibration can be very architecturally specific. Re-configurability was discussed in Chapter 1. Realising re-configurability in a converter is even more architecture dependent. Again re-configurability will be explained by examples due to the limited number of publications available in literature. In this section some of classical ADC architectures are used as short case studies to demonstrate and explain, calibration, time-interleaving and re-configurability where appropriate.

2.5.2 Flash converters and calibration

Figure 2.28 shows a simplified block diagram of a flash ADC. The comparator bank compares the input with all possible digital threshold levels simultaneously, the thermometer output generated from these comparators are converted into a binary code using digital logic. Like any mixed signal block there are many subtleties and details in implementing which should be considered to meet the specification. For flash ADCs there are 2 dynamic and 1 static requirements which are commonly the bottle neck for performance [22][23].

The first dynamic requirement is on the resistor ladder. The movement on the input, and firing of the comparators, couple back to the resistor ladder, taking current from the ladder and hence moving the references. This problem can be overcome by designing the current density of the ladder correctly.

The second dynamic problem is incorrect firing of comparators due to coupling noise, hysteresis and limited bandwidth of the comparator. Basically as the input signal moves, the comparator misses the signal or is subject to noise from their neighbouring comparators firing and produces an error. This can be improved with better comparator design, and implementation of a bubble correction block prior to the thermometer to binary converter.

The static error, which can be the biggest problem is the comparator offset. If the offset in any comparator is greater than half an LSB this causes a permanent error in the conversion and DNL error on the output. To improve this, for a given technology the size of the input pair may require increasing, but this would in term increase the input capacitance which may not be acceptable. To overcome the comparator offset problem in flash ADCs, calibration is commonly used [23].

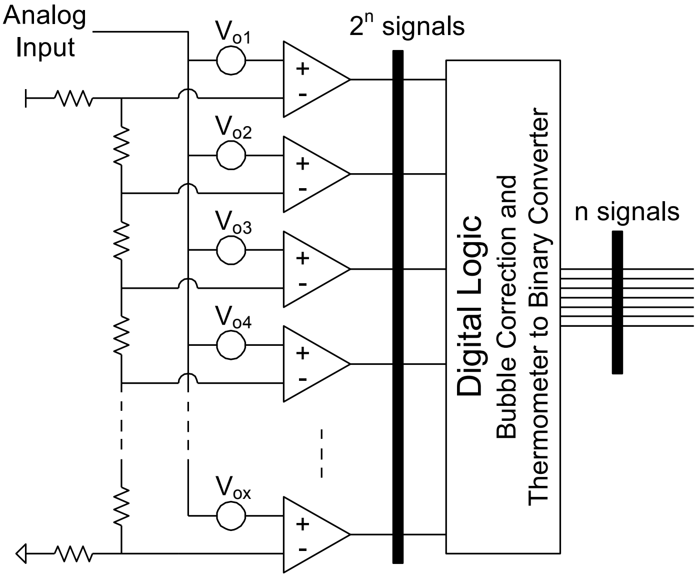


Figure 2.28 – Flash ADC, with comparator offset

Despite the output of the comparators being a digital signal, digital calibration is rarely used for correction of the comparator offsets, since due to the offset the information about the input is lost, since the comparators are digital devices, and it is difficult for a digital backend to recover the data. Analog calibration methods that involves correcting the offset of the comparator inside the comparator block is commonly used.

The offset error an input pair can be correct in many ways. Three methods are shown in Figure 2.29. The most obvious is to use a programmable device in parallel with the input device. In practice this can be very difficult solution to implement, can result in a difficult layout, very large size, and issues with noise and linearity in gain [24]. One of the other techniques is to effectively change the comparator crossing point by adding an artificial current to one side of the comparator, effectively giving the side with a smaller Vt a head-start. This method is one of the easiest to implement, however the adjustment currents effectively reduce the gain of the comparator, so using too much adjustment current can have other implications in performance [23]. The third method here is fixing of the input pair Vt mismatch by adjusting their bulk voltage. In practice building small, accurate and monotonic programmable voltage sources can be difficult, and for this technique commonly a programmable current source is used with a resistor to the common-node [25].

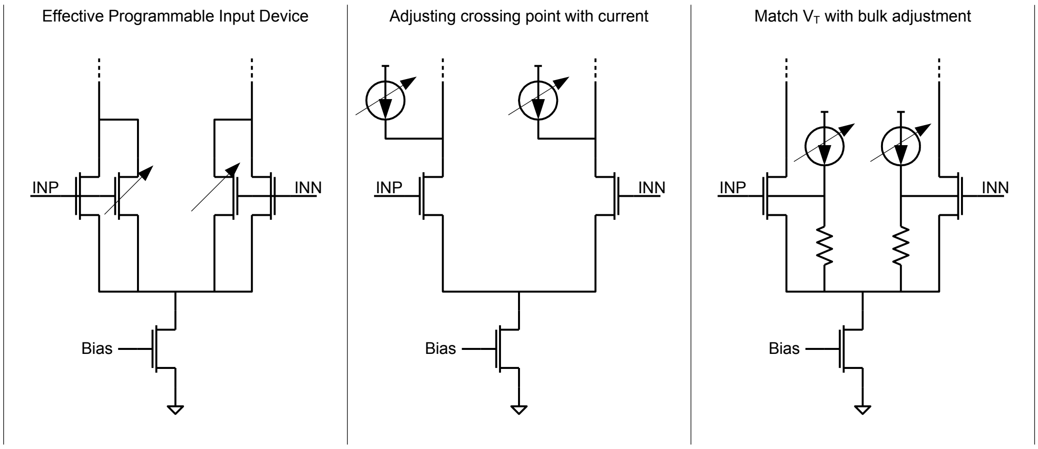

Effective Programmable Input Device
Adjusting crossing point with current
Match V_T with bulk adjustment
INP
INN
INP
INN
INP
INN
Bias
Bias
Bias

Figure 2.29 – Calibration methods for input pair

As you can see all methods introduce involve a small programmable element. The value of this programmable element needs to be set during calibration. The act of choosing the correct calibration value is referred to as the ‘calibration loop’. The calibration loop itself can be digital or analog in nature. The calibration loop is executed by the system at test time, startup, or in the background depending on the calibration type the system architecture chooses to apply, but the loop is always constant.

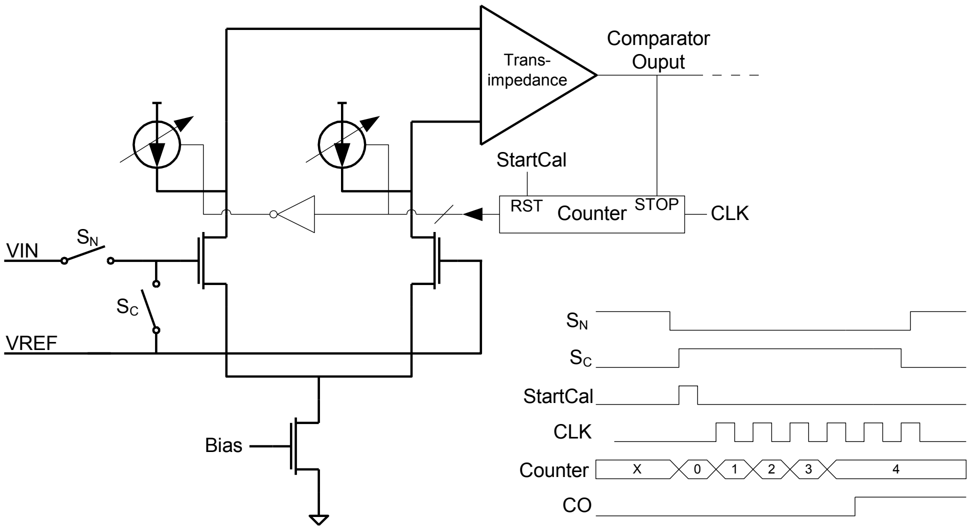


Figure 2.30 – Digital counter based calibration loop for input pair offset

Figure 2.30 shows an example implementation of a calibration loop [23]. In this system, the input device is disconnected from the input signal, and both inputs are connected the comparator reference voltage from the resistor ladder. This should be the normal firing voltage for the comparator in normal operation. The programmable current sources are controlled by the output of a counter. The counter is reset at the start of calibration, and the counter then starts counting up. As this counter counts up, the effective offset value of the comparator changes, until both inputs are equal and the comparator fires. This stops the counter, and the calibration value for the current mirrors is stored in the counter. This is a digital based calibration loop. The same system could be realised without digital counters. For example a fixed current mirror into a capacitor can be used to generate a ramp voltage which is used to drive the bias voltage of the programmable current mirrors. When the comparator fires, the value on the capacitor is held for normal operation. Analog basic calibration loops are less common due to issues with hold time, noise and temperature dependency, especially with the low cost of digital circuits, digital calibration is easier to implement and verify.

Once the calibration loop is defined, the regularity of the calibration can be chosen. For comparator offset, factory calibration (trimming) is rarely used, since the mismatch in input pair is related to the Vt mismatch in the input device which is temperature dependent. For that reason either foreground or background calibration should be used. Foreground calibration can be used if the application allows (dead time is available), and the intervals between calibration cycles is short enough that temperature variation does not cause the error to change more than the allowed specification. Background calibration can be used here with the introduction of redundancy [26]. If an extra comparator is added to the comparator bank, one comparator can always be in calibration. Once this is complete, it can be switched in to the main bank, and another comparator can be taken out of normal operation for calibration [26]. In practice performing background calibration with redundancy in a comparator array can be very difficult, since the redundant comparator requires access to all reference voltage levels, and the backend digital encoder requires full re-programmability. In practice, for background calibration, redundancy is done in smaller banks, for example groups of 4 or 8 comparators, where one extra comparator is added for every bank of 4 or 8 comparators, resulting in a more practical layout of the voltages and limits the complexity of the digital backend.

The offset calibration in a flash ADC was used here as an example to show the 3 main parts to building a calibration system. The first part is identifying the source of the error and choosing how it should be corrected, this can be with programmable current sources, or a digital mathematical function applied in a digital block. The second part is defining and building the calibration loop, which is the system that calculates the correct coefficients for the programmability defined in the first part. This can be with a loop inside the block, or signal that helps the digital backend estimate the coefficients. The third part the deciding how often the calibration loop is used, if it's applied only once at test, or at breaks in normal operation, or applied in the background during normal operation.

2.5.3 Pipeline converters, calibration, TI and re-configurability

In the previous section, flash ADCs were used to demonstrate calibration in the analog domain. Here pipeline ADCs will be used to show a fully digital calibration system. Also some time-interleaving and re-configurable work around pipeline ADCs will be shown, however the more detailed look at time-interleaving will be shown with SAR ADCs.

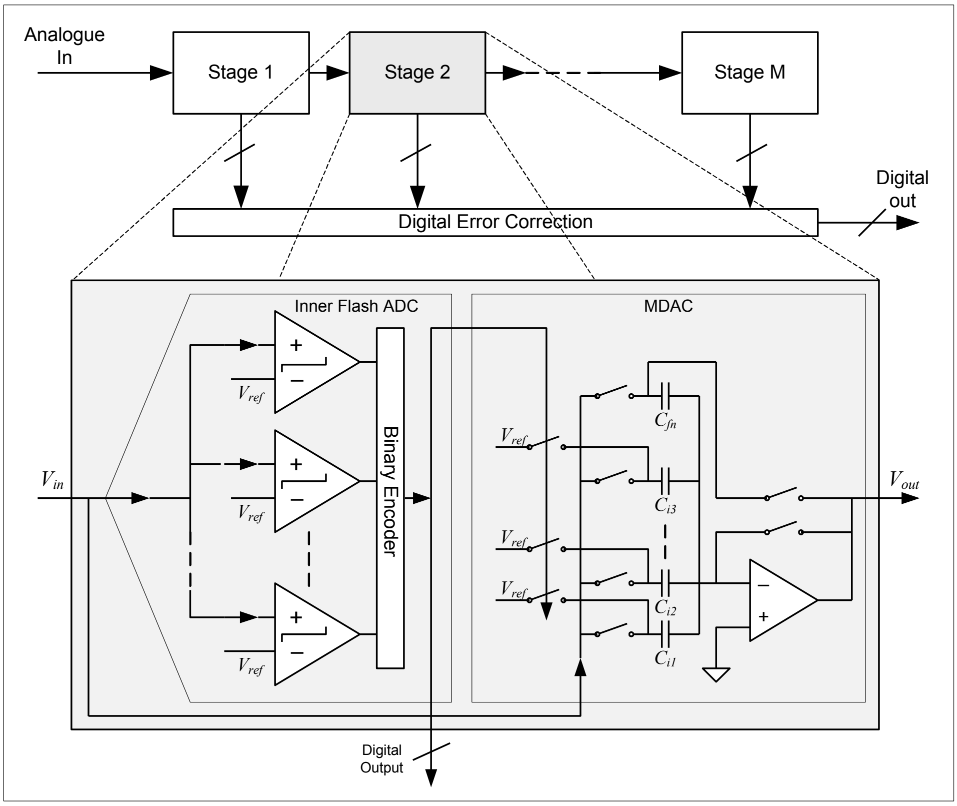


Figure 2.31 – Pipeline ADC basic block diagram

Figure 2.31 shows the basic block diagram of a pipelined ADC. Functionally the stages in the pipeline are identical. Simple 1.5-bits per stage pipelined ADCs obtain 1 bit of data at each stage. The input signal is passed to a stage, a sub-ADC in the stage (commonly a flash ADC) determines if the signal is above or below mid point, following this, the signal is multiplied by 2 by the MDAC and the digital value found by the Inner ADC is subtracted from the MDAC value. This process is repeated until all the bits are resolved. To gain some immunity to inner ADC offset error, small redundancy (overlap) between the stages is arranged, where each stage effectively calculates 1.5-bits worth of information, with 0.5-bit of redundancy used by the main digital error correction block to remove the effect of comparator offset [27].

The linearity performance of a conventional Pipelined ADC is commonly limited by the performance of the MDAC, and its ability to accurately apply the x2 function. This is determined by the open-loop gain of the amplifier, the gain-bandwidth product of the amplifier (which results in the effective settling accuracy) and the matching of the capacitors in the MDAC. The noise performance of the block is commonly limited by kT/C noise of the capacitors.

One can try to meet the requirements of the ADC by design, or alternatively meet the requirement via calibration. It must be noted that extending the resolution beyond 12-bits without any form of calibration in pipelined ADCs can be very difficult primarily due to capacitor matching characteristics in CMOS process. To extend the resolution beyond 12-bit, some form of capacitor calibration, commonly a factory calibration is required [28]. The other non-idealities which result in less that perfect gain of the MDAC can be calibrated either in foreground or background.

Background calibration of Pipelined ADCs is commonly done with use of a low-speed parallel A/D converter present in the system [29][30] as shown in Figure 2.32. The basic idea behind the calibration system is simple. Once the input signal is sampled, the samples are forward to the main ADC for normal digitisation, but in parallel is forwarded to a low-speed more accurate ADC as well. The low-speed ADC will do a more accurate job of digitising the input signal, but will have a lower sampling rate and hence will digitise much less samples compared to the main ADC. But from the few samples it digitises, the errors in the main pipelined ADC can be estimated, since for those samples the two blocks should produce identical outputs.

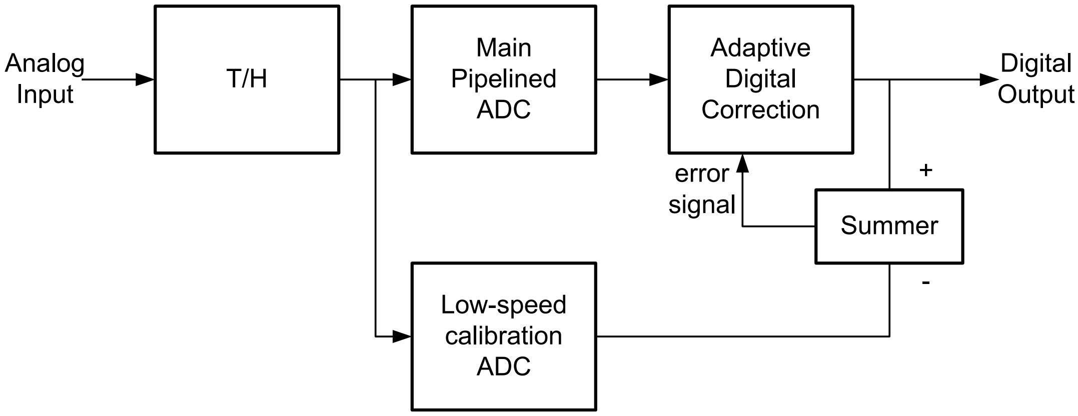


Figure 2.32 – Pipelined ADC is background calibration algorithm

In practice many challenges exist with implementing such systems, one of the biggest being the timing matching (both input bandwidth, and phase characteristics) of the two ADCs, and using efficient algorithms to correct of the main ADC nonlinearities efficiently, however the technique used has enabled performance close to 16-bit with 250MS/s sampling [29], and even some work has enabled the removal of the front T/H or S/H [30].

In half-duplex communication systems, where some dead-time in operation is available, one may choose to use a foreground calibration system. In pipelined ADCs foreground calibration is usually done by taking the 1.5 bits from each stage, mapping to a higher resolution space in the digital domain, and applying digital gain (to correct for the error in analog gain) to the values before adding them together. The computation engine to find the correct gain coefficients, and actual multiplier engine, all implemented in digital can be expensive in area and power. Commonly Least Mean Square (LMS) algorithms are used to estimate the digital coefficients [31][32]. Work in this area is commonly used to extend the sampling rate of the converter, where settling error dominates non-linearity. Using foreground calibration of this nature, 12-bit resolution with 200MS/s [31] and even 500MS/s sampling [32] have been demonstrated.

Re-configurability is possible and done often in Pipelined ADCs [33][34][35][36]. The ideal re-configurable system should allow trade-off of resolution and sampling rate for constant power consumption. This may not be possible as a direct trade-off, but each parameter can be manually changed, resulting in an effective reconfigurable system. To reduce resolution, commonly the earlier stages in the pipeline can be powered down, and the signal can be forwarded to the latter stages, trading power for resolution. To trade sampling rate for power, commonly time-interleaving is used. To achieve the maximum sampling rate all channels of the ADC are used, but to trade sampling rate for power, some channels can be powered down. Figure 2.33 shows such a system.

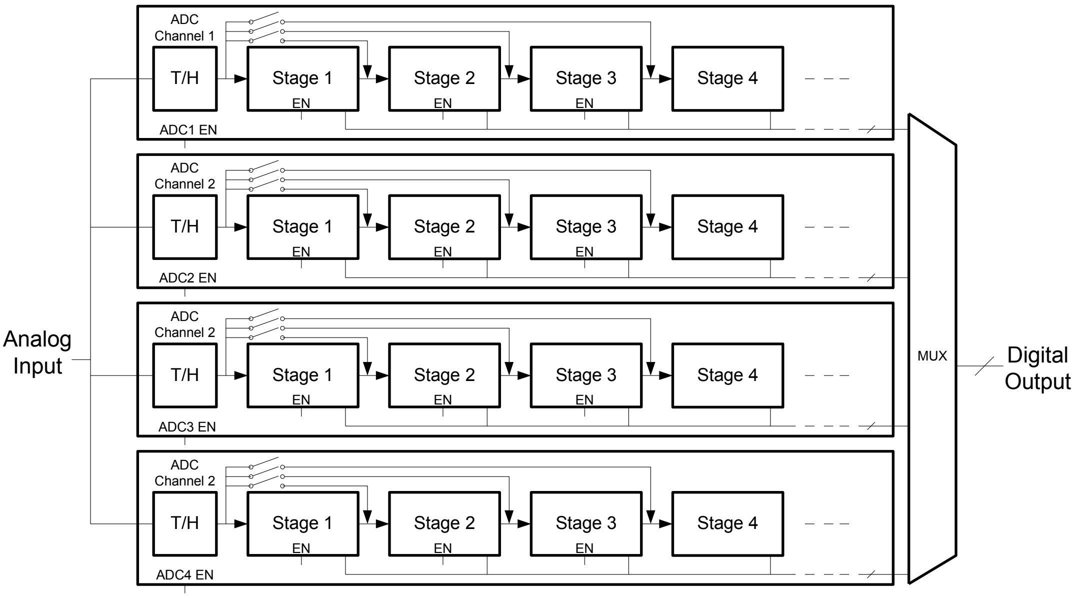


Figure 2.33 – Reconfigurable Pipelined ADC

Although examples of such systems have been implemented, they tend to not achieve similar performance comparable to standard pipelines primarily due to all the parasitics from the switches. More importantly, the reconfigurable system is achieved by implementing a super-set of all resolution and sampling rate configurations. If scaling resolution and sampling rate linearly is desirable, then this mode of operation uses all circuitry implemented on silicon, and the block is very inefficient in terms of area, and hence will also be for power consumption due to the excess parasitics. An alternative approach has been proposed to time-interleaving to achieve sampling rate re-configurability, by switching individual stages between pipeline mode and cyclic mode [33]. As explained briefly in Chapter 1, cyclic ADCs are similar to pipelined ADCs but the residue signal is looped around the same stage. If the pipeline is designed to operate at the highest possible application mode sampling rate, then for reducing sampling rate, one stage of the pipeline can operate in cyclic mode. This approach of course does not allow linear scaling of sampling rate and power, but can be seen as an alternative approach.

2.5.4 SAR converters and time interleaving

SAR ADCs were very briefly covered in Chapter 1, and it was shown that they are one of the popular choices of data converters for time-interleaving. A lot of the research and advancement in time-interleaving has been done around SAR ADCs, since they produce very power efficient time-interleaved systems [9]. SAR ADCs perform a binary search to digitise the input signal. They comprise a comparator, a DAC and control logic. The DAC drives the searching reference level, and the comparator compares this to the input. The DAC can be implemented as a resistor DAC [37], however commonly they require laser trimming to meet the resolution, since on chip calibration can be quite complex. Alternatively, a capacitor based DAC can be used based on charge re-distribution. For a capacitance DAC one can choose to use an independent S/H capacitor [38] or embed the S/H within the DAC [39]. Figure 3.34 shows a popular SAR architecture with a binary weighted DAC [10]. As previously explained, the SAR ADC performs a binary search starting from the MSB. The input signal is first sampled on to the capacitor array, following switching of the capacitors between ground and reference, the positive input of the comparator moves. The SAR logic moves these switches in order attempting to move the positive input of the comparator closer to the negative input. Alternatively a charge-sharing DAC can be used which requires pre-charging of the capacitance to the reference [40]

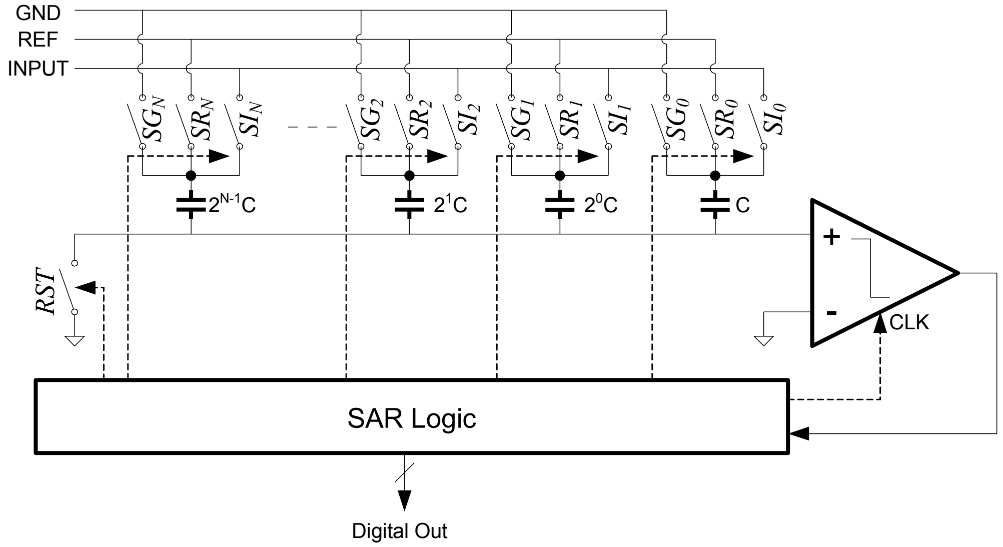


Figure 2.34 – SAR ADC with Binary Weighted Capacitor DAC

SAR ADCs are popular for time-interleaving since they are fundamentally low power architectures, since the only analog circuitry present is the comparator. On the other hand, due to their large capacitor array, when a number of channels are present, the overall size of the block can be quite big [10]. Using an attenuator capacitor, the binary array can be split into two banks, similar to an R2R architecture, however for capacitors [41][42]. A C2C architecture can reduce the overall size of the block.

Apart from area implications due to the large number of capacitors, many of the SAR DAC techniques require sampling of the input signal on the DAC capacitors directly. This results in input capacitance size issues with SAR ADCs. When used in time-interleaving system, this can become a big problem, both since the input capacitance increases and the nominal signal bandwidth increases. For this reason most high-performance SAR ADC use a dedicated frontend S/H circuit and amplifier per channel [9]. As explained previously, one global frontend sampler is not practical, so a time-interleaved sample and hold circuit is first realised [18] followed by channels of a time-interleaved ADC. This separation of T/H circuitry from the ADC circuitry, although their channels work in phase from one and another, is crucial for the performance of the circuit, mostly for practical layout and implementation reasons. Figure 2.35 shows the layout strategy for the time-interleaved T/H circuitry in conjunction with the ADC channels for a high-performance time-interleaved SAR ADC [9]. Here the T/H circuits have been separated from the main ADC channels, this allows careful design and layout of T/H circuits to a smaller pitch, allowing for easier distribution of matched clock signals to the channels. Once the input signal has been sampled, matching of clock and signal paths to each ADC is not as critical, since the signal is already sampled. If the T/H and ADC were combined, achieving such a small pitch in the circuit and clean distribution of clocks would have not been possible.

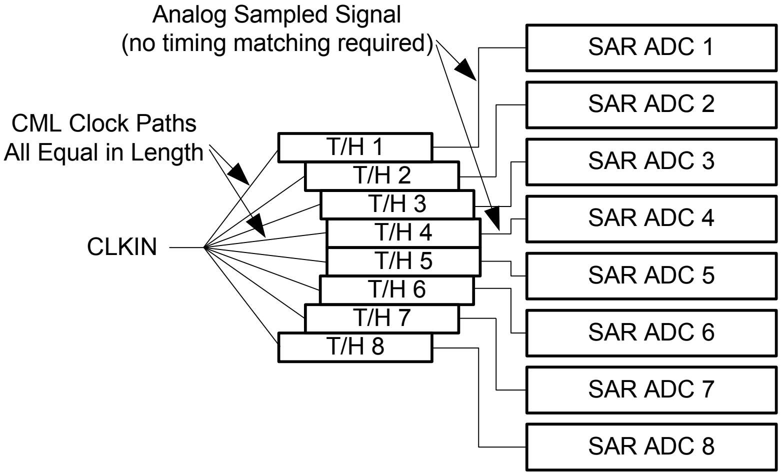


Figure 2.35 – Layout strategy and motivation for separation of T/H from ADC [9]

As the number of channels is increased, the time available for sampling can be setup differently. Figure 2.36 shows two possible timing options for a 4-channel time-interleaved ADC with T/H per channel. Option A has shorter sampling time, with longer time for conversion, while Option B has longer time for sampling eating away at the available time for conversion. Clearly with the increase in number of channels, the options in timing arrangement increases.

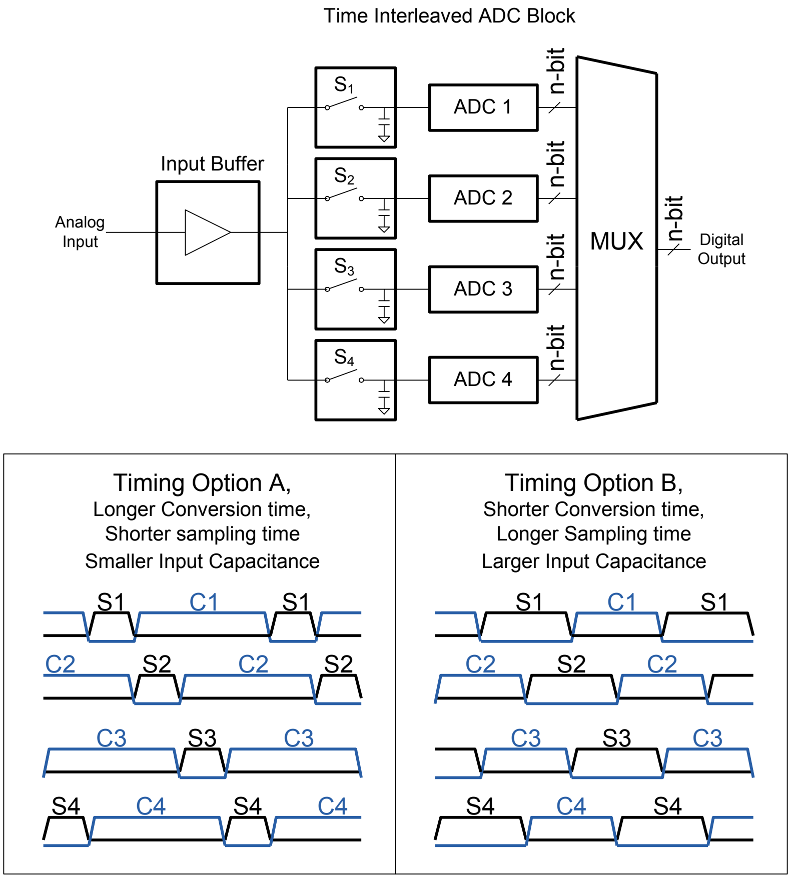


Figure 2.36 – Choice of T/H timing versus conversion timing

What is important is that Option B has a larger input capacitance, equal to 2 channel capacitances plus all parasitics, while Option A has only the capacitance contribution of 1 channel plus parasitics. Since input capacitance is such a problem for time-interleaved SAR ADCs, is common to try and reduce the ratio of sampling time to the comparison time, by improving the T/H circuitry [18].

SAR ADCs can be made reconfigurable for resolution, sampling rate, power consumption, and supply voltage range. The sampling rate of a SAR ADCs is proportional to the resolution, meaning for a given SAR ADC to double the effective sampling rate, the resolution (and hence the number of search loops) requires halving. As explained in Chapter 1, ideally halving of sampling rate should only result in 1-bit loss in resolution. For this reason SAR ADCs are not suited for direct resolution and sampling rate trade-off, however some re-configurability in their performance is possible, by introducing programmability and re-configurability in the DAC and comparator performance. For example in [43] the DAC and comparator are made programmable allowing the ADC to switch between a setup for Ultra-wideband and Bluetooth. While in [44] a re-configurability between 5 to 10-bits, while maintaining a near constant power FOM is achieved by allowing for an adjustable power supply between 0.4V to 1V. This shows that some level of programmability in SARs are possible, however this does come at the cost of more complex block design, since the algorithm of a SAR converters is not naturally suited to re-configurability.

2.5.5 Sigma-Delta converters and re-configurability

Sigma-Delta ADCs are a popular choice for cellular applications, because of the signal bandwidth present in those systems (in the 100s of KHz, to few MHz space), and their high-resolution requirement while maintaining low power operation. Re-configurability in Sigma-Delta ADCs is also possible with programmability in the system.

In the space of cellular, with many standards of communication such as EDGE, GSM, Bluetooth, UMTS, DVB-H, with signal bandwidths of 100KHz, 200KHz, 500KHz, 2MHz, and 4MHz respectively, multi-standard receivers are commonly realised as re-configurable Sigma-Delta ADC [45]. Also for WiFi and WiMAX the signal bandwidths and required resolution performance are comparable, and re-configurable Sigma-Delta ADCs are also used to build soft receivers [46][47]. Generally speaking sigma-delta ADCs are well suited to re-configurability, where the building blocks for a high-order modulator and DAC can be implemented as part of the system, and the over-sampling ratio, loop filter order, filter frequency, and DAC resolution can be adjusted for different modes of operation. Since thermal noise and modulator quantisation noise of systems scale differently for different over-sampling–ratio at different input bandwidths, it is common that parts of the system require over designing to meet the re-configurable nature of the system [48]. Making the quantiser levels programmable can help with this variability.

In recent years with the addition of LTE to the cellular standards, with bandwidths of 5MHz, 10MHz and 20MHz, the re-configurable sigma-deltas have become more challenging however examples of re-configurable implementations over this whole space have been demonstrated [48]. Due to the over-sampling nature of these ADCs, it is difficult to apply them for signal bandwidths in the 10s to 100s of MHz in frequency in today's technologies.

2.6 Conclusions

Having definitions to quantify the key non-idealities in A/D converters is important in enabling their design. Especially in time-interleaving systems, individual channel non-idealities can produce more complex error effects which can limit the system performance heavily. In time interleaving systems, assuming the non-idealities of each channel are equal, the overall converter will have the same characteristics as the individual channels, however this is practically impossible in any real implementation. It is important to understand how the mismatch of different channels affect the converter's overall performance. In most modern time-interleaving system, the use of one global frontend sample and hold circuit is omitted in favour of a more power efficient sample and hold circuit per channel. This, though more efficient, can put greater requirements on the matching between channels. Four parameters of channel mismatch were identified which can limit converter performance. These are offset mismatch, gain mismatch, input signal bandwidth mismatch, and sampling clock mismatch.

Calibration is becoming a big part of modern mixed signal design. Especially in SoC solutions, where the reducing supply voltage and relatively low cost of digital has meant that meeting the analog requirement in the blocks is become harder and some are choosing to relax these requirements allowing for digital based calibration of the performance at test, or in the field. It is important to not assume that calibration is for free, it commonly introduces an increase in area, and power consumption, and this cost should be compared against the cost of attempting to meet the performance criteria without the need for calibration in the first place.

Chapter 3 - Time Interleaved Counter ADC

3.1 Introduction

In this chapter the new A/D converter architecture developed, taking advantage of a high level of time-interleaving while achieving re-configurability between operation speed and resolution, will be presented. As previously discussed, time-interleaving enables extension of ADC architectures performance, specially for operation speed. In this work the natural progression of this idea is taking to its limit by exploring the implementation of a highly time-interleaved ADC of one of the slowest, and simplest A/D architectures. Counter ADCs, or Single Slope ADC are one of the slowest converters, and are commonly left out of architectural comparisons of A/D converters since they are very rarely used as individual ADC in any application. This work takes this extremely slow and simple ADC and applies a large level of parallelism to build a converter in the operation speed and resolution for high-speed communication applications. Due to the high-level of time interleaving used, and the nature of the counter ADC, the work realises a re-configurable converter that can trade resolution for speed at different configurations for use in multi-standard communication systems.

In this chapter the counter ADC will be first introduced, and some basic design rules associated with this type of converter will be explained. Counter ADCs are commonly used in parallel in CMOS image sensors to digitise the captured values for a row of pixels in an array simultaneously. The work done in this area will be looked at. This type of parallelism is very different from time-interleaving, since in parallelism all channels are working in phase with one another, simultaneously digitising a large number of samples, while in time-interleaving systems the data arrives in sequence, at a much higher rate, and converters work out of phase from one another. The chapter then moves to explain the proposed time-interleaving counter (TIC) ADC, and its main design challenges and considerations. In this chapter the TIC ADC will be introduced as a generic block, not attempting to meet any particular specification, but explaining the architecture, design techniques and specification critical areas which should be considered when attempting to build a TIC ADC to a particular specification. In the next chapter the implementation of a particular TIC ADC, to a given specification, which is manufactured, is presented.

3.2 Counter ADC

3.2.1 Description of the Counter ADC Architecture

The counter ADC or single slope ADC is one of the simplest and slowest A/D converter architectures. In a way it is the opposite of the flash ADC. In the flash ADC the input signal is compared to all possible digital reference levels simultaneously, while in the counter ADC the input signal is compared to all digital reference levels in sequence. Figure 3.1 shows the main elements and timing diagram of a simple counter ADC.

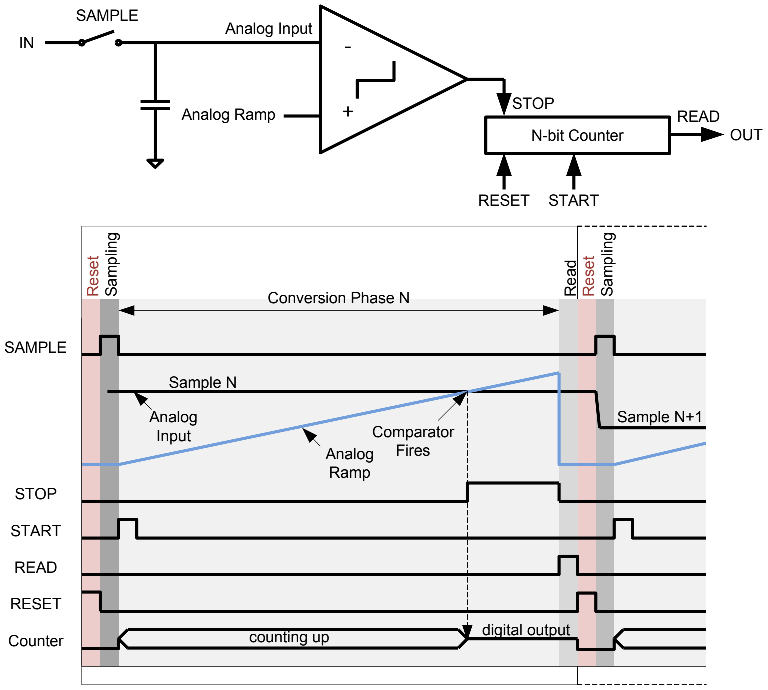


Figure 3.1 – Block diagram, and timing diagram of counter ADC

Here the input is first sampled, and then held at one input of a comparator. Then an analog ramp, travelling the possible full range of the input is fed to the other input of the comparator, while in parallel a digital counter in the back is reset and started. As the ramp moves closer to the analog input, the counter will count a code higher, eventually the ramp will reach the input signal, and the comparator will toggle, stopping the output counter. If the analog ramp is correctly synchronised to the digital counter in the back, the value in the counter after the comparator toggles is a digital representation of the analog input.

The comparator is comparing the input signal to all digital reference voltages in sequence. The ramp can be considered a sequence of discrete voltage levels representing the digital reference levels, synchronised to the backend counter. The counter ADC is in effect an analog to time, followed by a time to digital converter. The counter ADC has the slowest possible slow sampling rate in all converter architectures, since it performs sequential comparisons.

3.2.2 Implementation of an Counter ADC

Figure 3.2 shows a practical implementation of a Counter ADC system. The comparator backend counter has been replaced with a simple N-bit memory unit, which takes digital data from the global counter. The Ramp for conversion is made using a D/A converter, taking input from the same counter. The comparator output is high until the comparator fires. The N-bit memory is replicating the N-bit counter value, since the comparator output is connected to the Write signal of the memory, until the comparator fires, when then the value of the counter at that moment remains in the memory to be read out. A simple counter ADC can operate with almost only one digital control signal.

The DAC used for the ramp generation, requires a resolution equal to the over-all converter resolution, and in fact the linearity of the ramp can be directly correlated to the linearity of the transfer function of the whole ADC. Commonly some over-ranging is required in the ramp, and hence digital count value, to over-come settling time issues at the start of the ramp due to input capacitance of the comparator.

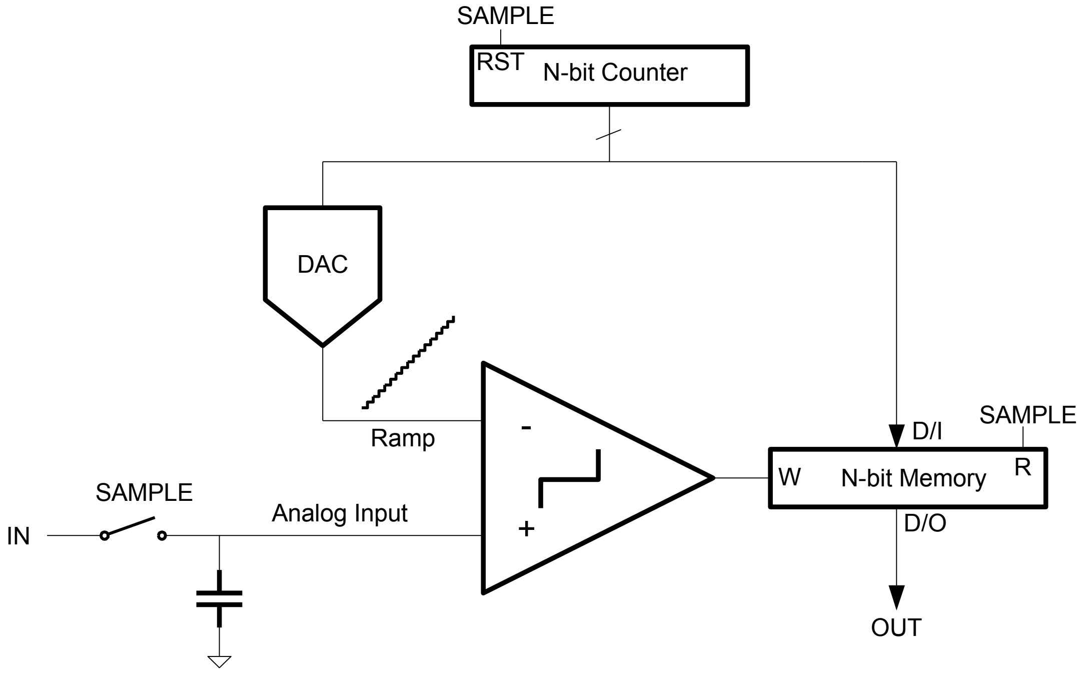


Figure 3.2 – Practical implementation of a Counter ADC system

The operation speed is related to the speed in which the DAC and digital counter and memory can be implemented at the back. Also the delay from the inputs to the comparator crossing one another, till the output of the comparator toggling must also be less than half an LSB code step, otherwise this will introduce offset in the output. In the later section we will talk about how this can be reduced or ignored. The comparator hence has a gain-bandwidth requirement tied to the offset of the ADC. Also the comparator requires to have the overall RMS noise matching that of the overall ADC resolution. There are many different comparator architectures and ramping strategies that can be used. These will be discussed later in the chapter and in the following chapter.

3.3 The use of parallel counter ADCs in imaging

Counter ADCs are commonly used in applications where a low sampling rate is acceptable however a high-level of monotonicity is required from the converter. Counter ADC were commonly used for instrumentation applications [49], however when implemented as individual units, the accuracy of the DAC, and requirements of the comparator results in an expensive implementation, and today commonly sigma-delta incremental ADCs are used for those applications [12]. Today counter ADCs are mostly used as column parallel ADCs in CMOS image sensor [50][51][52][53]. They are popular for this application since many converters, in the thousands, can be placed in parallel, where each unit has very little circuitry allowing for compact layout. Figure 3.3 shows a CMOS image sensor array and readout using a column ADC architecture.

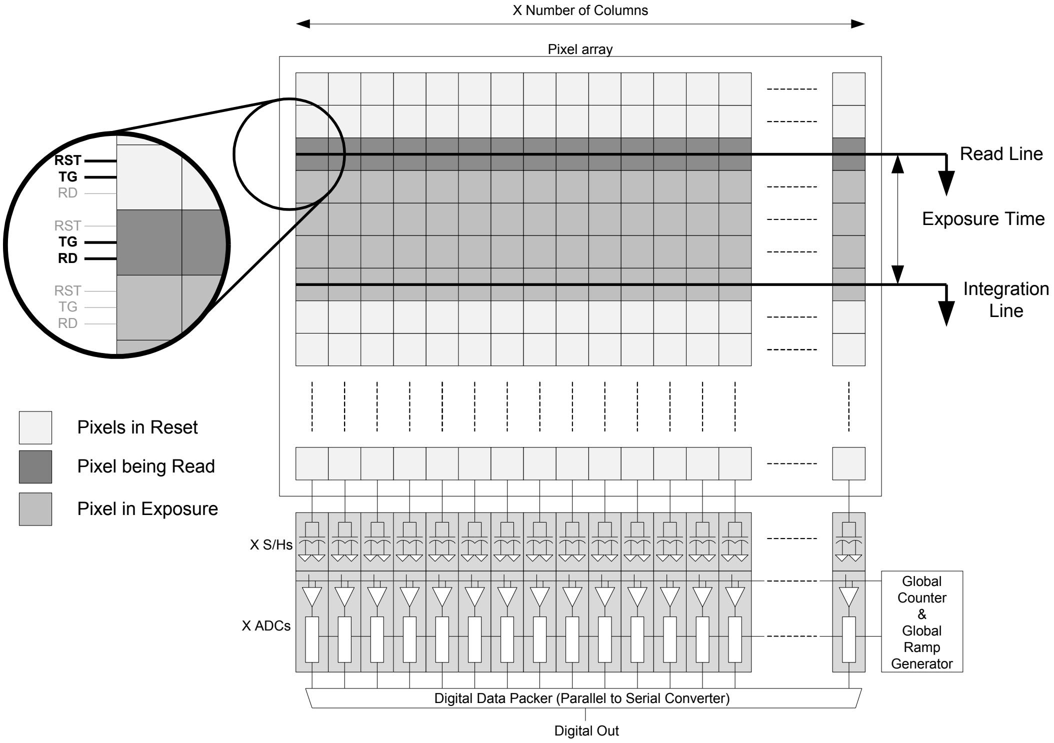


Figure 3.3 – CMOS image sensor with column ADC readout and converter

In a CMOS pixel array, a soft shutter system is commonly used, where each row of pixels is set to integration for a fixed amount of time, after the exposure time for that row is complete, that row of pixels is read out for conversion. This results in an effective integration and read line travelling through the rows. All the rows between the read line and integration line are active and exposured to light, while all other rows are held in reset. This form of operation means a whole row of pixels are read out simultaneously and require conversion.

A common architecture used for digitising of a whole row of pixels is the column parallel ADC architecture, where effectively a small ADC is implemented in each row of the pixel array which completes the digitisation of that row's read-out values in parallel with all other rows. A popular ADC to use in the column is a counter ADC due to its simple implementation, and possible sharing of common circuit elements such as the DAC and the digital counter [54]. Figure 3.4 takes a closer look at a counter ADC bank used in a column parallel architecture.

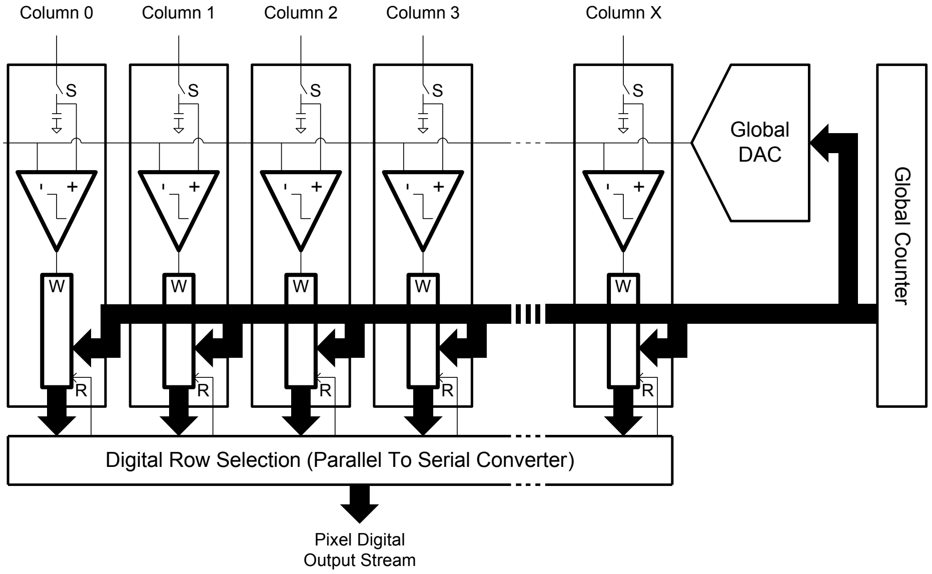
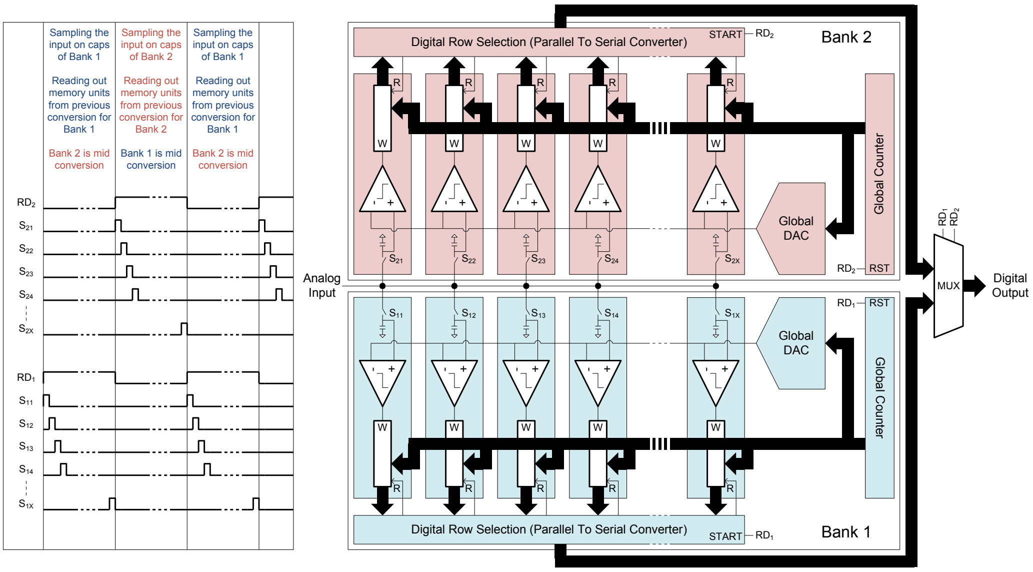

Column 0
Column 1
Column 2
Column 3
Column X
S
W
R
Global DAC
Global Counter
Digital Row Selection (Parallel To Serial Converter)
Pixel Digital Output Stream

Figure 3.4 – Counter ADC used in column parallel Architecture

Here each column is only made from a sampling capacitor, a comparator and a small memory unit. The DAC and counter for each converter has been replaced with a global counter which feeds the digital count value to all the columns, and a global DAC as a ramp generator which feeds a uniform ramp to all columns. The resolution requirements for the DAC remains the same as a unit DAC per ADC, however this DAC requires a lower output impedance since it drives many columns in parallel. Apart from area saving, the other benefit of the global counter and DAC is the matching and uniformity between different columns. It is important to emphasise, that a global ramp and counter work here since all columns perform conversion at the same time. They are all working in parallel with each other, with a global sampling signal, and hence can work with a global ramp, and global counter value. What is implemented in the column parallel counter ADC is not time-interleaving, but parallelism.

3.4 Parallelism with counter ADC block Ping-Ponging

Parallelism can be used for digitising a large number of samples, arriving simultaneously, for example in an image sensor, but cannot be used when samples arrive at a higher rate but in sequence. For example a bank of 1000 parallel counter ADCs, each sampling at 1MS/s can digitise 1 million samples a second, but they have to arrive in batches of 1000 samples, at 1MS/s intervals. Such a bank of parallel converters does not build a 1GS/s converter. A 1GS/s converter must manage 1 million samples arriving in sequence a second.



Sampling the input on caps of Bank 1
Reading out memory units from previous conversion for Bank 1
Bank 2 is mid conversion
Sampling the input on caps of Bank 2
Reading out memory units from previous conversion for Bank 2
Bank 1 is mid conversion
Sampling the input on caps of Bank 1
Reading out memory units from previous conversion for Bank 1
Bank 2 is mid conversion
RD₂
S₂₁
S₂₂
S₂₃
S₂₄
S₂X
RD₁
S₁₁
S₁₂
S₁₃
S₁₄
S₁X
Digital Row Selection (Parallel To Serial Converter)
START
RD₂
Bank 2
R
W
Global Counter
Global DAC
S₂₁
S₂₂
S₂₃
S₂₄
S₂X
RD₂
RST
Analog Input
RD₁
RD₂
MUX
Digital Output
S₁₁
S₁₂
S₁₃
S₁₄
S₁X
RD₁
RST
Global DAC
Global Counter
W
R
Digital Row Selection (Parallel To Serial Converter)
START
RD₁
Bank 1

Figure 3.5 – System diagram and timing diagram of a ping-pong counter ADC

To realise a real higher rate of sampling via parallelism, a technique known as bank ping-ponging can be used. Figure 3.5 shows the ping ponging concept implemented for parallel counter ADC system. In Ping-Ponging two identical banks of parallel ADCs are used, and two banks are working in a real time-interleaved arrangement from one another, however inside each bank parallelism is used. In this example, the input is sampled onto the sampling capacitors of Bank 1's ADCs in sequence. Once this is done, the whole bank is ready for parallel conversion. When this begins, the continuous stream of input sampling cannot be stopped, so the input continues sampling onto the sampling capacitors of Bank 2. While this is happening, Bank 1 is performing parallel conversion of all its inputs. Once the sampling onto Bank 2 is complete, the conversion of Bank 1 will be complete; the input stream can go back to sampling onto Bank 1. While new samples are being sampled on Bank 1, Bank 2 starts its parallel conversion of all inputs it previously sampled. Also while sampling on Bank 1 is occurring, the result of the conversions done in the previous conversion period for Bank 1 will be read out in sequence.

In this system although each bank is performing conversion in parallel (and not real time-interleaving) but since a higher level of time interleaving is realised between the two banks, the input stream is continuously sampled (in sequence) and the block generates a continuous stream of digital outputs. The overall converter achieves a sampling rate equal the sampling rate of each converter times the number of ADCs in each bank divided by 2. This is because each converter effectively spends half its time in 'dead-time', not performing conversions, waiting for its turn to sample, or waiting for other ADCs in its bank to sample, since conversion can only begin once all comparators in a bank have finished sampling. This also means that the block consumes twice as much static power as necessary for conversion, and also consumes twice the space it should really need for the conversion rate it achieves. This yields an inefficient system.

Apart from inefficiencies in power consumption and size, the architecture suffers from some non-idealities that are difficult to over come. Since a sequence of samples are captured in a bank, and then a parallel conversion is applied, channels earlier in the bank require to hold their analog samples for a longer period of time compared to the later channels in the system. At higher operation temperatures, the earlier channels will be subject to larger leakage compared to later channels resulting in relative errors between the channels. Also, since at times large blocks such as DACs and counter and bus drivers are held in reset, and then brought out of reset, a large amount of timing dependent noise appears on the power supply and substrate. This noise only affects certain samples which in terms of timing collide with the power-up events. This again produces non-ideal artefacts at the output. As a whole, although the ping-pong architecture does yield a correctly operating solution, it does result in a very inefficient system with some drawbacks in performance. A real time interleaving solution would be desirable.

3.5 The Time Interleaving Counter (TIC) ADC

Up to now the case has been made that exploring time interleaving to its extreme, interleaving the slowest and simplest converter, can result in an interesting, high-performance, and potentially versatile and reconfigurable new ADC architecture. There are two problems here, one to define the shape (the building blocks and timing) of a real time-interleaved counter ADC, and the other to implement the system and building blocks in an efficient and versatile way.

3.5.1 Defining the TIC ADC

Figure 3.6 shows the block diagram and timing diagram of the proposed time interleaved counter ADC. Each channel of the system, is an independent implementation of a counter ADC, similar to a parallel bank of ADCs used in an imager, however here, each channel works out of phase from its neighbour. Each row is one clock cycle, effectively 360-deg divided by the number of rows, out of phase from its neighbour. Two example time points, A and B, have been highlighted in Figure 3.6 as an example. For example at time A, row 1 is sampling the input, row 2 is in reset, row 3 is reading out the result of the conversion, and row M has just finished sampling in the previous clock cycle and is now busy converting. At time point B, row 1 is in reset, row 2 is reading out the value of the conversion, row 3 is almost at the end of its conversion period, and row M is sampling the input. Indeed, at any moment in time, one row is in reset, one is sampling the input, one row is reading out, and all other rows are mid conversion. At the next clock cycle the sampling, resetting, and reading move forward to the next rows in a circular way. This scheme is rotated in such a way that operations on row 1 directly follow those on row M. Here Ramp 1, Ramp 2 to Ramp M are out of phase from each other by one clock cycle.

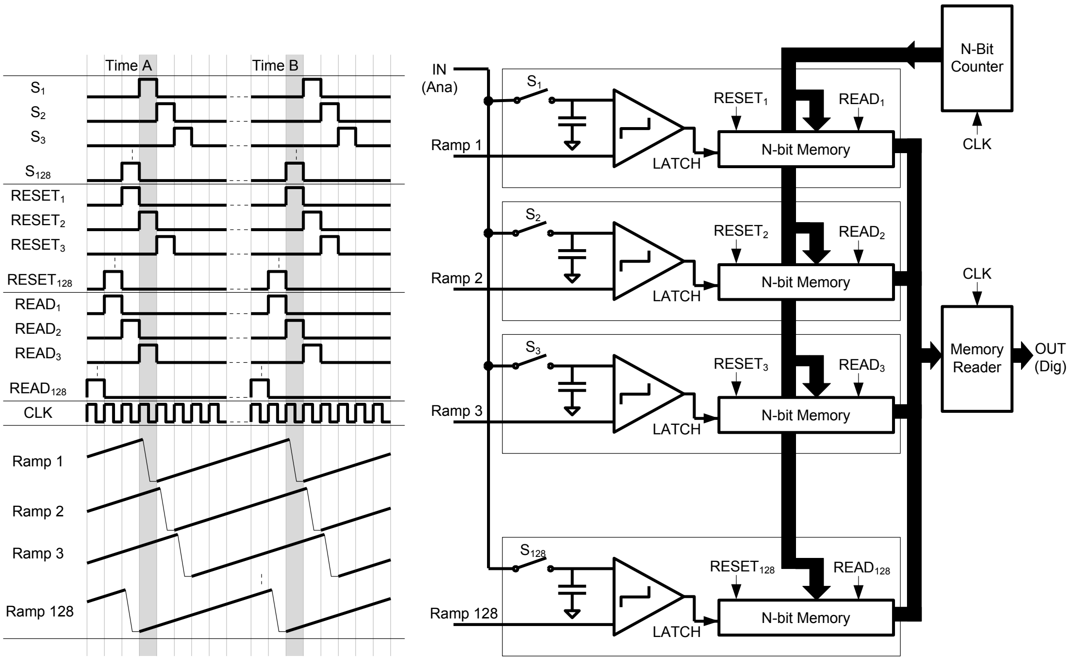


Figure 3.6 – Block diagram and timing diagram of proposed TIC ADC

In the proposed system, the digital counter for each row has been replaced by a global counter, however each row is operating out of phase from its neighbour, meaning it requires a counting sequence out of phase by a code from its neighbour. This problem can be overcome by placing an adder per row, or alternatively performing digital subtraction from the output of each row. The latter means that from the output of each row, for example row X, where X is a number between 1 to M, the value of X should be subtracted digitally from all its outputs, since the digital counting code that row X was using was effectively out of phase by X digital code steps.

3.5.2 The clock frequencies, sampling rate and resolution

It is important to note that the clock frequency of the digital counter, is independent of the sampling frequency. In this block description, the ramps are considered analog ramps. The sampling rate of such a converter is equal to each converter's sampling frequency multiplied by the number of converters, while the resolution of the converter is set by the ratio of the digital counter clock divided by the sampling rate. Note that each converter, similar to most time-interleaved systems, has an un-even duty cycle on its sampling clock. It spends a very short period of time sampling, and a long period of time in conversion. The sampling period for each converter compared to its conversion time is smaller by the number of rows. The overall time-interleaved converter sampling rate can be calculated as:

$$F_{S-TIC} = F_{S-ROW} \times M \tag{3.1}$$

Where $F_{S\text{-}TIC}$ is the overall converter sampling rate, $F_{S\text{-}ROW}$ is the sampling rate of each row, and M is the number of rows. As mentioned above, for the period of $1/F_{S\text{-}ROW}$ each row spends a short amount of time sampling, and a large amount of time converting. The period of time available for conversion for each row can be defined as:

$$T_{C-ROW} = \frac{M-1}{M} \times \frac{1}{F_{S-ROW}} \tag{3.2}$$

Where $T_{C\text{-}ROW}$ is the time available for conversion. The effective resolution of the overall converter is the resolution of each sub-ADC. The resolution of each sub-ADC is effectively the number of codes counted during the conversion period, and can be defined as:

$$Res_{TIC} = \frac{M-1}{M} \times \frac{F_{CLK-C}}{F_{S-ROW}} \quad \rightarrow \quad Res_{TIC} \approx \frac{F_{CLK-C}}{F_{S-ROW}} \tag{3.3}$$

Where $F_{CLK\text{-}C}$ is the counter clock frequency. The resolution can be estimated as the ratio of the counter clock frequency to the row sampling frequency, but to be exact has a losing factor related to the time of $F_{S\text{-}ROW}$ lost during sampling. Of course to actually meet the resolution, the analog circuitry requires designing to a given specification, and the ramps require to meet the linearity requirement for that resolution, but the frequencies calculated using the equations demonstrate the time required for the backend to enable the system to count to the required resolution, assuming the rest of the system meets the requirements.

The main weakness of the TIC ADC architecture is latency in conversion. An individual counter ADC has a slow conversion speed, exponentially proportional to the resolution of the conversion. The conversion speed of the block can be increased by time-interleaving, but unfortunately like all time-interleaving architectures the latency of conversion remains the conversion time of an individual channel. The implications of latency can be quite complex in communication systems, as latency may not be a problem when receiving a continuous stream of data, however if the converter is used inside an Automatic Gain Control (AGC) loop, the latency of the converter can set a hard limit on the speed the AGC algorithm can converge. The latency requirements on the block should be considered when choosing the number of channels and operation speed.

3.5.3 Changing the clock frequencies and re-configurability

Referring back to the proposed block diagram in Figure 3.6 and the equations in the previous section, Table 3.1 aims to show some different frequency settings as examples, resulting in different sampling rates and resolutions for the overall converter. Here $F_{C\text{-}B}$ is the base clock for the ADC, where all the clocks passed to each converter is effectively a divided and selective pulse combination of this clock. $F_{C\text{-}C}$ is $F_{CLK\text{-}C}$ the clock frequency of the backend counter, and $F_{S\text{-}R}$ is $F_{S\text{-}ROW}$ the effective sampling rate of each ADC channel, M is the number of rows, *eff* is the effective factor in each clock period for a ADC used for conversion, $F_{S\text{-}TIC}$ is the overall converter sampling rate, which is effectively the same as $F_{C\text{-}B}$, and Res_{TIC} is the theoretical converter resolution possible with these clock settings.

Table 3.1 – Examples of different clock frequency settings

$F_{C\text{-}B}$	$F_{C\text{-}C}$	M	$F_{S\text{-}R}$	eff	$F_{S\text{-}TIC}$	~Res_{TIC}	Res_{TIC}
Input	Input	Input	$\frac{F_{C-B}}{M}$	$\frac{M-1}{M}$	$F_{C\text{-}B}$	$\log_2\left(\frac{F_{C-C}}{F_{S-R}}\right)$	$\log_2\left(\frac{effF_{C-C}}{F_{S-R}}\right)$
1GHz	1GHz	128	~7.8MHz	0.992	1GS/s	7-bits	6.98-bits
500MHz	1GHz	128	~3.9MHz	0.992	500MS/s	8-bits	7.99-bits
250MHz	1GHz	128	~1.9MHz	0.992	250MS/s	9-bits	8.99-bits
2GHz	2GHz	64	~31.3MHz	0.984	2GS/s	6-bits	5.95-bits
1GHz	2GHz	64	~15.6MHz	0.984	1GS/s	7-bits	6.98-bits
500MHz	2GHz	64	~7.8MHz	0.984	500MS/s	8-bits	7.99-bits
250MHz	2GHz	64	~3.9MHz	0.984	250MS/s	9-bits	8.99-bits

The table shows effectively two different converters, one with 128 rows, and one with 64 rows. What is interesting here is that by dividing $F_{C\text{-}B}$ by 2, compared to $F_{C\text{-}C}$ for a given converter, the effective sampling rate of the converter halves while the resolution grows by 1 bit. This makes sense, since by halving the effective sub-ADC sampling rate, the conversion period for each converter doubles, allowing the backend counter to count to twice as big a number as before for a given input range, and this results in doubling of the dynamic range and hence the extra bit in resolution. This dividing can continue to trade sampling rate for resolution. Separately, a converter with 64-channels, producing a different sampling rate and resolution for matched $F_{C\text{-}B}$ to $F_{C\text{-}C}$ values can again trade sampling rate for resolution by dividing $F_{C\text{-}B}$.

What this shows is that for given implemented TIC, with a certain number of rows, the resolution and speed of the converter can be traded off actively by dividing the sub-ADC root clock compared to the counter clock frequency. Of course all of this is only true if the analog sub-blocks in the system, primarily the comparator, the sample-and-hold circuitry and the ramp all can operate to the highest clock frequency settings and also to the highest resolution settings.

Looking at the table, it can be seen that for example a 1GS/s 7-bit converter can be built using 128 time-interleaved ADC channels, or with 64 time-interleaved channels. To understand the trade-offs between these two implementations, and to appreciate the level of re-configurability possible with a given ADC, one must first appreciate the design constraints for the sub-sections of the system, how each of these sub-sections can be implemented, and how the top level requirements feed down to the specifications for these sub-blocks. The sub systems are,

> the ramp-generation circuitry,
>
> the Analog-Front-End (AFE) of each ADC made from sample-and-hold circuitry, the comparator, the method of applying the ramp for conversion,
>
> the top-level digital which is the global counter and the driver circuitry,
>
> each converters backend memory and global readout circuitry.

The remainder of this chapter looks at these individual sub-blocks of the system, and describes how their design and specification related to the top-level specifications. The following chapter puts the theory from this chapter into practice to build a high-performance re-configurable TIC ADC.

3.6 The Ramp Generation for the TIC

The system described in Section 3.5 is a true time-interleaved realisation of a counter ADC block, however to realise this block one of the key challenges is finding a way to generate the required number of ramps, all effectively out of phase from one another. In Table 3.1, using realistic clock frequencies obtainable in deep-sub-micron technologies, required resolution and sampling rates for communication systems of interest to this work showed the need for potentially 64 to 128 time interleaved channels, and hence many parallel ramps require generating.

A small ramp generator can be used per channel, for example a low frequency DAC, or an analog current into a capacitor per channel. The problem with implementing a ramp-generator per channel is the matching of these ramps. As described in Chapter 2, any gain error in each channel of a time-interleaving system, results in non-linearity at the top-level. Also the potential size of a 128 or more individual ramp generators can be very big. What will be proposed here is a novel new global ramp generator which is capable of generating many ramps simultaneously, all out of phase from one another. Since one ramp generator is used, all ramps will inherently match, and the resulting block due to sharing of components can be much smaller.

3.6.1 The Rotary Resistor Ring Concept

The concept of a Rotary Resistor Ring will be explained here, although to practically use the concept in a real circuit, many small modifications are required, but here the basic concept will be explained. Figure 3.7 shows a simple 8-piece rotary resistor ring system. A total of 8 resistors are connected in series, and hence 8 nodes are created. At any moment in time exactly one node is connected and driven by voltage V_{POS} and exactly one node is connected and driven by voltage V_{NEG}, all other nodes are un-driven. At the following clock edge, the driven nodes are moved forward by one step. For example at time zero, node N1 is driven by V_{POS} while node N5 is driven by V_{NEG}. In the following clock cycle, node N1 is no longer driven, while node N2 is now driven by V_{POS} and node N6 is driven by V_{NEG}. As the driven nodes rotate round the resistor ring, for each node, resistors R1 to R8 act as a potential divider between V_{POS} and V_{NEG} and hence the voltage seen on each node is effectively a step ladder climbing up to V_{POS} and down to V_{NEG} continuously where the total steps taken is the total number of resistors divided by 2. Importantly each node performs this ramping up and down exactly out of phase by its neighbour by 360-deg/number-of-resistors. If the number of resistors in resistor ring where to be increased, each node will effectively produce a ramp.

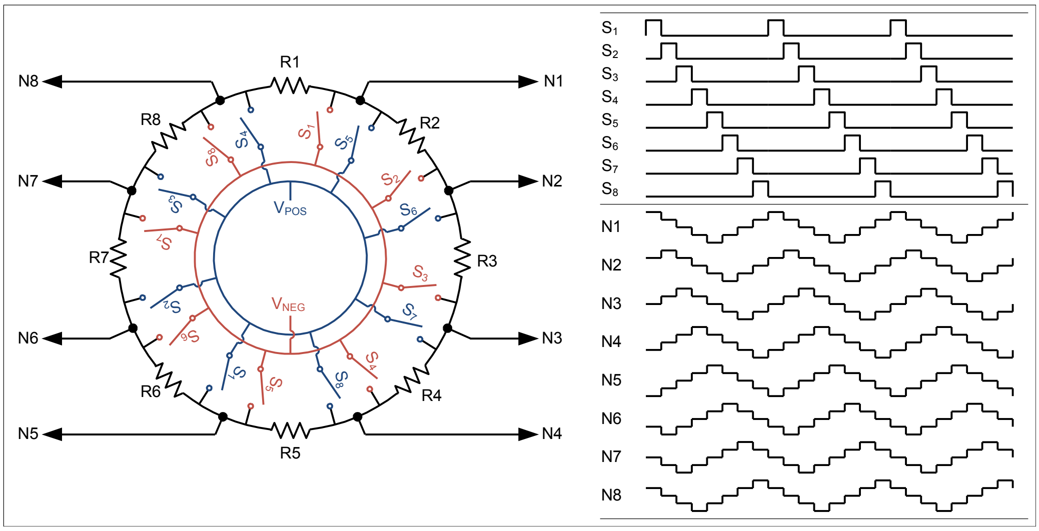


Figure 3.7 – Rotary Resistor Ring concept circuit and timing diagram

3.6.2 Building the figure-of-8 rotary resistor ring

There are a couple of adjustments required to the simple Rotary Resistor Ring before it can be of use in a TIC ADC. In a differential TIC, for each ADC we require two ramps, which are in phase, one which travels from V_{NEG} to V_{POS} while the other travels from V_{POS} to V_{NEG}. Once the travel is complete, they should reset to their start value and produce the same ramp again. This is not exactly what the Rotary Resistor Ring produces, since all nodes both ramp up and down. Referring back to Figure 3.7, it can be seen that node N1 and N8 are exactly 180-degrees out of phase from each other, when N1 is ramping up, N5 is ramping down and vice-versa. This means with a correctly timed analog multiplexer, which switches between Node 1 and Node 5 for a given ADC, one could have a continuous climbing and a continuous falling ramp which reset and then ramp in the same direction again. Unfortunately, node N1 and N5 are opposite ends of the ring, while we need to route these signal to the same ADC, and of course circular ring of resistor cannot be practically realised on silicon.

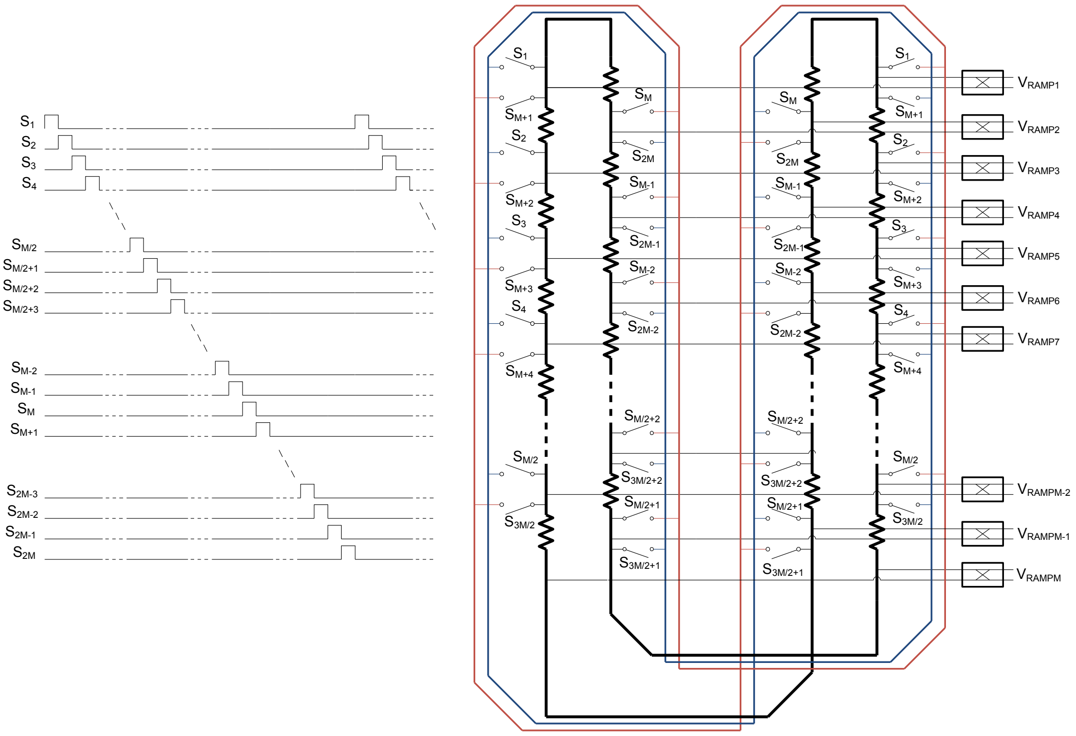


Figure 3.8 - Circuit diagram of figure-of-8 rotating resistor ring

To overcome these layout limitations, and to manage the multiplexing of the falling and rising ramps, the resistor ring can be folded in itself to realise a figure-of-8 resistor ring. Figure 3.8 shows the circuit diagram of the proposed figure-of-8 rotating resistor ring. Here a total of 2M equal unit resistors are used, if M is number of voltage steps required from each ramp. A total of 2M switch signals from S_1 to S_{2M} are used. Each switch signal is connected to two switches, one which connects a point of the ring to the positive reference, and a switch which connects a different point of the ring to the negative reference. At any moment of time only one of the S_1 to S_{2M} signals is high, tying one point of the ring to the positive reference, and one point of the ring to the negative reference. All other nodes on the ring are not driven, and are connected to created the positive and negative goings ramps for the each channel.

The exact operation of the ramp generator block can be difficult to understand, and careful study of the labelling of the switches on the rotating ring is required. To help with this understanding Figure 3.9 shows an example analog output for a few channels of the system, and the effective rotation of the resistors. Here the switches have been omitted from the diagram, and only the 2 switches that are closed at any moment are shown. It is worth emphasising that the resistors, and the connected point for each channel of the ADC to the ring does not move. What does move is the point in which the positive and negative reference connect to the ladder, causing the output voltages to appear as ramps.

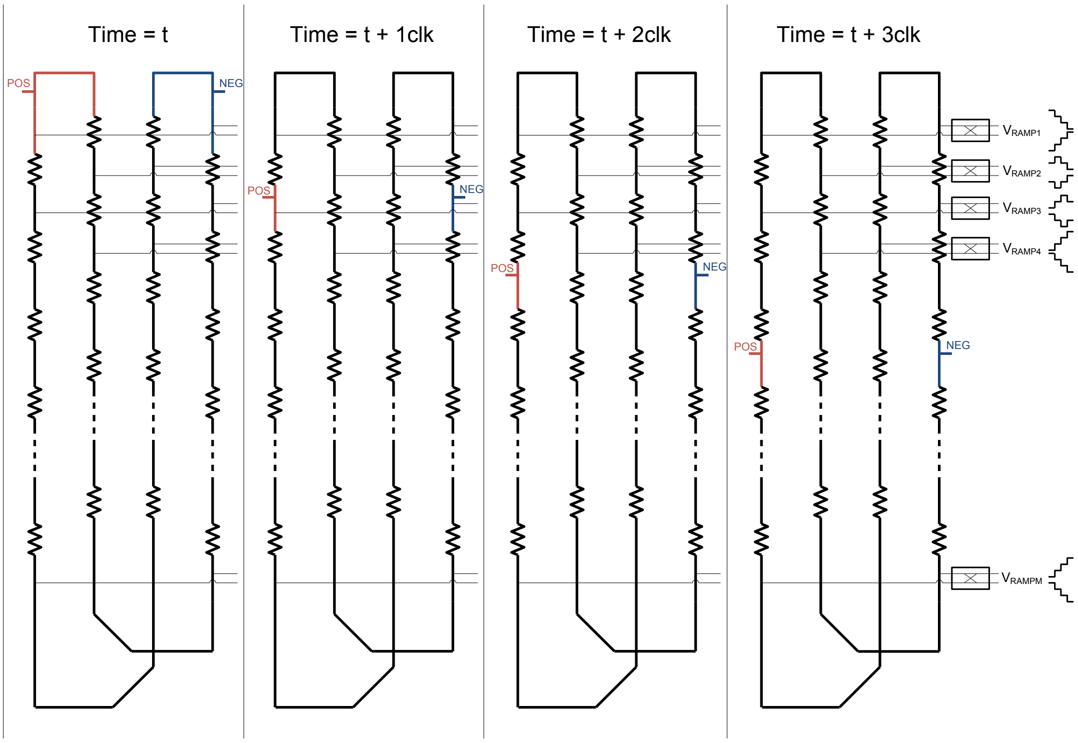


Figure 3.9 – Example of voltage rotation around resistor ring

The rotation of the Negative and Positive driven points can be realised with some basic timing circuitry. In the next chapter where an actual TIC ADC is implemented, the circuitry managing the timing of the switches is explained. The linearity of the ramps generated for each ladder, the INL and DNL of the ramp, plays an important part in defining the linearity of each ADC channel and hence the overall converter linearity. Each switch has a resistance which should not be ignored, and the mismatch in resistance of the switches results in the DNL errors in the ramp profile. As the size of the unit resistor is increased, the current consumption of the ladder is reduced, and the effective contributions of the switch resistance on the ladder voltage is reduced, for the same size of switches, however with smaller current in the system, and with the same capacitance on the ring, the ring will have longer RC settling requirements as it rotates. This trade off will be analysed further in the next chapter.

3.6.3 Choice of unit resistor and switch sizes

Generally speaking, in a given technology with certain switch Ron for device size, and mismatch characteristic, only a small amount of flexibility is available in the design of the resistor ladder. A practice design is shown in the next chapter.

The design of the sub-ADCs sets a requirement for the total voltage range expected from the ramp generator, this in turn sets a current and resistor product requirement on the ramp. This can be done with a large current and small unit resistor value, or a smaller current with larger unit resistor value. To determine the correct trade-off between resistance and current, two conflicting requirements should be traded off at the system level.

1. The choice of current and RC requirement

If the current is too small, there will be RC settling issues between steps, and more importantly at the start of the ramp. The RC settling at the start of the ramp, results in large non-linearity at the start of the ramp, this mean the full range of the ADC cannot be used, and hence some resolution of the system will be lost. If the current is set too big, the unit resistors will be small (since the voltage is pre-determined). This means the size of the switches must be increased, this in turn increases the capacitance on the ramp and defeats the point of increasing the current in the first place.

2. **The Mismatch of the switches**

The second issue is the mismatch of the Ron of the switches. In theory the current can be made smaller, and hence the unit resistors will get bigger, and the contribution of the switch resistances get smaller, and they can be made smaller, however only to a certain point, since the mismatch between the switches results in DNL errors in the ramp.

The two requirements are effectively working against each other and should be balanced for a given resolution requirement in a technology. In the next chapter this is practically done for the technology the ADC is implemented in, and the challenges in meeting the requirements for the ramp generator will be explained.

3.7 Analog Front End for channels of TIC ADC

The AFE in each channel is primarily made of the sample and hold circuitry, the circuitry which applies the ramp to the samples and the comparator, and the comparator itself. In the implementation of the TIC sub-ADC we use a technique known as bottom-plate-ramping[55] which will be explained. The TIC is implemented as a differential system.

3.7.1 Sample and Hold Circuit

The sample and hold circuit used in the TIC sub-channel comprises of a switch and capacitor. The capacitor is sized for kT/C requirements, and matching between positive input and negative input. The thermal noise requirements determining the size of the sampling capacitors due to kT/C noise is well understood [5]. Depending on the signal swing, in theory NMOS, PMOS or transmission gates can be used. For most applications, a high-level of linearity is required from the sample and hold circuitry. For this reason the *RC* settling should be made signal independent, for that reason bootstrap switches can be used which maintain a certain R_{on} over the signal range [56]. This R_{on} should be chosen to achieve the required settling time-constant which is again well understood in sample and hold and switched capacitor circuits and was derived in Chapter 2.

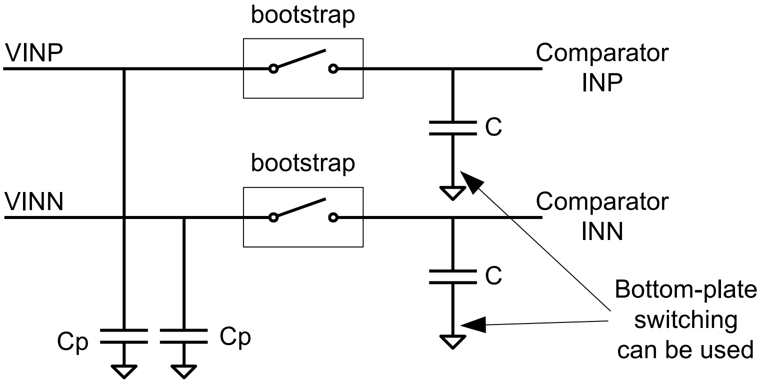


Figure 3.10 – Sample and Hold switch and capacitor

Figure 3.10 shows the basic sampling switches and capacitors. Cp represents the parasitic capacitance present on each input of the system. Cp is made from the routing capacitance, and the drain capacitance of all boot-strap switches in all other channel, while the switch is off. The total input capacitance of the TIC ADC (C_{in}) is the routing capacitance ($C_{routing}$) and the back-drain capacitance (C_{off}) of each switch which is off multiplied by the number of channel (M) minus 1, plus the sampling capacitor (C_{sample}) of one channel.

$$C_{in} = C_{routing} + M \times C_{off} + C_{sample} \tag{3.4}$$

3.7.2 Applying the Ramp

As mentioned above a technique known as bottom-plate-ramping [55] is used in the TIC AFE. Figure 3.11 shows the basic idea in a single ended configuration. Bottom-plate-ramping can be applied in many different ways, however the motivation of doing so is the same. Here the input is first sampled onto the capacitor through switches timed with signal S_S and S_E. Here S_E is opened slightly earlier than S_S. This technique is known as bottom-plate-switching [56], where the charge-injection sampled onto the capacitor is from the virtual earth side so it has a signal independent value. Following the sampling, the ramp is applied, here from the same side as the signal was originally sampled from. This moves node V_S from the signal value to GND, and hence moves V_E below GND by the value of the signal. As VRAMP climbs, V_E will eventually reach GND and the comparator will fire.

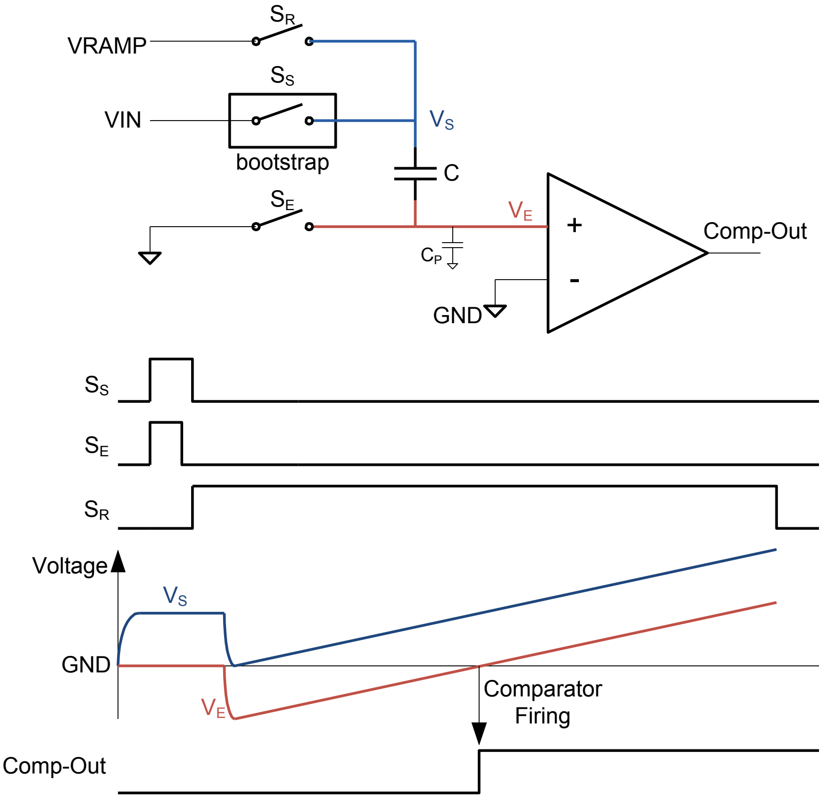


Figure 3.11 – Bottom-Plate-Ramping technique

The advantage of applying the ramp in this way is that the comparator will always fire when both its inputs are at GND, in other words the input voltages on the comparator at the time of firing are signal independent and always at the common-mode. This has the advantage that at the time of firing the internal DC conditions of the comparator is signal independent, meaning its open-loop-gain, bandwidth, delay, slew-rate is all signal independent. The comparator will always have a delay from inputs crossing one another, to the output firing, related to the analog characteristics of the comparator. The important advantage of bottom-plate-ramping is that this delay is signal independent.

When the switch controlled with S_R is closed, node V_E moves from GND to GND-VIN. This movement can only be achieved while charging the parasitic capacitances on node V_E, here marked as C_P. The charge required to charge this capacitor will come out of the main sampling capacitor C. Due to the type of bottom-plate-ramping circuitry used, at the time of firing V_E will have returned to GND, and all the charge previously taken from C to move this node will move back into C. This means the error due to the input capacitance of the comparator and bottom-plate of the sampling capacitor will be eliminated from the system.

Figure 3.12 shows the fully differential implementation of the sample and hold, and ramping circuitry with the comparator. As explained in the ramp generator section, the ramps produced by the main block require swapping at the end of every conversion cycle to maintain the same direction of ramping for consecutive conversions. This is done here with the addition of two switches. A new switch has been added which shorts the top and bottom of the capacitor prior to each conversion. This reduces the kick-back and more importantly signal dependent kickback, to the driver circuitry.

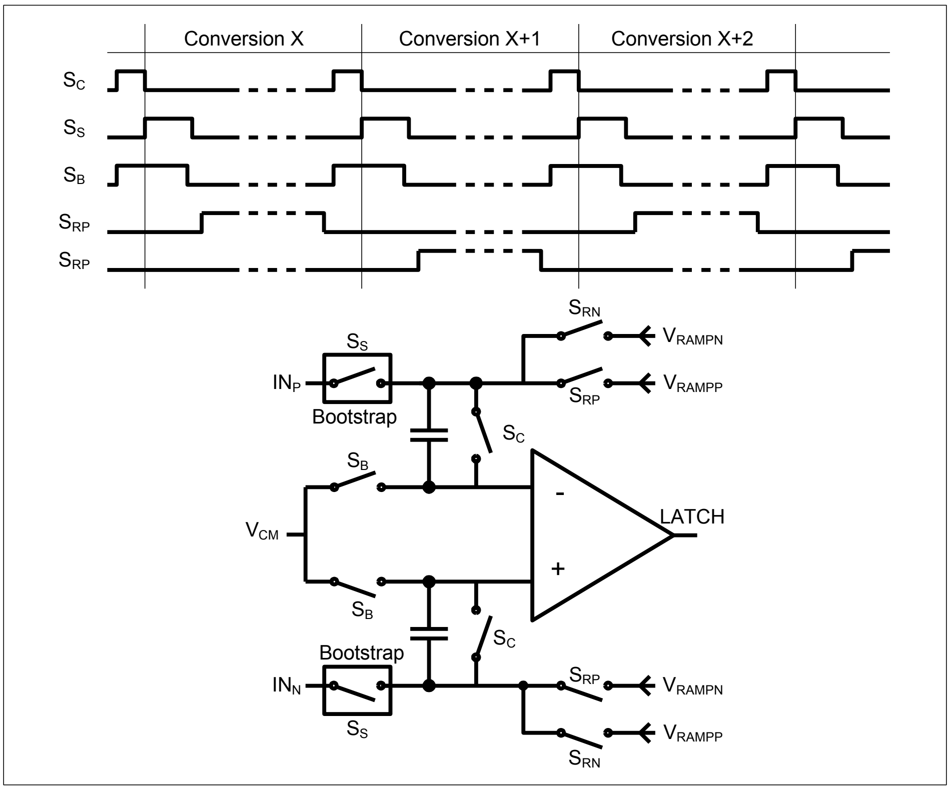


Figure 3.12 – Fully differential implementation of AFE

3.7.3 The requirements on the Comparator

There are two main requirements on the comparator, the speed in which the comparator reacts, and the noise of the comparator. Ideally the delay through the comparator should be less than the time taken for an LSB step on the ramp, and the output counter. If the delay is any larger than this an error will occur in the conversion. To achieve this small delay for a high performance converter, a large amount of current is required for each comparator, which may not be possible or desirable. As explained in the previous section, the delay through the comparator can be made signal independent using ramping techniques. If signal independent, this delay becomes an offset in the channel, as explained in Chapter 2, offset can be independently identified and calibrated for time-interleaving systems. Also as explained in Chapter 2, offset can limit the effective resolution of a data converter, due to loss of codes at the end of full-range, so despite the fact that the delay through the comparator is converted to signal independent offset which can be calibrated, care should be made to not lose too much of the effective resolution.

There is also another second-order effect that should also be considered when setting the gain-bandwidth of the comparator. In the previous section, the delay was claimed to be signal independent since the comparator always fires with its input at common-mode. One point which was ignored in the previous section was leakage on the capacitors. If this leakage in first order is assumed to be equal on both capacitors, the absolute magnitude of leakage is dependent on the length of time in conversion. The length of time for conversion is of course signal dependent, since for small signals the comparator fires early, capturing a small count value, while for large signals the comparator fires later capturing a large count value in memory. This signal dependent leakage manifests itself as common-mode movement, and common-mode error at the input of the comparator, so when considering second order effects, the comparator when firing does not strictly have its inputs at common-mode; a leakage dependent element can move this common-mode value. For a given capacitor size, this leakage and hence common-mode change, which results in signal dependent delay, can be calculated. This sets a hard-limit on the lowest value of gain-bandwidth and hence delay for the comparator.

Apart from power and size implications of choosing a gain-bandwidth higher then required, meeting the noise requirements can be harder with a higher bandwidth comparator, hence this bandwidth should be minimised. The comparator requires a noise figure matching the resolution of the whole converter. For frame-based communication applications, omitting flicker noise, the thermal noise of the comparator, is integrated over the bandwidth of the circuit. To design the comparator, the minimum bandwidth should be calculated, and the design should be improved to meet the noise requirement for this bandwidth. Band-limiting capacitors may be required to limit the bandwidth of the comparator, while larger (higher-gm) devices are used for noise reasons. It is key to understand that the reduction in bandwidth acceptable due to the use of bottom-plate-ramping in combination with offset calibration, heavily simplified the task of meeting the noise requirement, since bandwidth in which the noise should be integrated over is heavily reduced.

It is important to realise the comparator requirements for a TIC ADC are fundamentally different from that of other comparator based converters such as SAR and Flash ADCs. As explained in Chapter 1 and 2, the flash converter requires an individual comparator for each digital reference level. These comparators directly compare the un-sampled input signal to a pre-determined reference level. In the Flash ADC the comparators require a bandwidth greater than the input signal bandwidth to successfully compare the input signal to the reference levels. This is very different from the TIC ADC were the comparator bandwidth is de-coupled from the input signal bandwidth, since the input signal is sampled, and a ramp signal is effectively fed to the comparator.

SAR ADCs are also comparator-based converters, using a single comparator in conjunction with a DAC as part of a binary search algorithm. In the SAR ADC the input signal is also sampled, and the comparator does not directly require a bandwidth proportional to the input signal bandwidth, however in SAR ADCs a hard requirement on completion of comparison by the comparator and delay through the comparator exists. The comparator toggling is required to be completed within one clock cycle of the binary search algorithm, since this is the only time available before the DAC is adjusted to search for the next bit. However in the TIC ADC any delay from comparator input crossing to output toggling can be made input signal independent using the bottom plate ramping technique, and hence this delay only translated to signal independent offset in the conversion transformer function of the converter, which can be easily quantified, calibrated and removed. Meaning that for the SAR ADC, the comparator requires a much lower delay time, and hence greater GBW and power consumption compared to TIC ADC.

3.8 System Level Digital Counter

As explained earlier, a global counter is used to generate the count values for all sub-ADCs. Ideally the count value for each row should be out of sync by one code from its neighbour, but when using a global counter, this can be digitally corrected later. The global count sequence used here is a grey-code counter [57]. Figure 3.13 shows the motivations behind using a grey-code sequence.

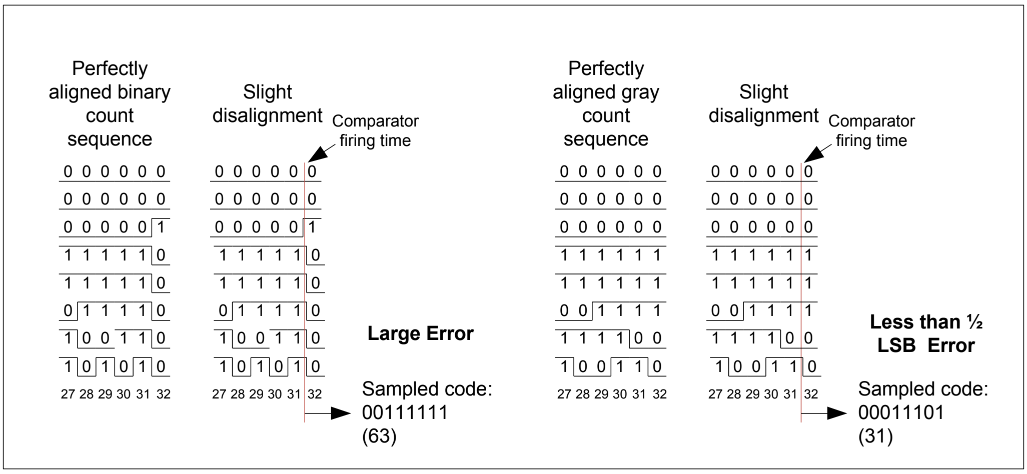


Figure 3.13 – Error at transition in binary code compared to gray code

If a normal binary sequence were to be used, there are times in the sequence that a large number of bits change value simultaneously. The arrival of bits of the count sequence to a particular row may not be perfectly aligned due to layout reasons, mismatch in devices or local power supply drops. If a sub-ADC fires exactly at a moment when a large code change is occurring, some bits of the new code could be combined with some bits of the previous code, since the code was half way through transition. This can result in a large error in the output code. When a binary code sequence is used, this mixing of bits between codes can result in a large error, while in gray code, in a sequence of numbers, no more than one code can change between sequential numbers, so due to misalignment no more than one code error can occur. Importantly, this problem only occurs when the comparator fires right at the point of code change, and in a way either code could represent the input analog value since the analog value is between the two codes, so using a gray-code sequence no real error or noise above an LSB level occurs even at transitions.

3.9 Backend Memory and Readout

In each ADC an N-bit memory unit is required, if N is the resolution of the ADC. The memory can be implemented as SRAM [58] or DRAM [50] unit cells. Both solutions can be used, an SRAM solution would consume more space at each ADC, and a tree (segmented) readout is required to buffer the SRAM outputs to drive the large capacitance on the output (from all the SRAM cells in parallel). A DRAM solution would be smaller per ADC, but would require a sense-amp based readout circuit, which could work with the large capacitance on the output. Both solutions are viable and produce overall similar results in terms of size and power consumption. It has been reported in literature that DRAM solutions tend to yield smaller overall circuits [50]. In this work the DRAM based solution is used, and the design trade-offs for which will be explained.

Figure 3.14 shows a unit DRAM. Each unit ADC would have N DRAM cells, and the output of each DRAM unit is in parallel with M other devices, when M is the number of channels. The output of the comparator is connected to the W signal, while the appropriate bit from the global count sequence is attached to the IN of the DRAM. The read signal for that row is connected to the R signal. The DRAM holds the value in the holding cap C_H. C_H can be removed if the parasitic capacitances in that node are sufficient for the hold. For the smallest input, the DRAM may require to hold its value for the full length of the conversion, this determines the size of the holding capacitor. When M_W is closed, the value on IN is stored on C_H. M_D is effectively a current pulling device, when R goes high, M_D has a path to discharge, and will pull the global bus line to ground. The Sense-Amplifier will detect this. Once this read cycle is done the line will be pulled up to VDD again, and it will be the turn of the next row to readout.

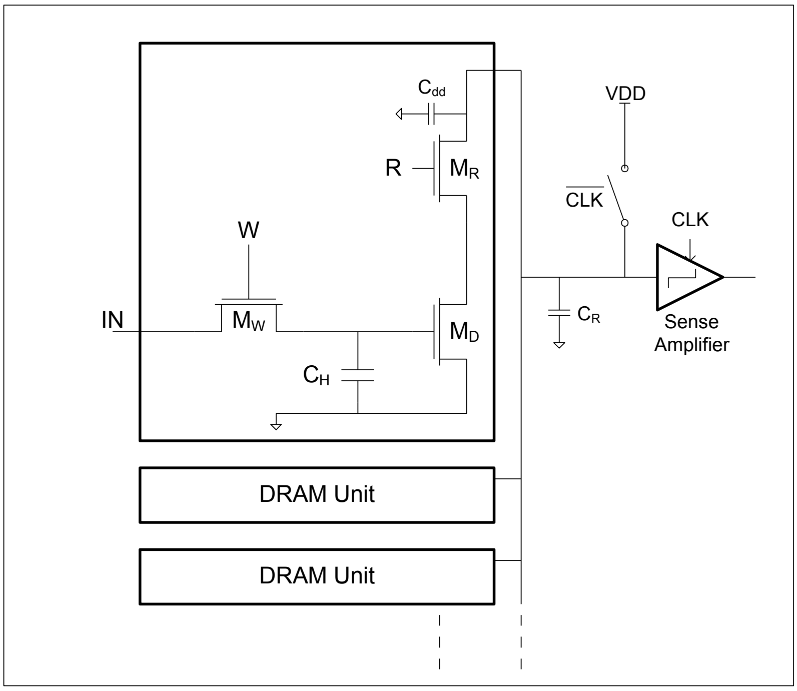


Figure 3.14 – Unit DRAM and readout circuit

The IN signal can be moving at a high-frequency, at the counter frequency, hence M_W should be sized in a way so this change can be seen on the holding cap. The combination of the M_W's R_{on} and C_H has a time constant. M_D determines the drive ability of the DRAM, and it should be made large enough to be able to pull the output node to ground within the read time. M_R may limit the speed in which the output node can move, if too small it can limit the peak current M_D can deliver, if too big it will add more capacitance to the output node and slow the system down. The output capacitance is the sum of all the C_{dd} of all the off DRAMs, the routing capacitance, and the input capacitance of the sense amplifier.

In practice there is a hard limit on how fast the DRAM can readout, since as M_D is increased, for speed, the value of C_H grows, and the input signal may not be able to operate at the required frequency. More importantly, to operate at a high-speed the size M_R must be increased, however this adds more capacitance and slows down the operation. To break the link between output capacitance and operation frequency a folding readout scheme has been proposed for TIC readout. Figure 3.15 shows the folding circuitry used.

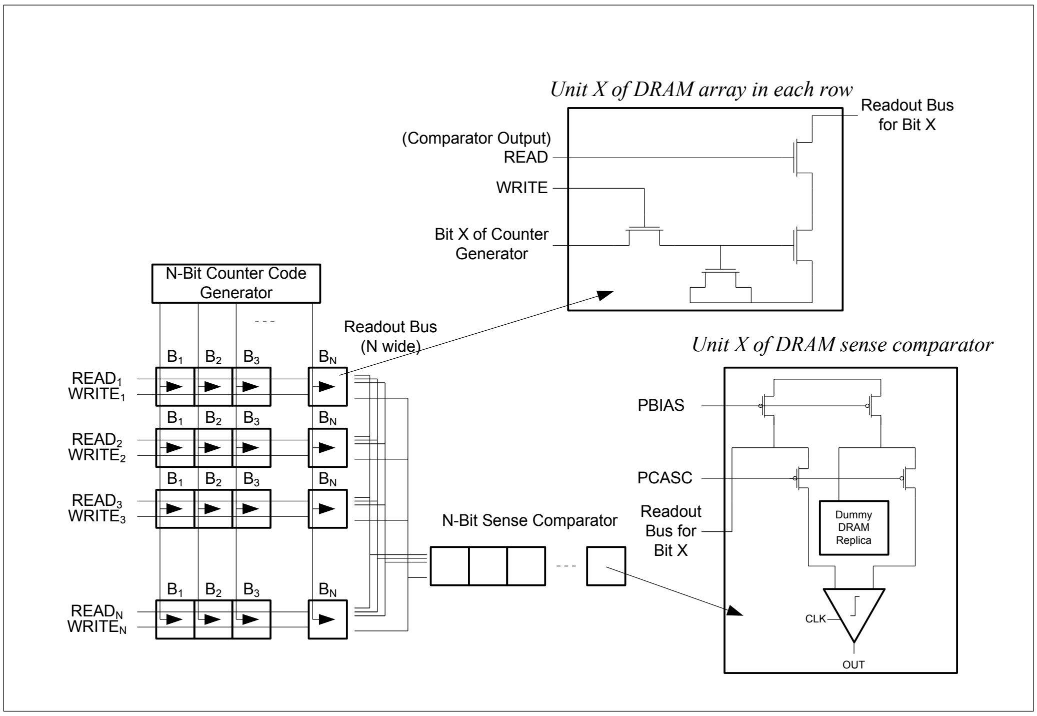


Figure 3.15 – DRAM unit, with readout circuit with folding

Here the sinking current each unit DRAM can produce at readout time, is effectively folded into a sourcing current using a current mirror and cascode device. The point of this is that the output bus of the DRAM cells, which has a lot of capacitance, does not require to move, while the other side of the current fold, below the cascode device is moved with the current. The magnitude of the current a DRAM cell produces is process and temperature dependent, hence a replica circuit is used for reference of the sense comparator. The folding current is designed to be bigger than the largest current a unit DRAM can produce, which is now much smaller since it only requires to move a node with a small capacitance compared to before.

3.10 Conclusions

The newly proposed Time-Interleaved Counter (TIC) ADC was proposed. Counter ADCs are commonly used in parallel in column parallel CMOS image sensors, where a large amount of data arrives in parallel, and the counter ADCs digitise the data in parallel. To adapt this architecture to process a sequential stream of data some modifications to the system is required. To enable such a time-interleaved system, an efficient global ramp-generator block is key. This work proposes such a block, made of a figure-of-8 folded resistor ring. This block is capable of generating many positive and negative going ramps out of phase from one another in an efficient way. Due to the sequential operation of the TIC ADC, re-configurability, trading of resolution for bandwidth is very practical. With the adjustment of clocks, effectively change in divider ratios, system sampling rate and effective resolution can be adjusted.

The other key part of the architecture is how the ramps are applied to each sub-ADC and the requirements on the comparator. The unconventional bandwidth of the comparator is key to the success of this architecture. The noise requirement on the comparator is the same as any of the other comparator based converters such as a flash or SAR ADC. However here, since the bandwidth of the comparator can be heavily reduced compared to the other architectures, this noise requirement can be meet with a much lower power consumption. The bandwidth and hence delay of the comparator is relaxed since the input signal to the comparator is a ramp, and any delay becomes offset which can be calibrated. This fact is key to this architectures success. In the next chapter a reconfigurable TIC ADC will be implemented.

Chapter 4 - Implementation of a TIC ADC

4.1 Introduction

In the previous chapter the Time Interleaving Counter (TIC) ADC was introduced. In this chapter the actual implementation of a re-configurable ADC in silicon will be discussed. The ADC realised can be configured to operate at 7-bit 1GS/s, 8-bit 500MS/s and 9-bit 250MS/s and was fabricated in 0.13µ standard CMOS process. The specifications were aimed for multi-standard wireline communication applications. This chapter first looks at defining the specifications, and from that and the technology available, choosing the internal clock frequencies of the system and number of rows, to meet the top-level specifications. Following this, each sub-section of the system, the ramp-generator, the AFE, the global counter, and the digital backend and readout will be specified and designed. Layout and shape constraints are a big part of the design of any block, in particular a highly time-interleaved ADC. For this reason prior to the discussion and design of each sub-block, top-level floor-planning and routing strategies will be discussed. The block is concluded with top-level assembly. To be able to test the chip, in an embedded mode of operation, some auxiliary blocks such as a clock receiver, digital backend packers and coder, digital control system and pad drivers are required. The design and implementation of these will be briefly discussed. In Chapter 5 the measurement results from the fabricated chip will be looked at.

4.2 Defining the specifications and clock frequencies

Flash converters cover sampling rates of 1GS/s and above, with usually resolutions limited to 6-bits. The architecture is popular for wide-band communication, with signal bandwidths in the 100s of MHz [13]. This resolution is limited to about 6-bits due to the exponential relationship between resolution and number of comparators and hence the limit on input capacitance. Extending this resolution has always been desirable to enable greater coverage for communication, however doing this can be difficult without increasing the input capacitance. Apart from re-configurability, it would be desirable if the TIC ADC could operate at sampling rates similar to the flash, but at a higher resolution. For this reason, the 1GHz, 7-bit performance range is targeted. This operation and resolution is desirable for wide-band communication systems with signal bands in the few 100s of MHz.

As previously explained, in the TIC ADC a 1-bit step in resolution can be traded off for doubling or halving of sampling rate. Most wireline standards are in the high 10s of MHz to 100s of MHz bandwidth space. For this reason, we aim to have the lowest resolution highest rate to be 1GHz, 7-bit, and work downwards to 500MHz, 8-bit, 250MHz, 9-bit, and potentially 125MHz 10-bit. This would cover most wireline standards, ITU based such as G.hn [59], and IEEE standards such as P1901 [60], and Private standards such as HomePlug and HomePlug AV [61], MediaXtream [13], and other closed wireline standards which operate up to 400MHz.

To fulfil this performance range, a number of different sampling clock frequencies, or number of rows can be used. As the sampling frequency is increased, for a given specification, the number of rows can be decreased. Increasing the clock frequency, increases the power-consumption of the digital backend, and also the analog front end, since each channel requires to operate faster, on the other hand the number of rows has decreased, which should reduce the power consumption. At first analysis, running the clocks slower, and increasing the number of channels, results in a more power efficient solution, on the other hand increasing the number of rows increases the overall area, and hence increase the parasitic routing capacitances. Apart from power consumption, there are requirements on input capacitance, and overall block size. The block will be implemented in a 0.13µ standard CMOS process. To be competitive compared to more conventional architectures, the block is required to maintain a size below 0.5mm^2. As previously explained, the flash maximum resolution is primarily limited by meeting the requirements for input capacitance. We aim to improve the input capacitance significantly with this architecture, and hence a 0.5pF input capacitance is targeted. This should enable any 50-ohm output characteristic driver to drive the block without bandwidth limitation for input signals up to the Nyquist for 1GS/s sampling.

The possible main clock frequency and number of row options, to meet the specifications proposed, are shown in Table 4.1. As a reminder here $F_{C\text{-}B}$ is the base clock for the ADC, where all the clocks passed to each converter is effectively a divided and selective pulse combination of this clock. $F_{C\text{-}C}$ is the clock frequency of the backend counter, and $F_{S\text{-}R}$ is the effective sampling rate of each ADC channel, M is the number of rows, $F_{S\text{-}TIC}$ is the overall converter sampling rate, which is effectively the same as $\mathrm{F}_{\mathrm{C\text{-}B}}$, and Res_{TIC} is the theoretical converter resolution possible with these clock settings.

Table 4.1 – Possible clock frequency and number of row options for specifications

	$\mathrm{F}_{\mathrm{C\text{-}B}}$	$\mathrm{F}_{\mathrm{C\text{-}C}}$	M	$\mathrm{F}_{\mathrm{S\text{-}R}}$	$\mathrm{F}_{\mathrm{S\text{-}TIC}}$	~$\mathrm{Res}_{\mathrm{TIC}}$
	Input	Input	Input	$\frac{F_{C-B}}{M}$	$\mathrm{F}_{\mathrm{C\text{-}B}}$	$\log_2\left(\frac{F_{C-C}}{F_{S-R}}\right)$
Option 1	1GHz	0.5GHz	256	~3.9MHz	1GS/s	7-bits
	500MHz	0.5GHz	256	~1.9MHz	500MS/s	8-bits
	250MHz	0.5GHz	256	~0.98MHz	250MS/s	9-bits
	125MHz	0.5GHz	256	~0.49MHz	125MS/s	10-bits
Option 2	1GHz	1GHz	128	~7.8MHz	1GS/s	7-bits
	500MHz	1GHz	128	~3.9MHz	500MS/s	8-bits
	250MHz	1GHz	128	~1.9MHz	250MS/s	9-bits
	125MHz	1GHz	128	~0.98MHz	125MS/s	10-bits
Option 3	1GHz	2GHz	64	~15.6MHz	1GS/s	7-bits
	500MHz	2GHz	64	~7.8MHz	500MS/s	8-bits
	250MHz	2GHz	64	~3.9MHz	250MS/s	9-bits
	125MHz	2GHz	64	~1.9MHz	125MS/s	10-bits
Option 4	1GHz	4GHz	32	~31.3MHz	1GS/s	7-bits
	500MHz	4GHz	32	~15.6MHz	500MS/s	8-bits
	250MHz	4GHz	32	~7.8MHz	250MS/s	9-bits
	125MHz	4GHz	32	~3.9MHz	125MS/s	10-bits

In the four options shown, Option 1 has the highest number of rows, while operating at the lowest backend clock frequency, while option 4 has the lowest number of rows with the highest backend clock frequency. In practice achieving clocking at 4GHz in 0.13µ technology was proven to be too difficult. Option 1 through an initial study showed to be the most power efficient solution for 0.13µ technology, however due to the large number or rows resulted in very large design (around 1mm^2) and maintaining the time-skew requirement between channels was not possible in this technology. Option 2 was found to not be as efficient for power compared to Option 1, in 0.13µ technology, consuming most of its power in the digital backend system, clocking at 1GHz, however meeting the requirement for size, input capacitance and timing skew with the extra power consumption in managing clock distribution. Option 3, had too large a power consumption in the digital, both front end and backend. Following this initial study by modelling, Option 2 was chosen as the implementation path. Option 2 is not as power efficient as an implementation of the same system as Option 1 for 0.13µ technology, however Option 1 would have had too high an input capacitance, resulting in the need for a input buffer, increasing the effective power consumption, and also the sampling clock skew over the full area would have not been achievable. It must be noted that as we move to smaller geometries, due to the improved performance of digital backend systems, the options with faster digital backend tend to be more power efficient, as well as having smaller input capacitance, and more manageable channel timing skew figures. This will be analysed further in the concluding chapter. Following the initial study, Option 2 was chosen as the 'clock' and 'number-of-channels' option to be implemented as part of this work.

4.3 Top-down Specification and Implementation

The main building blocks of the system have been explained in the previous chapter. In the previous section, the main clock frequencies of operation have been chosen. Here we will look at how these specifications feed-down to the specifications of the sub-sections of the system. In the implementation work here a top-down methodology was used, and that is reflected in the report. Knowing the top-level requirements, the specification for each sub-block as well as shape and layout size, and top-level routing for signals, clocks, and power are identified at the start. Here we will look at each of these requirements.

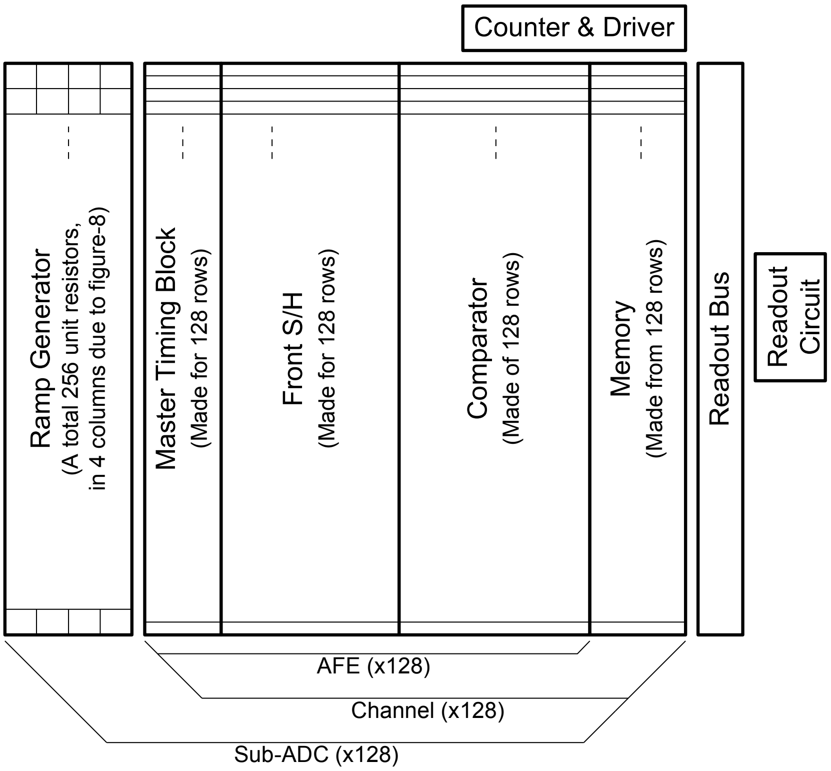


Figure 4.1 – Top-level building blocks, and layout floorplan

Figure 4.1 shows the main building blocks of the system, and how they can be realised in layout. The relative sizes of the block are accurate for layout, from initial study of relative size of sub-sections. A total of 128 sub-ADC rows are present. The global ramp generator connects to each ADC row. Each ADC row is controlled by a unit of a global master timing block, managing the timing of all the switches in that row, a front S/H circuit, a comparator, and a backend memory. In each row the S/H area is dominated by the sampling capacitor, and capacitor needed for the boot-strap circuitry. The global ramp generator generates a pair of ramps for each row. A global counter and driver circuitry is needed to generate and drive the grey code to all rows. The width of this bus will be 10-bits. The output of each row connects to a readout bus, which provides a path to a 10-bit readout (sense comparator) section.

The target total area is 0.5mm^2. A row pitch of 6.4µ is chosen, setting the total height of the block to 820µ, leaving 0.6µ for the width of the block, resulting in a near square shaped block. A row pitch of 6.4µ was chosen to allow space for the full signal routing per channel, including all ramps, references and control signals and digital outputs. Also allowing efficient use of space for building of metal capacitors in the row, without wasting too much space for shielding relative to the size of the capacitor. A section in this chapter is dedicated to the design of each sub-system of the ADC; the Global Ramp Generator, the Sample and Hold front-end, the Master Timing Block, the Comparator, the Backend Memory, the Global Count Generator and Backend Memory Readout Circuitry.

4.4 Global Ramp Generator

4.4.1 Design of a Variable Resolution Global Ramp Generator

In the previous chapter the figure-of-8 rotating resistor ring was introduced. Figure 4.2 shows this circuit again. On the left the original circuit with labels for the switches is shown. It can be seen, that each timed signal is used in two places, and only one signal should be high at any moment. The switch control can be implemented using a ring of latches. This is shown on the right hand side of Figure 4.2. The coloured squares with arrows in them are clocked latches. All latches have the same clock, when in reset all latches are reset to zero, except the 1st latch which is reset to 1. As the latch ring is clocked, the 1 will move forward down the ring through the 4 colours, and will switch on different switches as it rotates. Each one will tie one point of the resistor ring to the positive reference and the opposite point to the negative reference.

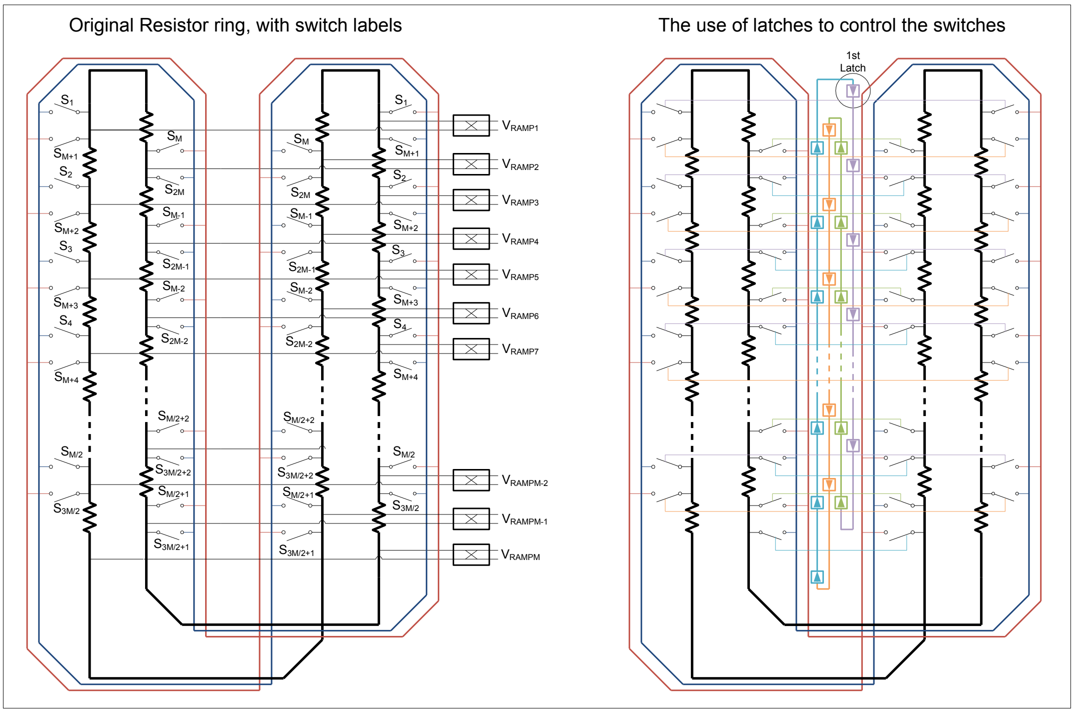


Figure 4.2 – Figure-of-8 rotating resistor ring, with latch timing circuitry

Based on the basic description of the resistor ladder, the ladder requires to operate at the counter clock frequency, since each code step in the backend ADC counter should be matched with a voltage step in the ramp. For the ADC implemented here, a total of 1024 codes can be used in the 10-bit operation mode, meaning 2048 unit resistors, and 2048 latches are required. The clock driver circuitry for 2048 latches proved to be very power inefficient. Apart from the number of unit resistors, there are a couple of other variables in the system:

- The total resistance of the ring (R x M), determined the unit resistor size R, and the number of units, M
- The size of the switches which contribute an on resistance in the path of the positive and negative reference voltages, and a parasitic capacitance to all other places in the ring.

The RC time constant found is a combination of the capacitance due to the switches and load capacitance on the reference lines, and the effective R of the ring at each point. The effective R of the ring, is dependent on the voltage of the ramp. If the ramp is at a voltage near the minimum or maximum point this R can be very small, near a unit resistor, while if the ramp is at mid-code, this R can be equal to the total ring resistance divided by 2. Figure 4.3 has simplified the ring as a single ring, (without the figure-of-8 fold) with only 16 elements, showing one moment in time, and the effective Rout for the different ramp outputs. It also shows the one example ramp from a 32-part resistor ring. This shows an important fact, that since the output impedance effectively changes, the smoothing and delay characteristics change through the voltage range of the ramp, and in fact if the load capacitance is too high this can result in major non-linearity in the ramp, where the mid-sections of the ramp will be subject to delays.

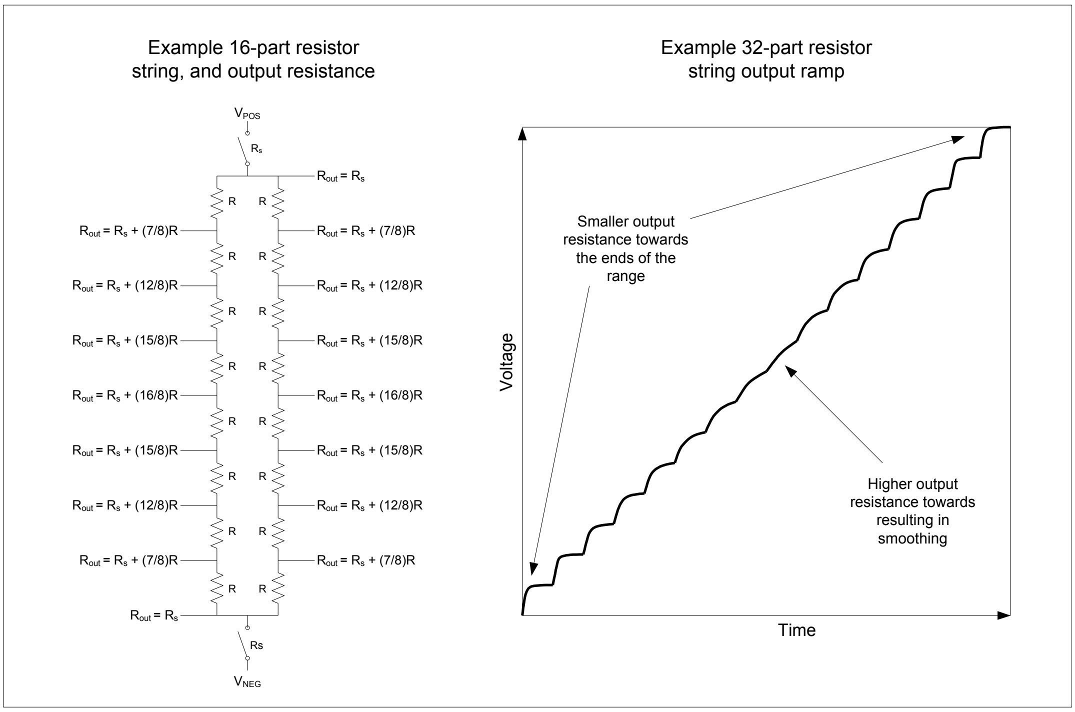

Example 16-part resistor string, and output resistance
Example 32-part resistor string output ramp
V_POS
R_s
R_out = R_s
R
R_out = R_s + (7/8)R
R_out = R_s + (12/8)R
R_out = R_s + (15/8)R
R_out = R_s + (16/8)R
R_out = R_s + (15/8)R
R_out = R_s + (12/8)R
R_out = R_s + (7/8)R
R_out = R_s
Rs
V_NEG
Smaller output resistance towards the ends of the range
Higher output resistance towards resulting in smoothing
Voltage
Time

Figure 4.3 – Example of output resistance effect on output ramp

If the smoothing effect was consistent throughout the ramp, the effect would only appear as an offset error due to the delay, however here the smoothing effect is different at different parts of the ramp. This is due to the output resistance being different and different points. Figure 4.4 shows the effect of the increasing the parasitic capacitance on each node.

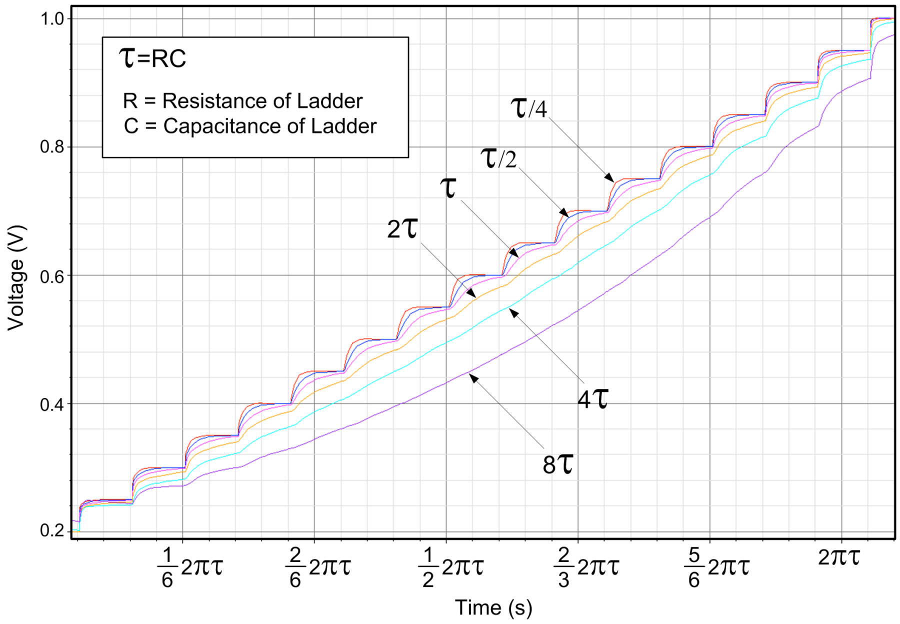


Figure 4.4 – The effect of different RCs on ramp linearity

Since each ramp step no longer meets the settling requirements, the errors accumulate. This does not result in much DNL (code to code non-linearity), however can result in very poor INL (over-all curvature of the transfer function). The overall error is related to the total capacitance on the ring, and effective difference in output resistance at the top and bottom points of the ladder vs. the mid point.

Since the contribution of R_{on} of a switch is a constant for all R_{out} values, referring to Figure 4.3, hence as R_{on} is increased, the mismatch between top and mid-point of the ramp is decreased. Further more the reduction of R_{on} is effectively reducing the size of the switches, which also reduces the capacitance. However very small switches cannot be used for two reasons:

1. The voltage drop across the switches is effectively lost voltage from the range, for example if R_{on} of the switches at the top and bottom of the

ring are each equal to the resistance of the whole ring, then the largest signal input range possible will be 1/3 of the supply range, since the largest ramp range possible would be 1/3 of the supply range.

2. The second reason is due to mismatch, the Ron of the switches is subject to mismatch. The positive and negative references are held constant, but as the switch sizes are decreased, the mismatch between the switches is increased, and also since the switch now takes a larger part of this voltage range the effect of mismatch on the ramp accuracy will be increased.

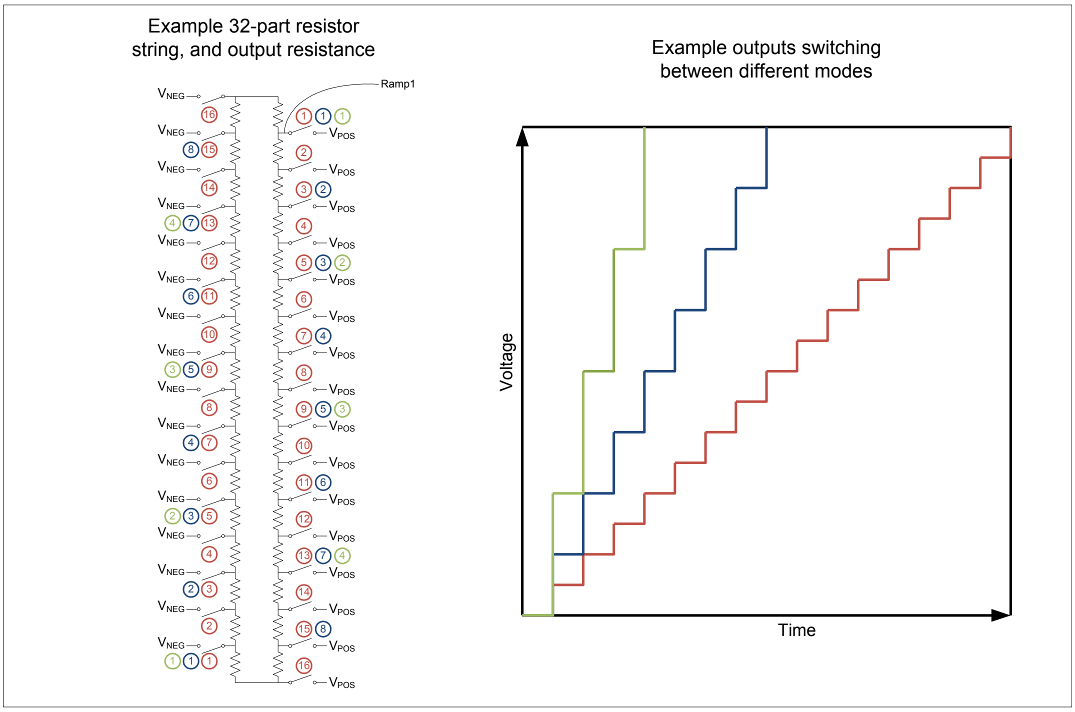
In practice leaving enough of the supply range for signal operation becomes the dominant factor in choosing the size of the switches in relation to the unit resistors. Here we design so that the sum of the top and bottom switch resistance is equal to the ring resistance, leaving half the supply range for the signal. This is important in deep-sub-micron design. It is interesting to note that this sets a hard relationship between the Ron of the switches and the total resistance of the ring. The linearity of the ring, related to the relative resistance difference in the ring and total capacitance, cannot be improved, for a given ramping speed, by reducing the resistance of the ring, since for example if the unit resistor is halved, halving the total ring resistance, the switches require doubling in size to maintain the ratio of the total ring resistance to R_{on}, which would then double the capacitance on the ladder and hence maintain the same non-linearity profile. For linearity reasons, the resistance of the ring should be reduced up to a point where the capacitance of the switches dominates the total capacitance of the ring, compared to other routing and parasitic resistances. Beyond that reducing the ring resistance only increases the power consumption, and will have no positive effect on linearity. To improve linearity, once can only choose to ramp slower, allowing more settling time, but this in effect reduces the sampling rate of each sub-ADC, and hence the overall converter ratio. This in effect sets a hard ratio between signal swing, resolution and operation speed, for a given technology.

Once this relationship is set, the total ring resistance can be broken into the required units for different system resolutions. We are looking to building a re-configurable ADC with 128-channel, operational in 7-bit 1GS/s, 8-bit 500MS/s, 9-bit 250MS/s and 10-bit 125MS/s. In these modes of operation, the ramps should have inherent linearity matching the resolution, and should cover the full signal range in 128ns, 256ns, 512ns and 1024ns, while in each mode the ramps should have 128-steps, 256-steps, 512-steps and 1024-steps.

Focusing on individual modes of operation, for example for a 7-bit system, the total resistance is broken into 128 units, and 128 switch pairs are used. For the 8-bit system the total resistance is broken in 256 units, with 256 switch pairs. It is important to realise, that the unit switch sizes between the two systems is the same, determined by the R_{on} to total ring resistance ratio, while the 8-bit system has twice as many switches compared to the 7-bit system. This results in twice as much capacitance on the ring. If we assume the 7-bit system, in this technology was meeting the 7-bit linearity, moving to the 8-bit system, the capacitance has doubled, so the ramp needs to operate at half the rate of the 7-bit system, to meet the same linearity, but we require further linearity in the ramp in the 8-bit mode compared to the 7-bit mode, so the ramp requires slowing down by another factor of 2. In effect comparing a 7-bit 128 unit ring, in isolation to an 8-bit 256 unit ring, the ramp has to be operated 4 times slower to meet the resolution. Operating at 2 times slower to allow better settling, and a further 2 times slower to accommodate for the extra capacitance due to double of the switches. This is a worrying conclusion, since we were looking to trade 1-bit for halving of sampling rate.

Assuming the 7-bit system to start with had much greater inherent linearity than 7-bits or even 8-bits (meaning it was over designed for settling), then moving to the 8-bit system, halving the ramp rate by one would accommodate the doubling in capacitance, and the 8-bit system would have the same inherent linearity of the 7-bit 128-element ring, which we claimed was as good as 8-bits in the first place. Of course this inherent linearity is technology dependent, not one that we can choose. Assuming we have a technology which enables this, we're looking to build a re-configurable ADC, which only has one global ramp generator, not one for each configuration mode, meaning that the resistor ring requires breaking into units equal to the highest possible resolution to enable the high-resolution operation, and when we require to lower the resolution, and have a high-speed ramp, we can just choose the skip a certain number of switches. For example if 1024 units are present, and we require faster ramps, say for the 7-bit modes, we can either clock the whole system 8 times faster, or just jump to every 8th switch. Note that in lower resolution modes, the extra capacitance of the higher resolution mode (extra switches) is still present, however we require clocking at a higher rate. This can be possible if an independent 128-element (with 128 switch pair) ring, clocked at 1GHz used for the 7-bit mode, had a much greater inherent linearity than 7-bits, in fact it would require to be linear to 10-bits, so that when we add the extra switches taking it up to 1024 switches it still meets the linearity required at 7-bit mode, and when the ramp-time requires doubling for 8-bit mode, by applying the clock to more switches (reducing the step size), the linearity of the system is increased to 8-bits (since the capacitance is un-changed, but settling time has doubled). Figure 4.5 shows how such a system would work, however as explained can only be realised, if the technology can meet the specification of linearity.

Unfortunately, such a system cannot be realised in the technology chosen. A 128-element resistor ring, normally used for a 7-bit system, was implemented, and clocked at 1GHz, building ramps which complete within 128ns. The inherent linearity of this ramp was only found to be 7.8-bits, and not as high as 10-bit as required by the proposed system. This cannot be increased by increasing the current in the ring as previously explained, as it is limited by the technology, and the relationship between Cdd and Ron. Meaning the doubling of resolution for halving of sampling rate cannot be achieved up to 10-bit resolution in this technology, by building a re-configurable rotating resistor ring with variable clocking rate, or unit skipping. It was found to extended the inherent linearity of the 128-element ring to 10-bits, to allow the re-programmability, the 7-bit ramp should be clocked at 200MHz. This would result in a major compromise to the overall specifications of the converter.



Example 32-part resistor string, and output resistance
Ramp1
VNEG
VPOS
Example outputs switching between different modes
Voltage
Time

Figure 4.5 – Example of reconfigurable resistor ring with switch skipping

It is important to note that a 128-element 1GHz ramp generator, resulting in ramp time of 128ns, can be realised with an inherent linearity of 7-bits, and is possible with this architecture in this technology. The realisation of a 7-bit 1GS/s converter in isolation, without reconfigurability, is possible in the technology. However due to the addition of the non-used switches for reconfigurability to higher resolutions, the 7-bit mode would no longer meet the specification, and in general, even in isolation, a 256-element, 512-element or 1024-element ramp generator, clocked at 1GHz, resulting in ramp times of 256ns, 512ns, and 1024ns, cannot be realised to the linearity levels of 8-bits, 9-bits and 10-bits, in this technology, using this architecture. This is primarily due to the exponential growth in number of switches required for high resolution ramps, loading the ring and limiting the settling time.

Looking back at the basic 128-element ring, clocked at 1GHz, the ramp generator produces a ramp in 128ns, with an inherent linearity of 7-bits. If we choose to clock the 128-element ring at 500MHz, this would result in a 256ns ramp, which we require for the 8-bit system. Since each step has twice as much time to settle, and the capacitance on the ring has not changed, the linearity of this ramp will extend the 8-bits. The problem here is that although the linearity is extended to 8-bits, the ramp only has 128 unique steps, in effect the quantisation noise of the ramp is still at 7-bits, slowing the system down has increased the linearity, but without the introduction of new step levels. This quantisation noise is of course saw-tooth shaped, and can be heavily reduced by filtering, without affecting the linearity of the ramp. Figure 4.6 shows how applying filtering to different resolution ramps can result in similar ramps.

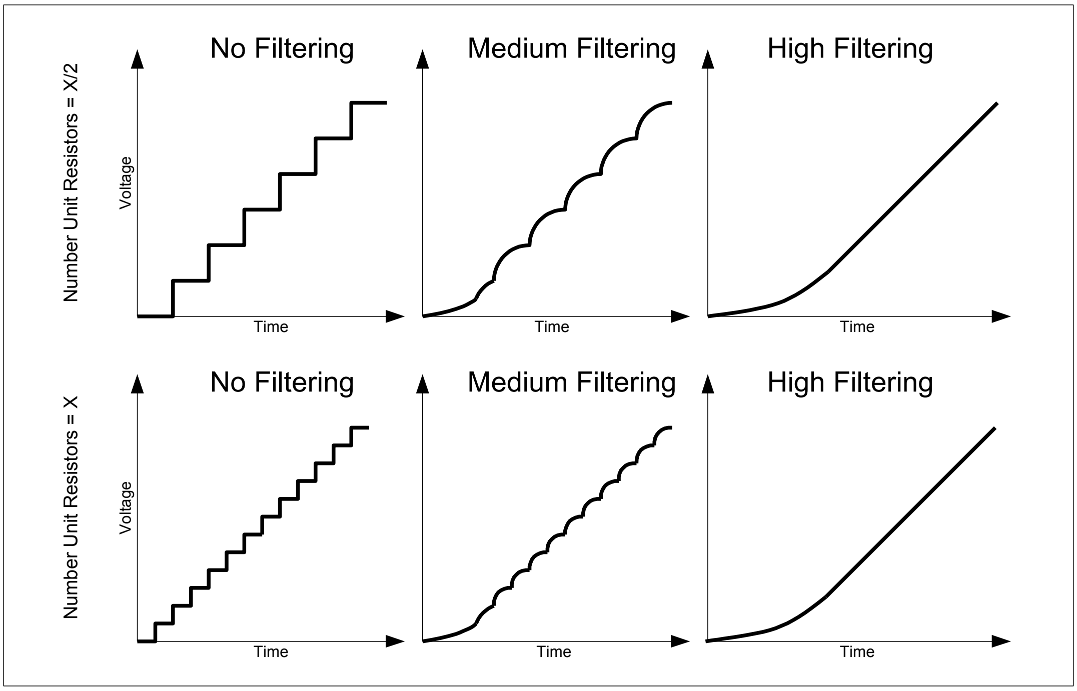


Figure 4.6 – Different ramp resolutions, and the effect of filtering

For example, the ramp produced by 128 levels, which has 8-bits or greater effective linearity, can be used in an 8-bit system following some filtering, and in fact figure 4.6 shows how there is almost no difference between a ramp with more quantisation levels compared to one with less after filtering. The act of filtering does add a greater lag at the start of the ramp, this has two effects, the delay in the ramp which effectively produces an offset, and also the loss (or error) at the start of the ramp. Figure 4.7 shows these effects comparing a ramp with no filtering to one with filtering applied, relative to the digital code count at the back. The first 5 codes represent a small range of the input signal, all effectively within 1 LSB. These are effective lost codes from the resolution space. Also some minor non-linearity can be seen for a few codes.

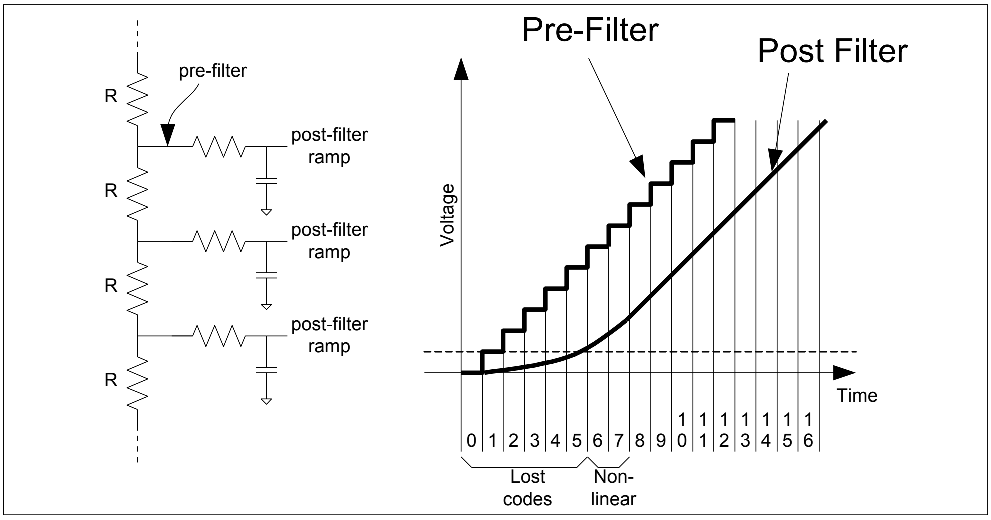


Figure 4.7 – The effect of filtering on ramp accuracy

Different filters are required for the different modes of operation, and hence the resistor in the interpolating filter has been made programmable to be adjusted for 8-bit, 9-bit and 10-bit modes of operation. For 7-bit mode no filtering is required. It was found that with basic interpolation filter, the output of the 128-level resistor ring could not be made linear enough for the 10-bit operation without dismissing a large amount of the signal range. The 10-bit operation mode remains in the system, however performance of 10-bit linearity is not expected when operating at 125MS/s. To increase the resolution of the output ramp, the clock for the ramp-generator is divided, and the filter frequency is moved down by a factor of 2.

To conclude, the global ramp generator is an integral part of the TIC ADC. The timing of the rotating switches is managed by a shift-register architecture. Ideally one would like to unitise the resistor ring to the highest resolution of the system, however in practice this grows the number of switches, and hence parasitic capacitance on the ring exponentially, quickly limiting how fast the ring can be rotated while maintaining the required linearity due to settling requirements. To overcome this problem, a smaller number of units are used, 128 here, and to produce slower ramps with high-resolutions, the rotating clock frequency is reduced and interpolating resistors are place in series with the output ramps to remove the quantisation steps on the ramp. This results in a small loss of resolution, due the loss of the first few codes in each mode of operation.

4.4.2 Top-Level Implementation of Global Ramp Generator

Figure 4.8 shows the top-level building blocks for the programmable global ramp generator. From left to right, the block receives the reference clock, at 1GHz, 500MHz or 250MHz. This clock is then treed and passed to an array of non-overlapping clock generators. These are needed since to control the switching (rotating) of the resistor-ring, rather than d-types, custom latches are used. These custom latches are arranged as four columns of 64 elements, generating 4 sets of 64-bit wide signals. At each moment in time only one of these 256 signals is high. These signals are connected to 4 columns of 64 switch pairs, which connect appropriate points of the resistor ring to either the Positive reference voltage or the Negative reference voltage. Each of the 256 control signals connects exactly one switch to the positive reference, and one switch to the negative reference. These two switches will be at opposite ends of the resistor ring, but because of the hold in the resistor ring, these points are always next to each other.

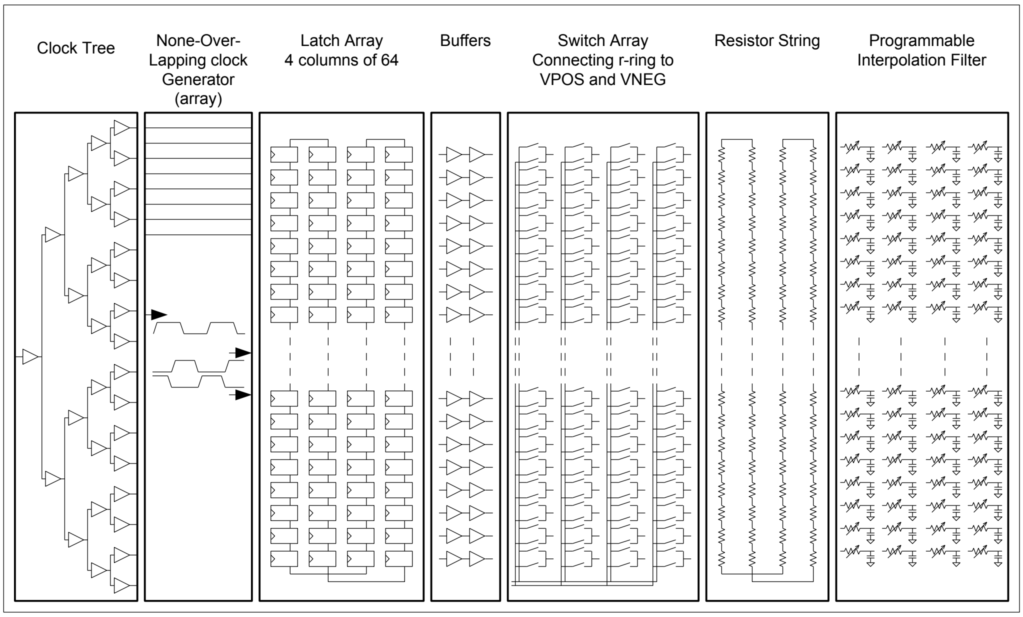

Clock Tree
None-Over-Lapping clock Generator (array)
Latch Array 4 columns of 64
Buffers
Switch Array Connecting r-ring to VPOS and VNEG
Resistor String
Programmable Interpolation Filter

Figure 4.8 - Top level diagram of rotating resistor ring system

The actual resistor ring is constructed from 4 columns of 64 unit resistors. The top of each resistor is connected to the driving switches to the left, and also connected to programmable interpolation filters to the right. The global ramp generator produces 256 different ramp voltages.

4.4.3 Custom Latch for Timing of Resistor Ring

Figure 4.9 shows the circuit diagram of the custom latch. This latch is a semi-dynamic structure for logic 0. It requires two clock signals which should be non-over-lapping to avoid transparency failure. Since device M1 will have a V_{gs} drop when passing high voltages of D, device M4 is used to pull this voltage up to VDD if it is higher than the V_t of the device.

Devices M2, M3, M8 and M9 are used at reset to set the initial state of the latch. As previous discussed, at initialisation a latch may require the value 1 or 0 depending on its position in the ring. In normal operation SET0 and SET1 are low. To reset the latch into the value of 0, SET0 should be taken high, and to reset the last into value 1, SET1 should be taken high. The appropriate bar signal should be managed accordingly.

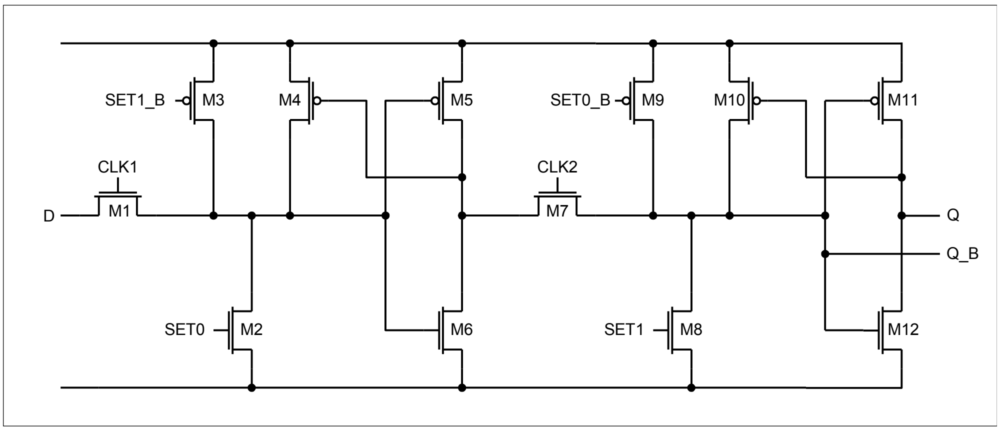


Figure 4.9 - Circuit diagram of custom latch

The layout of the latch is shown in Figure 4.10. Only metal 1 was used for the layout to allow routing above the device. The pitch of the latch is 1.6μ, and is done in a way to allow simple arraying of the cell. The cell can be folded and placed at the purple boundary line, allowing lining of contacts without the generation of any DRC errors.

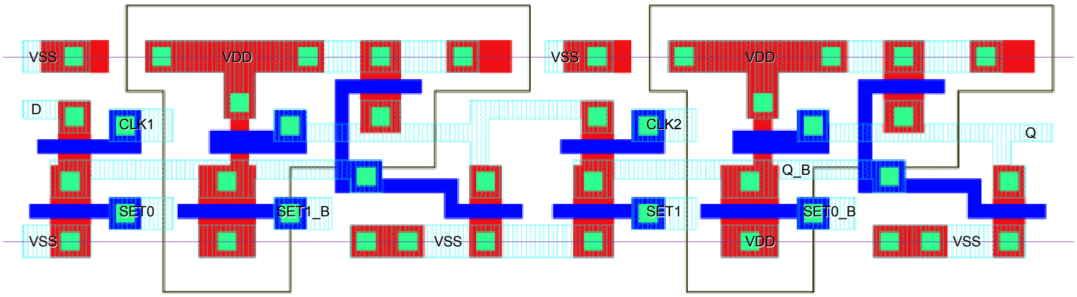
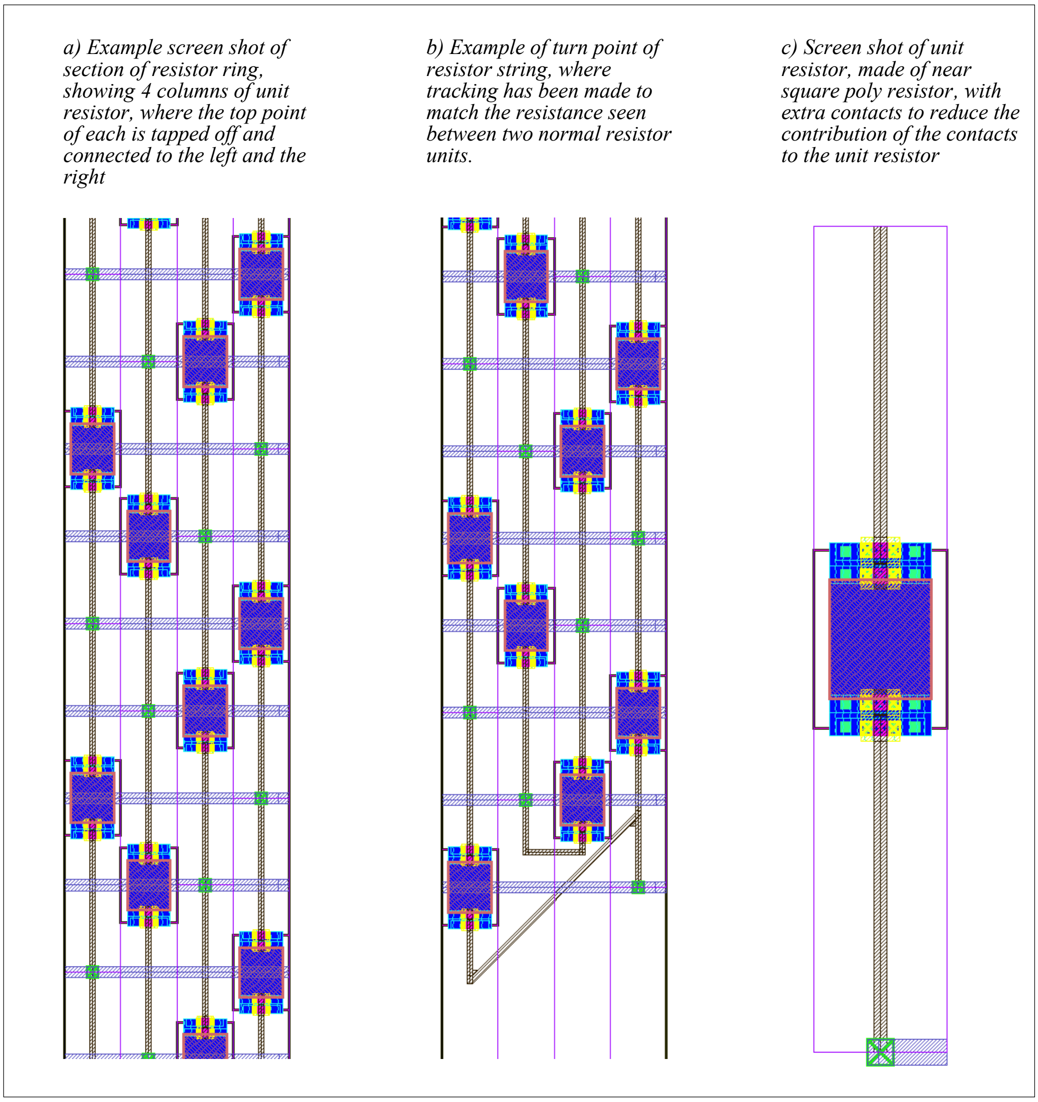


Figure 4.10 - Layout of custom latch

4.4.4 Unit Resistor and Resistor Ring

Using the design explanation presented earlier, unit resistors of about 9-ohms are required in the ring. A total of 4 columns of 64 unit resistors should be arranged in a ring. One approach for optimum layout would to build the resistor ring as a continuous strip of poly, only tapping off at intervals required to obtain the required resistance. The problem with this method is the turn-around-points. For electro-migration rules determine the minimum width of the poly, which hence sets a minimum distance the turn-around-points require to travel. It is critical that the tracking resistance of the turn around points is exactly matching the tracking resistance between two unit resistors. A unit resistor comprising of a new square poly resistor unit, and metal tracking was developed. Due to this structure, the turn around points can be easily realised using the metal already present in the unit resistor, maintaining uniformity throughout the ring. Figure 4.11 shows the unit resistor designed, and how it was used in the resistor ring. Figure 4.11-b shows how turn around points were done to maintain matching of unit resistors.



a) Example screen shot of section of resistor ring, showing 4 columns of unit resistor, where the top point of each is tapped off and connected to the left and the right

b) Example of turn point of resistor string, where tracking has been made to match the resistance seen between two normal resistor units.

c) Screen shot of unit resistor, made of near square poly resistor, with extra contacts to reduce the contribution of the contacts to the unit resistor

Figure 4.11 - Examples of unit resistor and connection in different parts of the ring

4.4.5 Programmable Interpolation Filter

The programmable interpolation filter comprises of a programmable resistor that in combination with the parasitic capacitance driven by the ramp performs a filtering function. In this process high resistance poly resistors are available with a resistance close to 1kOhm/square. The programmable resister used has a maximum resistance of 100kOhms, programmable in steps of 12.5kOhms. In can be seen in simulation that a 100kOhm resistor is not required in any of the modes of operation, but was implemented to allow experimentation on the chip after manufacturing.

4.5 Sample and Hold and Ramp Front End

Figure 4.12 shows the master timing diagram of the sample-hold and ramp front end. It is important to point out that a front sample and hold amplifier is not present in the implemented TIC ADC. Bootstrap switches are used to maintain series resistance and hence RC profile independent of signal amplitude.

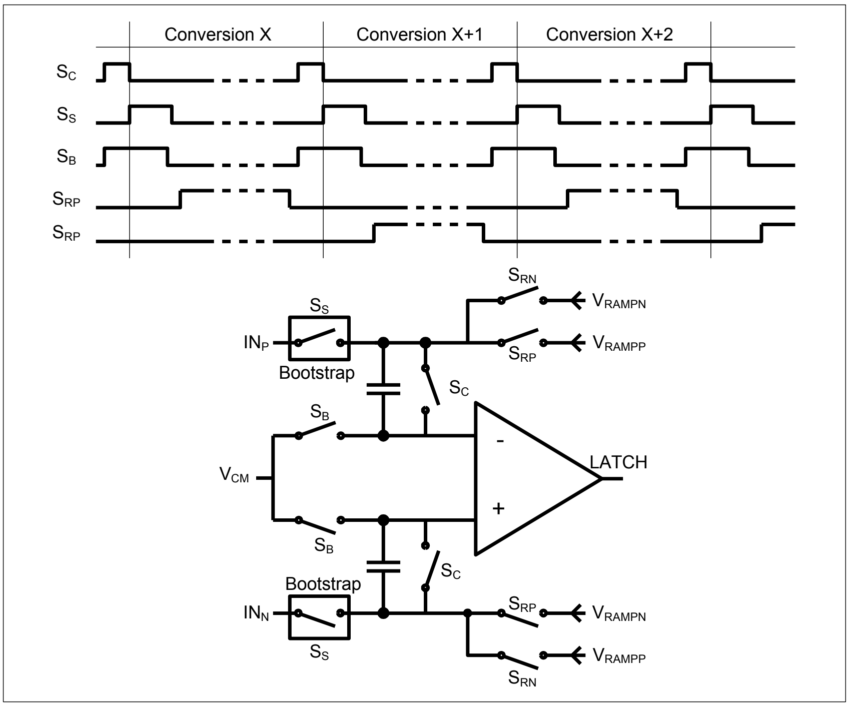


Figure 4.12 - Sample and hold, and ramp front end

The design cycle for the sizing of the switches and capacitors in the sample-and-hold front end, begins with the capacitors. kT/C sampling noise puts a hard limit on the lowest possible size for the sampling capacitors. In this design programmable sampling capacitors with a maximum value of 480fF are used, to allow for 10-bit operation, however in normal modes of operation this maximum value of capacitance is disabled, since 10-bit mode of operation was not an achievable specification for the ramp generator. These capacitors are constructed from custom metal fringe capacitors, using the higher metal layers to minimise parasitic capacitance to ground.

Following initial sizing of the sampling capacitors, the series on-resistance of switches S_S and S_B are the next most important design element. The choice of Ron and hence device size of these devices, in first order is driven by two constraints:

- Firstly, the combination of this on-resistance and the size of the sampling capacitor, produces a settling time requirement on the front end of the circuit. This was explained in Chapter 2, and based on the resolution the sample requires to settle to a certain number of time-constants.
- Secondly, inevitably some front-end bandwidth mismatch between the channels will exist, and as discussed in Chapter 2, based on the matching possible between channels, the bandwidth of the front-end circuit is required to be made a certain factor greater than the largest signal component to ensure despite the mismatch in bandwidth, the error due to the phase change in different channels is below the requirements of the system. This bandwidth figure is the combination of the sampling capacitor, the on-resistance of the switches, all parasitic resistance and capacitance, and the output resistance of the signal driver.

As custom capacitors were designed, no exact matching data for this process was available for them. Using information published in literature [62] some estimates were made of the expected capacitor matching. It is important to note that the higher resolution performance (9-bit mode) has the hardest specification for the channel bandwidth matching, while lower resolution mode (7-bit mode) has the hardest specification for settling time, due to the higher operational frequency. In this design, meeting the settling requirement was found to be the dominant source of error, and set the lower limit on on-resistance of the switch. This is primarily due to the fact that the bandwidth of the path with the switch is dominated by parasitic capacitance and output resistance of the driver, which is common for all channels. Also referring back to Chapter 2, the shape of the curves from bandwidth mismatch to SNR drop and remain at a given resolution for a given mismatch, meaning the dominant method of reducing this noise contribution is the improvement of matching rather than increase in bandwidth. The switches were designed for an on resistance of 40-ohms to meet all timing requirements.

In the default configuration of the TIC, each channel uses one clock cycle for sampling the input, and the remainder of the time for conversion. As explained in Chapter 2, in theory each stage can use a longer stretch of time for sampling at the expense of conversion time. Here this will in effect reduce the number of possible codes, and hence theoretical resolution of the converter, however will improve the settling time requirement. This choice of timing was made programmable in the implementation to allow for adjustment for better understanding of the limitations of the circuit after fabrication.

There is also a further secondary constraint on the size of the switches, primarily the bottom plate switch. As will be explained in the following section, meeting timing skew requirement between channels can be challenging for such a large number of channels. The timing skew error is dominated by mismatch in devices in the clock path, but also the switch used for sampling. Commonly the opening of the boot-strap switch acts as the sampling element in S/H circuits, and in time-interleaving systems a lot of work can be seen on attempting to improve the timing skew of boot-strap switches [9]. Here since bottom-plate-switching is used, the opening of this switch, driven by S_B, determines the moment of opening, so some care should be made firstly to achieve the required matching in the device itself, but without increasing its size unnecessarily to then require further buffering for the clock driver which inevitably will increase clock skew. Achieving the right balance of sizing, while maintain the required Ron is key to efficient design and operation of the circuit.

As explained in the previous chapter, bootstrap switches are commonly used in S/H circuits, to minimise signal dependent settling and help with matching of the positive and negative input of differential circuits. Many different bootstrap circuits are available and are used. A popular architecture is the one first proposed by Abo and Gray [56], however due to the number of capacitors and device arrangement was dismissed here due to its difficult layout in the pitch available. With the small row pitch, the number of different devices, particularly in different well voltages, and total number capacitors with required spacing, have great implications for size. In time-interleaving system some designs make modifications to the boot-strap switch to reduce the timing skew [18], however here the sampling takes place by the bottom place switch hence such modifications are not required. In this implementation a slightly modified version of [63] was used which has been shown in Figure 4.13.

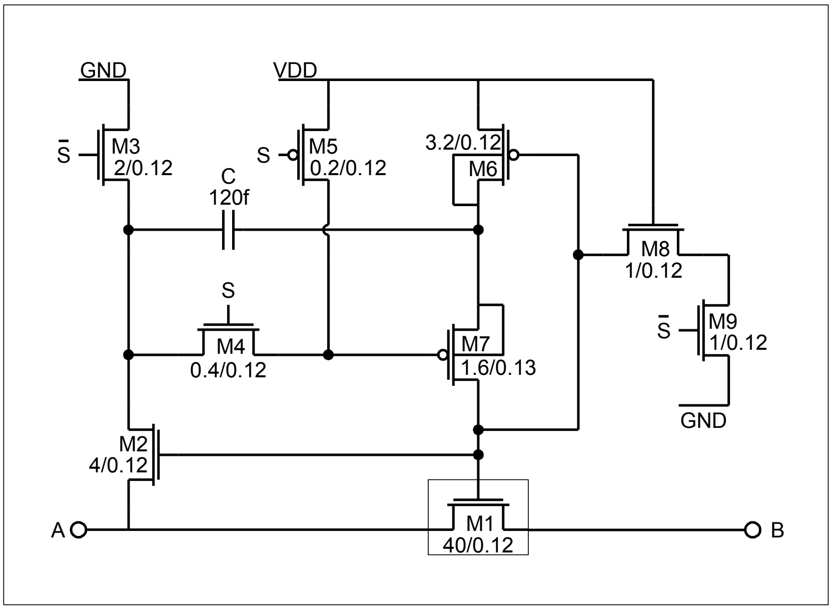


Figure 4.13 - Boot-strap switch circuit diagram

This architecture was chosen due to its smaller device count and number of capacitors. Device M1 should be sized to meet the ron requirements of the sample and hold circuit, primarily settling and phase requirements as discussed in the previous two chapters. Capacitor C is first charged to VDD, and at the time of sampling is placed across the Vgs of M1. This Vgs is required to be signal independent to achieve signal independent ron for the switch to meet the linearity requirements of the sample and hold. Capacitor C is charged to VDD through devices M3 and M6. These devices should be sized to achieve the charging of the capacitor to the required accuracy in the 0.5ns time available. Following this charging phase, the capacitor is place across the Vgs of M1 via devices M2 and M7. These devices on one had require the settling time requirement for the fast operation of the circuit, but also contribute to parasitic capacitance. The bootstrap capacitor must be chosen in a way, that the input signal dependent loss of charge, due to charging of parasitic capacitances in the circuit, still maintains a Vgs sufficiently input signal independent to maintain the linearity required for the system. Here a bootstrap capacitor of 120fF was used in the boot-strap switch, and the devices M2 and M7 were sized to allow for settling of the boot-strap voltage across the Vgs of M1.

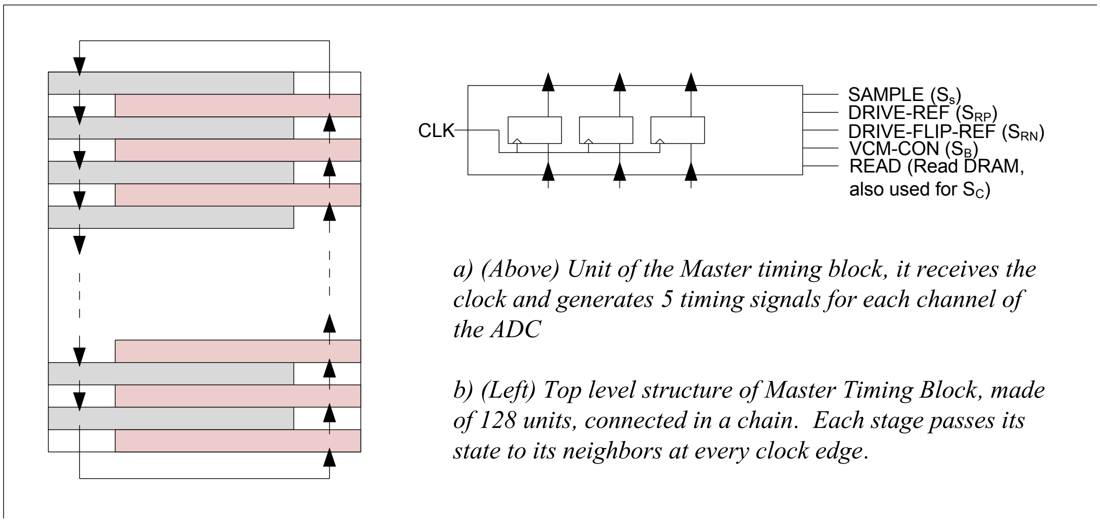
Referring back to Figure 4.12, S_c is added to empty the charge on the sampling capacitors before they are connected to the input, this is to reduce the signal dependent kick-back to the input driver, especially since a sample-and-hold amplifier is not used. S_B is always connected to the common-mode voltage so simple NMOS switches can be used. S_{RN} and S_{RP} connect the ramp voltages to the capacitor, and alternate on different ramping periods. Transmission switches were used here to pass the full range of the signal, however it must be noted that their resistance is in series with the high-resistance interpolation filter.

In the layout of the block, isolating row to row coupling in the analog front end is critical to performance. For capacitors, metal finger capacitors were used. The side of the capacitors driven by the ramp generator was place on the outside of the capacitors, closer to the neighbouring row, while the more sensitive side which connects to the input of the comparator was placed on the inside. This further helps to minimise row-to-row coupling of signals.

4.6 Master Timing Block

The Master Timing Block (MTB) generates the unique timing signals used by each channel of the TIC. Referring back to Figure 4.12 for example, all the timing signals shown are generated by the master timing block. Looking at the whole ADC array, and each clock edge, one row is sampling, one row is reading out, and all other rows are mid conversion. To manage this timing, the MTB operates in a similar way to the control circuitry for the resistor ring.

Each row holds 3 memory units (d-types, or flip-flops). If the row is sampling this first memory unit is 1, the others are 0. If the row is performing readout or is mid-conversion the 2nd or 3rd memory units will be 1 respectively, and the others will be 0. At every clock edge, the value of these memory units are clocked to the neighbouring row, and in a way the next row will now be performing the task this row was originally performing, and this row will move into the state of the previous row. Assuming the state of the memory units are correctly initialised, so that memory unit 1 and 2 are only 1 in 2 of the rows, adjacent, and in the correct order, and memory 3 is 1 in all other rows, then clocking of the MTB should produce all the correct timing signals.


CLK
SAMPLE (S_s)
DRIVE-REF (S_{RP})
DRIVE-FLIP-REF (S_{RN})
VCM-CON (S_B)
READ (Read DRAM, also used for S_C)

a) (Above) Unit of the Master timing block, it receives the clock and generates 5 timing signals for each channel of the ADC

b) (Left) Top level structure of Master Timing Block, made of 128 units, connected in a chain. Each stage passes its state to its neighbors at every clock edge.

Figure 4.14 - Top level view, and unit view of Master Timing Block

Figure 4.14 shows the proposed units of the MTB, and the top-level connection strategy. Here architecturally the same latches used in the resistor ring are used for the MTB system, and are wired to be initialised correctly. For correct operation of an ADC row more than just these three sample, read and convert signals are required. Figure 4.15 shows the internal structure of a unit MTB which generates the desired 5 signal from the 3 main signals. Since bottom-plate-switching is used within the main sampling array (to minimise signal dependent charge injection) the VCM-CON and SAMPLE signal is modified to have matching rising edges, while VCM-CON has a slightly earlier fall time. As previous discussed, each ADC receives a positive going ramp and negative going ramp, but after a conversion cycle these two signals should be switched, since their rising direction is switched due to the implementation of the resistor ring. This is also managed by the MTB. A d-type is as a memory unit to manage this connection. At every new sampling period this d-type is clocked and the direction is changed. At each conversion period either DRIVE-REF or DRIVE-FLIP-REF is high but never both.

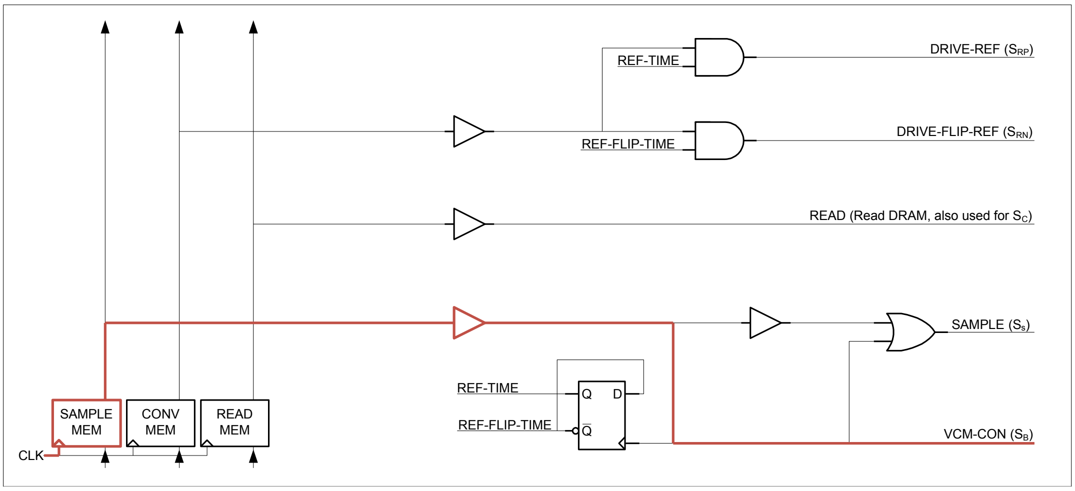


Figure 4.15 - Circuit diagram of unit MTB

As mentioned previously, one of the challenging aspects of meeting the performance in time-interleaved ADCs, in particular ones with high number of channels is meeting the timing skew-requirement between channels. In Chapter 2, it was shown how by increasing the number of channels, the requirement on timing-skew converges to be modelled in a way similar jitter, and from the perspective of manufacturability, increasing the number of channels can be beneficial. In practice, with the increase in number of channels, the clock is required to be distributed over a greater area, and hence through more individual, channel (or channels) specific devices, increasing the effective timing skew between channels. If systematic clock skew were to be eliminated by good design, the sigma timing skew between channels is the timing skew though each channel specific unit circuit added in quadrature. Clearly a clock tree will be used to distribute the sampling clock to all channels, to eliminate systematic timing skew, but further to this as the depth of the clock tree is increased, and the number of devices following the clock tree is increased, the sigma timing skew between channels will increase. Using equations from Chapter 2, this converter is required to meet a 2ps sigma timing skew between channels for the 7-bit operation mode. As was shown in Chapter 2, the noise contribution from timing skew scales linearly with sampling rate and resolution, meaning the same

2ps sigma timing skew is required for the 8-bit and 9-bit mode of operation as well as the 7-bit mode of operation.

This timing skew is the result of mismatch in the clock tree buffers and all devices in the MTB in the path of the clock going to the sampling switch. In Figure 4.15 the red path highlighted shows this path and the timing critical devices. On the clock tree to minimise the timing skew, sharp edges with extra large custom buffers were used to improve the skew. The sharp edges reduce the contribution of V_t mismatch to timing skew, while the increased buffer devices improve matching. The net effect is an increase in power consumption in the clock tree, which unfortunately is necessary to meet this requirement. Following this the timing skew inside the timing latch circuitry should be minimised. As previously explained a similar architecture to that used in the global ramp generator is also used in the MTB, however some device sizes in the latch used for sampling have been adjusted to improve the timing-skew requirement.

Figure 4.16 shows the circuit diagram of the latch used inside the MTB, similar in architecture used in the Global Ramp Generator, however here some device sizes are adjusted. The critical timing is when the gate of the bottom-plate-switch is pulled low, and the input signal is sampled onto the capacitor. The polarity of the signals is arranged in a way so that Q_B is used to drive the buffer which drives this switch, meaning the timing critical moment is the skew on node B moving low. Prior to the moment of the switch opening, on the previous half clock cycle, node A is pulled low, however CLK2 is still low, and node B is high driving the sampling switch which is closed. The moment CLK2 rises, it will connect node A to B, and device M6 will pull node B to ground. The timing skew of node B pulling to ground is critical, for this reason device M7 and M6 are enhanced in size to improve this performance. M6 to improve the drive ability, and M7 to minimise the effect of mismatch. To increase the speed of this action, device M10 can be eliminated, however it was found that this introduces implications to the turn on time of the sampling switch, since node B then only increases to a Vt drop from VDD. This then has implications for settling time inside the S/H circuit due to the delay in turn-on of the switch. Instead the relative drive strength of M6 was increased. It is clear to see that the use of a NMOS device as M7 is beneficial over a PMOS, since the falling edge, i.e. passing of a zero, is the timing critical event. In fact in the whole latching only the matching of this one device determines the timing skew of the circuit to first order, meaning it achieves timing matching similar to an inverter.

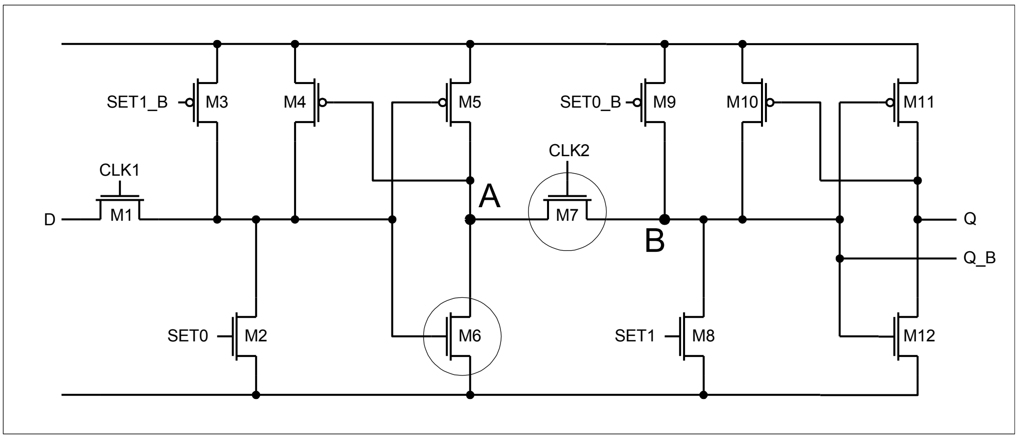


Figure 4.16 – Circuit diagram of MTB latch, with highlighted timing critical device

Referring back to Figure 4.14, to enable greater programmability and debug, the MTB unit loop is closed using a multiplexer, meaning at start-up rather than resetting the MTB loop to the design reset values, with one row in sample, read, and all others in conversion, un-conventional and test patterns can be shifted inside the MTB memory ring. Using these extra elements of debugging the system can be modified to for example sample two rows at a time as mentioned in the previous section, or close the sampling switches slightly earlier. This can be useful to understand the limitation of different timed signals after chip manufacturing.

4.7 Comparator Design

The design requirements on the comparator were explained in Chapter 3. The comparator requires a RMS noise performance matching that of the overall converter resolution, which can be achieved by limiting the bandwidth of the comparator and hence low power consumption since the comparator does not require a bandwidth related to the input. The reduced bandwidth results in a delay from input crossing to output toggling which if greater than the time taken for an LSB change in the counter, will result in offset error in the conversion. However offset errors can be calibrated so back-off on bandwidth is acceptable, however this lowering results in lost codes at the end of the signal range and hence small loss in effective resolution.

Architecturally, in counter based converters, two type of comparators can be used, clock or non-clocked comparators:

- Clocked comparators commonly comprise an input pre-amp, followed by a timing regenerative latch [64]. These comparators are continuously latched at the frequency of the output counter. As each new count value is generated, the comparator is latched to see if the inputs have crossed. These type of comparators can be efficient for power consumption and noise, since they achieve effective wide-band gain with very little cost.
- Non-clocked comparators comprise of a high-gain amplifier, followed by digital buffers [65]. The output of this comparator, following buffers is fed directly to the write signal of the following DRAM. This architecture has a higher static power consumption due to the need for a high-gain amplifier, commonly multistage.

Despite the efficiencies of clocked comparators, a non-clocked, open-loop high-gain amplifier was used as a comparator. The motivation for this is to control and manage noise on power supplies and biasing. All comparators share the common power supply and bias line. Clocked comparators are more efficient in overall power consumption, but introduce a lot of noise to the supply as they are clocked. Furthermore an extra clock tree and distribution is also required for them. A high-gain OTA has static power consumption, but this power consumption is constant. As the comparator for one channel toggles, the power supply profile does not change and this eliminates the possibility of neighbouring misfires through the supply. Furthermore care is taken to minimise the capacitance coupling paths between neighbouring comparators. Clocked comparators are commonly used in image sensors [66], however there, although column coupling is well understood problem, the negative side-effect of column coupling in images can be limited in some scenes, however for communication, such coupling can result in major error in conversion. For this reason extra care is taken, and all such effects are simulated with extracted views of the layout.

As mentioned before, the comparator requires a noise floor meeting the overall resolution of the ADC. The noise can be improved by band-limiting the comparator. The 9-bit mode has the highest requirement on noise. On the other hand band-limiting the comparator, increases the delay in the comparator. The delay in the comparator, relative to the rate of change of the counter, limits the possible number of output codes. In 7-bit mode, only 128 possible codes exist, changing at 1ns intervals. In 7-bit mode the delay in the comparator results in a much greater number of relative lost codes compared to the 8-bit or 9-bit mode. Clearly the requirement in 7-bit mode and 9-bit are different, and hence some basic level of programmability is require in the comparator.

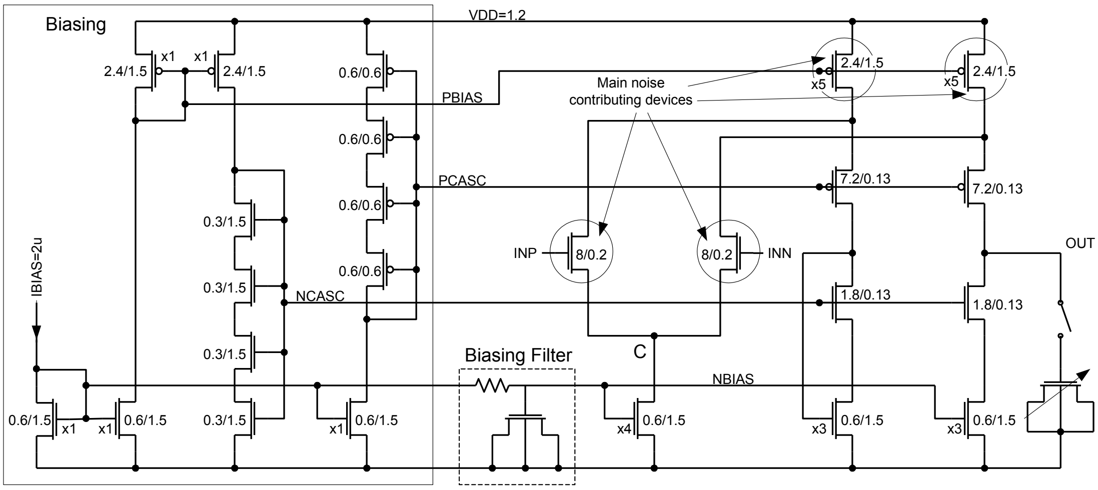


Figure 4.17 - Circuit diagram of folded cascode amplifier used as comparator

In this design a simple folded cascode comparator was used to achieve high gain required. The comparator is a low power design consuming only 28µA of current off a 1.2V supply. Figure 4.17 shows the well-known circuit diagram of a folded cascode amplifier. The choice of transistor size for the input pair is one of the critical aspects of the design. On one hand the size of these devices should be minimised to help minimise the input capacitance of the comparator, since this input capacitance is both driven by the input signal, and also the ramp generator. On the other hand the comparator has a requirement on speed of transition, and noise. As extensively explained in Chapter 3, the comparator does not require bandwidth proportional to the input signal bandwidth, however the limited bandwidth of the comparator introduces delay in the transition. The comparator requires to meet the input referred noise floor required by the resolution of the system. The noise figure can be improved by increasing the gm of the input pair and further limiting the bandwidth of the comparator and hence the integrated noise bandwidth. This however is in conflict with the requirement on delay. The input pair should be sized to meet this balance of requirement. The target delay should be chosen to allow for a good balance between the limited bandwidth and hence delay induced loss of resolution, and the noise requirement on the comparator. An inverter follows the continuous time comparator. An inverter commonly has an open loop gain of 20dB. For a full digital transition to take place, the output of the full comparator system is required to swing by 1.2V, and hence the input of inverter, which is the output of the amplifier, is required to swing by 120mV. The speed in which this movement of 120mV can be achieved is proportional to the gm of the input pair and the output capacitance. In 7-bit mode, each step of the ramp, which is equal to 1-LSB, is 9.37mV (which is 1.2/128, where 1.2V is the full scale input signal). This step in voltage is converted to a current at the output of the amplifier by the gm of the input pair. The gm of the input pair, and hence GBW of the amplifier should be chosen to meet the delay requirement.

The circuit has been designed to have about 120MHz gain-bandwidth, which results in a 10ns delay from input crossing to output firing for the ramp-speeds used in 7-bit mode. The gain was designed to meet the input referred RMS noise requirement of less than 3mV for 7-bit mode of operation. For higher modes of operation, the bandwidth of the amplifier is reduced with a programmable load capacitor, band-limiting the amplifier, reducing gain-bandwidth and hence input referred noise to meet the requirement of 8-bit and 9-bit. Slight over-design for 7-bit mode allowed this adjustment to be made with relatively small MOS capacitors.

In some column parallel architectures rail-to-rail input comparators are used to increase the input signal swing [65]. This is not necessary here since the input is always at common-mode at the time of firing. Care should be made that the node labelled C does recover in time for a ramp-input for when the inputs are at common-mode.

A 10ns delay results in a loss of 10-codes at the output in 7-bit mode (and all other modes of operation), results in a possible 118 codes at the output rather 128, limiting the converter to a theoretical 6.88-bit converter in 7-bit mode. The delay through the comparator was found to change by 14% from 0-deg to 125-deg at worse corner. This information can be used to determine the regularity of calibration needed, depending on the change in temperature expected in the application. It should be noted that separate to the comparator delay, the offset of the comparator, dominated by the input pair offset, also contributes to the offset of the channel, which further minimises the effective output code range. This was found to be relatively small compared to the offset introduced by the delay however should still be quantified. The offset was found to be 4.1mV sigma, which is equivalent to just under 1-code of the 7-bit mode. The clipping of the input signal should be avoided, for all fabricated parts. Further to the 10-code back-off required due to the delay of the comparator, at least a further 3-sigma back-off due to the offset of the input pair, equivalent to 3-codes, should be budgeted to confirm the input signal range is always within the count value of a channel ADC.

Figure 4.18 shows a layout snapshot of the comparator laid out to the row pitch of 6.4µ. Here extra care has been taken to isolate rows from one another. Vertical power and ground lines in high metal layers are used to deliver power, all sensitive nodes are kept to the middle of the comparator.

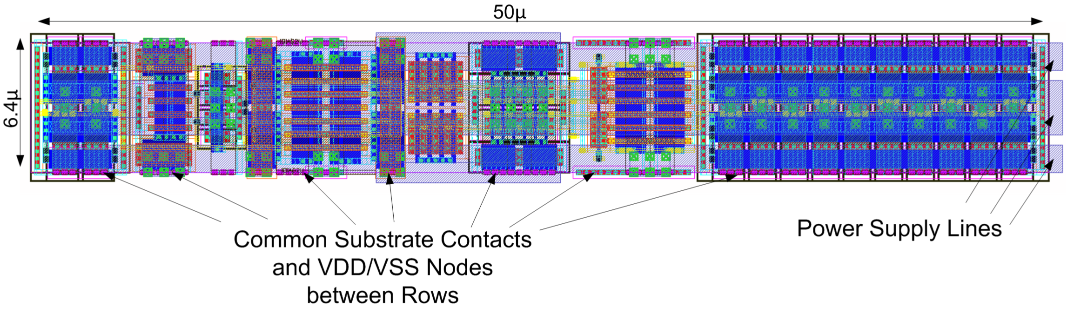


Figure 4.18 - Circuit diagram of folded cascode amplifier used as comparator

4.8 Backend Memory

As discussed in the previous chapter, a backend memory is used to capture the count values generated by the global count generator. This can be realised as an SRAM or DRAM. In this implementation DRAM was used, but equally an SRAM structure could have been used. Figure 4.19 shows the circuit diagram and transistor sizes for the unit DRAM used. The input is connected to the count values generated by the global counter, and the output of the comparator is connected to the W signal. The R signal is driven by the READ signal generated by the MTB. The signal is sampled onto capacitor M2. When R goes high, if a 1 was stored in memory, device M3 will attempt to pull the output node low. To readout the value of a DRAM unit, the output node is pre-charged to VDD, then the Read signal is enabled, if the output node is pulled low, then a high value had been sampled in memory. If the output node remains high, the value zero was sampled. M3 is the main pulling device, and is commonly made as powerful as possible, however here this device is sized for different reasons, primarily for matching, and predictability under process corners to the replica readout circuit. The motivation for this will be explained when covering the backend memory readout circuit. A total of 10 DRAM units are present in each sub-ADC row.

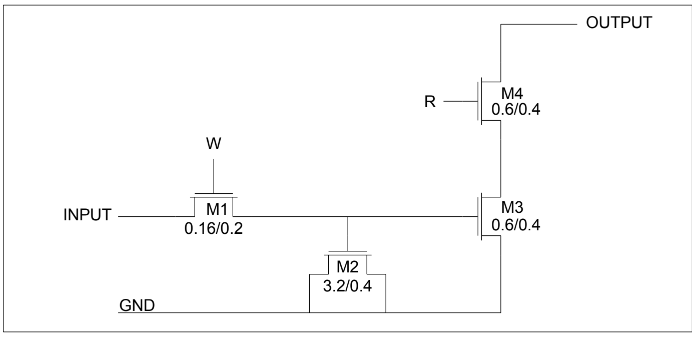
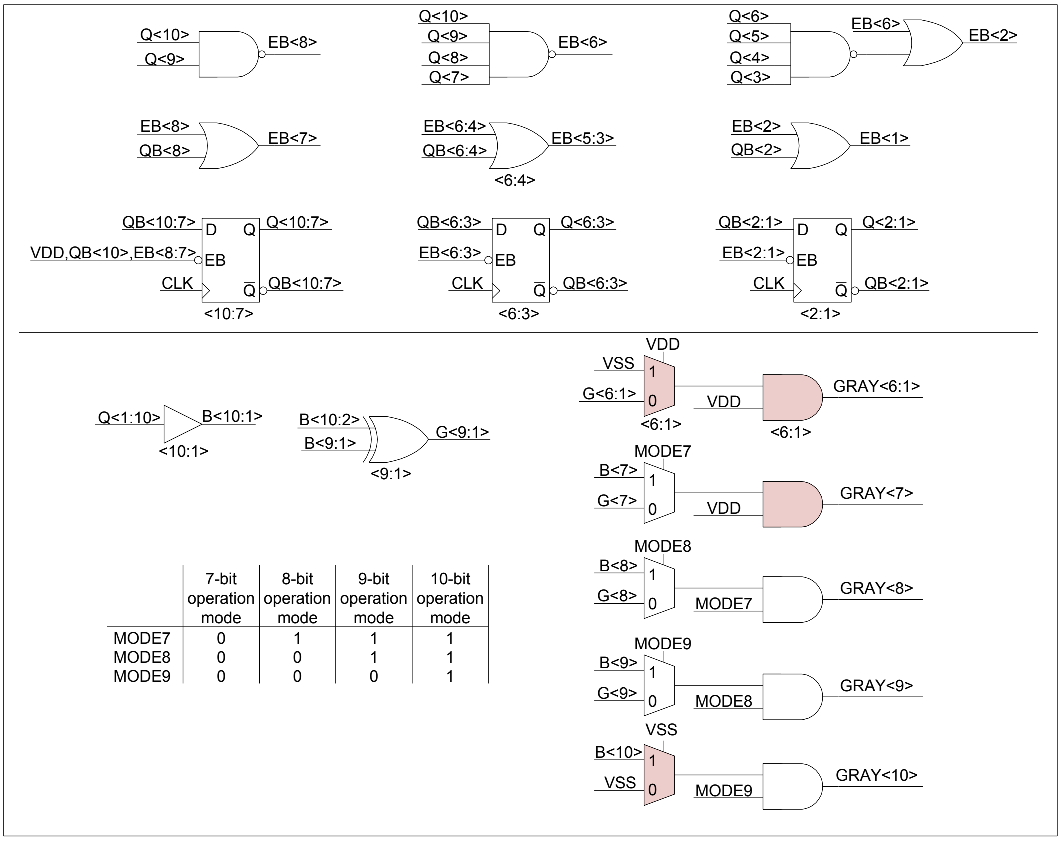


Figure 4.19 - Unit DRAM, and device sizes

4.9 Global count generator

As discussed in the previous chapters, a global gray code pattern is fed to all ADC rows. The global count generator block generates this pattern. Figure 4.20 shows the logic associated with this generation. The top half of the figure shows a synchronous binary count generator. The generator has been pipelined 3 times to enable operation at 1GHz. The largest element EB<1> has a clk-to-clk gate delay of 4. The bottom half of the figure converts this binary clock to gray code. This is primarily done using XOR logic gates, however the added complexity is to allow for 7-bit, 8-bit, 9-bit or 10-bit modes of operation. The MSB of gray code and binary code are the same. To do this conversion correctly, depending on the mode of operation the appropriate MSB should be forwarded to the output and other more significant bits not used should be grounded. The input of this block is only the clock, and the 3 mode select bits. The outputs are GRAY<10:1> which are driving to the ADC array. The devices in red are redundant in function, and are used for timing matching.



	7-bit operation mode	8-bit operation mode	9-bit operation mode	10-bit operation mode
MODE7	0	1	1	1
MODE8	0	0	1	1
MODE9	0	0	0	1

Figure 4.20 - Circuit of Global Counter Generator

4.10 Backend Memory Readout circuit

4.10.1 Task of memory readout

As previously discussed, at readout of each unit DRAM, the output node is pre-charge to VDD, then the read signal is enabled, and the output node is monitored. Since 128 unit DRAMs are present on every output node, and the required operation speed can be as high as 1GHz, the capacitance on the output node is too high to allow full change and discharge of the node with the DRAM cell. To increase the drive ability of the unit DRAM cell, its size requires increasing, which in turn increases the capacitance on the output node. For this technology, with the size of the array used, and the 1GHz operation speed, implementation of a simple pre-charge based readout circuit was not found to be possible.

As discussed at the conclusions of Chapter 3, an analog folding technique has been used to fold the pull current of the DRAM cell. Figure 4.21 shows the circuitry. Here node X, which is the high capacitance output node of the parallel DRAM units, is held by the cascode device M3, and should not move. The current pulled by the DRAM unit I_m is folded by device M1, and is passed through M3 to the latch, meaning the drain of M3 moves rather than node X. The drain of M3 would be a much smaller capacitance node.

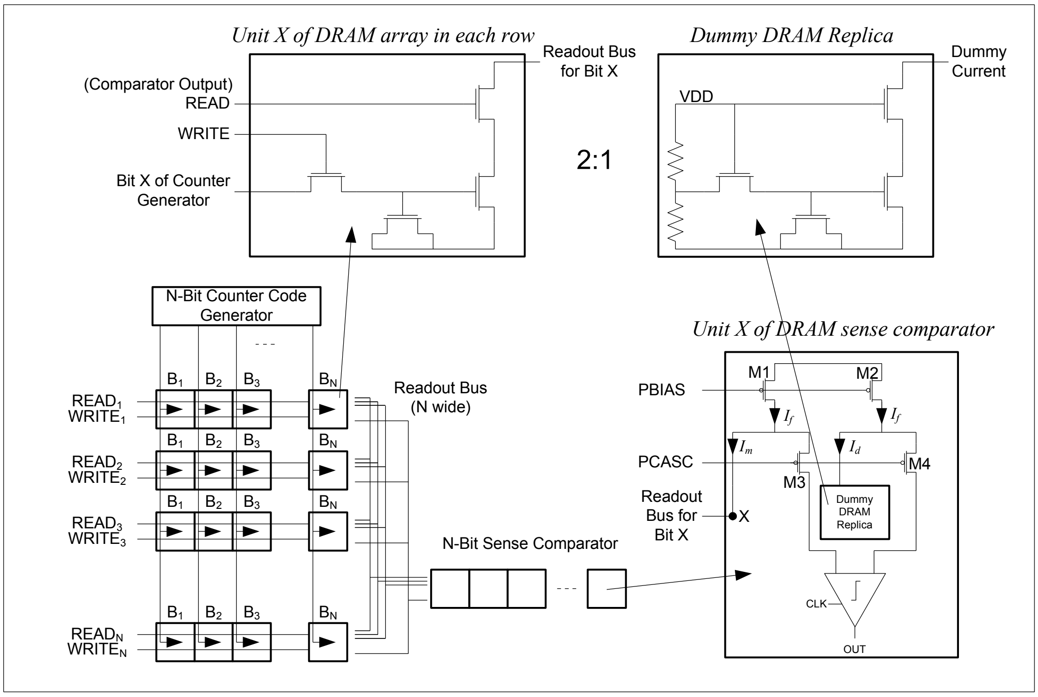


Figure 4.21 - DRAM readout circuit

The current the DRAM produces is related to the size of the DRAM pulling device, and the particular process corner, and temperature of operation. The size of the DRAM pulling device, and I_f has been chosen in a way to guarantee this current is always bigger than the I_m, the current the DRAM unit can pull. The Dummy DRAM Replica has been design to pull current close to half the DRAM pulling current, and to match over process and temperature.

One of the great challenges to enable correct operation of readout circuit is timing. The latch shown in Figure 4.21 should be reset between every continuous readout operating at 1GHz, and this latch signal requires near perfect timing alignment with the read signal used inside every ADC row. Figure 4.22 shows how this is achieved. All MTB units are timing matching from a global clock tree. From this tree, a clock is taken off to a block with exact same devices and layout of the MTB unit, to generate the latch signal from the DRAM readout circuitry. This signal is then routed to the readout circuit in a way matched to the main circuit read signal routing. This path was extracted and checked for timing matching.

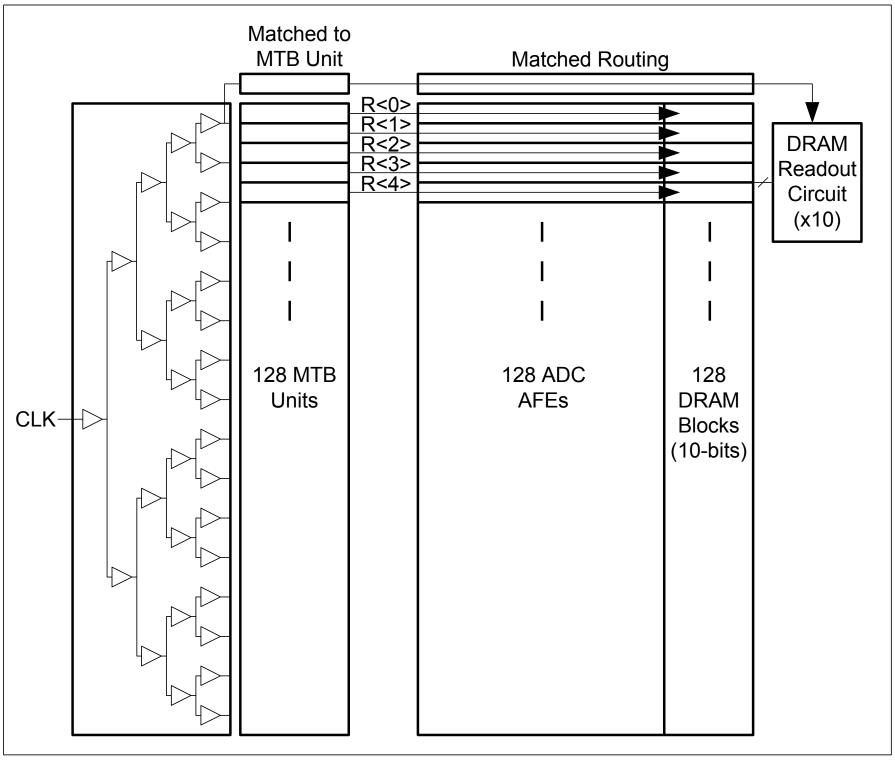


Figure 4.22 - Matching of timing accuracy between DRAM read and readout

4.11 TIC Top-Level

Figure 4.23 shows the top-level view of the TIC ADC showing all 128-columns and the relative placement of all the blocks. Green is the colour of top-level metal which is used for distribution of power through the array. Meeting IR requirements throughout the array, especially for the clock tree blocks is critical, and these effects were extracted and simulated. The area is dominated by metal capacitors and logic gates. The total circuit comes to 820μ x 600μ.

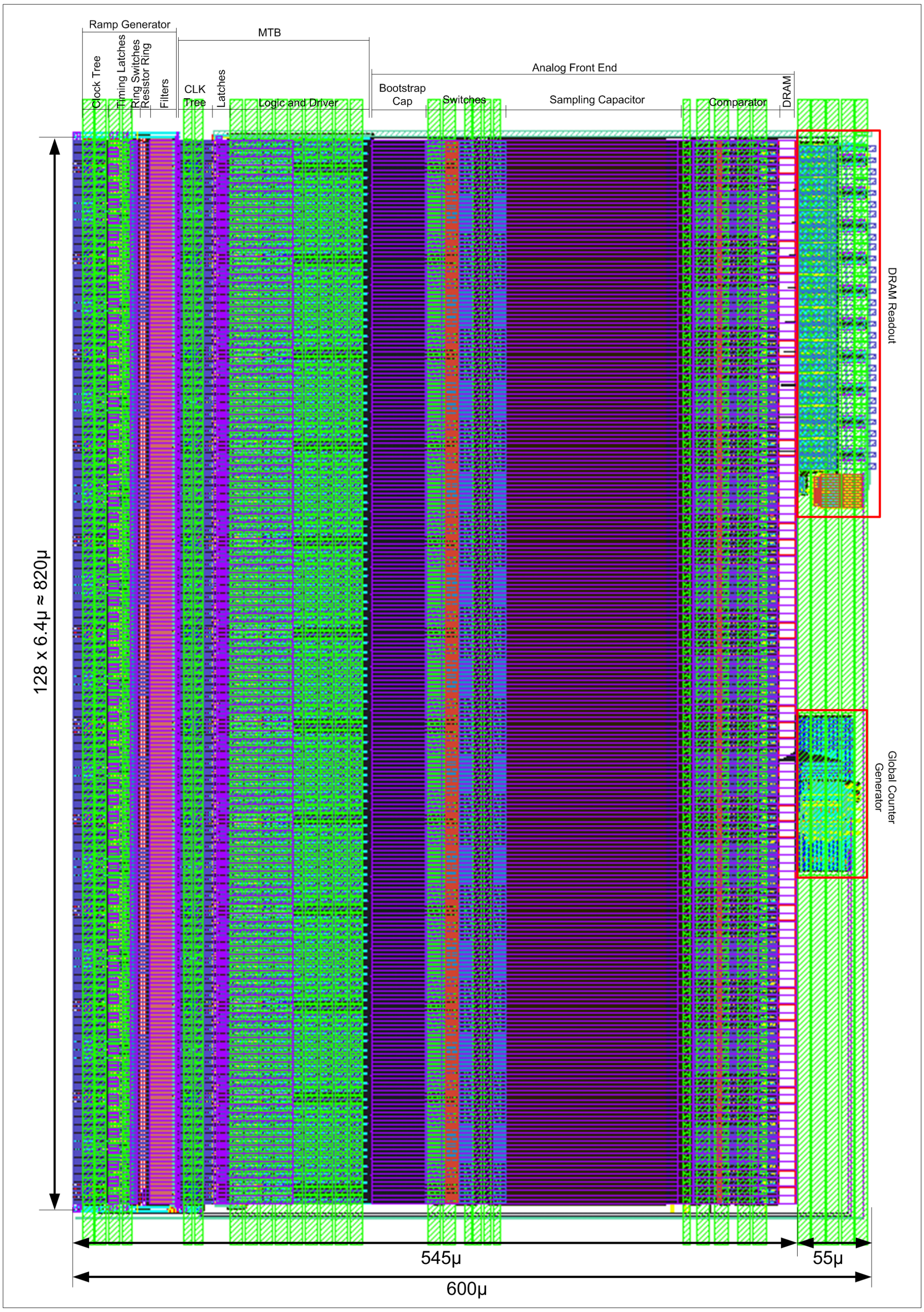


Figure 4.23 - Top Level view of TIC ADC, showing 128 rows and readout circuitry

4.12 Chip-Level Auxiliary blocks

4.12.1 Introduction

To be able to test the ADC on chip at full speed a number of auxiliary blocks are also required. The main tasks are to enter a clock on chip for the ADC to function, and to get the digital output data from the ADC off chip.

To output the digital data commonly two approaches are used; One is to implement memory on chip, with high speed write rate, allowing the ADC to operate at full rate, capturing the digital output of the ADC to memory. Once this is done, the data from the memory can be read off the chip at a much slower rate, and even in a serial fashion requiring very few pins. The second approach is to implement digital IOs for the chip that can operate at the rate the ADC generates data, for example LVDS pads. If the second approach is used then a digital capture device with similar digital communication standards is required. In the second approach the data can be partially parallelised to reduce its data rate, and then taken off chip in real time with more conventional IO devices.

If memory cells and memory compilers are available, the first approach is commonly favoured for many reasons. It simplifies the test setup considerably, but most importantly reduces the possibility of cross talk from IOs to the main ADC. The output drivers are operating at a much lower rate, but more importantly are not operating when the main ADC is in operation. If the second approach is used, extra care should be taken to reduce the cross talk from IO drivers to the main ADC.

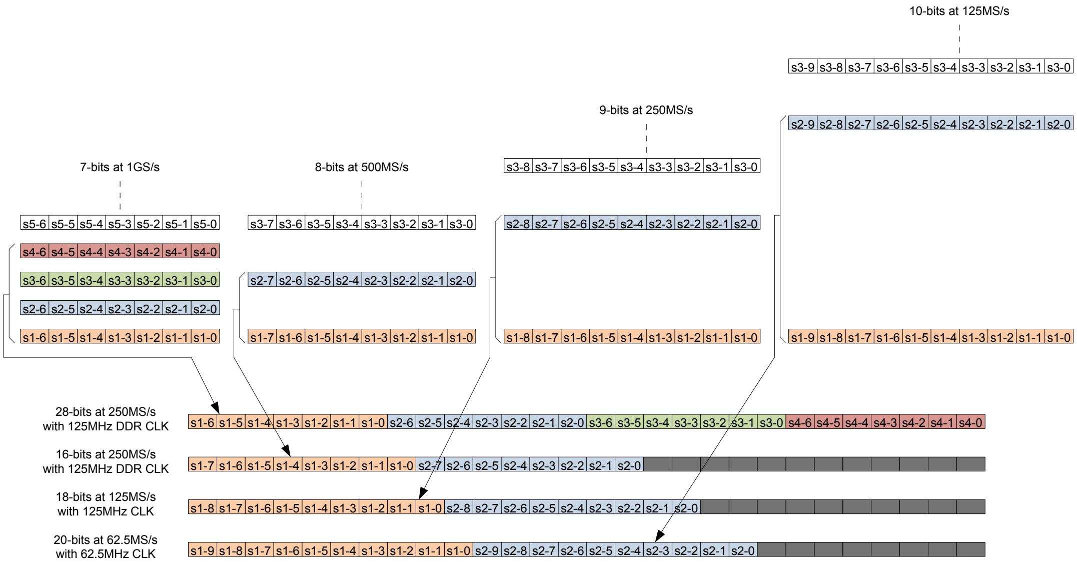
Unfortunately memory compilers for this technology were not available at the time of this project. Furthermore due to limited available area on the die, LVDS drivers could not be used and hence to allow data to be outputted from the chip, the output of the ADC would first be parallelised to a lower rate, and then driven off chip at CMOS level.

4.12.2 Data Packer

As will be explained in the next section, only a limited number of pins could be used for the digital readout, since the chip was PAD limited. A total of 28-pins were assigned for digital readout. The fastest rate data can be taken off chip using standard CMOS levels for the chosen package with expected bond-wire inductance, with the part soldered down, was found to be around 130MHz. The data from the ADC can come in 4 different modes:

7-bit data, at 1GS/s, with a 1GHz reference clock
8-bit data, at 500MS/s, with a 500MHz reference clock
9-bit data, at 250MS/s, with a 250MHz reference clock
10-bit data, at 125MS/s with a 125MHz reference clock

To get the data off chip, a packer block was developed which packs the data into a 28-pin output with reference clock. For each of the 4 modes of operation the packing algorithm is different. Figure 4.24 graphically shows the different packing algorithms.



10-bits at 125MS/s
s3-9 s3-8 s3-7 s3-6 s3-5 s3-4 s3-3 s3-2 s3-1 s3-0
s2-9 s2-8 s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
s1-9 s1-8 s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0
9-bits at 250MS/s
s3-8 s3-7 s3-6 s3-5 s3-4 s3-3 s3-2 s3-1 s3-0
s2-8 s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
s1-8 s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0
7-bits at 1GS/s
s5-6 s5-5 s5-4 s5-3 s5-2 s5-1 s5-0
s4-6 s4-5 s4-4 s4-3 s4-2 s4-1 s4-0
s3-6 s3-5 s3-4 s3-3 s3-2 s3-1 s3-0
s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0
8-bits at 500MS/s
s3-7 s3-6 s3-5 s3-4 s3-3 s3-2 s3-1 s3-0
s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0
28-bits at 250MS/s with 125MHz DDR CLK
s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0 s3-6 s3-5 s3-4 s3-3 s3-2 s3-1 s3-0 s4-6 s4-5 s4-4 s4-3 s4-2 s4-1 s4-0
16-bits at 250MS/s with 125MHz DDR CLK
s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0 s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
18-bits at 125MS/s with 125MHz CLK
s1-8 s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0 s2-8 s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0
20-bits at 62.5MS/s with 62.5MHz CLK
s1-9 s1-8 s1-7 s1-6 s1-5 s1-4 s1-3 s1-2 s1-1 s1-0 s2-9 s2-8 s2-7 s2-6 s2-5 s2-4 s2-3 s2-2 s2-1 s2-0

Figure 4.24 – Packing algorithms for different modes of operation

Mode 1 (7-bit, 1GHz): In this mode the data is paralleled up 4 times, converting into a stream of 28-bits and 250MHz reference clock. The reference clock is divided by 2 again, building a 125MHz clock, but operating in DDR (double-data-rate), while the data is sampled on both rising edge and falling edge of the block. When the digital data has an effective sampling rate of 250MHz, it means the fastest the value of each bit changes is at 4ns intervals, similar to a 125MHz clock, meaning each of the 28-bit data streams at worse change rate can appear like a 125MHz clock, and the digital IOs are capable of handling this data rate.

Mode 2 (8-bit, 500MHz): Similar to Mode 1, however the data is only paralleled up to 2 times, resulting in a 16-bit data stream with a 250MHz reference clock. The DDR clock at 125MHz is outputted with the data again similar to Mode 1.

Mode 3 (9-bit 250MHz): The data is paralleled up twice, resulting in an 18-bit data stream with reference clock of 125MHz. This is directly outputted from the chip.

Mode 4 (10-bit 125MHz): The data is parallel up twice, resulting in a 20-bit data stream with reference clock of 62.5MHz. This is directly outputted from the chip.

The data packer was realised using standard-cell logic, and then buffered to be driven off chip using custom CMOS level drivers.

4.12.3 Clock Receiver

Another important part of the testability of the ADC at full rate is the delivery of reference clock to the block. Here again, two approaches are common; One is to implement a high-speed PLL on chip which meets the requirements for short-term and long-term jitter, and to use an external reference crystal as input for the VCO. The PLL can then multiply this clock up to the required frequency. The second approach is to deliver the high-speed clock directly to the chip from an external high-speed clock generator. The first approach is prone to less error, but requires the availability of a PLL. Here the second approach was used due to the lack of availability of a PLL in this technology which meets the jitter requirements needed for characterisation of the system.

A high-speed current mode clock receiver was implemented to take a 1GHz clock into the chip from an external current mode clock source. Figure 4.25 shows the circuit diagram used. It is based around a high-speed amplifier with hysteresis. The 50-ohm termination resistors are made of 10 500-ohm resistors in parallel to meet the current handling requirements. The output clock is buffered, and then routed to the main ADC.

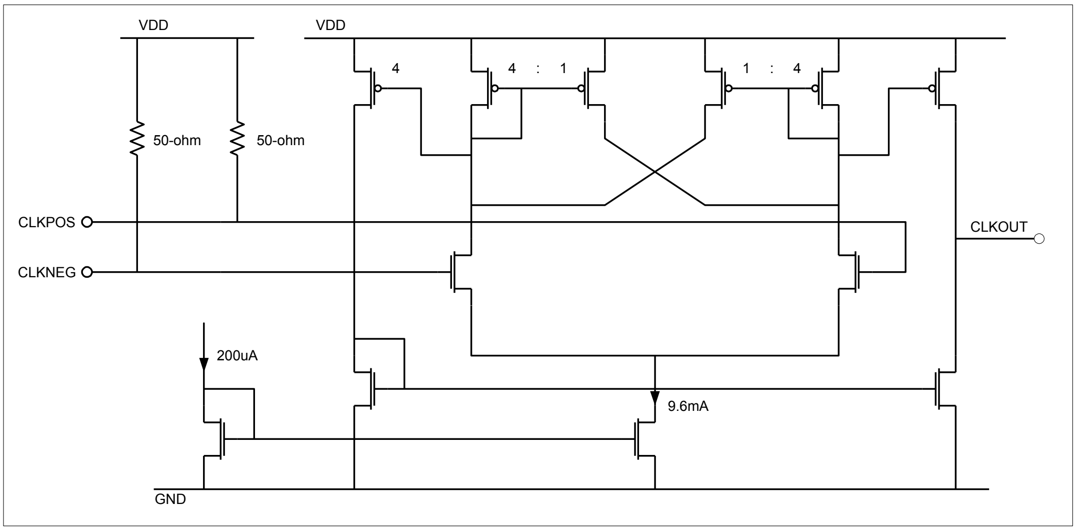


Figure 4.25 - Clock Receiver circuit diagram

A number of programmabilities in current, input mode, divide options on the output clock were implemented on silicon. A bypass mode to pass in a low frequency clock, and termination options were also implemented. A number of ESD and clamping circuits have also been added in the path of the input clock.

4.12.4 Serial Interface

The main ADC block has over 50 digital control signals for programmability, mode select and testing. Unfortunately as it will be explained in the next section connecting each of these to a dedicated pin on the package is not possible, and hence a serial control interface was implemented on the chip. Here the required values for these control signals are slowly shifted into the serial interface and then strobed to the main ADC array. Figure 4.26 shows the basic operation of the serial control interface. The desired values for CONTROL<63:0> are slowly shifted in through SR_IN while SR_CLK is clocked in sync. When all 64 values have been shifted in, SR_STROBE is taken high, and all the values drop down into the lower shift-register bank, and the value of all control signals are simultaneously changed. SR_OUT is added for debug purposes. Using this technique here for example 64 control signals are realised using only 5 signal pins.

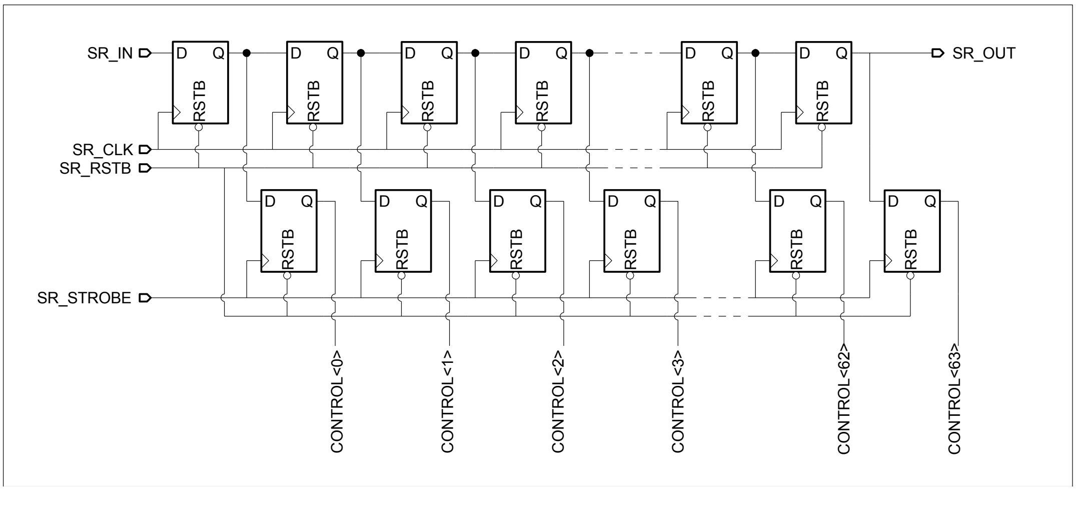


Figure 4.26 - Circuit diagram of Shift-Register based serial interface

4.13 Top Level Chip Assembly

4.13.1 Number of Pads, Pin-out, and Package choice

The ADC was fabricated on a 1.5mm x 1.5mm die. Due to bonding machine limitations, and available pad opening options, pad no closer than 72μ in distance could be used on the chip, limited to 18 pads per side and hence 72 in total. An 84-pin TQFP package with small cavity, was chosen for the device. Pins were chosen to allow individual supplies for different sections of the main TIC ADC to be routed out to measure the power consumption of different sub-sections. Figure 4.27 shows the top level pin-out plan. The digital backend supply (VDD12_BE) and ground is distributed between the digital outputs. A split ring strategy is used to help with noise isolation, where the analog signals come in through a clean section of the ring which is powered by its dedicated supply. As mentioned previously isolating the digital output drivers from the ADC is very important.

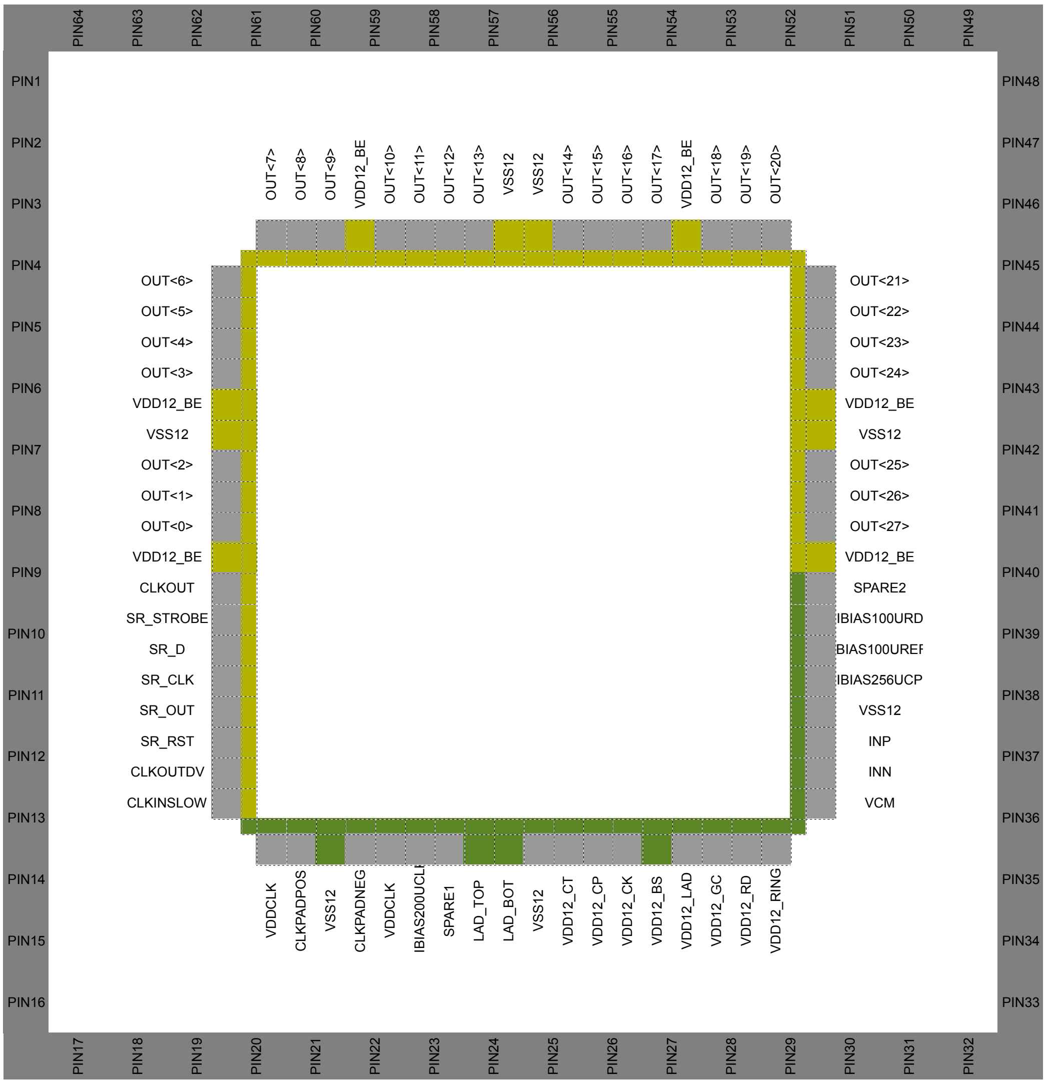


Figure 4.27 – Chip pin-out and bonding diagram

Figure 4.28 shows a screenshot of the chip top-level layout, and the relative position of the blocks. Figure 4.29 shows the die photograph of the fabricated chip. The space between the blocks has been filled with decoupling capacitors. The clock receiver is placed at the bottom left of the chip, meaning it requires to be routed all the way around the chip to the input ports of the ADC. At first this may seem undesirable, however the clock receiver has been purposely placed as far away as possible from the digital out Pads to minimise signal dependent cross talk. Meeting the jitter requirement for characterisation of the block is quite challenging. The clock is then routed in a shielded clock routing path, regularly buffered with unit inverters in isolated deep-nwell on a dedicated supply.

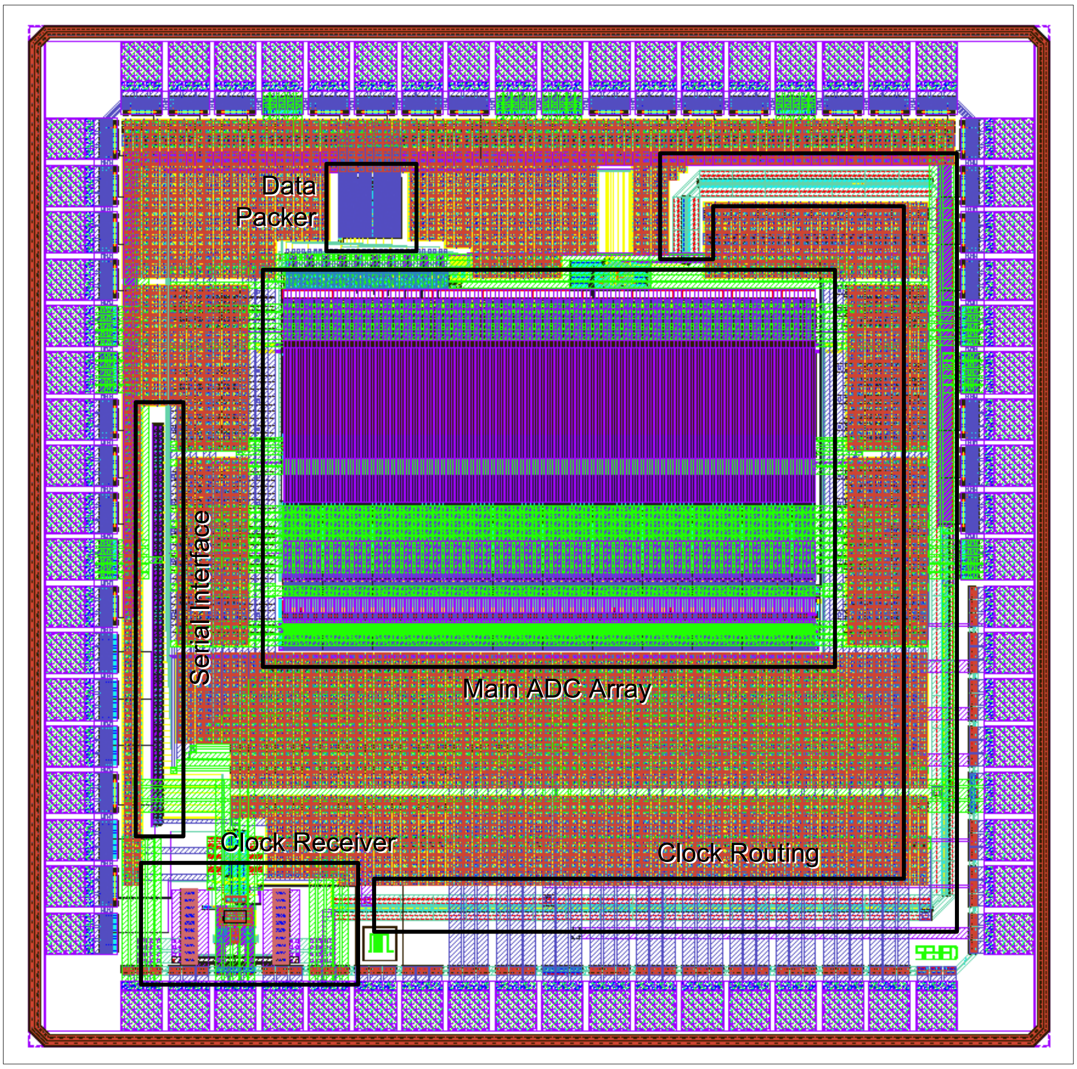


Figure 4.28 – Chip top-level layout, and block positions

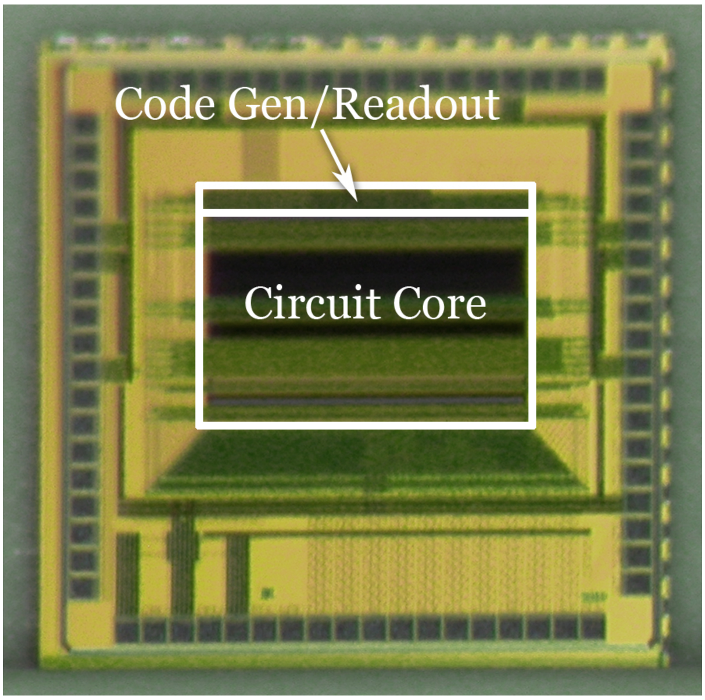


Figure 4.29 – Die photograph of chip

4.14 Conclusions

This chapter looked at the actual implementation of a Time-Interleaved Counter ADC implemented in 0.13µ CMOS process. A number of practical challenges were realised and overcome:

The realisation of the global ramp generator was found to be challenging since the chosen technology sets a hard limit on the resolution/speed range the architecture can achieve. In reality for high resolution applications, greater than say 8-bit, realistically implementing a ramp-generator with greater than 256 individual steps is not practical with this architecture. Here interpolation was used to trade-off a small loss of range for resolution, however it was seen that despite the use of interpolating, resolutions close to 10-bit were not practical to achieve.

Meeting the timing skew between channels was difficult. Initial study revealed the more power efficient 256-channel converter could not be realised due to the practicalities of meeting the timing-skew requirement. Custom latches were designed and a large amount of power was consumed in clock trees to meet this requirement.

The front-end bandwidth requirement for channel matching and settling time requirements was found to be challenging over the full range of operation modes. Some extra programmability was introduced in this section to allow fulfilment of all different operational modes. These programmable elements resulted in inefficiencies.

The requirements on the comparator for the different modes of operation were conflicting. For high-speed mode a larger bandwidth comparator was required, while in high-resolution a band-limited low noise comparator was required. This was realised by some basic re-programmability in the comparator, however the implemented solution leaves much room for improvement.

In terms of top-level functionality, and implementation of the ADC function, the block is a sensitive timed machine, and distribution of evenly timed clocks, and generation of timing signals is an important and challenging part of the design. Apart from good architectural design, verification of the implementation work with detailed extraction simulations was required.

Observing the layout of the full ADC highlights the relative area taken by both the metal capacitors for the sampling capacitors of each ADC, and the bootstrap capacitors. Also a large part of the unit ADC is dominated by standard cells used as part of clock trees, timing logic, or buffers to drive switches.

The ADC was completed to be fabricated on 1.5mm x 1.5mm die in 0.13µ CMOS, where the active circuitry consumes 820µ by 600µ. The following chapter will look at measurement results.

Chapter 5 - Measurement Results

5.1 Introduction

This chapter looks at the results of chip testing performed on the fabricated chip described in the previous chapter. The basic test setup will be explained. Prior to normal operation, a calibration cycle is required of the block. This will be explained and data collected during calibration will be analysed. Following this the block will be characterised in different modes of operation, the 7-bit, the 8-bit and the 9-bit mode. An attempt will be made to understand the limiting factors in performance in the block, and to quantify the contribution of all the non-idealities, timing-skew, bandwidth skew, settling time, code-loss and noise. As explained in the previous chapter, some parameters in the system are programmable allowing for adjustments to help understand the limiting factors in the design. A summary of performance in normal operational modes will be presented.

5.2 Test Setup

Figure 5.1 shows the test setup developed for the testing of the TIC ADC test chip. The setup comprises of a two board solution. The main test-board holds the test-chip, analog signal coupling components, and the clock generator and driver circuitry. The second (daughter board) is an Opal Kelly XEM3010 which comprises a FPGA part, and USB interface to communicate to the computer.

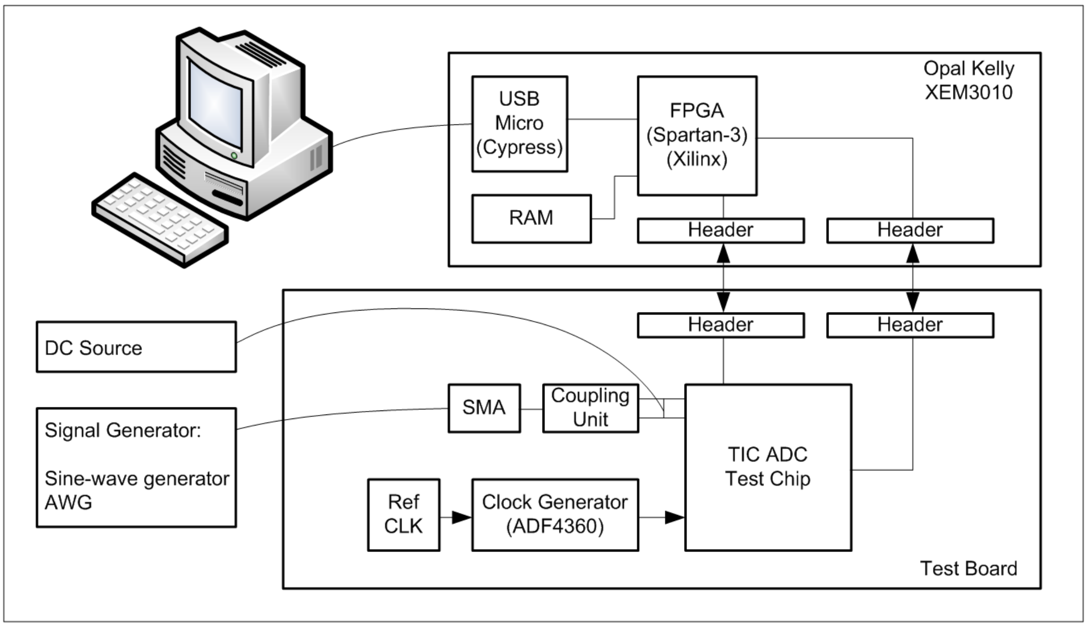


Figure 5.1 – Test setup developed for testing chip

A hardware abstraction layer is generated on the computer in conjunction with the FPGA part, where on the computer the low speed control and enable signals can be toggled using a python interface. These signal changes are then packed appropriately and translated to changes in the 4 serial interface input pins. This sequence is then forwarded to the FPGA part through the USB interface. The FPGA then forwards these commands through the board-to-board headers to the TIC-ADC. The appropriate values for all bits are then shifted into the shift-register and strobed to apply to the block. Other devices on the board such as clock-generator and reference generators are also controlled using a similar interface, originally driven by python to the FPGA to the part with the appropriate communication standard, which was SPI for the clock-generator and references.

As input stimuli a linear sine-wave generator, and an Arbitrary Waveform Generator (AWG) were used. As mentioned an on-board PLL and driver from Analog Devices was used. As a reference clock for the VCO an accurate crystal was used on the board. When performing FFT testing, a zero-ohm link was removed from the path of this crystal and a 10MHz reference clock was taken from the appropriate signal generator to the board via an SMA connector. This is to insure the matching frequency of the sampler (the ADC) and the signal generator.

When a capture is taking place, the TIC chip can generate data 28-bits wide at 125MHz. This data travels across the high-speed header to the FPGA part, and using the provided clock is latched in the FPGA part and stored in memory at full rate. Careful design of the gates in the FPGA was required to latch this data correctly. Extra care was taken on the routing of these digital signals on the board and over to the FPGA board to maintain timing alignment. Once a capture is completed, the data is slowly passed to the computer through the USB interface. The computer can then de-pack the data and do required analysis on the data.

5.3 Calibration

Calibrating the device before normal operation and at regular intervals is an important part of the operation of the ADC. The calibration involves learning the offset of each channel during the foreground calibration cycle, and then applying correction to all channel outputs during normal operation, to correct for this offset mismatch. Individual ADC channels require individual calibration information which should be applied to their output. The calibration process comprises a simple subtraction operation only, which is applied in software after the output of the ADC is captured. The number which is subtracted from each row comprises of two parts, one the fix row offset which is dependent on the row number and is due to the use of a global count generator, the second part is due to the delay and offset of the comparator. The row offset is pre-determined, and the delay induced offset should be equal between different rows to first order, so the main difference between the rows should be dominated by the comparator offset and any mismatch in gain-bandwidth between the channels. The calibration routine is applied before each capture. The inputs to the ADC are grounded, and a total of 4 full runs of the system are captured into memory on the computer. The results are averaged and used for subtraction for all future captures. The calibration cycle and how the capture data is applied is shown graphically in Figure 5.2.

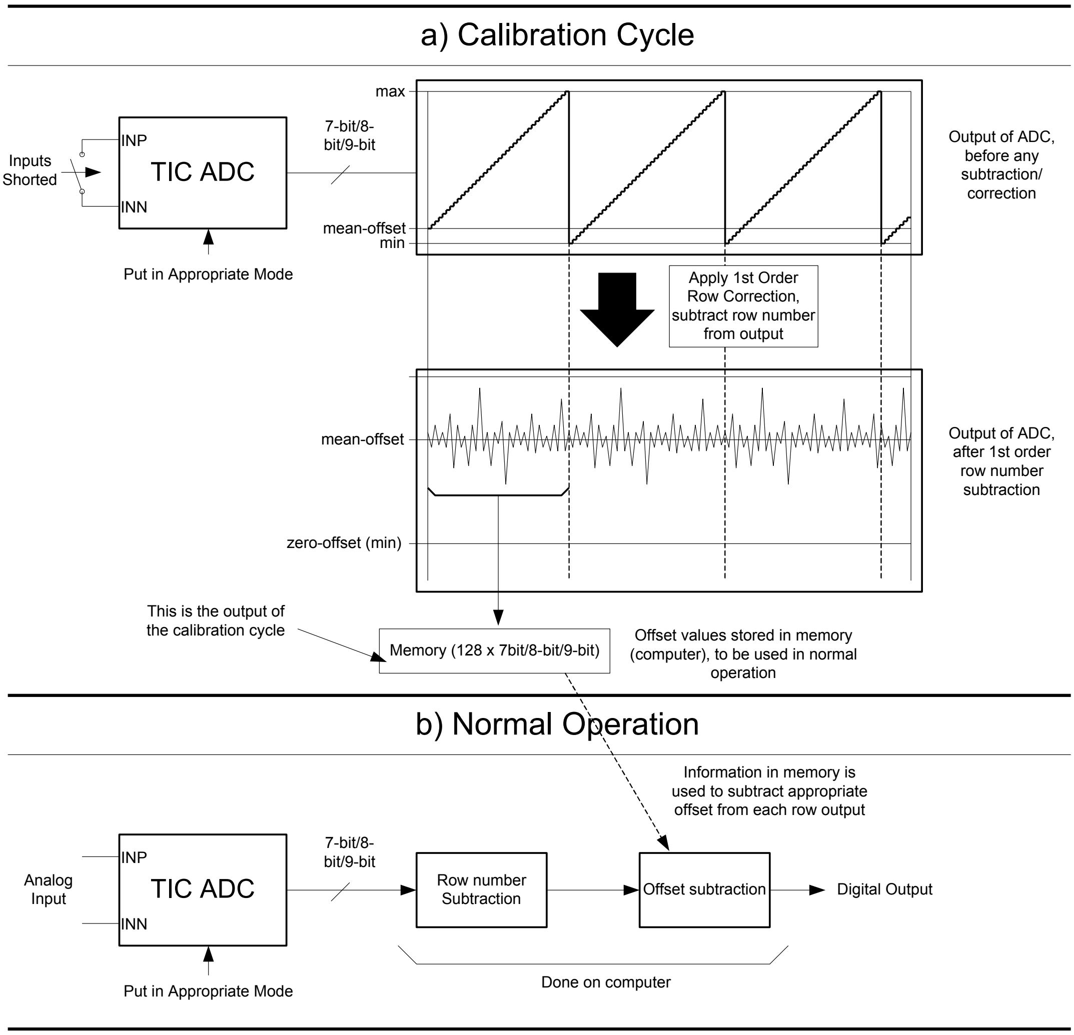

a) Calibration Cycle
Inputs Shorted
INP
INN
TIC ADC
Put in Appropriate Mode
7-bit/8-bit/9-bit
max
mean-offset
min
Output of ADC, before any subtraction/ correction
Apply 1st Order Row Correction, subtract row number from output
mean-offset
zero-offset (min)
Output of ADC, after 1st order row number subtraction
This is the output of the calibration cycle
Memory (128 x 7bit/8-bit/9-bit)
Offset values stored in memory (computer), to be used in normal operation
b) Normal Operation
Information in memory is used to subtract appropriate offset from each row output
Analog Input
INP
INN
TIC ADC
7-bit/8-bit/9-bit
Row number Subtraction
Offset subtraction
Digital Output
Put in Appropriate Mode
Done on computer

Figure 5.2 – Calibration cycle, and how information is used in normal operation

Of course the data from calibration is stored in memory on the computer and then applied to all following normal operational mode captures. Two issues should be considered regarding the regularity of calibration, drift and variation over temperature, and 1/f noise. Based on measurements, 1/f noise was found to be an order of magnitude smaller than the highest resolution mode LSB, and does not play a part in determining the regularity of calibration. In 7-bit mode, at room temperature the mean offset was found to be 9.2 output codes. This is believed to be dominated by the limited bandwidth, and hence delay of the comparator converting to offset, and matches data from simulation of an expected offset of 10-codes. This offset was found to change between 8.9 to 10.4 codes mean for a temperature range of -10deg to 80deg. This variation is believed to be related to the change in gain-bandwidth of the comparator over temperature. This also matches simulation. This data can be used to determine the regularity of calibration required in production. On top of the mean, row-to-row mismatch is also present, and was measured as a sigma of 1.2-codes. The histogram of the offsets for a part has been shown in Figure 5.3. The offset's show no particular pattern, and appear random. The offset is believed to be dominated by the comparator input offset, as measured in simulation, rather than mismatch in gain-bandwidth between channels. For all further testing of the ADC, an appropriate signal back-off is applied to avoid signal clipping for all manufactured parts.

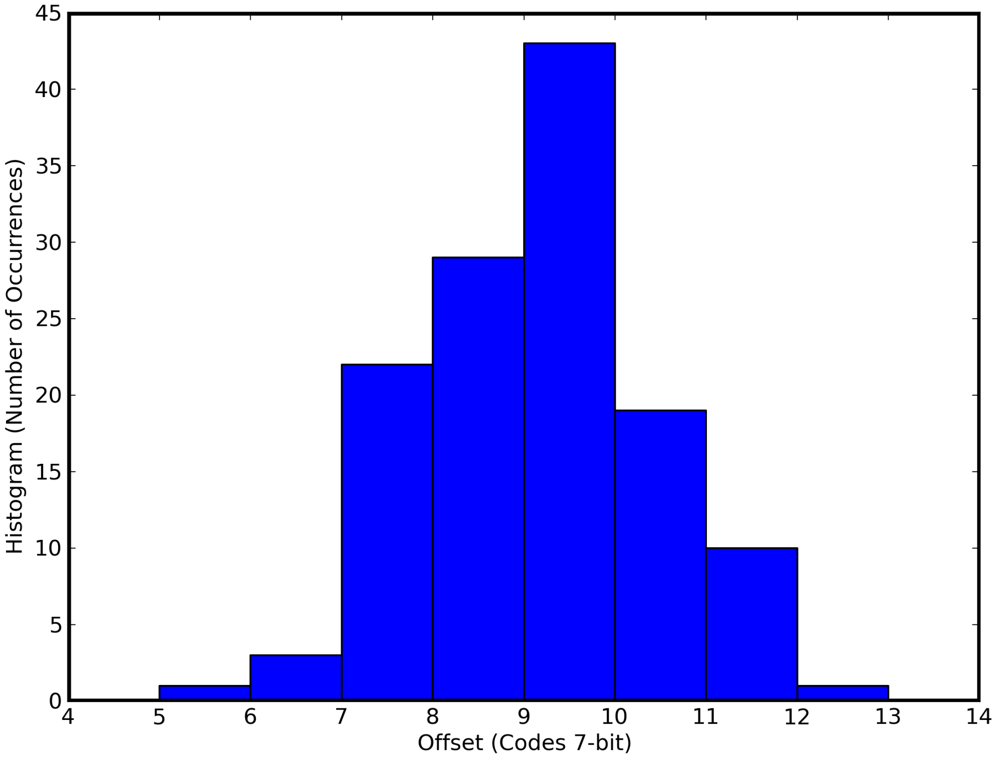


Figure 5.3 – Histogram of Offsets in codes of 7-bit mode

5.4 Block Performance Analysis

5.4.1 DNL and INL, static non-linearities

As explained in chapter 2, DNL and INL are common methods to measure a converter's linearity performance. This is commonly done with the statistical method explained in Chapter 2. Unfortunately, for time-interleaving ADCs, especially one with as many channels as this, INL and DNL may not be representative of the converters performance. Any mismatch in static non-linearities and errors between channels will be modulated to higher frequencies, and hence will not be reflected in the INL and DNL plot. These plots only show static non-linearities common between the channels.

In the case of the TIC ADC, the INL and DNL plots should show the characteristics of the ramp generator, since the profile of the ramps should be common for all channels of the system. Figure 5.4 shows the DNL measured for the ADC in 9-bit mode of operation, using the statistical method. An input signal frequency of below 100kHz was used to measure a near static linearity performance. The DNL profile is very repetitive, and clean for a time-interleaved ADC since any mismatch in devices will be dithered into noise due to the presence of multiple channels. A number of codes are missing from the start of the DNL plot. This is due to the delay in the comparator and non-linearity introduced due to the interpolating filter. Early codes never occur, since even for the smallest input signal the delay through the comparator means the backend counter has already counted to higher values.

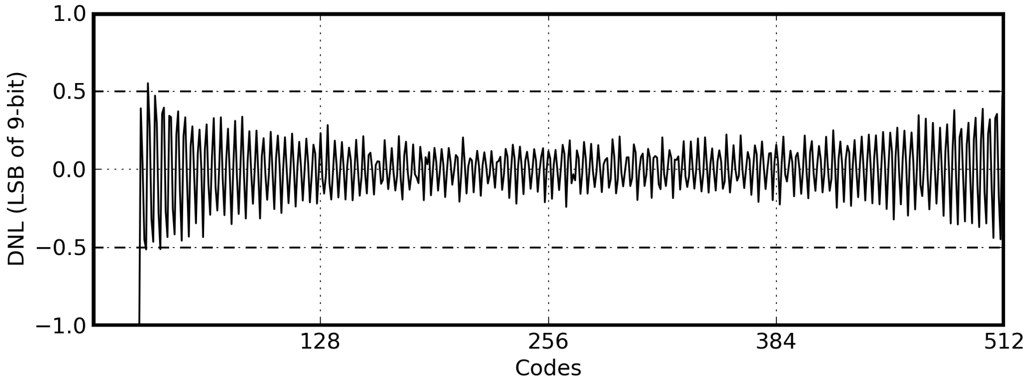


Figure 5.4 – Static DNL performance of TIC ADC in 9-bit Mode

Figure 5.5 shows a close up of the DNL plot. Clearly, a pattern is repeated every 4 codes. This is due to the interpolating filter used in the ramp-generator. The ramp generator only generates 128 unique levels, these are for example at code 104, 108, 112, 116 and so on. The filter interpolates the codes between these for the 9-bit system. The size of the filter was chosen in a way to assure 1/2 LSB performance in DNL.

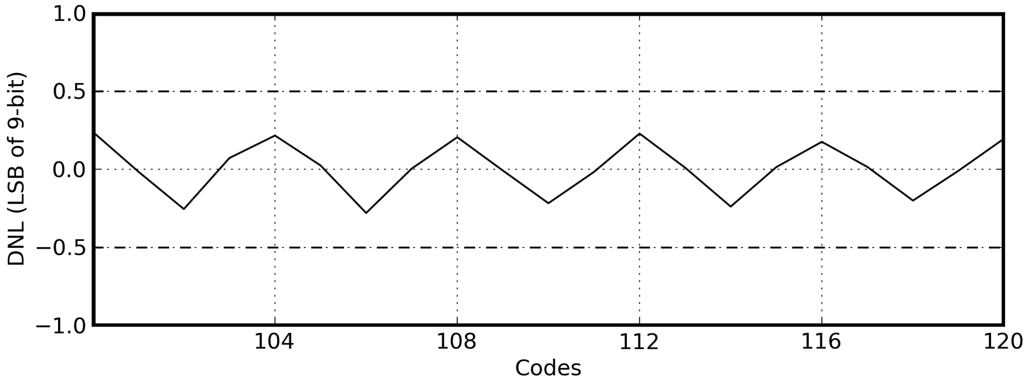


Figure 5.5 – Static DNL performance of TIC ADC in 9-bit Mode

Figure 5.6 shows the INL of the ADC in 9-bit mode. Apart from the lost codes at the start of the output range due to the delay in the comparator, the first few codes have very poor INL performance. This is due to the curving at the start of the ramp due to the RC time constant used in the filter. As the filter frequency is reduced, the DNL performance between 4 consecutive bits is improved, however the curvature and size of the poor INL part at the start of the sequence will increase. For practical use of the converter, the use of these first few codes are avoided since the performance degradation due to using these codes out ways the benefit they bring by the small extension in range.

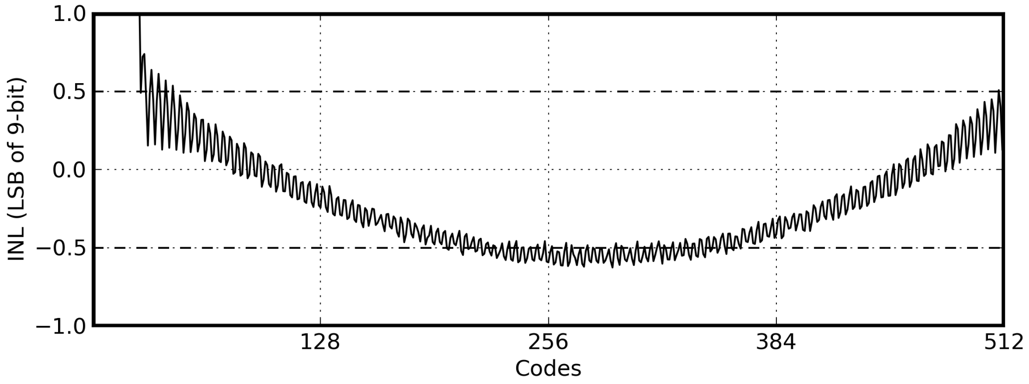


Figure 5.6 – Static INL performance of TIC ADC in 9-bit Mode

As the interpolating frequency is reduced, the DNL performance improves, since the ramp becomes smoother, closer to a real 9-bit ramp made of 512 unique levels. However when this frequency is reduced, the INL performance degrades, since greater overall bowing in the ramp will appear. This frequency should be chosen in a way to balance the INL and DNL performance.

5.4.2 Frequency domain performance

For analysing the performance of this data converter, performance measured in the frequency domain is of much greater interest. Since non-linearities in the system, even if different between the channels will still appear at the output of a frequency domain analysis. The analysis here will begin with looking at the circuit in 7-bit mode to identify the different sources of error, and then the results for all modes of operation will be presented.

Figure 5.7 shows the output spectrum of the ADC in 7-bit mode, with an input tone at 468.75MHz. A high number of bins are used in the FFT to show the harmonic tones in the output. A number of them can be seen above the noise floor. With an ADC with such a high number of channels, distinguishing between THD and SNDR can be very difficult, since the spurious tones due to various channel mismatch characteristics and non-idealities can be at many frequency bins. Figure 5.6 only shows the output spectrum for an input signal at a certain frequency. These different output spectrums were generated for input signals in steps of 25MHz, and the SNDR of the converter at each of these inputs was measured and has been plotted on Figure 5.8.

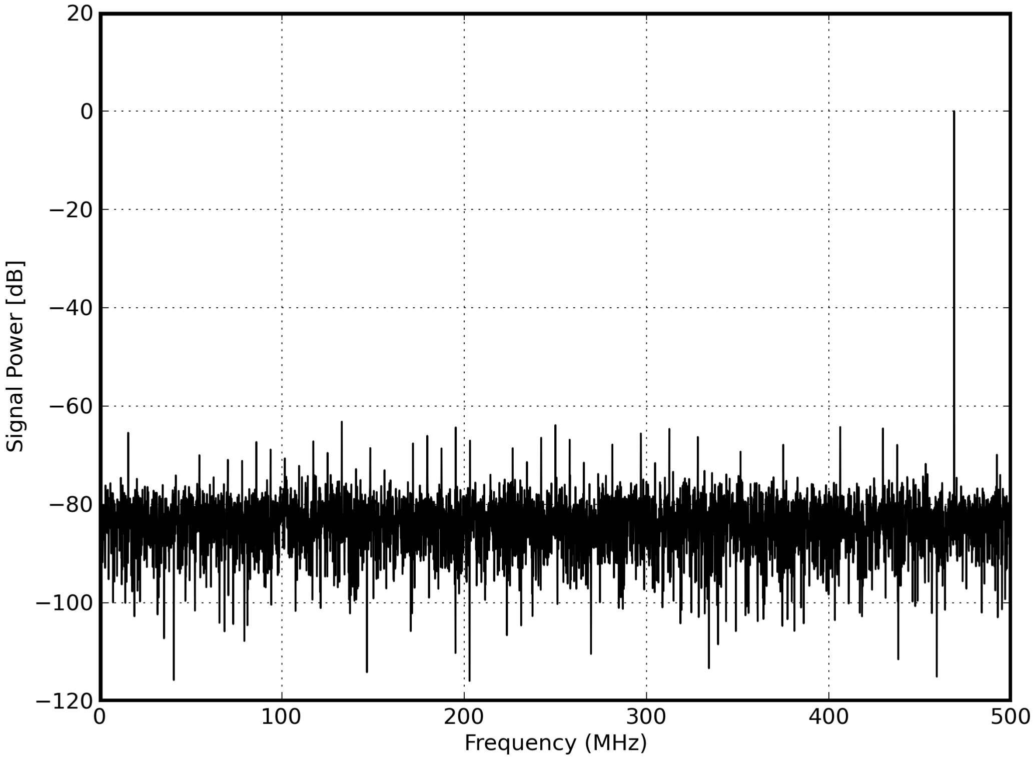


Figure 5.7 – Output spectrum of ADC with 468.75MHz input frequency

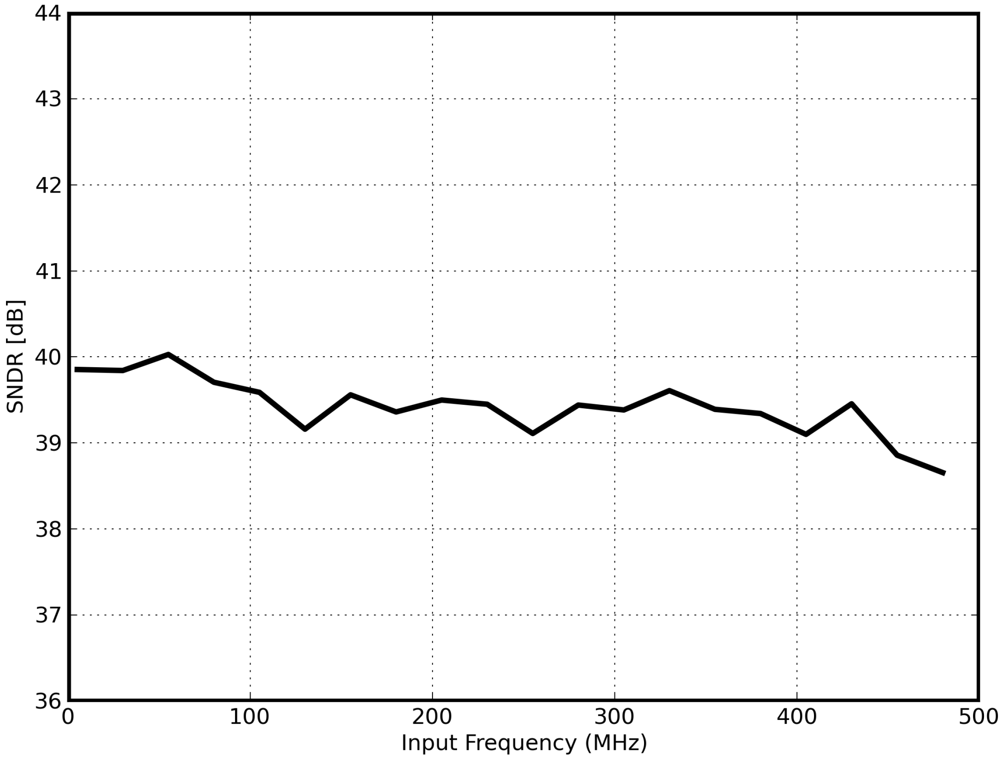


Figure 5.8 – SNDR of ADC vs. input signal frequency in 7-bit mode

It is interesting to see the SNDR curve is relatively constant over input signal band. Most non-idealities which were believed to be the limiting factor in performance, such as timing-skew between channels, sampling bandwidth mismatch between channels, and settling time errors, all have strong dependence on input frequency. However looking at this result one could conclude none of the above are limiting factors in performance. Therefore frequency independent sources of error, such as thermal noise of the comparator, kT/C sampling noise, and mismatch in ramp-generator switches must be some of the more dominant sources of noise in the system.

The sampling capacitor in the S/H circuit is programmable in magnitude. This capacitor sized can be doubled, hence improving the kT/C sampling noise. By doing this, the settling time noise is expected to increase. Figure 5.9 shows this change in sampling capacitor size. The low frequency performance has improved, but not by a huge amount, suggesting kT/C noise was not the dominant source of noise, however the high-frequency performance has degraded significantly, due to the block no longer meeting the settling time requirements.

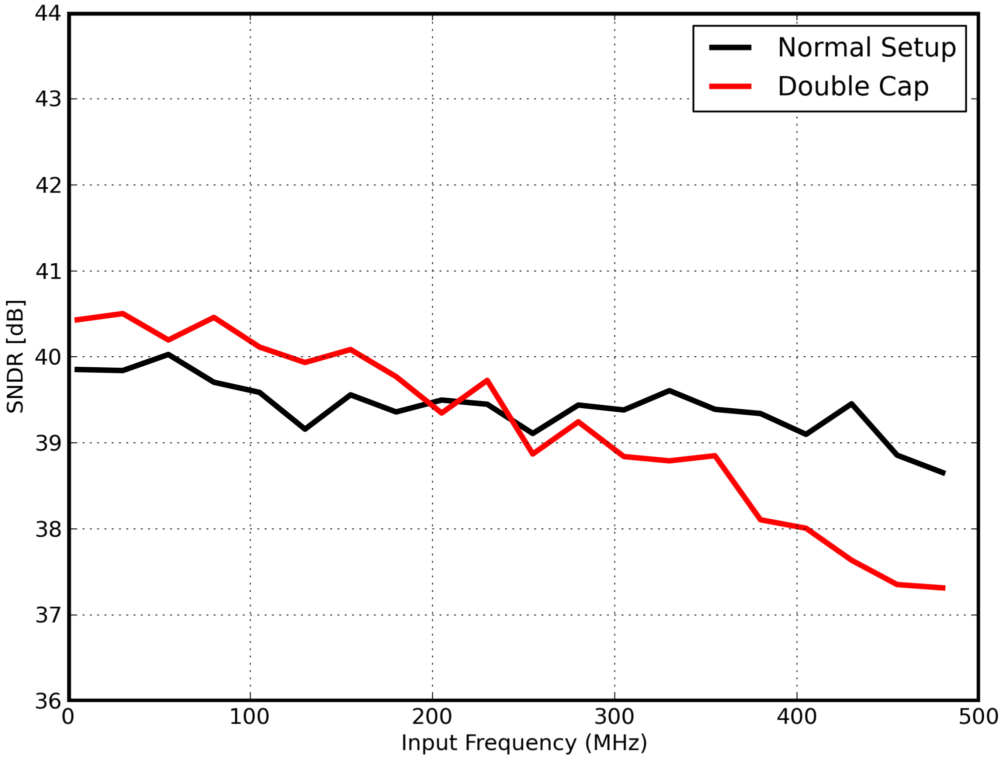


Figure 5.9 – 7-bit SNDR vs. input signal frequency with capacitor adjustment

Meeting the timing skew requirements between channels was well understood as one of the great challenges in meeting the requirements of the overall ADC. Looking at the output spectrum, it would appear that this requirement has been met, and that the output is dominated by non-frequency dependent noise and channel mismatch effects. However it is important to understand and confirm this further. The power supply for the front-end clock-trees was separated and brought to its own pin on the package. This then allows for us to adjust this supply. Lowering this supply will reduce the clock edge speed of the buffers, and hence allow the Vt mismatch in devices to create larger timing skew between the channels. This supply was reduced to 0.9V rather than the nominal 1.2V and the sweep of input frequency was repeated.

Figure 5.10 shows the SNDR performance vs. input frequency with this supply reduced.

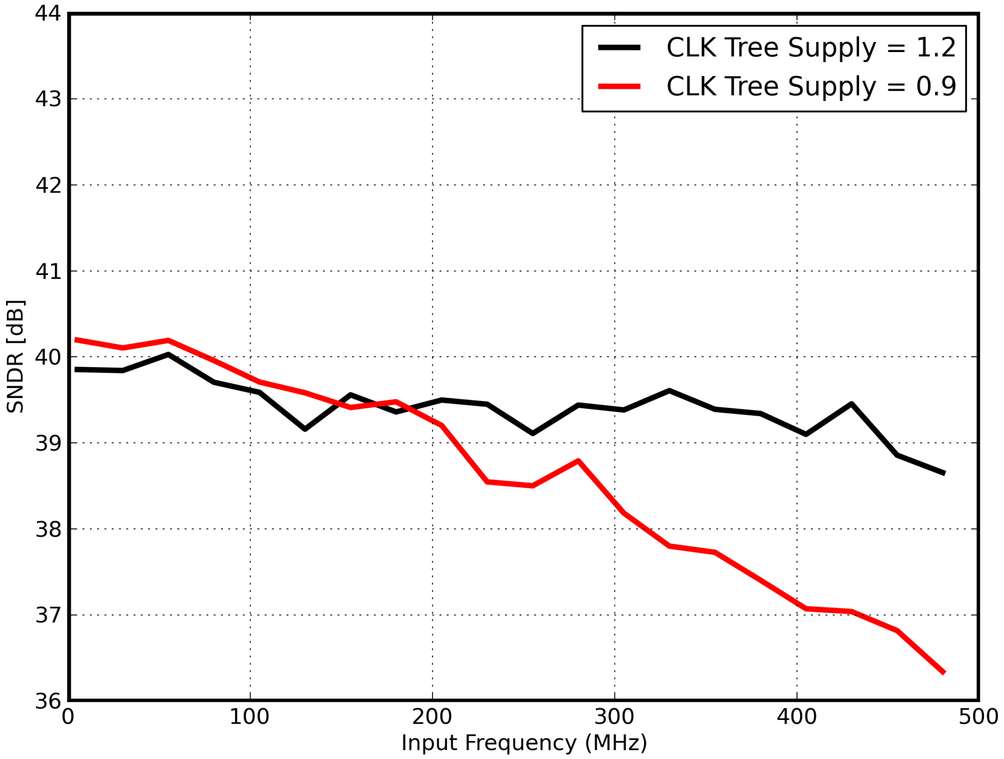


Figure 5.10 – High-frequency SNDR with reduction in clock-tree supply voltage

Clearly as this supply is reduced, the clock edge speed has been reduced and the effective timing-skew between channels has increased. It should be noted that with the slower clock edges, the clock is also more sensitive to supply noise, and this may have also played a part in the degradation in performance.

Figure 5.11 shows the output spectrum of ADC setup in 9-bit mode with an 110MHz input signal tone at full power. The output characteristic is different to the 7-bit mode. Firstly there are more low power tones, apart from the frequency dependent elements. This is due to the ripple in the ramp signal. Secondly, the spurious tones are dominated by the 3rd harmonic at 80MHz. This shows that in 9-bit the linearity performance is limited by the non-linearity of the ramp-generator.

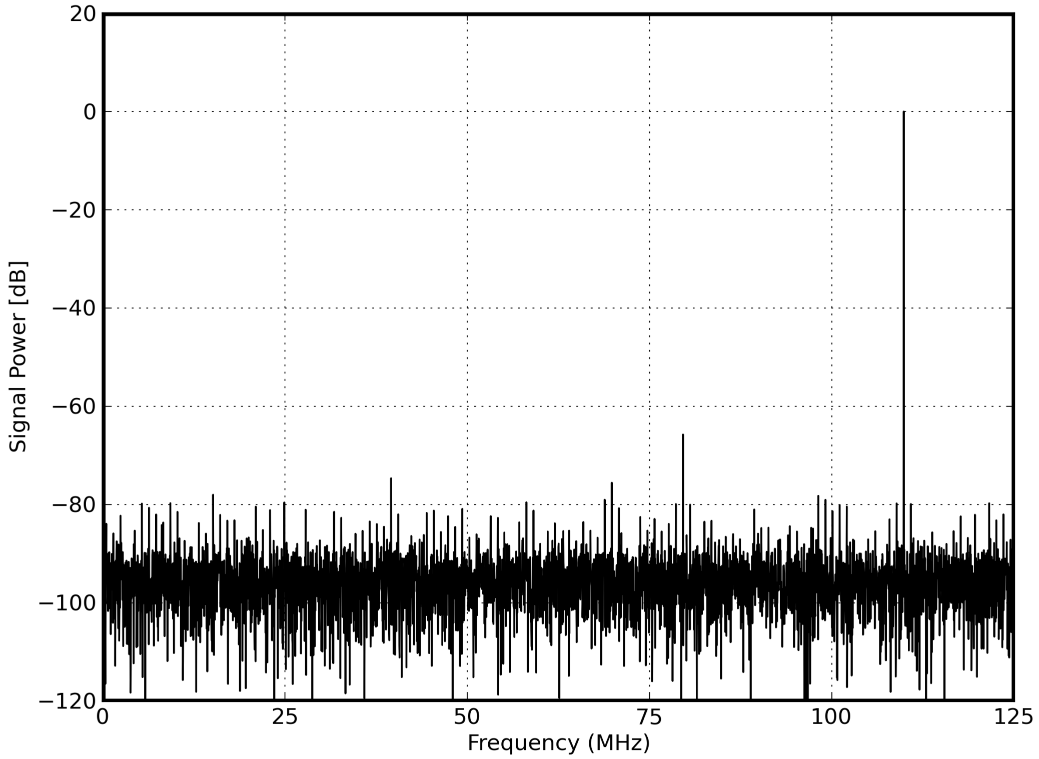


Figure 5.11 – Output spectrum of ADC in 9-bit mode with 110MHz input

The flat SNDR performance of the converter for an input frequency sweep highlights that the non-linearity elements of main concern in time-interleaving ADCs have not limited the performance in this block. This test is then repeated for 8-bit and 9-bit mode operation formally for the default configurations of the system. The results are shown in Figure 5.12. It can be seen that this flat profile is achieved in different modes of operation.

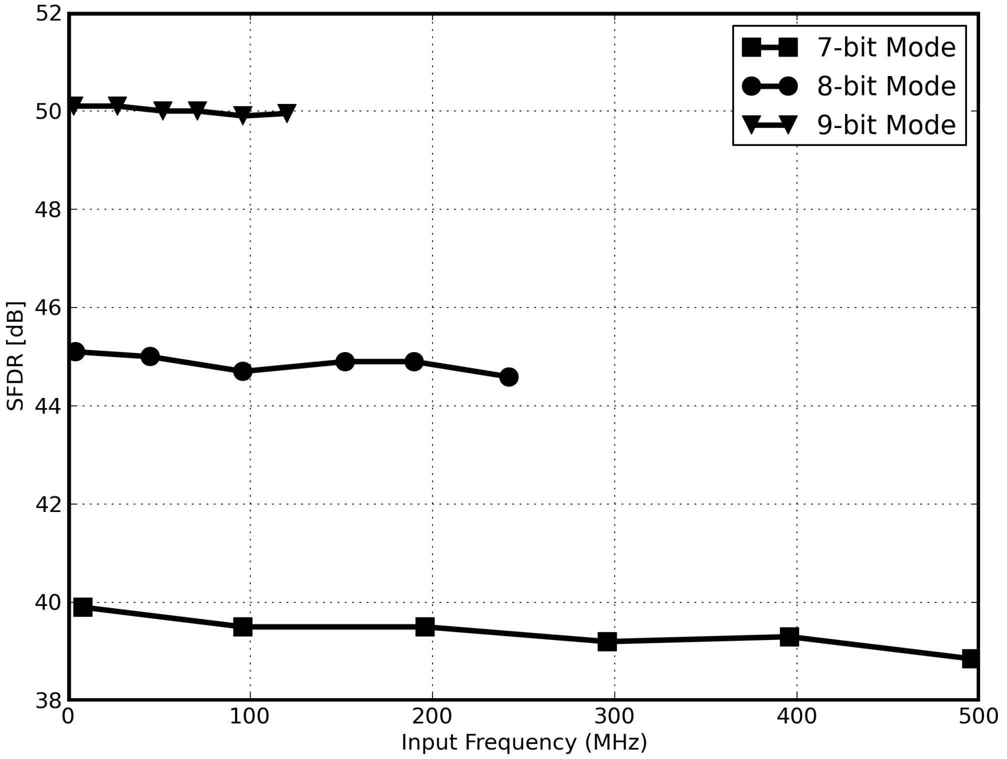


Figure 5.12 – SFDR vs. input frequency for all modes of operation

5.5 Summary of Performance

Table 5.1 summarises the performance and specification of the implemented ADC in different modes of operation. The digital power consumption is dominated by the global counter drivers, and the DRAM readout circuit including the folding circuit. In lower modes of operation the number of counter bits are reduced, however the system requires operating at a much higher rate. The count generator operates at the same frequency of 1GHz in all modes of operation. The majority of the analog power consumption is static and mode independent, however due to some dynamic elements and programmability, this value is slightly different between different modes of operation. The ADC has the ability to operate at 10-bit mode, however in this mode due to need for further smoothing from the interpolation filter, a much greater signal range is lost, and the effective gain of using the resolution is diminished. In all tests, and for normal operation the ADC input range has been reduced to 1.1V to accommodate for the nonlinearity at the start of the range. In 7-bit mode, the full 1.2V input range can be used, since no interpolation filter is required, however in characterisation of the ADC the same 1.1V swing was used for all modes to allow a fair comparison between the modes of operation. The converter achieves a sub 400fJ/step in all modes of operation. Crucially a very small input capacitance is achieved in all modes of operation.

Table 5.1 – Summary of performance

Resolution	7-bits	8-bits	9-bits
Conversion Rate	1 GS/s	500 MS/s	250 MS/s
INL	0.28 LSB	0.37 LSB	0.63 LSB
DNL	0.26 LSB	0.31 LSB	0.58 LSB
Input Range	1.1 Vp-p differential		
SNDR (@ ~Nyq.)	38.85dB	44.59dB	49.95dB
Voltage Supply	1.2V		
Analog Power Consumption	8.3mW	8.4mW	8.9mW
Digital Power Consumption	18.2mW	17.6mW	16.4mW
Total Power Consumption	26.5mW	26mW	25.3mW
FOM	364fJ/step	380fJ/step	399fJ/step
Embedded Input Capacitance	430 fF	430 fF	560 fF
Active Die Area	850μm x 650μm (0.55mm^2)		
Technology	0.13u CMOS		

5.6 Conclusions

A total of 5 parts were packaged and available for testing. The converter manages to meet the expected performance in different modes of operation. It is interesting to see how the converter meets the requirements for most signal frequency dependent artefacts. From the measured results, it can be argued that the noise performance, or generally wide-band signal independent non-linearity has room for improvement.

The non-linearity of the ramp-generator, due to the self-capacitance of the array, was much smaller on silicon than expected, however it is still the limiting factor in the linearity performance of the converter. Higher resolution modes were tested with higher interpolation filter frequencies, since the ramp generator was generally found to be very linear. In the INL plot of the 9-bit mode operation, the non-linearity of the ramp-generator itself can be seen, however this performance was better than simulation, though the chip is in a particular process corner. On the other hand the mismatch in the switches in the ramp-generator, which would result in near white-noise on the output, seems larger than expected. The dominant sources of noise are believed to be the mismatch of the switches in the ramp-generator and the comparator thermal noise.

The converter meets the specifications originally set out, achieving expected results functioning in 7-bit at 1GS/s, 8-bit at 500MS/s and 9-bit at 250MS/s. This is done with meeting the input capacitance requirements, with a competitive 400fJ/step Figure of Merit in all mode of operation.

Chapter 6 - Conclusions

6.1 Introduction

This chapter looks to critically assess the work carried out during the project, and the proposed architecture of the Time-interleaved Counter ADC. First the silicon implementation, as an example implementation of this architecture, will be looked at more critically and the performance and possible improvements to the sub-blocks will be discussed. If the implemented ADC can be considered an example implementation of the proposed architecture, its different modes of operation will then be compared with the current state of the art.

The case was made that this architecture is suited to work to thinner geometries then 0.13µ. The effect of process scaling to 40nm and beyond, and its effect on choice of frequencies, number of rows and general design strategies will be presented. Some estimations on how this architecture would scale in technology, and what performance should be expected from this architecture in 40nm technology and beyond, will be looked at. The thesis will then be concluded.

6.2 Critical Assessment of Work

A reconfigurable TIC ADC was implemented in 0.13µ CMOS and the performance of the block was measured on silicon and presented. Now that this initial design cycle is complete, this section looks over some of the design decisions made during the project, assessing if architecturally correct decisions were made, and at the circuit level what changes can be made to the current design to improve performance.

6.2.1 Choice of number of rows and clock speeds

With the target performance specification for resolution and sampling rate, a number of different 'number-of-row' and 'clock-speed/channel-speed' options were available. In this implementation we chose 128-rows, with a 1GHz backend counter clock. The neighbouring alternatives were 256-rows, with a 500MHz backend clock, or alternatively 64-rows with a 2GHz backend clock. This choice of breakdown plays a key part in the overall performance and efficiency of the converter for size, power, input capacitance and inevitably FoM. Of course this decision is required to be made from the start, since all other block specifications depend on this.

The initial study suggested that the 256-row option would have been more power efficient, primarily due to the saving in the digital backend and counter driver. This option was dismissed due to the larger expected area and the challenges in meeting clock-skew between the channels. On the first point, as the pitch of a row is reduced, the area usage of each row becomes less efficient. This would have implications in the overall area of the block, and hence parasitic capacitances. As a second order effect this would then increased the power consumption. The second issue of clock-skew was a major worry in the performance of the block prior to design. As the number of rows, and area of the block is increased, the number of devices in the clock-tree and clock-distribution requires increasing which in turn increases the statistical clock skew between channels. Minimising the clock skew was deemed critical in the design, and the choice of 256-rows was deemed unable to meet this requirement and was dismissed. Now that the block has been implemented, it would appear that the clock-skew requirements have been met with some margin, suggesting that the 256-row option might have been feasible. However due to the secondary effects of power consumption, the merit in choosing this breakdown is questionable.

Alternatively the 64-row option, with the 2GHz backend clock could have been considered. Operating 0.13μ logic at 2GHz can be challenging, and would have an overhead in power consumption. This approach would arguably lead to a smaller area, however the power consumption of all digital and dynamic parts of the circuit would grow considerably. It would appear that the choice of number of rows and backend counter was correct, however some requirements such as timing-skew might have been over designed, increasing the power consumption to a higher number than what would have been required, however insufficient parts have been measured, especially corner lots, to confirm this.

6.2.2 Design of Ramp Generator

Looking at the chip results, at least for the process corner parts we have received, it would appear that variable output resistance problem discussed in Chapter 4 which led to the use of interpolation filters has not been as big as predicted in simulation.

Going back over the design to try and understand this, the better performance than expected can be due to two reasons. Firstly, the study of the possible performance capabilities of the Ramp Generator was based on models of the parasitic capacitances expected in the ramp-generator and AFE of a row. Once implemented, since this was perceived to be a problem, extra care was taken to reduce these parasitics, and the overall size of the analog section of the block, and effective switches driving the ring in practise was much smaller than the original estimates had predicted, hence the better performance can be seen. Secondly the fabricated parts are only in a particular process corner, while the design was implemented for functionality under full process corners. This includes transistor process corners, and parasitic extraction process corners. Looking back over the performance, potentially rather than only 128-unique levels in the ramp-generator, 256 unique levels could have been implemented, allowing native operation in 8-bit mode without the need for interpolation, and improving the performance of 9-bit mode, and potentially enabling useful operation in the 10-bit mode as well. It is difficult to be exact without full implementation, but such a modification is expected to increase the power consumption by around 3mW to 4mW, however if successful would allow for an extra mode of operation.

This change can only be proposed, if it can be shown that the 8-bit ramp-generator can achieve the linearity requirement under all corners. It would be incorrect to try and conclude from one wafer measurement that we have over designed the ramp-generator, and can make this adjustment. There is no question that designing for process corners yields less efficient solutions at typical runs, but design is done in this way for a reason, to allow real manufacturability of blocks. For a fair comparison of this architecture the margin in design for process variation should be left-in when comparing to more conventional architectures.

Separate to the linearity issue, the mismatch of the switches in the resistor ring contributes to reduction of DNL in the ramp signal. Of course this loss of linearity cannot be seen at the DNL of the ADC, since the linearity profile for each channel will be different since each switch resistance contribute towards a different part of a ramp for each channel, and this mismatch in a way acts similar to dynamic-element-matching, where the mismatch is dithered (due to the number of channels) into noise. As seen from measurement the performance can be improved by improving the noise floor of the converter. This mismatch will increase this noise floor. Based on the linearity results, the size of the switches could have been increased, allowing for further curving on the ramp, firstly better matching and secondly less contribution from the switches to the ramp resistance, to improve this noise contribution from the ramp-generator switches.

6.2.3 Comparator Design

Based on the measurement, the noise is one of the main limiting factors in the performance of the block. This is not necessarily fully limited by the comparator, but never the less the comparator plays a big part in this. Apart from limited performance due to the noise, the design of the comparator is not efficient for use of power and other performance criteria. Its delay contributes to lost codes, while at higher resolution modes, band-limiting is used to be meet the noise requirements at the cost of power. The comparator is an area with room for improvement.

As explained in the design chapter, a folded cascode amplifier was used as the comparator, achieving high-gain, however the bandwidth required limiting at high resolution modes. Alternatively a clocked comparator can be used [64]. Another approach is the use of a zero-crossing detector comparator, or basic dynamic comparator which does not require static current for biasing. In recent years zero-crossing based pipelined ADCs have been researched [67][68]. These pipelines use a zero-crossing comparator, and an effective ramp to perform the multiplication-by-two operation. A similar style comparator can be used in this ADC. A trial implementation of such a comparator shows good performance at a much lower power consumption, with potentially smaller delay. The downside to these comparators are the poor power supply rejection, and introduction of power supply noise. In such a parallel time-interleaved system, row to row isolation is critical. Ideally a comparator with constant power consumption with high power supply rejection is desirable. The dynamic comparators have variable power consumption and low power supply rejection. This does not necessarily mean that they cannot be used, however to reduce the power consumption of this block through these means, extra care is requires around power supply routing and isolation of the rows from one another.

6.3 Performance Summary

A highly time interleaved ADC was realised in 0.13µ CMOS which can operate in a 7-bit 1GS/s, 8-bit 500MS/s and 9-bit 250MS/s operational modes. In the introduction the need for such converters which can operate over a number of different modes of operation, was explained. But crucially such a converter should achieve these cross architecture specifications in comparable performance. Figure 6.1 shows the implemented ADC in purple verses published architectures in the last 10 years. The ADC managed to travel across 3 different architectures, and achieve performance (balance of sampling rate vs. resolution) comparable to the state of the art. This would suggest that the architecture proposed is a real candidate for use in applications as an alternative to the more conventional architectures, but more importantly can actively trade off resolution and speed to cover a range of performance specifications which previously could only be covered by A/D converters of different architectures.

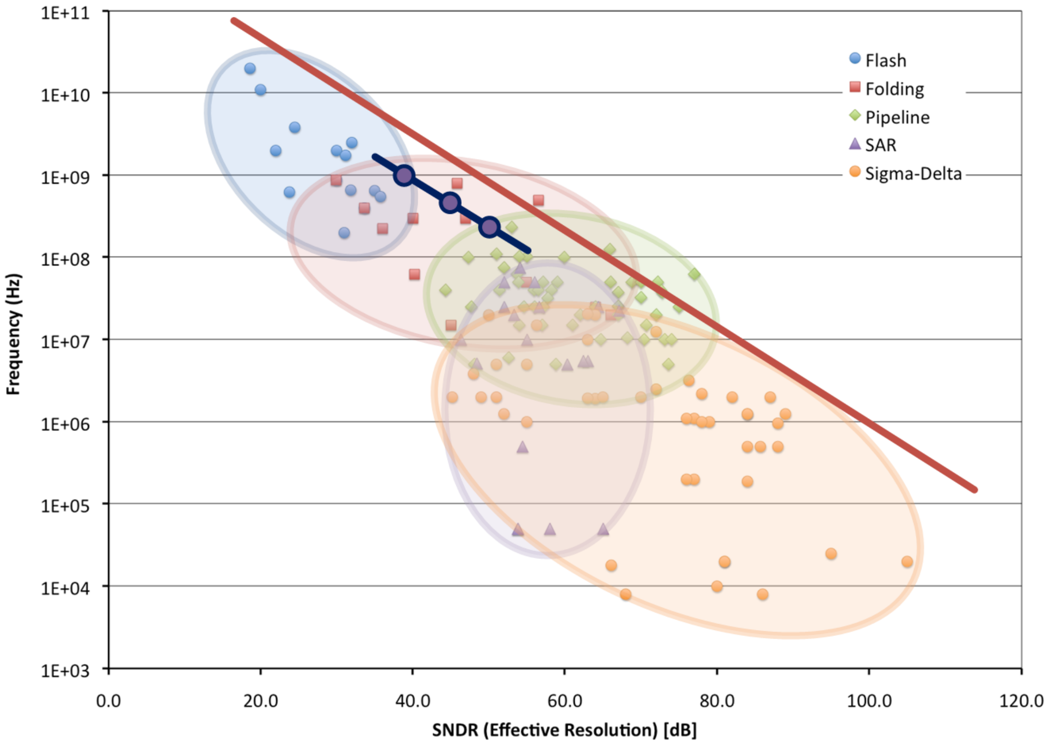


Figure 6.1 – Comparison of implemented ADC verses publish work

Figure 6.1 shows the strength of the architecture as a whole, and how it performs compared to the more conventional architectures. Figure 6.2 compares the performance of the new converter in terms of power efficiency. Here we can see that the TIC ADC again is very close to the red-line, and in fact out performs almost all architectures it competes against in terms of performance and speed. Looking at both Figure 6.1 and 6.2 one can conclude that the proposed TIC ADC firstly achieves the right ratio of sampling frequency versus resolution for today's communication applications, and secondly it does this with great power efficiency. Maybe most importantly is the fact that this converter is no longer a dot on the graph but a line, achieving a range of performances.

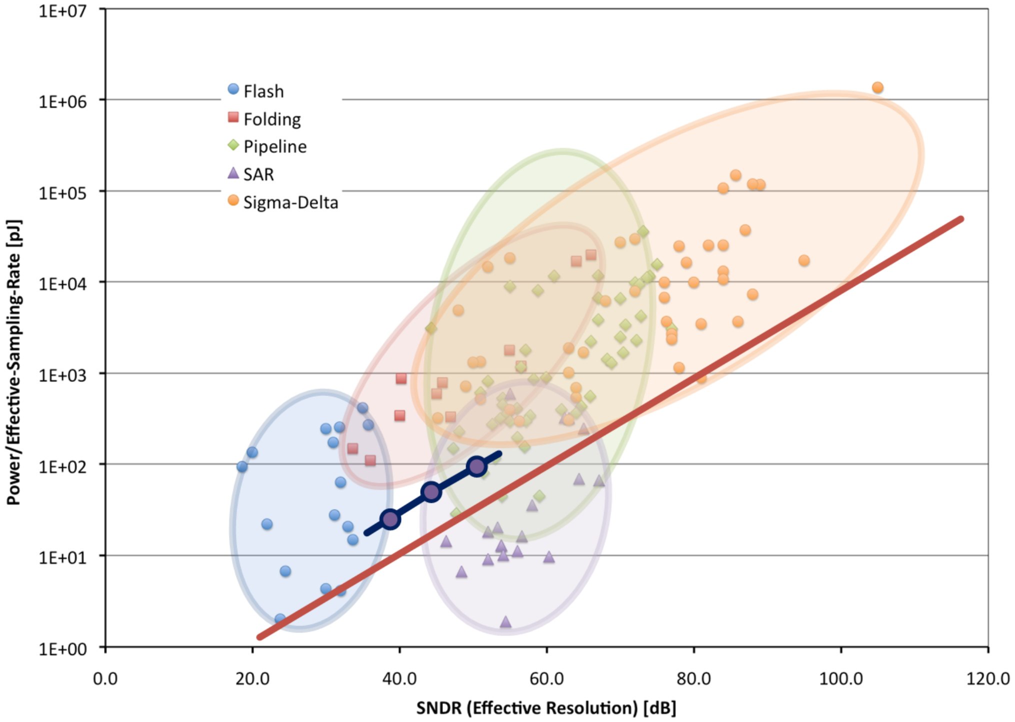


Figure 6.2 – Comparison of TIC ADC with other architectures for power efficiency

6.4 Technology Scaling

The architecture proposed is a heavily digital based system, with very few analog circuits. Clearly this architecture would greatly benefit from technology scaling. This section will use data from a 40nm G process as a reference for scaling. The effect of this scaling from 0.13µm to 40nm will be looked at in two ways: Firstly, if we assume in the architecture, number of rows and clock speeds remain unchanged from 0.13µm to 40nm, how this architecture would perform, and what figure of merit would it achieve in 40nm. Secondly, if aiming for 40nm as a fresh design, how could the top-level segmentation be changed to improve performance, or realise new specifications.

Moving from 0.13μ to 40n, a 3x improvement in gate capacitance for relative standard-cell drive strength is expected. Furthermore the supply voltage will move from 1.2V to 0.9V. Meaning most digital circuitry dominated by driving gate devices will scale by a factor of 4 in power. Over 2/3 of the total power consumption of the implemented block is in digital circuitry. On area, for the implemented ADC, for each row, 30% and 35% of the area is used for metal capacitors (for sampling and boot-strap switches) and standard cell logic respectively. This is close to 2/3 of the total area. Both these types of devices would heavily scale in technology, and major reduction in area should be expected. This again reduces the routing distance and parasitics and would help with the reduction in power consumption. Moving down in geometry, the Ron of switch devices, relative to their parasitic capacitances will improve. This will have huge positive implications for the global ramp generator, where its ramp swing, and hence signal swing is limited by the Ron of the switches, while its linearity is related to parasitic capacitances. This improvement would suggest, despite the reduction in power supply voltage, the actual signal swing can be maintained, meaning the sampling capacitor in terms of size, and noise requirement on the comparator remain unchanged. If the current design was directly ported to 40nm, with no architectural changes, it is not unfeasible to expect a 2-digit FoM for power, effectively in the sub 100fJ/step space. While the area of the block could scale by a factor of 3, to near 0.15mm^2.

These possible performance specifications are quite impressive, but what is potentially more interesting is how the architecture can be adjusted to take advantage of process scaling with the aim to improve the performance, or simplify design and also allow for more trade-off in the specification space. As we move to thinner geometries, it becomes possible to clock the backend counters at much higher rates, in 40G at 2GHz, or even 4GHz. This would allow a reduction in the number of rows to 64 or 32 for the same specification space. This would reduce the parasitics on the block, and allow for more efficient layout, generally taking advantage of the technology speed characteristics better. One of the weaknesses of this architecture is the latency from sample to digital output. This latency is exponentially proportional to the resolution of operation. By increasing the backend clocking frequency, this latency can be improved. Figure 6.3 shows the power efficiency of the implemented TIC ADC compared to published work from the last decade, and also shows the predicted performance of the TIC ADC, if the implemented block in 0.13μ was directly ported to 40nm, and also how the performance could be improved further by changing the clock frequency and number-of-channels to suit the 40nm technology.

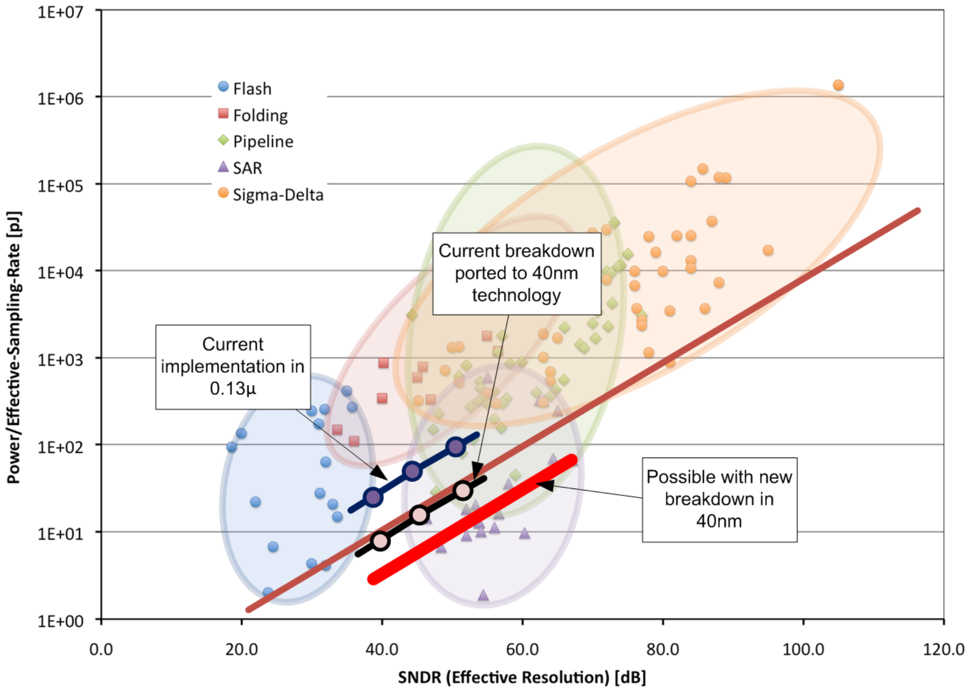


Figure 6.3 – Performance improvement of TIC ADC with technology scaling

6.5 Architectural Directions

As mentioned above, one of the weaknesses of the TIC architecture is the latency of the block. The latency is proportional to the number of counter values each channel requires to complete. In a single channel counter ADC, the conversion rate is proportional to this. There to increase the sampling rate, or to reduce the conversion time, multi-slope architectures are commonly used [54]. This can be realised in two ways, with the use of a coarse and fine ramp, or with parallel conversion. In the TIC architecture, parallel ramps can be used to reduce the conversion time per channel, and in effect reduce the latency. The concept can be thought of as sub-ranging of the TIC ADC.

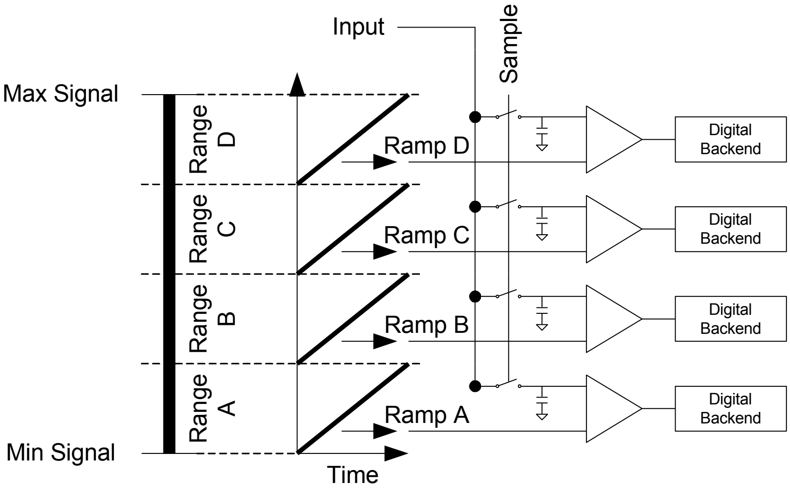


Figure 6.4 – Circuit diagram of single channel for sub-ranging concept

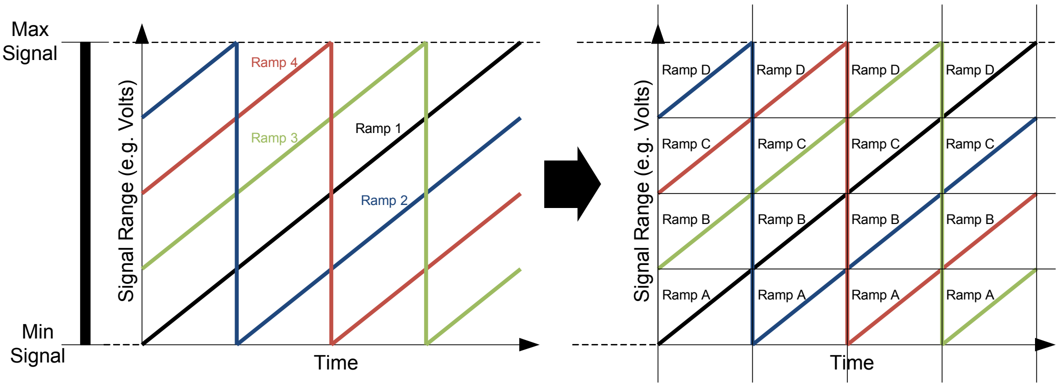


Figure 6.5 – Ramp generation for sub-ranging concept

Figure 6.4 shows a single channel of the proposed sub-ranging TIC ADC. One sample and hold circuitry has been replaced with 4 in this example. The four samples of the input are simultaneously compared to 4 different sub-ranges of the full-signal range, using 4 sub-ramps. In this example, the effective conversion rate of each channel has been increased by 4, while consuming 4 times the area and power. In a way for the converter implemented in this work, if such an architecture were to be adapted, the overall resolution and conversion rate of whole converter would not change, however the latency would be reduced by a factor of 4. In effect 4 rows simultaneously do the job of one row, but at 4 times the rate. Figure 6.5 shows how the output of the current ramp-generator, generating 4 out-of-phase ramps can be arranged to generate Ramp A to D, using multiplexing switches.

This architecture increases the input capacitance of the block, since multiple sampling circuits are present at the input simultaneously, however can be used to trade-off input capacitance for latency, if needed for certain architectures.

6.6 Conclusions

This work set out exploring the concepts of time-interleaving but applied to one of the slowest and simplest ADC architectures, the counter ADC. The architecture of a time-interleaving counter (TIC) ADC was described, and a prototype data converter in the architecture was realised in silicon and measured. The architecture allows for simple re-configurability by trading resolution for sampling rate using basic adjustments of clock frequency and some changes in the analog sub-blocks. The implemented block achieves re-configurability in a performance space never previously possible. Furthermore each individual performance node achieves good FoM compared to more conventional architectures, and it can be shown if this architecture were to be scaled to thinner geometries, it will achieve a FoM at state-of-the-art over its entire conversion space. A large percentage of the design is made of logic elements, and the architecture fundamentally, compared to more conventional architectures, moves more of the design from analog to digital, becoming a prime candidate to take advantage of technology scaling.

The architecture can also achieve relatively low area, and despite the use of time-interleaving, very low input capacitance, especially compared to alternatives in the high-frequency space of operation. The architecture's weakness is in latency, exponentially proportional to the resolution, however moving down in geometries, greater backend clocking frequencies will help to reduce this number. The architecture was originally proposed for multi-standard multi-PHY communication systems in 10s to 100s of MHz of signal bandwidth. These systems are commonly implemented as an SoC. This architecture achieves such re-configurability with great efficiency in power and area, and directly benefits from technology scaling commonly found in SoC applications. Also porting from technology node to the next can be done quickly, since very little custom design blocks are needed. This architecture can be considered a serious alternative to conventional A/D converter architectures.

References:

[1] Shannon, C. E.; “A Mathematical Theory of Communication”, The Bell System Technical Journal, Vol 27, pp. 379-423, July 1948

[2] Van Nee. R.; Prasad, R.; “OFDM For Wireless Multimedia Communication”, Artech House Publishers, 2000, ISBN 0-89006-530-6

[3] Proakis, J.; “Digital Communications”, McGraw-Hill Science; 4th Edition, 2000, ISBN 0-07232-111-3

[4] Lin, C.-H.; van der Goes, F.; Westra, J.; Mulder, J.; Lin, Y.; Arslan, E.; Ayranci, E.; Liu, X.; Bult, K; “A 12b 2.9GS/s DAC with IM3 ≪-60dBc beyond 1GHz in 65nm CMOS”, Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International, p-p 74-75

[5] V. D. Plassche, R.; “CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters”, Springer; 2nd Edition, 2003, ISBN 1-44195-367-1

[6] Demler, M. J.; “High-Speed Analog-to-Digital Conversion”, Academic Press, 1991, ISBN 0-12209-048-9

[7] Gustavsson, M.; Wikner, J.; Tan, N.; “CMOS Data Converters for Communications”, Springer International Series in Engineering and Computer Science, 2000, ISBN 0-79237-780-X

[8] Murmann, B.; “ADC Performance Survey”, Available online: http://www.stanford.edu/~murmann/adcsurvey.html

[9] Louwsma, S. M.; van Tuijl, A.J.M.; Vertregt, M.; Nauta, B.; “A 1.35 GS/s, 10 b, 175 mW time-interleaved AD converter in 0.13 um CMOS”, IEEE J. Solid-State Circuits April 2008, p-p 778-786

[10] Alpman, E.; “A 7-bit 2.5GS/sec Time-Interleaved C-2C SAR ADC For 60GHz Multi-Band OFDM-Based Receivers”, PhD Thesis, Carnegie Mellon University, August 2009

[11] Kurosawa, N.; Kobayashi, H.; Maruyama, K.; Sugawara, H.; Kobayashi, K.; “Explicit Analysis of Channel Mismatch Effects in Time-Interleaved ADC Systems”, IEEE Transcations on Circuits and Systems – I: Fundamental Theory and Applications, Vol. 48, No. 3, March 2001, p-p 261-271

[12] Quiquempoix, V.; Deval, P.; Barreto, A.; Bellini, G.; Markus, J.; Silva, J.; Temes, G.C.; “A low-power 22-bit incremental ADC”, Solid-State Circuits, IEEE Journal of, Volume: 41 , Issue: 7, 2006, p-p 1562-1571

[13] Findlater, K.; Bailey, T.; Bofill, A.; Calder, N.; Danesh, S.; Henderson, R.; Holland, W.; Hurwitz, J.; Maughan, S.; Sutherland, A.; Watt, E.; “A 90nm CMOS Dual-Channel Powerline Communication AFE for Homeplug AV with a Gb Extension”, International Solid-State Circuits Conference, 2008, Digest of Technical Papers, p-p 464-628

[14] Deergha Rao, K.; Murthy, T.S.N.; “Analysis of Effects of Clipping and Filtering on the Performance of MB-OFDM UWB Signals”, Digital Signal Processing, 2007 15th International Conference on, 2007, p-p 559-562

[15] Da Dalt, N.; Harteneck, M.; Sandner, C.; Wiesbauer, A.; “On the jitter requirements of the sampling clock for analog-to-digital converters”, Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on Volume: 49 , Issue: 9, p-p 1354-1360

[16] Arkesteijn, V.J.; Klumperink, E.A.M.; Nauta, B.; “Jitter requirements of the sampling clock in software radio receivers”, Circuits and

Systems II: Express Briefs, IEEE Transactions on, Volume: 53 , Issue: 2, 2006, p-p 90-94

[17] Manoj, K.N.; Thiagarajan, G.; “The effect of sampling jitter in OFDM systems”, Communications, 2003. ICC '03. IEEE International Conference on Volume: 3, 2003, p-p 2061-2065

[18] Louwsma, S.M.; van Tuijl, E.J.M.; Vertregt, M.; Nauta, B.; “A Time-Interleaved Track & hold in 0.13 µm CMOS sub-sampling a 4 GHz signal with 43 dB SNDR”, Custom Integrated Circuits Conference, 2007. CICC '07. p-p 329-332

[19] Iroaga, E.; Murmann, B.; “A 12-Bit 75-MS/s Pipelined ADC Using Incomplete Settling”, Solid-State Circuits, IEEE Journal of, Volume: 42, Issue: 4, 2007, p-p 748-756

[20] Varzaghani, A.; Yang, C.-K.K.;”A 4.8 GS/s 5-bit ADC-Based Receiver With Embedded DFE for Signal Equalization”, Solid-State Circuits, IEEE Journal of, Volume: 44 , Issue: 3, p-p 901-915

[21] Murmann, B.; “A/D converter trends: Power dissipation, scaling and digitally assisted architectures”, Custom Integrated Circuits Conference, 2008, p-p 105-112

[22] Choi, M.; Abidi, A.A.; “A 6-b 1.3-Gsample/s A/D converter in 0.35-µm CMOS”, Solid-State Circuits, IEEE Journal of, Volume: 36 , Issue: 12, p-p 1847-1858

[23] Danesh, S.; Holland, W.; Hurwitz, J.; Findlater, K.; Henderson, R.; Renshaw, D.; “A non-uniform resolution step GHz 7-bit flash A/D converter for wideband OFDM signal conversion”, International Symposium on Circuits and Systems, 2009, Proceedings of, p-p 964-967

[24] Yoo, J.; “A TIQ Based CMOS Flash A/D Converter For System-on-Chip Applications”, PhD Thesis, The Pennsylvania State University, Department of Computer Science and Engineering, 2003

[25] Junjie Yao; Jin Liu; Hoi Lee; “Bulk Voltage Trimming Offset Calibration for High-Speed Flash ADCs”, Circuits and Systems II: Express Briefs, IEEE, Transactions on, Volume: 57 , Issue: 2, p-p 110-114

[26] Kijima, M.; Ito, K.; Kamei, K.; Tsukamoto, S.; “A 6b 3GS/s flash ADC with background calibration”, Custom Integrated Circuits Conference, 2009. CICC '09. IEEE, p-p 283-286

[27] Goes, J.; Vital, J. C.; “Systematic Design for Optimisation of Pipelined ADCs”, Springer, International Series in Engineering and Computer Science, 2001, ISBN 0-79237-291-3

[28] Devarajan, S.; Singer, L.; Kelly, D.; Decker, S.; Kamath, A.; Wilkins, P.; “A 16-bit, 125 MS/s, 385 mW, 78.7 dB SNR CMOS Pipeline ADC”, Solid-State Circuits, IEEE Journal of, Volume: 44 , Issue: 12, p-p 3305-3313

[29] Ali, A.M.A.; Morgan, A.; Dillon, C.; Patterson, G.; Puckett, S.; Hensley, M.; Stop, R.; Bhoraskar, P.; Bardsley, S.; Lattimore, D.; Bray, J.; Speir, C.; Sneed, R.; “A 16b 250MS/s IF-sampling pipelined A/D converter with background calibration”, Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010, p-p 292-293

[30] Haoyue Wang; Xiaoyue Wang; Hurst, P.J.; Lewis, S.H.; “Nested Digital Background Calibration of a 12-bit Pipelined ADC Without an Input SHA”, Solid-State Circuits, IEEE Journal of, Volume: 44 , Issue: 10, p-p 2780-2789

[31] Razavi, B.; Sahoo, B.D.; “A 12-Bit 200-MHz CMOS ADC”, Solid-State Circuits, IEEE Journal of Volume: 44 , Issue: 9, p-p 2366-2380

[32] Verma, A.; Razavi, B.; “A 10b 500MHz 55mW CMOS ADC”, Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International, p-p 84-85

[33] Anderson, M.; Norling, K.; Dreyfert, A.; Yuan, J.; “A reconfigurable pipelined ADC in 0.18 µm CMOS”, VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on, p-p 326-329

[34] Audoglio, W.; Zuffetti, E.; Cesura, G.; Castello, R.; “A 6-10 bits Reconfigurable 20MS/s Digitally Enhanced Pipelined ADC for Multi-Standard Wireless Terminals”, Solid-State Circuits Conference, 2006. ESSCIRC 2006. p-p 496–499

[35] Cheng-Chung Hsu; Chen-Chih Huang; Ying-Hsi Lin; Chao-Cheng Lee; Soe, Z.; Aytur, T.; Ran-Hong Yan; “A 7b 1.1GS/s Reconfigurable Time-Interleaved ADC in 90nm CMOS”, VLSI Circuits, 2007 IEEE Symposium on, p-p 66-67

[36] Young-Ju Kim; Hee-Cheol Choi; Si-Wook Yoo; Seung-Hoon Lee; Dae-Young Chung; Kyoung-Ho Moon; Ho-Jin Park; Jae-Whui Kim; “A Re-configurable 0.5V to 1.2V, 10MS/s to 100MS/s, Low-Power 10b 0.13um CMOS Pipeline ADC”, Custom Integrated Circuits Conference, 2007, p-p 185-188

[37] Mortezapour, S.; Lee, E.K.F.; “A 1-V, 8-bit successive approximation ADC in standard CMOS process”, Solid-State Circuits, IEEE Journal of, Volume: 35 , Issue: 4, p-p 642-646

[38] Sauerbrey, J.; Schmitt-Landsiedel, D.; Thewes, R.; “A 0.5-V 1-μW successive approximation ADC”, Solid-State Circuits, IEEE Journal of, Volume: 38, Issue: 7, 2003, p-p 1261-1265

[39] Suarez, R.E.; Gray, P.R.; Hodges, D.A.; “All-MOS charge-redistribution analog-to-digital conversion techniques.”, Solid-State Circuits, IEEE Journal of, Volume: 10 , Issue: 6, 1975, p-p 379-385

[40] Craninckx, J.; Van der Plas, G.; “A 65fJ/Conversion-Step 0-to-50MS/s 0-to-0.7mW 9b Charge-Sharing SAR ADC in 90nm Digital CMOS”, Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers, p-p 246-247

[41] Ying-Min Liao; Tai-Cheng Lee; “A 6-b 1.3Gs/s A/D Converter with C-2C Switch-Capacitor Technique”, VLSI Design, Automation and Test, 2006 International Symposium on, 2006, p-p 1-4

[42] Yee, Y.S.; Terman, L.M.; Heller, L.G.; “A two-stage weighted capacitor network for D/A-A/D conversion”, Solid-State Circuits, IEEE Journal of, Volume: 14 , Issue: 4, 1979, p-p 778-781

[43] Zhao, K.-Q.; Amir, S.; Meng, X.-Z.; Ali, M.; Gustafsson, M.; Ismail, M.; Rusu, A.; “A reconfigurable successive approximation ADC in 0.18μm CMOS technology”, Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on, 2008, p-p 646-649

[44] Yip, M.; Chandrakasan, A.P.; “A resolution-reconfigurable 5-to-10b 0.4-to-1V power scalable SAR ADC”, Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011, p-p 190-192

[45] Yi Ke; Peng Gao; Craninckx, J.; Van der Plas, G.; Gielen, G.; “A 2.8-to-8.5mW GSM/bluetooth/UMTS/DVB-H/WLAN fully reconfigurable CTΔΣ

with 200kHz to 20MHz BW for 4G radios in 90nm digital CMOS", VLSI Circuits (VLSIC), 2010 IEEE Symposium on, 2010, p-p 153-154

[46] Malla, P.; Lakdawala, H.; Kornegay, K.; Soumyanath, K.; "A 28mW Spectrum-Sensing Reconfigurable 20MHz 72dB-SNR 70dB-SNDR DT ΔΣ ADC for 802.11n/WiMAX Receivers", Solid-State Circuits Conference, 2008. ISSCC 2008, p-p 496-631

[47] Jabbour, C.; Camarero, D.; Van Tam Nguyen; Loumeau, P.; "A 1 V 65 nm CMOS reconfigurable time interleaved high pass ΣΔ ADC", Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on, 2009, p-p 1557-1560

[48] Christen, T.; Qiuting Huang; "A 0.13µm CMOS 0.1–20MHz bandwidth 86–70dB DR multi-mode DT ΔΣ ADC for IMT-Advanced", ESSCIRC, 2010 Proceedings of the, 2010, p-p 414-417

[49] Kochan, R.; Berezky, O.; Karachka, A.; Maruschak, I.; Bojko, O.; "Development of the integrating analog to digital converter for distributive data acquisition systems with improved noise immunity", Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, International Workshop on, 2001, p-p 193.196

[50] Findlater, K.; Henderson, R.; Baxter, D.; Hurwitz, J.E.D.; Grant, L.; Cazaux, Y.; Roy, F.; Herault, D.; Marcellier, Y.;, "SXGA pinned photodiode CMOS image sensor in 0.35 um technology", IEEE ISSCC Dig. Tech. Papers, 2003, p-p 218–219.

[51] Nitta, Y.; Muramatsu, Y.; Amano, K.; Toyama, T.; Yamamoto, J.; Mishina, K.; Suzuki, A.; Taura, T.; Kato, A.; Kikuchi, M.; Yasui, Y.; Nomura, H.; Fukushima, N.;., "High-speed digital double sampling with analog

CDS on column parallel ADC architecture for low-noise active pixel sensor," IEEE ISSCC Dig. Tech. Papers, 2006, p-p 500–501.

[52] Sugiki, T.; Ohsawa, S.; Miura, H.; Sasaki, M.; Nakamura, N.; Inoue, I.; Hoshino, M.; Tomizawa, Y.; Arakawa, T.; "A 60 mW 10b CMOS image sensor with column-to- column FPN reduction", IEEE ISSCC Dig. Tech. Papers, 2000, p-p 108–109.

[53] W. Yang, O.-B. Kwon, J.-I. Lee, G.-T. Hwang, and S.-J. Lee, "An integrated 800x600 CMOS imaging system", IEEE ISSCC Dig. Tech. Papers, 1999, p-p 304–305.

[54] Snoeij, M.F.; Theuwissen, A.J.P.; Huijsing, J.H.; Makinwa, K.A.A.; "Power and Area Efficient Column-Parallel ADC Architectures for CMOS Image Sensors", IEEE Sensors, 2007, p-p 523-526

[55] Sugiki, T.; Ohsawa, S.; Miura, H.; Sasaki, M.; Nakamura, N.; Inoue, I.; Hoshino, M.; Tomizawa, Y.; Arakawa, T.; "A 60 mW 10 b CMOS image sensor with column-to-column FPN reduction", IEEE ISSCC Dig. Tech. Papers, 2000, p-p108-109

[56] Abo, A.M.; Gray, P.R.; "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter", Solid-State Circuits, IEEE Journal of, Volume: 34 , Issue: 5, 1999, p-p 599-606

[57] Cohn, M.; Even, S.; "A Gray Code Counter", Computers, IEEE Transactions on, Volume: C-18, Issue: 7, 1969, p-p 662-664

[58] Storm, G., Henderson, R., Hurwitz, J.E.D., Renshaw, D., Findlater, K., Purcell, M., "Extended Dynamic Range From a Combined Linear-Logarithmic CMOS Image Sensor", Solid-State Circuits, IEEE Journal of, Volume: 41 Issue: 9, 2006, p-p 2095-2106

[59] Oksman, V.; Galli, S.; “G.hn: The new ITU-T home networking standard”, Communications Magazine, IEEE, Volume: 47 , Issue: 10, 2009, p-p 138-145

[60] Goldfisher, S.; Tanabe, S.; “IEEE 1901 access system: An overview of its uniqueness and motivation”, Communications Magazine, IEEE, Volume: 48 , Issue: 10, 2010, p-p 150-157

[61] Yonge, L.; “The HomePlug Powerline Alliance and HomePlug AV Overviews”, Power Line Communications and Its Applications, 2006 IEEE International Symposium on, 2006, p-p 9-10

[62] Aparicio, R.; Hajimiri, A.; “Capacity limits and matching properties of integrated capacitors”, Solid-State Circuits, IEEE Journal of Volume 37, Issue 3, 2002 p-p 384-393

[63] Vecchi, D.; Azzolini, C.; Boni, A.; Chaahoub, F.; Crespi, L.; “100-MS/s 14-b track-and-hold amplifier in 0.18-μm CMOS”, Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European, 2005, p-p 259-262

[64] Snoeij, M.F.; Theuwissen, A.J.P.; Huijsing, J.H.; “A 1.8 V 3.2 μW comparator for use in a CMOS imager column-level single-slope ADC”, Circuits and Systems, 2005. ISCAS 2005, p-p 6162-6165

[65] Kiyoyama, K.; Ohara, Y.; Lee, K.-W.; Yang, Y.; Fukushima, T.; Tanaka, T.; Koyanagi, M.; “A parallel ADC for high-speed CMOS image processing system with 3D structure”, 3D System Integration, 2009. 3DIC 2009. IEEE International Conference on, 2009, p-p 1-4

[66] Snoeij, M.F.; Theuwissen, A.J.P.; Makinwa, K.A.A.; Huijsing, J.H.; “Multiple-Ramp Column-Parallel ADC Architectures for CMOS Image Sensors”, Solid-State Circuits, IEEE Journal of, Volume: 42 , Issue: 12, 2007, p-p 2968-2977

[67] Brooks, L.; Hae-Seung Lee; “A Zero-Crossing-Based 8-bit 200 MS/s Pipelined ADC”, Solid-State Circuits, IEEE Journal of, Volume: 42 , Issue: 12, 2007, p-p 2677-2687

[68] Brooks, L.; Hae-Seung Lee; “A 12b, 50 MS/s, Fully Differential Zero-Crossing Based Pipelined ADC”, Solid-State Circuits, IEEE Journal of, 2009, p-p 3329-3343

[69] Choi, M.; M.; Asad, A.; "A 6b 1.3Gsample/s A/D converter in 0.35μm CMOS", IEEE International Solid-State Circuits Conference, 2001, Paper number: 8.1

[70] Geelen, G.; "A 6b 1.1Gsample/s CMOS A/D converter", IEEE International Solid-State Circuits Conference, 2001, Paper number: 8.2

[71] Lin, J.; Haroun, B. "An embedded 0.8 V/480 μW 6b/22 MHz flash ADC in 0.13 μm digital CMOS process using nonlinear double-interpolation technique", IEEE International Solid-State Circuits Conference, 2002, Paper number: 18.2

[72] Cheng, W.; Ali, W.; Moon-Jung Choi, Liu, K.; Tat, T.; Devendorf, D.; Linder, L.; Stevens, R.; "A 3b 40GS/s ADC-DAC in 0.12μm SiGe", IEEE International Solid-State Circuits Conference, 2004, Paper number: 14.6

[73] Van der Plas, G.; Decoutere, S.;Donnay, S.; "A 0.16pJ/Conversion-Step 2.5mW 1.25GS/s 4b ADC in a 90nm Digital CMOS

Process", IEEE International Solid-State Circuits Conference, 2006, Paper number: 31.1

[74] Park, S.; Palaskas, Y.;Flynn, M.; "A 4GS/s 4b Flash ADC in 0.18µm CMOS", IEEE International Solid-State Circuits Conference, 2006, Paper number: 31.3

[75] Schvan, P.; Pollex, D.; Wang, S-C.; Falt, C.; Ben-Hamida, N.; "A 22GS/s 5b ADC in 130nm SiGe BiCMOS", IEEE International Solid-State Circuits Conference, 2006, Paper number: 31.4

[76] Daly, D.; Chandrakasan, A.; "A 6b 0.2-to-0.9V Highly-Digital Flash ADC with Comparator Redundancy", IEEE International Solid-State Circuits Conference, 2008, Paper number: 30.8

[77] Ono, K.; Shimizu, H.; Ogawa, J.; Takeda, M.; Yano, M.; "A 6bit 400Msps 70mW ADC using interpolated parallel scheme", IEEE VLSI Symposium on Circuits, 2002, Paper number: 23.1

[78] Paulus, C.; Bluthgen, H.-M.; Low, M.; Sicheneder, E.; Bruls, N.; Courtois, A.; Tiebout, M.; Thewes, R.; "A 4GS/s 6b flash ADC in 0.13um CMOS", IEEE VLSI Symposium on Circuits, 2004, Paper number: 25.1

[79] Deguchi, K.; Suwa, N.; Ito, M.; Kumamoto, T.; Miki, T.; "A 6-bit 3.5-GS/s 0.9-V 98-mW Flash ADC in 90nm CMOS", IEEE VLSI Symposium on Circuits, 2007, Paper number: 7.2

[80] Chen, C.-Y.; Le, M.; Kim, K.Y.; "A Low Power 6-bit Flash ADC with Reference Voltage and Common-Mode Calibration", IEEE VLSI Symposium on Circuits, 2008, Paper number: 2.1

[81] Verbruggen, B.; Wambacq, P.; Kuijk, M.; Van der Plas, G.; "A 7.6 mW 1.75 GS/s 5 Bit Flash A/D Converter In 90nm Digital CMOS", IEEE VLSI Symposium on Circuits, 2008, Paper number: 2.2

[82] Choi, M.; Lee, J.; Lee, J.; Son, H.; "A 6-bit 5-GSample/s Nyquist A/D Converter in 65nm CMOS", IEEE VLSI Symposium on Circuits, 2008, Paper number: 2.3

[83] Chung, H.; Rylyakov, A.; Deniz, Z-T.; Bulzacchelli, J.; Wei, G-Y.; Friedman, D.; "A 7.5-GS/s 3.8-ENOB 52-mW flash ADC with clock duty cycle control in 65nm CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 26.2

[84] Moreland, C.; Elliott, M.; Murden, F.; Young, J.; Hensley, M.; Stop, R.; "A 14b 100Msample/s 3-stage A/D converter", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.1

[85] Choe, M-J.; Song, B-S.; Bacrania, K.; "A 13b 40Msample/s CMOS pipelined folding ADC with background offset trimming", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.2

[86] Pan, H.; Segami, M.; Choi, M.; Cao, J.; Hatori, F.; Abidi, A.; "A 3.3V, 12b, 50Msample/s A/D converter in 0.6μm CMOS with over 80dB SFDR", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.4

[87] Sushihara, K.; Matsuzawa, A.; "A 7b 450MSample/s 50mW CMOS ADC in 0.3mm^2", IEEE International Solid-State Circuits Conference, 2002, Paper number: 10.3

[88] Taft, R.; Menkus, C.; Tursi, M.R.; Hidri, O.; Pons, V.; "A 1.8V 1.6GS/s 8b self-calibrating folding ADC with 7.26 ENOB at Nyquist frequency", IEEE International Solid-State Circuits Conference, 2004, Paper number: 14.1

[89] Geelen, G.; Paulus, E.; "An 8b 600MS/s 200mW CMOS folding A/D converter using an amplifier preset technique", IEEE International Solid-State Circuits Conference, 2004, Paper number: 14.2

[90] Verbruggen, B.; Cranickx, J.; Kuijk, M.; Wambacq, P.; Van der Plas, G.; "A 2.2mW 5b 1.75GS/s Folding Flash ADC in 90nm Digital CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.8

[91] Taft, R. C.; Francese, P. A.; Tursi, M. R.; Hidri, O.; MacKenzie, A.; Hoehn, T.; Schmitz, P.; Werker, H.; Glenny, A.; "A 1.8V 1.0 GS/s 10b Self-Calibrating Unified-Folding-Interpolating ADC with 9.1 ENOB at Nyquist Frequency", IEEE International Solid-State Circuits Conference, 2009, Paper number: 4.3

[92] Yoon, K.; Lee, J.; Jeong, D-K.; Kim, W.; "An 8-bit 125 MS/s CMOS folding ADC for Gigabit Ethernet LSI", IEEE VLSI Symposium on Circuits, 2000, Paper number: 16.2

[93] Sigenobu, T.; Ito, M.; Miki, T. "An 8-bit 30 MS/s 18 mW ADC with 1.8 V single power supply", IEEE VLSI Symposium on Circuits, 2001, Paper number: 19.2

[94] Blum, A.S.; Engl, B.H.; Eichfeld, H.P.; Hagelauer, R.; Abidi, A.A.; "A 1.2 V 10-b 100-MSamples/s A/D converter in 0.12um CMOS", IEEE VLSI Symposium on Circuits, 2002, Paper number: 23.2

[95] Wang, Z-Y.; Pan, H.; Chang, C-M.; Yu, H-R.; Chang, M.F.; "A 600 MSPS 8-bit folding ADC in 0.18um CMOS", IEEE VLSI Symposium on Circuits, 2004, Paper number: 25.2

[96] Makigawa, K.; Ono, K.; Ohkawa. T.; Matsuura, K.; Segami, M.; "A 7bit 800Msps 120mW Folding and Interpolation ADC Using a Mixed-Averaging Scheme", IEEE VLSI Symposium on Circuits, 2006, Paper number: 16.3

[97] Singer, L.; Ho, S.; Timko, M.; Kelly, D.; "A 12b 65Msample/s CMOS ADC with 82dB SFDR at 120MHz", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.3

[98] Ming, J.; Lewis, S. H.; "An 8b 80Msample/s pipelined ADC with background calibration", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.5

[99] Chen, H-S.; Bacrania, K.; Song, B-S.; "A 14b 20MSample/s CMOS pipelined ADC", IEEE International Solid-State Circuits Conference, 2000, Paper number: 2.7

[100] Park, Y-I.; Karthikeyan, S.; Tsay, F.; Bartolome, E.; "A 10b 100Msample/s CMOS pipelined ADC with 1.8V power supply", IEEE International Solid-State Circuits Conference, 2001, Paper number: 8.3

[101] Kelly, D.; Yang, W.; Mehr, I.; Sayuk, M.; Singer, L.; "A 3V 340mW 14b 75MSPS CMOS ADC with 85dB SFDR at Nyquist", IEEE International Solid-State Circuits Conference, 2001, Paper number: 8.5

[102] Miyazaki, D.; Furuta, M.; Kawahito, S.; "A 16mW 30MSample/s 10b pipelined A/D converter using a pseudo-differential architecture", IEEE International Solid-State Circuits Conference, 2002, Paper number: 10.5

[103] Kulhalli, S.; Penkota, V.; Asv, R.; "A 30mW 12b 21MSample/s pipelined CMOS ADC", IEEE International Solid-State Circuits Conference, 2002, Paper number: 18.4

[104] Min, B-M.; Kim, P.; Boisvert, D.; Aude, A.; "A 69mW 10b 80MS/s pipelined CMOS ADC", IEEE International Solid-State Circuits Conference, 2003, Paper number: 18.4

[105] Yoo, S-M.; Park, J-B.; Yang, H-S.; Bae, H-H.; Moon, K-H.; Park, H-J.; Lee, S-H.; Kim, J-H.; "A 10b 150MS/s 123mW 0.18µm CMOS pipelined ADC", IEEE International Solid-State Circuits Conference, 2003, Paper number: 18.5

[106] Murmann, B.; Boser, B. E.; "A 12b 75MS/s pipelined ADC using open-loop residue amplification", IEEE International Solid-State Circuits Conference, 2003, Paper number: 18.6

[107] Hernes, B.; Briskemyr, A.; Andersen, T. N.; Telstø, F.; Bonnerud, T. E.; Moldsvor, O.; "A 1.2V 220MS/s 10b pipeline ADC implemented in 0.13µm digital CMOS", IEEE International Solid-State Circuits Conference, 2004, Paper number: 14.3

[108] Siragusa E.; Galton, I.; "A digitally enhanced 1.8V 15b 40MS/s CMOS pipelined ADC", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.1

[109] Liu, H-C.; Lee, Z-M.; Wu, J-T.; "A 15b 20MS/s CMOS pipelined ADC with digital background calibration", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.2

[110] Nair K.; Harjani, R.; "A 96dB SFDR 50MS/s digitally enhanced CMOS pipeline A/D converter", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.3

[111] Chiu, Y.; Gray, P.; Nikolic, B.; "A 1.8V 14b 10MS/s pipelined ADC in 0.18µm CMOS with 99dB SFDR", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.4

[112] Carl R. Grace, Paul J. Hurst, Stephen H. Lewis; "A 12b 80MS/s pipelined ADC with bootstrapped digital calibration", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.5

[113] Olaf Stroeble, Victor Dias, Christoph Schwoerer; "An 80MHz 10b pipeline ADC with dynamic range doubling and dynamic reference selection", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.6

[114] Seung-Tak Ryu, Sourja Ray, Bang-Sup Song, Gyu-Hyeong Cho, Kanti Bacrania; "A 14b-linear capacitor self-trimming pipelined ADC", IEEE International Solid-State Circuits Conference, 2004, Paper number: 25.7

[115] Wang, R.; Martin, K.; Johns, D.; Burra, G.; "A 3.3 mW 12 MS/s 10b pipelined ADC in 90 nm digital CMOS", IEEE International Solid-State Circuits Conference, 2005, Paper number: 15.2

[116] Yoshioka, M.; Kudo, M.; Gotoh, K.; Watanabe, Y.; "A 10 b 125 MS/s 40 mW pipelined ADC in 0.18 µm CMOS", IEEE International Solid-State Circuits Conference, 2005, Paper number: 15.4

[117] Hwi-Cheol Kim; Deog-Kyoon Jeong; Wonchan Kim; "A 30mW 8b 200MS/s pipelined CMOS ADC using a switched-opamp technique", IEEE International Solid-State Circuits Conference, 2005, Paper number: 15.5

[118] Geelen, Govert; Paulus, Edward; Simanjuntak, Dobson; Pastoor, Hein; Verlinden, Ren; "A 90nm CMOS 1.2V 10b Power and Speed Programmable Pipelined ADC with 0.5pJ/Conversion-Step", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.1

[119] Ryu, Seung-Tak; Song, Bang-Sup; Bacrania, Kanti; "A 10b 50MS/s Pipelined ADC with Opamp Current Reuse", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.2

[120] Bogner, Peter; Kuttner, Franz; Kropf, Claus; Hartig, Thomas; Burian, Markus; Eul, Hermann; "A 14b 100MS/s Digitally Self-Calibrated Pipelined ADC in 0.13µm CMOS", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.6

[121] Choi, Hee-Cheol; Kim, Ju-Hwa; Yoo, Sang-Min; Lee, Kang-Jin; Oh, Tae-Hwan; Seo, Mi-Jung; Kim, Jae-Whui; "A 15mW 0.2mm2 10b 50MS/s ADC with Wide Input Range", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.7

[122] Ray, Sourja; Song, Bang-Sup; "A 13b Linear 40MS/s Pipelined ADC with Self-Configured Capacitor Matching", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.8

[123] Yoshioka, M.; Kudo, M.; Mori, T.; Tsukamoto, S.; "A 0.8V 10b 80MS/s 6.5mW Pipelined ADC with Regulated Overdrive Voltage Biasing", IEEE International Solid-State Circuits Conference, 2007, Paper number: 25.1

[124] Young-Deuk Jeon; Seung-Chul Lee; Kwi-Dong Kim; Jong-Kee Kwon; Jongdae Kim; "A 4.7mW 0.32mm2 10b 30MS/s Pipelined ADC Without a Front-End S/H in 90nm CMOS", IEEE International Solid-State Circuits Conference, 2007, Paper number: 25.3

[125] Seung-Chul Lee; Young-Deuk Jeon; Kwi-Dong Kim; Jong-Kee Kwon; Jongdae Kim; Jeong-Woong Moon; Wooyol Lee; "A 10b 205MS/s 1.0mm2 90nm CMOS Pipeline ADC for Flat Panel Display Applications", IEEE International Solid-State Circuits Conference, 2007, Paper number: 25.4

[126] Hernes, B.; Bjornsen, J.; Andersen, T.N.; Vinje, A.; Korsvoll, H.; Telsto, F.; Briskemyr, A.; Holdo, C.; Moldsvor, O.; "A 92.5mW 205MS/s 10b Pipeline IF ADC Implemented in 1.2V/3.3V 0.13um CMOS", IEEE International Solid-State Circuits Conference, 2007, Paper number: 25.6

[127] B. Lee, B. Min, G. Manganaro, J. W. Valvano; "A 14b 100MS/s Pipelined ADC with a Merged Active S/H and first MDAC", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.6

[128] M. Boulemnakher, E. Andre, J. Roux, F. Paillardet; "A 1.2V 4.5mW 10b 100MS/s Pipeline ADC in 65nm CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.7

[129] B. Gregoire, U-K. Moon; "An Over-60dB True Rail-to-Rail Performance Using Correlated Level Shifting and an Opamp with 30dB Loop Gain", IEEE International Solid-State Circuits Conference, 2008, Paper number: 30.1

[130] K-W. Hsueh, Y-K. Chou, Y-H. Tu, Y-F. Chen, Y-L. Yang, H-S. Li; "A 1V 11b 200MS/s Pipelined ADC with Digital Background Calibration in 65nm CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 30.4

[131] Ashutosh Verma, Behzad Razavi; "A 10b 500MHz 55mW CMOS ADC", IEEE International Solid-State Circuits Conference, 2009, Paper number: 4.6

[132] Siddharth Devarajan, Larry Singer, Dan Kelly, Steven Decker, Abhishek Kamath, Paul Wilkins; "A 16b 125MS/s 385mW 78.7dB SNR CMOS Pipeline ADC ", IEEE International Solid-State Circuits Conference, 2009, Paper number: 4.7

[133] Andrea Panigada, Ian Galton; "A 130mW 100MS/s Pipelined ADC with 69dB SNDR Enabled by Digital Harmonic Distortion Correction", IEEE International Solid-State Circuits Conference, 2009, Paper number: 9.1

[134] Imran Ahmed, Jan Mulder, David A. Johns; "A 50MS/s 9.9mW Pipelined ADC with 58dB SNDR in 0.18µm CMOS Using Capacitive Charge-Pumps", IEEE International Solid-State Circuits Conference, 2009, Paper number: 9.2

[135] Y-C. Huang, T-C. Lee; "A 10b 100MS/s 4.5mW Pipelined ADC with a Time Sharing Technique", IEEE International Solid-State Circuits Conference, 2010, Paper number: 16.5

[136] Shabra, A. Hae-Seung Lee; "A 12-bit mismatch-shaped pipeline A/D converter", IEEE VLSI Symposium on Circuits, 2001, Paper number: 19.3

[137] Dong-Young Chang; Gil-Cho Ahn; Un-Ku Moon; "A 0.9 V 9 mW 1MSPS digitally calibrated ADC with 75 dB SFDR", IEEE VLSI Symposium on Circuits, 2003, Paper number: 6.1

[138] Anderson, M. Norling, K. Dreyfert, A. Yuan, J.; "A reconfigurable pipelined ADC in 0.18um CMOS", IEEE VLSI Symposium on Circuits, 2005, Paper number: 21.2

[139] Matsui, H. Ueda, M. Daito, M. Iizuka, K.; "A 14bit digitally self-calibrated pipelined ADC with adaptive bias optimization for arbitrary speeds up to 40MS/s", IEEE VLSI Symposium on Circuits, 2005, Paper number: 21.3

[140] D.-L. Shen T.-C. Lee; "A 6-Bit 800-MS/s Pipelined A/D Converter with Open-Loop Amplifiers", IEEE VLSI Symposium on Circuits, 2006, Paper number: 16.1

[141] K. Honda, Z. Liu, M. Furuta and S. Kawahito; "A 14b Low-power Pipeline A/D Converter Using a Pre-charging Technique", IEEE VLSI Symposium on Circuits, 2007, Paper number: 19.2

[142] K.-J. Lee, E.-S. Shin, H.-S. Yang, J.-H. Kim, P.-U. Ko, I.-R. Kim, S.-H. Lee, K.-H. Moon and J.-W. Kim; "A 90nm CMOS 0.28mm2 1V 12b 40MS/s ADC with 0.39pJ/Conversion-Step", IEEE VLSI Symposium on Circuits, 2007, Paper number: 19.3

[143] J. Shen, P. Kinget; "A 0.5V 8bit 10Msps Pipelined ADC in 90nm CMOS", IEEE VLSI Symposium on Circuits, 2007, Paper number: 19.5

[144] H. Van de Vel, B. Buter, H. van der Ploeg, M. Vertregt, G. Geelen, E. Paulus; "A 1.2V 250mW 14b 100MS/s Digitally Calibrated Pipeline ADC in 90nm CMOS", IEEE VLSI Symposium on Circuits, 2008, Paper number: 8.2

[145] J. Hu, N. Dolev, B. Murmann; "A 9.4-bit, 50-MS/s, 1.44-mW Pipelined ADC Using Dynamic Residue Amplification", IEEE VLSI Symposium on Circuits, 2008, Paper number: 22.1

[146] H.-C. Choi, Y.-J. Kim, M.-H. Lee, Y.-L. Kim, S.-H. Lee; "A 12b 50MS/s 10.2mA 0.18µm CMOS Nyquist ADC with a Fully Differential Class-AB Switched OP-AMP", IEEE VLSI Symposium on Circuits, 2008, Paper number: 22.3

[147] M. Anthony, E. Kohler, J. Kurtze, L. Kushner, G. Sollner; "A Process-Scalable Low-Power Charge-Domain 13-bit Pipeline ADC", IEEE VLSI Symposium on Circuits, 2008, Paper number: 22.4

[148] Franz Kuttner; "A 1.2V 10b 20MSample/s non-binary successive approximation ADC in 0.13µm CMOS", IEEE International Solid-State Circuits Conference, 2002, Paper number: 10.6

[149] Verma, Naveen; Chandrakasan, Anantha; "A 25µW 100kS/s 12b ADC for Wireless Micro-Sensor Applications", IEEE International Solid-State Circuits Conference, 2006, Paper number: 12.5

[150] Craninckx, J.; Van der Plas, G.; "A 65fJ/Conversion-Step, 0–50MS/s 0–0.7mW 9bit Charge Sharing SAR ADC in 90nm Digital CMOS", IEEE International Solid-State Circuits Conference, 2007, Paper number: 13.5

[151] V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Vander Plas, J. Craninckx; "An 820uW 9b 40MS/s Noise-Tolerant Dynamic0SAR ADC in 90nm Digital CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.1

[152] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. Klumperink, B. Nauta; "A 1.9uW 4.4fJ/conversion-step 10b 1MS/s Charge-Redistribution ADC", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.4

[153] A Agnes, E. Bonizzoni, P. Malcovati, F. Maloberti; "A 9.4-ENOB 1V 3.8uW 100kS/s SAR ADC with Time-Domain Comparator", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.5

[154] W. Liu, P. Huang, Y. Chiu; "A 12b 22.5/45MS/s 3.0mW 0.059mm2 CMOS SAR ADC Achieving Over 90dB SFDR", IEEE International Solid-State Circuits Conference, 2010, Paper number: 21.2

[155] M. Yoshioka, K. Ishikawa, T. Takayama, S.; "A 10b 50MS/s 820µW SAR ADC with On-Chip Digital Calibration", IEEE International Solid-State Circuits Conference, 2010, Paper number: 21.4

[156] C-C. Liu, S-J. Chang, G-Y. Huang, Y-Z. Lin, C-M. Huang, C-H. Huang, L. Bu, C-C. Tsai; "A 10b 100MS/s 1.13mW SAR ADC with Binary-Scaled Error Compensation", IEEE International Solid-State Circuits Conference, 2010, Paper number: 21.5

[157] P. Harpe, C. Zhou, X. Wang, G. Dolmans, H. de Groot; "A 30fJ/Conversion-Step 8b 0-to-10MS/s Asynchronous SAR ADC in 90nm CMOS", IEEE International Solid-State Circuits Conference, 2010, Paper number: 21.6

[158] Chun-Cheng Liu, Soon-Jyh Chang, Guan-Ying Huang, Yin-Zu Lin; "A 0.92mW 10-bit 50-MS/s SAR ADC in 0.13µm CMOS Process", IEEE VLSI Symposium on Circuits, 2009, Paper number: 23.1

[159] Pierluigi Nuzzo, Claudio Nani, Costantino Armiento, Alberto Sangiovanni-Vincentelli, Jan Craninckx and Geert Van der Plas; "A 6-bit 50-MS/s Threshold Configuring SAR ADC in 90-nm Digital CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 23.2

[160] Joshua J. Kang and Michael P. Flynn; "A 12b 11MS/s Successive Approximation ADC with two comparators in 0.13µm CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 23.3

[161] Seon-Kyoo Lee, Seung-Jin Park, Yunjae Suh, Hong-June Park, and Jae-Yoon Sim; "A 1.3uW 0.6V 8.7-ENOB Successive Approximation ADC in a 0.18um CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 23.4

[162] Chun-Cheng Liu, Soon-Jyh Chang, Guan-Ying Huang, Yin-Zu Lin; "A 0.92mW 10-bit 50-MS/s SAR ADC in 0.13μm CMOS Process", IEEE VLSI Symposium on Circuits, 2009, Paper number: 26.1

[163] Joshua J. Kang and Michael P. Flynn; "A 12b 11MS/s Successive Approximation ADC with two comparators in 0.13μm CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 26.3

[164] Seon-Kyoo Lee, Seung-Jin Park, Yunjae Suh, Hong-June Park, and Jae-Yoon Sim; "A 1.3uW 0.6V 8.7-ENOB Successive Approximation ADC in a 0.18μm CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 26.4

[165] C. Lee, M. Flynn; "A 12b 50MS/s 3.5mW SAR Assisted 2-Stage Pipeline ADC", IEEE VLSI Symposium on Circuits, 2010, Paper number: 23.2

[166] C.-C. Liu, S.-J. Chang, G.-Y. Huang, Y.-Z. Lin, C.-M. Huang; "A 1V 11fJ/Conversion-Step 10bit 10MS/s Asynchronous SAR ADC in 0.18μm CMOS", IEEE VLSI Symposium on Circuits, 2010, Paper number: 23.3

[167] Y.-Z. Lin, C.-C. Liu, G.-Y. Huang, Y.-T. Shyu, S.-J. Chang; "A 9-bit 150-MS/s 1.53-mW Subranged SAR ADC in 90-nm CMOS", IEEE VLSI Symposium on Circuits, 2010, Paper number: 23.4

[168] Geerts, Y.; Steyaert, M.; Sansen, W.; "A 2.5 MSample/s multi-bit ΔΣ CMOS ADC with 95 dB SNR", IEEE International Solid-State Circuits Conference, 2000, Paper number: 20.2

[169] Fujimori, I.; Longo, L.; Hairapetian, A.; Seiyama, K.; Kosic, S.; Cao, J.; Shu-Iap Chan; "A 90 dB SNR, 2.5 MHz output rate ADC using cascaded multibit ΔΣ modulation at 8x oversampling ratio", IEEE International Solid-State Circuits Conference, 2000, Paper number: 20.3

[170] Tabatabaei, A.; Kaviani, K.; Wooley, B.; "A two-path bandpass ΣΔ modulator with extended noise shaping", IEEE International Solid-State Circuits Conference, 2000, Paper number: 20.5

[171] Burger, T.; Qiuting Hueng; "A 13.5mW, 185 MSample/s ΔΣ-modulator for UMTS/GSM dual-standard IF reception", IEEE International Solid-State Circuits Conference, 2001, Paper number: 3.1

[172] Oliaei, O.; Clement, P.; Gorisse, P.; "A 5 mW ΣΔ modulator with 84 dB dynamic range for GSM/EDGE", IEEE International Solid-State Circuits Conference, 2001, Paper number: 3.2

[173] Vleugels, K.; Rabii, S.; Wooley, B.A.; "A 2.5 V broadband multi-bit ΣΔ modulator with 95 dB dynamic range", IEEE International Solid-State Circuits Conference, 2001, Paper number: 3.4

[174] Salo, T.; Hollman, T.; Lindfors, S.; Halonen, K.; "A dual-mode 80 MHz bandpass ΔΣ modulator for a GSM/WCDMA IF-receiver", IEEE International Solid-State Circuits Conference, 2002, Paper number: 13.3

[175] Ruoxin Jiang; Fiez, T.S.; "A 1.8 V 14 b ΔΣ A/D converter with 4MSamples/s conversion", IEEE International Solid-State Circuits Conference, 2002, Paper number: 13.4

[176] Gupta, S.K.; Brooks, T.L.; Fong, V.; "A 64 MHz ΣΔ ADC with 105 dB IM3 distortion using a linearized replica sampling network", IEEE International Solid-State Circuits Conference, 2002, Paper number: 13.6

[177] Gomez, G.; Haroun, B.; "A 1.5 V 2.4/2.9 mW 79/50 dB DR ΣΔ modulator for GSM/WCDMA in a 0.13 µm digital process", IEEE International Solid-State Circuits Conference, 2002, Paper number: 18.1

[178] Reutemann, R.; Balmelli, P.; Qiuting Huang; "A 33mW 14b 2.5MSample/s ΣΔ A/D converter in 0.25µm digital CMOS", IEEE International Solid-State Circuits Conference, 2002, Paper number: 18.6

[179] Blanken, P.G.; Menten, S.E.J.; "A 10uV-offset 8kHz bandwidth 4th-order chopped Sigma-Delta A/D converter for battery management", IEEE International Solid-State Circuits Conference, 2002, Paper number: 23.5

[180] Feng Chen; Ramaswamy, S.; Bakkaloglu, B.; "A 1.5V 1mA 80dB passive ΣΔ ADC in 0.13µm digital CMOS process", IEEE International Solid-State Circuits Conference, 2003, Paper number: 3.1

[181] YuQing Yang; Chokhawala, A.; Alexander, M.; Melanson, J.; Hester, D.; "A 114 dB 68 mW chopper-stabilized stereo multi-bit audio A/D converter", IEEE International Solid-State Circuits Conference, 2003, Paper number: 3.2

[182] Dezzani, A.; Andre, E.; "A 1.2-V dual-mode WCDMA/GPRS ΣΔ modulator", IEEE International Solid-State Circuits Conference, 2003, Paper number: 3.3

[183] Tabatabaei, A.; Onodera, K.; Zargari, M.; Samavati, H.; Su, D.K.; "A dual channel ΣΔ ADC with 40MHz aggregate signal bandwidth", IEEE International Solid-State Circuits Conference, 2003, Paper number: 3.7

[184] Balmelli, P.; Qiuting Huang; "A 25 MS/s 14 b 200 mW ΣΔ modulator in 0.18 μm CMOS", IEEE International Solid-State Circuits Conference, 2004, Paper number: 4.2

[185] Putter, B.M.; "ΣΔ ADC with finite impulse response feedback DAC", IEEE International Solid-State Circuits Conference, 2004, Paper number: 4.3

[186] Yao, L.; Steyaert, M.; Sansen, W.; "A 1 V 88 dB 20 kHz ΣΔ modulator in 90 nm CMOS", IEEE International Solid-State Circuits Conference, 2004, Paper number: 4.5

[187] Gaggl, R.; Inversi, M.; Wiesbauer, A.; "A power optimized 14-bit SC ΔΣ modulator for ADSL CO applications", IEEE International Solid-State Circuits Conference, 2004, Paper number: 4.6

[188] Ying, F.; Maloberti, F.; "A mirror image free two-path bandpass ΣΔ modulator with 72 dB SNR and 86 dB SFDR", IEEE International Solid-State Circuits Conference, 2004, Paper number: 4.7

[189] Jiang Yu; Maloberti, F.; "A low-power multi-bit ΔΣ modulator in 90nm digital CMOS without DEM", IEEE International Solid-State Circuits Conference, 2005, Paper number: 9.2

[190] Jinseok Koh; Yunyoung Choi; Gomez, G.; "A 66dB DR 1.2V 1.2mW single-amplifier double-sampling 2nd-order ΔΣ ADC for WCDMA in 90nm CMOS", IEEE International Solid-State Circuits Conference, 2005, Paper number: 9.3

[191] Brewer, R.; Gorbold, J.; Hurrell, P.; Lyden, C.; Maurino, R.; Vickery, M.; "A 100dB SNR 2.5MS/s output data rate ΔΣ ADC", IEEE International Solid-State Circuits Conference, 2005, Paper number: 9.4

[192] Kwon, S.; Maloberti, F.; "A 14mW Multi-bit Delta-Sigma Modulator with 82dB SNR and 86dB DR for ADSL2+", IEEE International Solid-State Circuits Conference, 2006, Paper number: 3.4

[193] Goes, J.; Vaz, B.; Monteiro, R.; Paulino, N.; "A 0.9V Delta-Sigma Modulator with 80dB SNDR and 83dB DR Using a Single-Phase Technique", IEEE International Solid-State Circuits Conference, 2006, Paper number: 3.7

[194] Fujimoto, Y.; Kanazawa, Y.; Lore, P.; Miyamoto, M.; "An 80/100MS/s 76.3/70.1dB SNDR Delta-Sigma ADC for Digital TV Receivers", IEEE International Solid-State Circuits Conference, 2006, Paper number: 3.8

[195] Christen, T.; Burger, T.; Qiuting Huang; "A 0.13µm CMOS EDGE/UMTS/WLAN Tri-Mode ΔΣ ADC with -92dB THD", IEEE International Solid-State Circuits Conference, 2007, Paper number: 13.2

[196] Y. Chae, I. Lee, G. Han; "A 0.7V 36uW 85dB-DR Audio delta-sigma Modulator Using a Class-C Inverter", IEEE International Solid-State Circuits Conference, 2008, Paper number: 27.2

[197] R. Veldhoven, R. Rutten, L. Breems; "An Inverter-Based Hybrid delta sigma Modulator", IEEE International Solid-State Circuits Conference, 2008, Paper number: 27.3

[198] P. Malla, H. Lakdawala, K. Kornegay, K. Soumyanath; "A 28mW Spectrum-Sensing Reconfigurable 20MHz 72dB-SNR 70dB-SNDR DT delta sigma ADC for 802.11n/WiMax Receivers", IEEE International Solid-State Circuits Conference, 2008, Paper number: 27.5

[199] Lynn Bos, Gerd Vandersteen, Julien Ryckaert, Pieter Rombouts, Yves Rolain, Geert Van der Plas; "A Multirate 3.4-to-6.8mW 85-to-66dB DR GSM/Bluetooth/UMTS Cascade DT ΔΣM in 90nm Digital CMOS", IEEE International Solid-State Circuits Conference, 2009, Paper number: 9.8

[200] Balmelli, P. Qiuting Huang Piazza, F.; "A 50-mW 14-bit 2.5-MS/s Σ-Δ modulator in a 0.25 μm digital CMOS technology", IEEE VLSI Symposium on Circuits, 2000, Paper number: 11.1

[201] Kye-Shin Lee Maloberti, F.; "A 1.8 V, 1 MS/s, 85 dB SNR 2+2 mash Sigma-Delta modulator with +/-0.9 V reference voltage", IEEE VLSI Symposium on Circuits, 2003, Paper number: 6.2

[202] Jae Hoon Shim Beomsup Kim ; "A third-order Sigma-Delta modulator in 0.18um CMOS with calibrated mixed-mode integrators", IEEE VLSI Symposium on Circuits, 2004, Paper number: 6.2

[203] Sunyoung Kim, Jae-Youl Lee, Seong-Jun Song, Namjun Cho, Hoi-Jun Yoo; "An energy-efficient analog front-end circuit for a sub-1V digital hearing aid chip", IEEE VLSI Symposium on Circuits, 2005, Paper number: 12.2

[204] Libin Yao Steyaert, M. Sansen, W.; "A 1-V, 1-MS/s, 88-dB sigma-delta modulator in 0.13um digital CMOS technology", IEEE VLSI Symposium on Circuits, 2005, Paper number: 12.3

[205] Fornasari, A.; Borghetti, F.; Malcovati, P.; Maloberti, F.; "On-line calibration and digital correction of multi-bit sigma-delta modulators", IEEE VLSI Symposium on Circuits, 2005, Paper number: 12.4

[206] C. Tsang; Y. Chiu; B. Nikolic; "A 1.2V, 10.8mW, 500kHz Sigma-Delta Modulator with 84dB SNDR and 96dB SFDR", IEEE VLSI Symposium on Circuits, 2006, Paper number: 19.2

[207] J. Paramesh; R. Bishop; K. Soumyanath; D. Allstot; "An 11-Bit 330MHz 8X OSR Sigma-Delta Modulator for Next-Generation WLAN", IEEE VLSI Symposium on Circuits, 2006, Paper number: 19.4

[208] Z. Zhang, J. Steensgaard, G.C. Temes* and J.-Y. Wu; "A Split 2-0 MASH with Dual Digital Error Correction", IEEE VLSI Symposium on Circuits, 2007, Paper number: 23.3

[209] H. Park, K. Nam, D. Su, K. Vleugels, B. Wooley; "A 0.7-V 100-dB 870-µW Digital Audio Σ Δ Modulator", IEEE VLSI Symposium on Circuits, 2008, Paper number: 18.2

[210] Robert H.M. van Veldhoven, Nicolo Nizza, Lucien J. Breems; "Technology portable, 0.04mm2, Ghz-rate SD modulators in 65nm and 45nm CMOS", IEEE VLSI Symposium on Circuits, 2009, Paper number: 7.3

[211] Dieter Draxelmayr; "A 6b 600MHz 10mW ADC array in digital 90nm CMOS", IEEE International Solid-State Circuits Conference, 2004, Paper number: 14.7

[212] S-W. Chen, R. Brodersen; "A 6b 600MS/s 5.3mW Asynchronous ADC in 0.13µm CMOS", IEEE International Solid-State Circuits Conference, 2006, Paper number: 31.5

[213] Hesener, M.; Eichler, T.; Hanneberg, A.; Herbison, D.; Kuttner, F.; Wenske, H.; "A 14b 40MS/s Redundant SAR ADC with 480MHz Clock in 0.13pm CMOS", IEEE International Solid-State Circuits Conference, 2007, Paper number: 13.6

[214] B. Ginsburg, A. Chandrakasan; "Highly-Intereleaved 5b 250MS/s ADC with Redundant Channels in 65nm CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 12.2

[215] Z. Cao, S. Yan, Y. Li; "A 32mW 1.25GS/s 6b 2b/step SAR ADC in 0.13um CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 30.2

[216] P. Schvan, J. Bach, C. Falt, P. Flemke, R. Gibbins, Y. Greshishchev, N. Ben-Hamida, D. Pollex, J. Sitch, S-C. Wang, J. Wolczanski; "A 24GS/s 6b ADC in 90nm CMOS", IEEE International Solid-State Circuits Conference, 2008, Paper number: 30.3

[217] Erkan Alpman, Hasnain Lakdawala, L. Richard Carley, K. Soumyanath; "A 1.1V 50mW 2.5GS/s 7b Time-Interleaved C-2C SAR ADC in 45nm LP Digital CMOS", IEEE International Solid-State Circuits Conference, 2009, Paper number: 4.2

[218] Wenbo Liu, Yuchun Chang, Szu-Kang Hsien, Bo-Wei Chen, Yung-Pin Lee, Wen-Tsao Chen, Tzu-Yi Yang, Gin-Kou Ma, Yun Chiu; "A 600MS/s 30mW 0.13µm CMOS ADC Array Achieving Over 60dB SFDR with Adaptive Digital Equalization", IEEE International Solid-State Circuits Conference, 2009, Paper number: 4.5

[219] Y. M. Greshishchev, J. Aguirre, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan, S-C. Wang, M. Besson; "A 40GS/s 6b ADC in 65nm CMOS", IEEE International Solid-State Circuits Conference, 2010, Paper number: 21.7

[220] B. Ginsburg, A. Chandrakasan; "A 500MS/s 5b ADC in 65nm CMOS", IEEE VLSI Symposium on Circuits, 2006, Paper number: 16.4

[221] Louwsma, S.M.; van Tuijl, E.J.M.; Vertregt, M.; Nauta, B.; "A 1.35 GS/s, 10b, 175 mW Time-Interleaved AD Converter in 0.13µm CMOS", IEEE VLSI Symposium on Circuits, 2007, Paper number: 7.1