UNIVERSITA’ DEGLI STUDI DI PAVIA
FACOLTA’ DI INGEGNERIA
DOTTORATO DI RICERCA IN MICROELETTRONICA, XXIII CICLO

CMOS CIRCUITS FOR SERIAL LINKS BEYOND 10Gb/s

Tutor:
Chiar.mo Prof. Francesco Svelto

Ph. D. dissertation
of Giorgio Spelgatti

Index

Introduction

Data rates in serial communications have been steadily increasing with on-chip processing rate and logic density both in network applications and in hard-disk interconnects. The data rates are now exceeding 10Gbits/second (Gb/s). A widely used approach is based on continuous-time or discrete-time forward (FFE) and feedback (DFE) equalizers. This analog approach shows its limits with the increase of the data rates resulting to be completely not adequate for certain specific high-demand applications (optical links and high cross-talk links).

A common trend has been the increasing use of digital signal processing. There has been a growing interest in incorporating CMOS analog-to-digital converters (ADCs) as the frontend of high-speed serializers/deserializers (SerDes). By quantizing the signal with an ADC, the digital signal representation enables greater flexibility and more powerful signal processing techniques to achieve lower bit-error rates (BER), ease of programmability and extensibility for different channel characteristics, and robustness to process variations.

This thesis focuses on both the analog and digital approaches. In particular we describe the digital circuits of an analog receiver and a high speed flash ADC which can be used as front end in a digital receiver.

In chapter one an overview of the main characteristics of high-speed serial communication is presented. First of all, the signal modulations, used in this type of communication (NRZ and RZ), and the serial link standards are analyzed. After, we describe the main system metrics and the source of non-idealities that limit the performances. The analog and the digital equalization approaches are described and finally, clock and data recovery (CDR) architectures are analyzed both for analog and digital receiver.

In chapter two we describe an analog receiver for a 12 Gb/s data rate. Data equalization is based on a linear equalizer (LE) followed by a continuous-time DFE (CTDFE). The chapter analyzes in detail the digital circuits that have been designed: a digital bang-bang CDR, the circuits to auto-adapt the CTDFE in order to maximize the equalization efficiency and some auxiliary circuits. Finally we show the measurement results and the literature comparison. This comparison shows that the presented receiver has the better equalization capability between the first in class serial interface receivers, published in the last three years at ISSCC.

In chapter three a 6bit, 5GS/s CMOS flash ADC is described. This ADC can be used as digital front-end of an ADC-based receiver for SerDes applications. The sampling rate of the ADC is the data rate of the SerDes. CMOS ADC implementations have reached multi-GSamples/s (GS/s) sampling rates. Such high sampling rates are typically achieved using a flash ADC architecture. To exceed 10GS/s, multiple ADCs are time-interleaved using multiple clock phases.

We first define the specifications of each flash ADC block. After, we perform a comparison between two flash ADC architectures and we select the better one in terms of power consumption. We also describe the digital encoder, which converts the thermometer output in a binary format and the digital algorithm for the offset calibration. Finally simulated results are shown. Measured results are not available because the ADC is not yet been integrated. The tape-out is planned in mid-november 2010. The partials simulated results show that the designed ADC is comparable with the best-in-class 6 bit ADCs in terms of energy consumption per conversion step.

Chapter 1

HIGH SPEED SERIAL LINK

1.1 INTRODUCTION

In this chapter we analyze the main characteristics of high-speed serial communication. First of all we analyze the signal modulations used in this type of communication (NRZ and RZ) and the serial link standards, referring both to copper links and optical links. After, we describe the main system metrics and the sources of non-idealities limiting the system performances.

We also analyze the analog and digital signal equalization techniques. An analog system is usually based on analog equalization of the transmitted distorted data, and on the detection of the signal zero crossing by a slicer. On the contrary, the digital approach is based on analog to digital conversion of the input distorted signal, and equalization in the digital domain. A digital equalization is not always based on the signal zero crossing detection. In both approaches, a clock recovery circuit is necessary to recover the correct sampling clock, used for reading the received signal. The last part of this chapter deals with clock recovery circuits for both analog and digital receivers. The serial received signal, correctly recognized, is de-multiplexed and made available in a parallel form, as shown in Figure 1. A digital receiver is able to correctly equalize some kind of data links (such as the multimode optical ones) which an analog receiver is not able to equalize.

We focus on analog and digital receivers because the next chapters deal with circuit blocks for both of them. Chapter two describes the digital circuits used in an analog receiver. Chapter three focuses on the design of a flash ADC, which can be used as the front-end for a digital receiver.

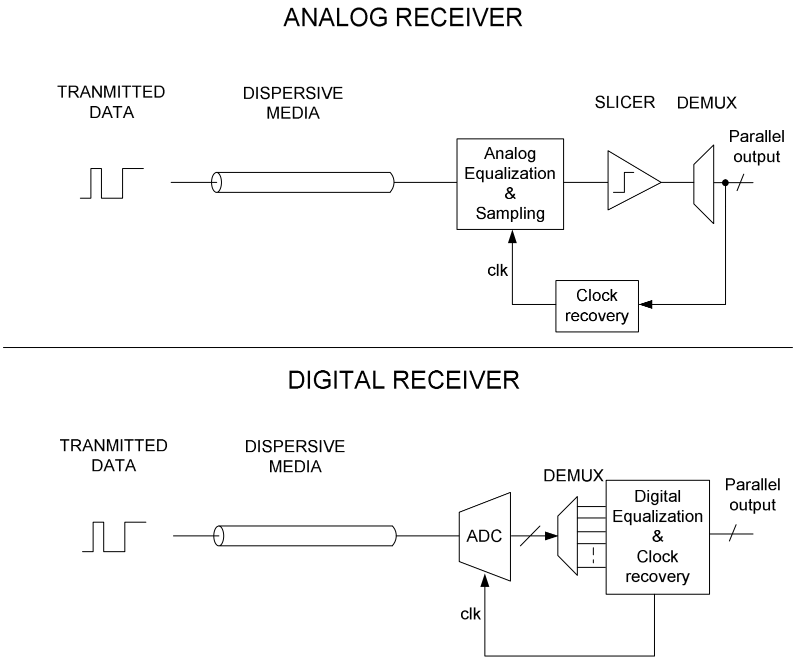


Figure 1: Analog and digital receivers.

1.2 BINARY CODING

Most serial communication systems employ simple binary amplitude modulation of the signal for easy of detection.

A random binary sequence consists of logical ONEs and ZEROs that carry the information and usually occur with equal probability.

1.2.1 NRZ data

In “Non return to zero” (NRZ) coding the logical symbols (ONE and ZERO) are simply represented by high or low voltage (or current) signals, with a duration of a bit period Tb. We refer also to the bit period with the term unit-interval (UI)

The bit rate is equal to 1/Tb per second. It is important to note that the bit stream does not give explicit information on Tb value. A pure random data stream may contain arbitrarily long strings of the same logic value (also called “runs”) exhibiting a local low “transition density”.

These strings create difficulties in particular operation as clock recovery. For this reason, communication standards specify the maximum "run length" as the maximum number of consecutive ONEs or ZEROs.

It is also interesting to examine a random binary data p(t) in the frequency domain assuming that the single bit is simply a rectangular pulse of width Tb.

In this case the power spectral density is given by:

$$\textbf{1.1} \quad S_x(f) = \frac{1}{Tb}\left|P(f)\right|^2$$

Where P(f) represents the Fourier transform of p(t):

$$\textbf{1.2} \quad P(f) = Tb \cdot \frac{\sin(\pi \cdot f \cdot Tb)}{\pi \cdot f \cdot Tb}$$

And so the spectrum of a random sequence is expressed as:

$$\textbf{1.3} \quad S_x(f) = Tb \cdot \left[\frac{\sin(\pi \cdot f \cdot Tb)}{\pi \cdot f \cdot Tb}\right]^2$$

Noting that the $\sin(\pi \cdot f \cdot Tb)$ vanishes at f = n/Tb for integer values of n: this spectrum is shown in Figure 2.

The above analysis yields an important attribute of random binary sequences. For bit rate of 1/Tb the spectrum exhibits no power at frequencies equal to 1/Tb, 2/Tb, etc.

The spectrum exhibits a delta in the zero frequency due to the non-zero mean of the NRZ signal (the considered signal is between zero and one).

This observation proves critical in the task of clock recovery.

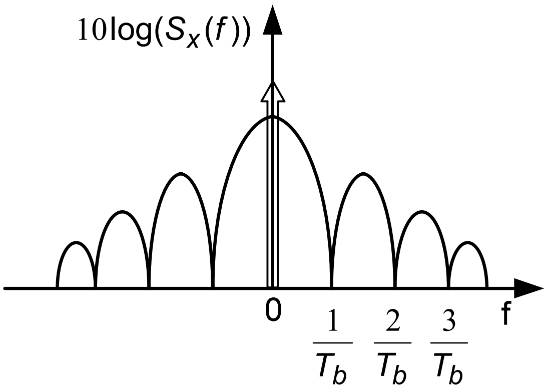


Figure 2: Spectrum of a NRZ random sequence.

1.2.2 RZ data

In “return to zero” (RZ) data each bit consist of two sections: the first section assumes a value, that represents the bit value, and the second section is always equal to a logical zero. So that, every two symbol carrying information are separated by a redundant zero symbol.

The RZ signal can be viewed as the product of a NRZ sequence and a periodic square wave. The spectrum of RZ is the convolution of the two spectrums.

In contrast to NRZ data RZ waveforms exhibit a spectral line at a frequency equal to the bit rate, thereby simplifying the task of clock recovery.

Comparing the spectra of NRZ versus RZ reveals the drawback of RZ: it occupies about twice the bandwidth of NRZ, Figure 3.

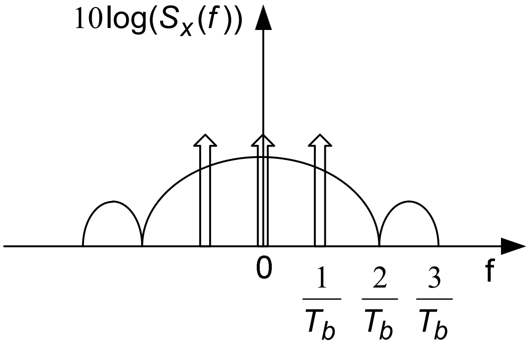


Figure 3: Spectrum of RZ random sequence.

1.2.3 Generation of random data

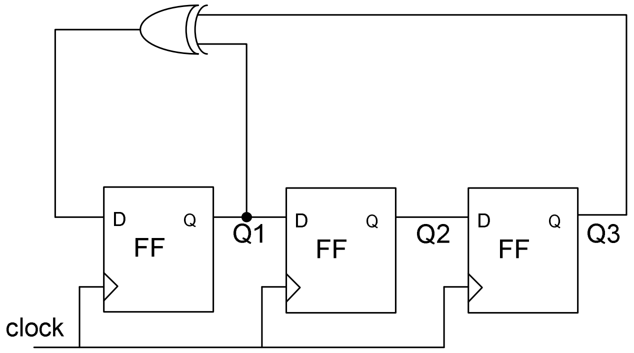
In simulation and characterization, it is difficult to generate completely random binary waveforms because of randomness to manifest itself, the sequence must be very long.

For this reason, it is common to employ standard “pseudo random” binary sequences (PRBS).

Each PRBS is in fact a repetition of a pattern that itself consist of a random sequence of a number of bits. As an example of PRBS generation consider the circuit shown in Figure 4, where three flip flop form a shift register and a XOR gate senses Q1 and Q3, returning the result to the input of the first flip flop.

In that figure the flip flop outputs are also indicated: the generated pattern repeats every 2^3-1 = 7 clock cycles. We also note that if the initial condition is 000, the register remains in a degenerate state; thus some means of initialization is necessary. The waveform produced by the above circuit is an example of a relatively random sequence.

One attribute of randomness is “dc balance”: the total number of ONEs in each period differs from that of ZEROs by only one; note also that the maximum run length is equal to three. Other properties of the PRBSs are described in [1].



Q1	Q2	Q3
1	0	0
1	1	0
1	1	1
0	1	1
1	0	1
0	1	0
0	0	1

Figure 4: PRBS generator.

This technique can be extended to an m-bits system so as to produce a sequence of length 2^m-1. For example many serial communication circuits are tested with a PRBS of length 2^7-1, 2^{15}-1, 2^{23}-1, 2^{31}-1, with a maximum run length of 7, 15, 23 and 31 respectively.

It is important to note that the spectrum of pseudo-random data sequences is quite different from that of really random data sequence. Since the random pattern is repeated periodically, we expect the spectrum to contain only impulses.

Its shape is similar to the really random data one but consists of only spectral line appearing at integer multiple of $[(2^m\text{-}1)Tb]^{-1}$. Of course, for long random pattern, the impulses are very closely spaced, creating an almost continuous spectrum.

The correctness of the data stream, generated through a PRBS, can be tested through a PRBS checker. Figure 5 shows the checker for the PRBS 3 shown in Figure 4: we perform a logical XOR between the input data, generated by the PRBS, and the Q1 XOR Q3 gate output. This signal is called ERROR. If the received sequence is correct the ERROR signal is zero. Moreover ERROR signal is right only after receiving at least 3 bit. The main advantage in using a PRBS data stream is that the generated sequence is self-aligning. This means that the sequence correctness can be verified without the need to find an align sequence.

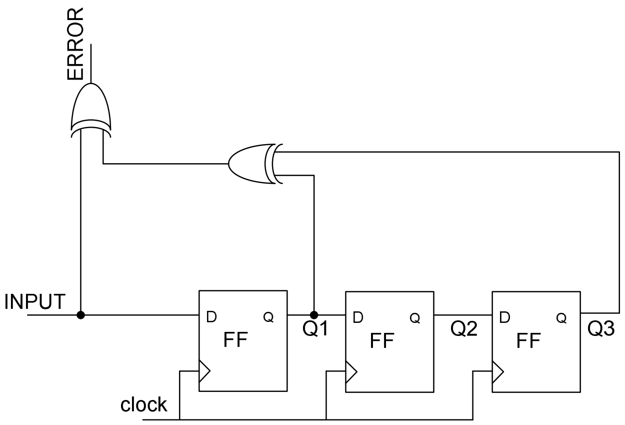


Figure 5: PRBS checker.

1.3 LINK OVERVIEW

For over 20 years, the parallel bus interface has been the mainstream storage interconnects for most storage systems [2]. Increasing bandwidth and flexibility demands have exposed inefficiencies in the two main parallel interface technologies: SCSI (Small Computer System Interface) and ATA (Advanced Technology Attachment). The lack of compatibility between parallel ATA and SCSI increases costs for inventory management, R&D, training and product qualification. Continued demands for higher speeds, more robust data integrity, smaller designs and wider standardization cast doubt on the ability of parallel technology to economically keep pace with increasing CPU processing power and disk drive speeds.

In addition, shrinking budgets are making it increasingly difficult to sustain the costs of developing and managing multiple backplane types, validating multiple interfaces and stocking multiple I/O connections.

Terminating parallel signals is also difficult, requiring individual lines to be terminated, usually by the last drive, to avoid signal reflection at the end of a cable.

Finally, parallel's large cable and connector size make it unsuitable for increasingly dense computing environments.

A possible way to avoid this constrain is in a serial solution.

This technology draws its name from the way it transmits signals, that is, in a single stream (serially) compared with the multiple streams found in parallel technology.

Serial technology wraps many bits of data into packets and then transfers the packets up to 30 times faster than parallel down the wire to or from the host.

1.3.1 Link description

The system's physical layer defines the passive interconnect, the transmitter and receiver device electrical characteristics.

A general serial data link, valid for all the described standards, is shown in Figure 6.

The input data are provided to the transmitter in a parallel form, they are multiplexed and transmitted in a serial form through a copper link or an optical link; the transmitted data are so received by a receiver and converted in a parallel form, the output data.

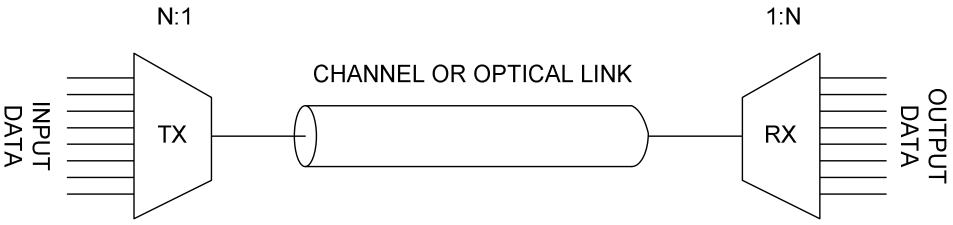


Figure 6: Generic serial data link.

1.3.2 Serial channel link

The goals of a transmitter for high-speed communication system are to convert an incoming parallel data stream to a serial data stream and send it to the receiver, with the appropriate slope and amplitude, through a channel [3]. For this reason, often the transmitter is called serializer.

A general block diagram is presented in Figure 7.

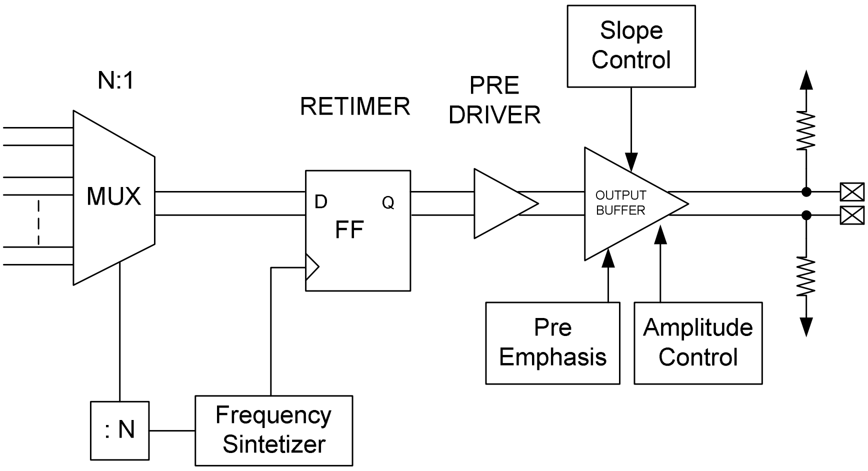


Figure 7: Transmitter block diagram.

Increasing the data rate up to several Gb/s over cables of several meters or long on-chip interconnects, the propagation of the signal is affected by non-idealities due to circuit performance, limited channel's bandwidth and reflection.

The attenuation of cable increases with cable length and frequency. Therefore pre-emphasis techniques in the transmitters have been adopted to compensate for the inter-symbol interference (ISI) distortions.

Traditional interconnect design required numerous data pins, address pins, control signal and clocking signals. These bus based designs could provide point to multipoint interconnections but as the number of devices connected to a bus increased, so did the associate capacitance, reducing achievable data rate.

Recent trends in high-speed system interconnect concentrates on reducing the pin count, increasing the overall throughput and decreasing complexity and cost. To obtain the highest data throughput while reducing the pin count or data bus width, the clock frequency must be increased. Increasing the clock frequency makes the channel more sensitive to the capacitive loading effects, including reflections as would be experienced by adding multiple drop nodes on the interconnect, and crosstalk, caused by poor electrical interface [4].

Moreover long, high-speed pc-board traces operate in a zone influenced by both skin-effect and dielectric losses. Both mechanisms attenuate the high-frequency portion of your signals but in slightly different ways. The skin effect is the tendency of current flow in a conductor to be confined to a layer in the conductor close to its outer surface. As frequency is increased the depth to which the current flow can penetrate is reduced according to skin depth:

1.4 $SKIN\_DEPTH = \sqrt{\frac{2\rho}{\omega\mu}}$

Where μ is the product of μr (relative permeability) and μ0 (vacuum permeability), ρ is the conductor resistivity (ohm*meters) and ω is the frequency (rad/s).
Consequently we can see that the apparent resistance of a line increases with frequency:

1.5 $R(\omega) = \frac{\rho}{SKIN\_DEPTH}\left(\frac{L}{\pi D}\right)$

where L and D are the line's sizes.
The dielectric losses too are related to the media characteristic. Each media is characterized by a dielectric constant ε; this constant is the product between ε0, the vacuum dielectric constants (expressed in F/m), and εr the **relative** dielectric constants. Dielectric loss tangent (Tanδ) is the ratio between the imaginary and the real part of the dielectric constant and determines the losses of the media.

1.6 $\tan(\delta) = -\frac{\varepsilon_{im}}{\varepsilon_{real}} = \frac{\sigma}{\omega\varepsilon}$

Both the skin-effect and the dielectric-loss effect degrade digital signals in the same fundamental way, by smearing the rising and falling edges. The difference is that dielectric attenuation varies directly with frequency, and skin-effect attenuation varies only in proportion to the square root of frequency. Moreover, in backplane as well as in integrated circuit environments, a premium on space and costs precludes creating completely shielded links. Therefore, moving to faster data rates, high frequency signal components couple more electromagnetic energy into neighboring channels.
This coupling manifests as near-end crosstalk (NEXT) and far-end crosstalk (FEXT).
NEXT is interference between two pairs of a cable measured at the same end of the cable as the transmitter; FEXT is interference between two pairs of a cable measured at the other end of the cable from the transmitter.
The channel attenuation increases with frequency, while near-end crosstalk grows up very fast with frequency and at high frequency becomes wider than received signal.

The receiver must detect the input signal and its clock, conditioning the signal itself to maximize the effectiveness of the decision circuit.

Finally the receiver converts the serial data received in a parallel flow and sends it to the logic core. Generic receiver architecture is shown in Figure 8.

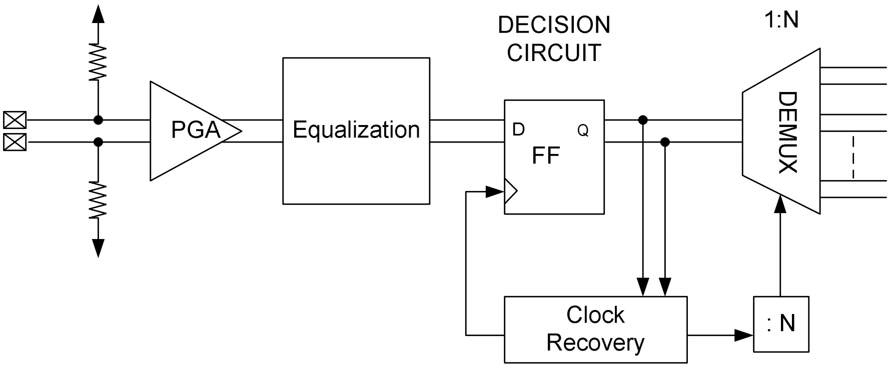


Figure 8: Receiver block diagram.

There is an input matching network followed by a programmable gain amplifier (PGA). The PGA regulates the input data swing, different for each standard, to match the optimal one for the following stages. The received data, amplitude adapted, must be equalized to correctly recognize it.

There are several ways to perform this: an analog equalization, a digital equalization or a mixed mode one. An analog equalization is usually based on a linear equalizer with adaptable boost and peak frequency followed by a decision feedback equalizer (DFE), while a digital equalization is based on an ADC and a successive digital signal processing. The equalized data is correctly recognized by a decision circuit and then de-muxed by a demultiplexer (so the receiver is called de-serializer and the join of transmitter and receiver is called SerDes).

To properly recover the data, the best sampling phase must be chosen and this task is called clock recovery (CR).

1.3.3 Serial channel link standard

The serial links described in this section are: Serial Attached SCSI (SAS), Serial ATA (SATA) and Fiber Channel. SAS is an evolution of parallel SCSI into a point-to-point serial peripheral interface in which controllers are linked directly to disk drives. SAS is a performance improvement over traditional SCSI because SAS enables multiple devices (up to 128) of different sizes and types to be connected simultaneously with thinner and longer cables ; its full-duplex signal transmission supports up to 3.0Gb/s. In addition, SAS drives can be hot-plugged.

SATA extends the ATA technology roadmap by delivering disk interconnect speeds starting at 1.5 Gb/s, moving up to 3, 6 and maybe 12 Gb/s in the next future. Due to its lower cost per gigabyte, SATA will continue as the prevalent disk interface technology in desktop PCs, sub-entry servers and networked storage systems where cost is a primary concern.

An important Serial ATA standard characteristic is Spread Spectrum Clocking (SSC), used to improve the EMI performance of the interface. It takes the form of frequency modulation (FM) of the data clock. This has the effect of spreading the radiated energy across more of the frequency spectrum and lowering the power at any one frequency. SSC also reduces the likelihood that your device will interfere with the operation of some other piece of equipment. The parameters that describe FM are deviation and rate; the standard considers a deep of 5000 ppm and a frequency of 30 kHz. Applying a FM, we are varying the frequency of our signal and the period and the position of our edges. Conversely, the varying edge position claims more challenging during the receiver design.

Fiber Channel (FC) is a technology for transmitting data between computer devices at high data rates. Also Fiber Channel, thanks to his speed, has begun to replace the SCSI as the transmission interface between servers and clustered storage devices. Fiber channel electrical signals are sent over a duplex differential interface. The link may consist of electrical transmission lines such as coaxial cable and shielded twisted pair or two optical fibers. The serial data stream is independent of the transmission medium used; the pattern of ones and zeros is exactly the same whether bits are sent by means of light or electrical signals.

The fiber channel standard is described in the optical link standards section more in depth.

1.3.4 Serial optical link

The idea of using light as a carrier for signal has been around for more than a century, but it was not until the mid 1950s that researches demonstrate the utility of the optical fiber as a medium for light propagation.

A simple optical communication system consists of three components: an electro-optical transducer, for example a laser diode, which converts the electrical data to optical form; a fiber which caries the light produced by the laser (the optical link); and a photodetector, for example a photodiode, which senses the light at the end of the fiber and converts it to an electrical signal. With long or low cost fiber, the light experience considerable attenuation as it travels.

Thus the laser must produce a high light intensity, the photodiode must exhibit a high sensitivity to light and the electrical signal generated by the photodiode must be amplified with low noise.

These observation lead to the system shown in Figure 9: a laser driver deliver a large current to the laser and a transimpedance amplifier (TIA) amplifies the photodiode output with low noise and sufficient bandwidth, converting it to a voltage.

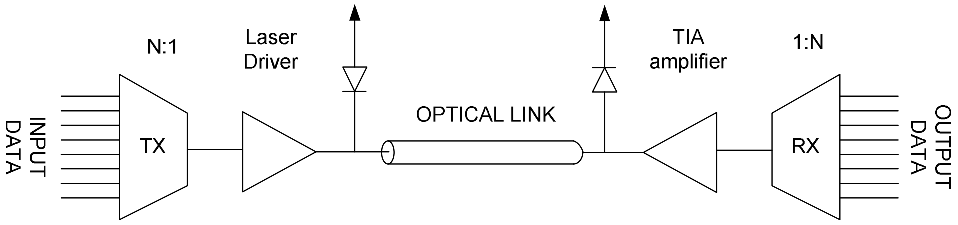


Figure 9: Optical link.

The transmitter and the receiver are equal to the ones described in the preceding paragraph, without the output and input matching network.

The optical fiber consists of a core that carries the light from the transmitter to the receiver; surrounding the core is another layer called the cladding, its function is to confine the light to the core and prevent it from escaping the fiber. The cladding accomplishes this by taking advantage of an optical phenomenon that occurs when light encounters a boundary between two medium with different transmission characteristics: the reflection.

Optical fiber can be categorized as either single mode fiber or multimode fiber based upon the manner in which the light propagates through the fiber. In single-mode fiber, all the light propagates along the same path in the fiber. This is accomplished by reducing the diameter of the core to such a degree that all the light is constrained to follow the same path (the core diameter is approximately 9um).

In multimode fiber, the core diameter is much larger (approximately 50um) resulting in multiple propagation modes, or path, that the light can follow, Figure 10. This results in a phenomena called modal dispersion. Modal dispersion results in spreading of the pulse and ultimately limits the distance and data rate that can be achieved with multimode fiber.

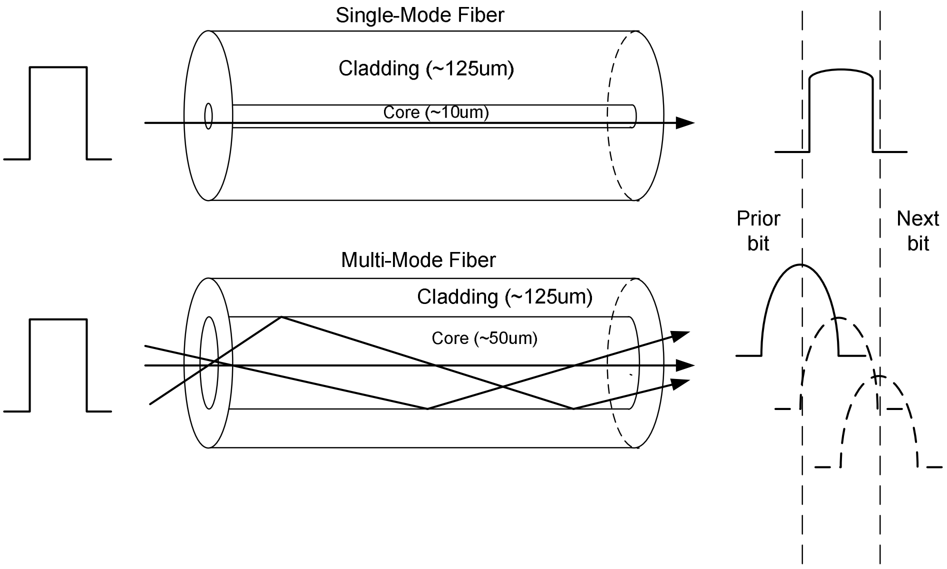


Figure 10: Single-mode and Multimode propagation.

Optical fiber provides several distinct advantages over copper transmission lines that make it very attractive medium for many applications. Among those advantages are:

- Greater distance capability than is generally possible with copper at the same data rate;
- Insensitive to induced electromagnetic interference (EMI);
- No emitted electromagnetic radiation (RFI);
- No electrical connection between the two ports;
- Not susceptible to crosstalk;
- Compact and lightweight cables and connectors.

On the other hand, optical link have some drawbacks:

- Optical links tend to be more expensive than copper links over short distances;
- Optical connector don't lend themselves to backplane printed circuit wiring;
- Optical connector may be affected by dirt and other contamination.

1.3.5 Serial optical link standard

Fibre channel is an example of serial optical link standard [5].

Fibre channel is a technology for transmitting data between computer devices at data rates up to 8.5 Gbit/s, compatible with lower data rate as 1.0625 Gbit/s, 2.125 Gbit/s, 4.25 Gbit/s

This allows different products and configurations to use different signalling rates as appropriate.

Fibre Channel defines multiple optical variants using both single-mode and multimode optical fibers. The purpose in defining multiple options is to allow for flexibility in making cost versus performance trade-offs for different applications.

Using multimode optical fiber the information is sent for a maximum distance of 500m.

Using single-mode optical fiber the information can be sent for a maximum distances in excess of 50 km between a transmitter and a receiver. This distance can be extended even further by using repeaters or proprietary links.

Optical data transmission is accomplished by using electrical signals to control an optical emitter such as a LASER diode. The resultant optical pulses are injected into a fiber optic cable.

Within the fiber optic cable, light pulses are carried via a glass strand approximately the diameter of a human hair. At the receiver end of the optical fiber, the optical pulses are converted back into electrical signals by use of a photo detector.

While it is possible to send signals in both directions simultaneously through a single optical fiber, it is generally simpler and more cost effective (at least for computer interfaces) to use two separate fibers, one carrying information in each direction.

Using two fibers also makes it easier to convert between optical and electrical media because it is not practical to send electrical signals in both directions simultaneously through a single electrical transmission line.

1.4 SYSTEM CHARACTERIZATION

1.4.1 Eye diagram

The eye diagram is a useful tool for the qualitative analysis of signal used in digital transmission. It provides at-a-glance evaluation of system performance and can offer insight into the nature of channel imperfections [7].

Careful analysis of this visual display can give the user a first-order approximation of signal-to-noise, clock timing jitter and skew.

The eye diagram is an oscilloscope display of a digital signal, repetitively sampled to get a good representation of its behaviour: It is a composite view of all the bit periods (UI) of a captured waveform superimposed upon each other.

The eye diagram can be used to examine signal integrity in a purely digital system: such as fiber optic transmission, network cables or on a circuit board.

A good use of the eye diagram is to evaluate the received signal quality. The diagram in Figure 11 illustrates the type of information that can be determined from the eye diagram.

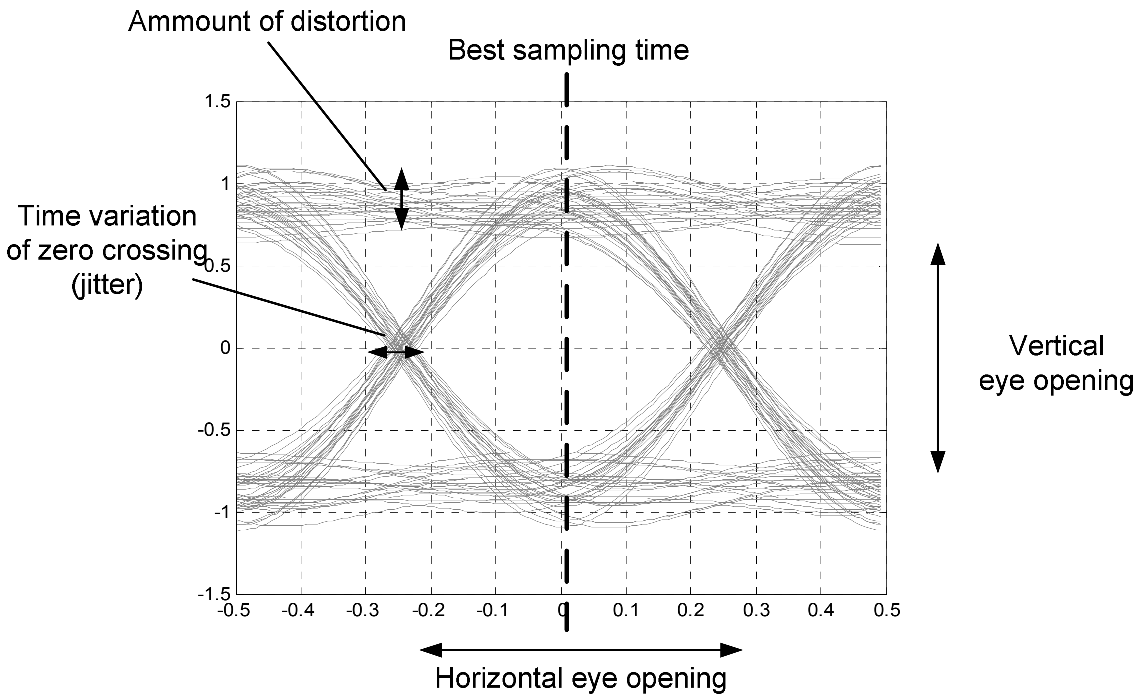


Figure 11: Eye diagram.

1.4.2 Bit error ratio (BER)

Bit error rate, BER is a key parameter that is used in assessing systems that transmit digital data from one location to another. Systems for which bit error rate is applicable include radio data links as well as fibre optic data systems, Ethernet, or any system that transmits data over a network of some form where noise, interference, and phase jitter may cause degradation of the digital signal.

As the name implies, a bit error rate is defined as the rate at which errors occur in a transmission system.

The definition of bit error rate can be translated into a simple formula:

$$\textbf{1.7} \quad BER = \frac{Number\ of\ errors}{Total\ number\ of\ bits\ sent}$$

Typical BER values in data storage application are 10^{-12} or 10^{-15}. In some applications, called error-free, the BER must be 10^{-17}.

1.4.3 Bathtub curves

A bathtub curve is a graph of BER versus sampling point throughout the Unit Interval, Figure 12.

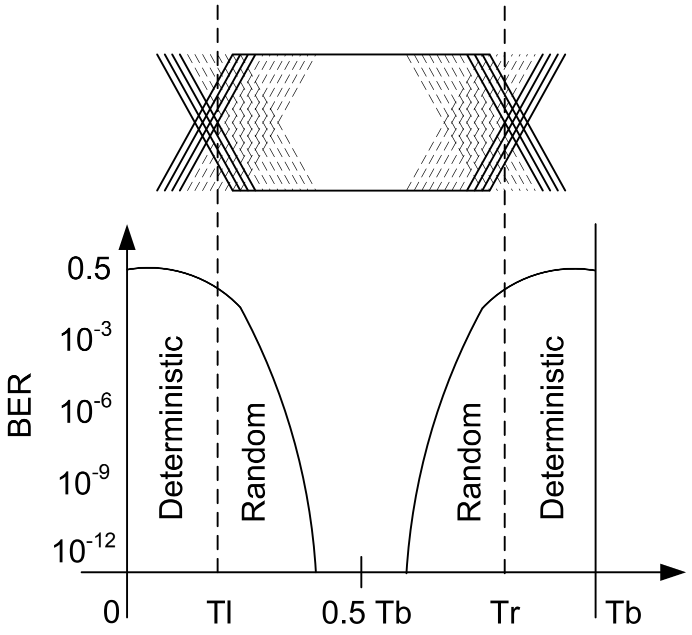


Figure 12: Bathtub plot.

It is so named because its characteristic curve looks like the cross-section of a bathtub.

A bathtub plot is typically shown with a log scale that illustrates the functional relationship between sampling time and BER [8].

When the sampling point is at or near the transition points, the BER is 0.5: equal probability for success or failure of a bit transmission. The curve is fairly flat in these regions, which are dominated by deterministic jitter phenomena.

As the sampling point moves inward from both ends of the unit interval, the BER drops off precipitously. These regions are dominated by random-jitter phenomena and the BER is determined by the sigma of the Gaussian processes producing the random jitter. As one would expect, the center of the unit interval provides the optimum sampling point.

1.4.4 Jitter tolerance

Serial data communication embeds the clock signal in its transmitting data bit stream. At the receiver side, this clock needs to be recovered through a clock recovery (CR) device where phase-locked loop (PLL) circuits are commonly used. It is well known that a PLL typically has certain frequency response characteristics. Therefore, when a receiver uses the recovered clock to time or retime the received data, the jitter seen by the receiver follows certain frequency response functions.

The clock recovery circuit typically has a low-pass frequency response function with a pole at f=fc. This suggests that a receiver can track more low-frequency jitter at frequencies of f < fc than at higher frequencies of f > fc.

Jitter tolerance is defined as the maximum jitter amplitude, typically sinusoidal jitter, in the input data stream of a receiver not producing errors or synchronization anomalies. The jitter amplitude corresponds with the closure of the incoming data's eye.

The sinusoidal jitter's amounts at different frequencies are often specified by the recommendations in the form of masks.

In Figure 13 is shown an example of jitter tolerance: the jitter amplitude is specified in UI versus frequency.

Jitter tolerance is used to define a specific about the "strength" of the receiver.

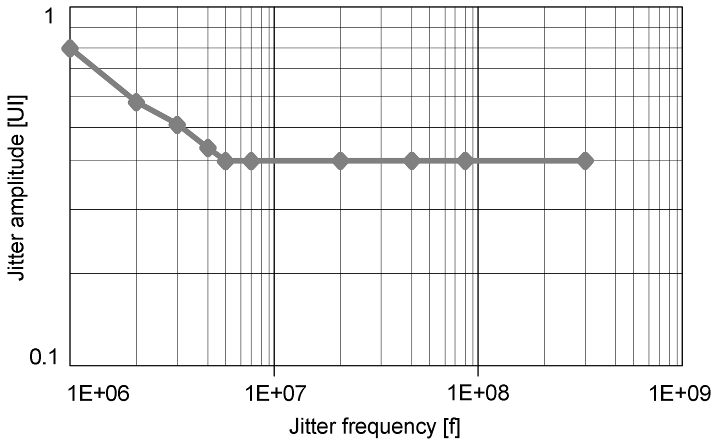


Figure 13: Jitter tolerance.

1.5 PERFORMANCES LIMITS

The serial link performances (the ability to correctly recognize the transmitted data) are limited by a number of non idealities.

The channel bandwidth limitation or the modal dispersion of an optical fiber introduces intersymbol interference (ISI) in the binary data and so deteriorate the detection of data, degrading both the amplitude and the time resolution.

The noise of the receiver can also significantly impact the detection of the data and affect, like the channel bandwidth limitation, both horizontal and vertical eye opening.

1.5.1 Intersymbol Interference (ISI)

Intersymbol interference is a form of distortion of a signal in which one symbol interferes with subsequent and preceding symbols. This is an unwanted phenomenon as the subsequent and preceding symbols have similar effect as noise, thus making the communication less reliable. ISI is usually caused by bandlimited channels or multipath propagation causing successive symbols to "mix" together. The presence of ISI in the system introduces errors in the decision device at the receiver output. Therefore, in the design of the transmitting and receiving filters, the objective is to minimize the effects of ISI, and thereby deliver the digital data to its destination with the smallest error rate possible.

A communication link is usually characterized by its frequency response (of low pass type) and its impulse response.

Figure 14 shows a generic impulse response of a cable with bandwidth limitation: C0 is called the cursor and normally represent the maximum amplitude of the pulse, C1, C2 ecc. are called post-cursor and represent the pulse amplitudes at distances multiples of the bit duration (Tb) after the cursor, C-1 is called the pre-cursor and is the pulse amplitude at a distance Tb before the cursor.

If a series of bits are sent through a channel the time response of the individual bits can be summed with the correct time shift due to the time invariance of the transmission medium.

Figure 15 shows this behaviour: if the current bit is considered in the sampling instant, the value is given by the sum of the current cursor plus the preceding bits post-cursors and the subsequent bit pre-cursor.

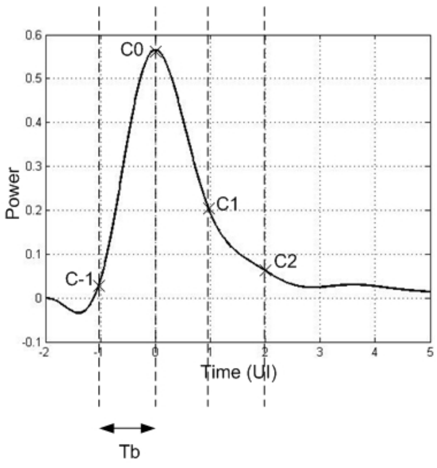


Figure 14: Channel impulse response.

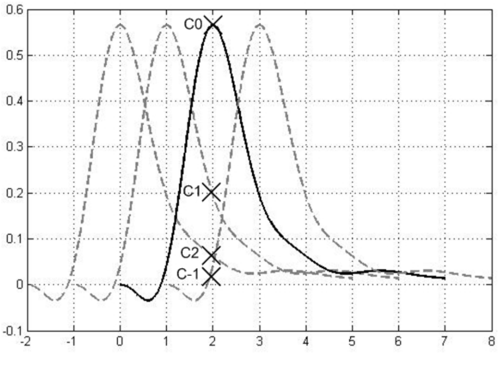


Figure 15: ISI in a bandlimited channel.

The pulses response of a multimode optical fiber can have a different shape with respect to the cable tipical one shown in Figure 14. This behaviour is due to the modal dispersion.

The pulse response can present a big pre-cursor, a symmetric shape with a pre and post cursor bigger than the cursor or a big post-cursor. Figure 16 shows these types of impulse response in a multimodal fiber as defined by the IEEE 802.3aq 10GBASE-LRM standard [6].

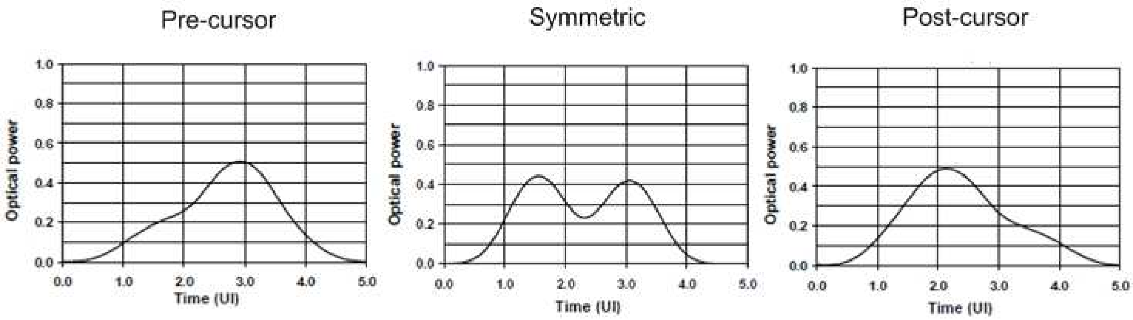


Figure 16: Optical pulses shape.

1.5.2 Noise

Random data propagation through a cable or optical link may experience considerable attenuation. Thus the noise at the receiver can significantly impact the detection of the data. Since noise directly trades with gain, bandwidth and power dissipation of the receiver circuits, it is important to determine how much noise can be tolerated for given performances. The data bits must ideally be sampled by the clock at their midpoint so as to provide maximum distance from the decision. It's possible to derive the error rate in terms of the additive noise amplitude.

If the binary sequences toggle between -V_0 and V_0 with equal probabilities its probability density function (PDF) is Px(x) and consists of two impulses at -V_0 and V_0, each having a weight of 1/2; the PDF of the noise n(t) is Pn(n) and exhibits a Gaussian distribution with zero mean and RMS value σ_n.

We can see that the PDF of the signal + noise, Figure 17, is the convolution of the two PDF and consist of two Gaussian distribution centered around –V_0 and V_0, the shaded tails represent samples of -V_0 +n(t) that are positive and samples of V_0 +n(t) that are negative.

We must now calculate the probability of error: the probability that the actual bit is a logical ZERO but the received level -V_0 +n(t) is positive is given by the shaded area in Figure 17 from 0 to +∞:

$$\textbf{1.8} \quad P_{0->1} = \frac{1}{2}\int_{0}^{+\infty} \frac{1}{\sigma_n\sqrt{2\pi}} \cdot \exp\left(\frac{-(n+V_0)^2}{2\sigma_n^2}\right) dn$$

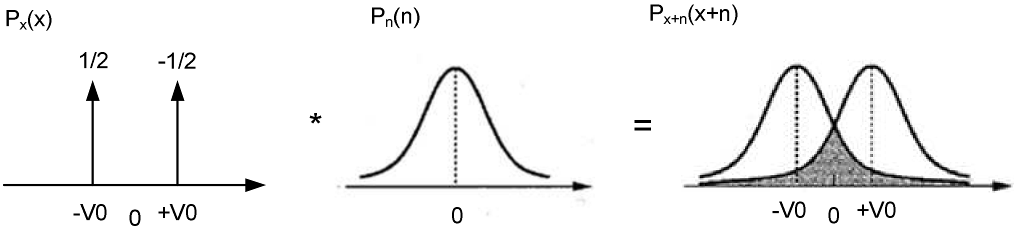


Figure 17: PDF of the signal + noise.

If ONEs and ZEROs arrive with equal probabilities and the noise corrupts high and low levels equally than $P_{0->1} = P_{1->0}$ and we need calculate only one and multiply by two this result to obtain the total probability:

$$\textbf{1.9} \quad P_{TOT} = Q\left(\frac{V_0}{\sigma_n}\right)$$

Where Q(x) is the Q function defined as:

$$\textbf{1.10} \quad Q(x) = \int_{x}^{+\infty} \frac{1}{\sqrt{2\pi}} \cdot \exp\left(\frac{-u^2}{2}\right) du$$

How can we expect the BER decrease as the $V_0/\ \sigma_n$ increase.

Figure 18 shows the BER versus V_0/σ_n. If we want a BER of 10^{-12}, the ratio between half the signal range and the noise standard deviation must be at least equal to 7.

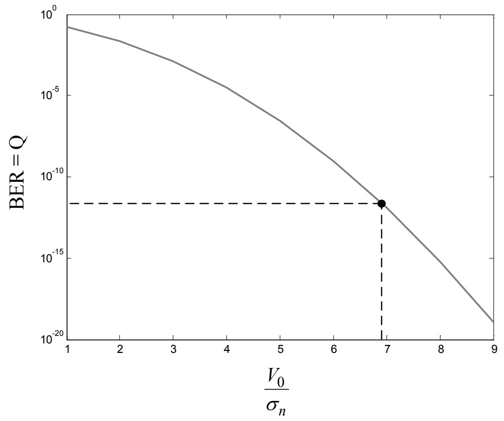


Figure 18: BER versus V_0/σ_n.

1.5.3 Jitter

Jitter is the deviation of the zero crossing from their ideal position in a signal (clock or data). The goal of the receiver is to sample the incoming signal at the data center in order to maximize the eye opening. Both the clock and data jitter deteriorate the horizontal eye opening.

In the analysis of serial data system, it's useful to distinguish two categories of jitter: random jitter and deterministic jitter.

The deterministic jitter is mainly due to ISI caused by the transmission media, Duty Cycle Distortion (DCD) in the transmitted data, and Periodic Jitter caused by cross-coupling or EMI problem. The deterministic jitter is referred to the data.

This jitter reduces the horizontal eye diagram by a limited amount and for this reason is said to be bounded.

The random jitter arises from phase noise random processes typically found in VCO structures or clock sources. VCO phase noise varies the period of oscillation randomly as if the oscillator occasionally operates at frequencies different from its nominal and the zero crossing may not occur at integer of the period. In general all the circuit noise is a cause of random jitter. The random jitter is referred to the clock.

The Probability Density Function (PDF) of the random jitter follows a Gaussian distribution and is said to be un-bounded, Figure 19.

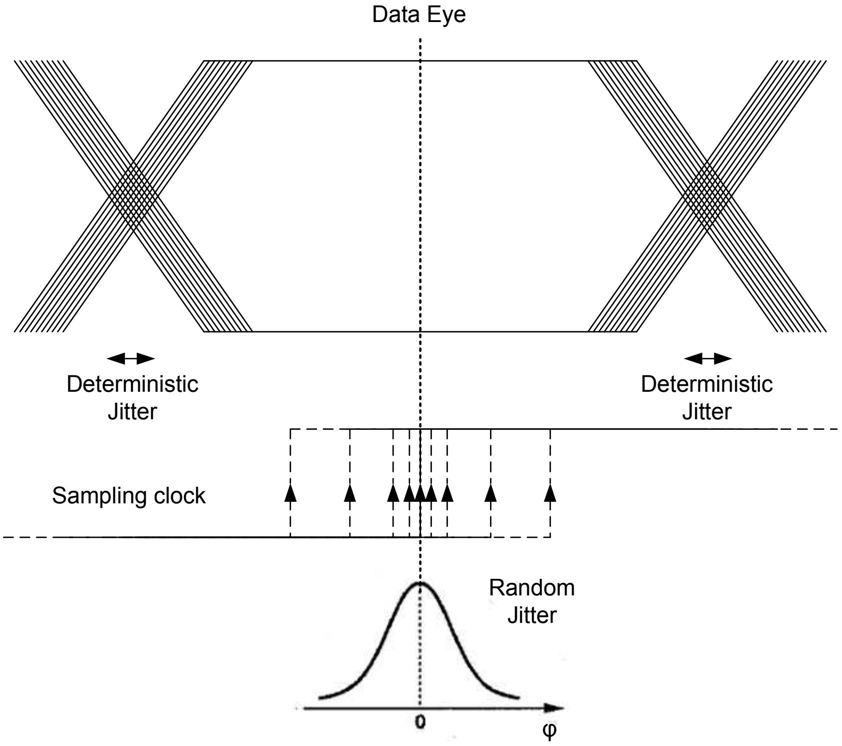


Figure 19: Deterministic and Random jitter.

The peak-to-peak value depends on the observation time and its standard deviation is commonly used to describe its magnitude.

1.6 ANALOG EQUALIZATION TECHNIQUES

1.6.1 Introduction

The two main sources of signal distortion in a digital communication channels are ISI and additive noise. The ISI is due to bandlimited channels or multipath propagation and can be characterized by the channel transfer function in the frequency domain or by the impulse response, in the time domain.

The noise can be internal to the system or external to the system; if the noise is introduced primarily by electronic components and amplifiers at the receiver, it may be characterized as thermal noise.

Hence at the receiver the distortion must be compensated in order to reconstruct the transmitted symbols. This process of suppressing channel induced distortion is called channel equalization.

An analog equalization approach uses usually a linear equalizer followed by a DFE (decision feedback equalizer).

1.6.2 Linear equalizer

A possible solution of the channel induced ISI problem is to compensate or reduce it with a high-pass filter called equalizer. The channel is a band-limited transmission media of low-pass type.

From the communication theory it is well know that the optimum receiver can be realized as a filter matching the inverse of the channel transfer function in the frequency domain, followed by a sampler operating at the symbol rate [9]. In a digital communication system the channel frequency response is not known with sufficient precision to design optimum filter.

Time-continuous analog filters are extensively employed in systems for analog processing of high-frequency signals. These circuits synthesize the requested transfer function in analogic way.

In order to obtain a high pass filter transfer function, a number of zeros must be introduced at low frequency adequate to match the channel shape, Figure 20.

The equalizer boost is usually placed at the Nyquist frequency, half the data rate one. This frequency is a good compromise because the signal power spectral density is mainly contained under this frequency and the high frequency noise contributions are not too amplified.

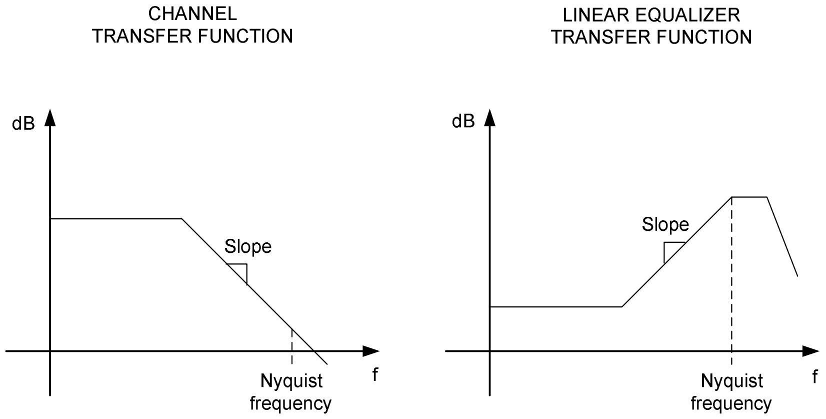


Figure 20: Channel and linear equalizer transfer function.

The described transfer function can be obtained using the circuit shown in Figure 21. The differential gain voltage can be easily calculated neglecting the finite value of the drain-source resistance.

1.11 $$G = \frac{Vout}{Vin} = \frac{gm}{1+gmR} \cdot \frac{1+S2CR}{1+\left(\frac{S2CR}{1+gmR}\right)} \cdot \frac{R_L}{1+SR_L C_P}$$

Cp is the parasitic capacitance of the circuit that loads the equalizer output.

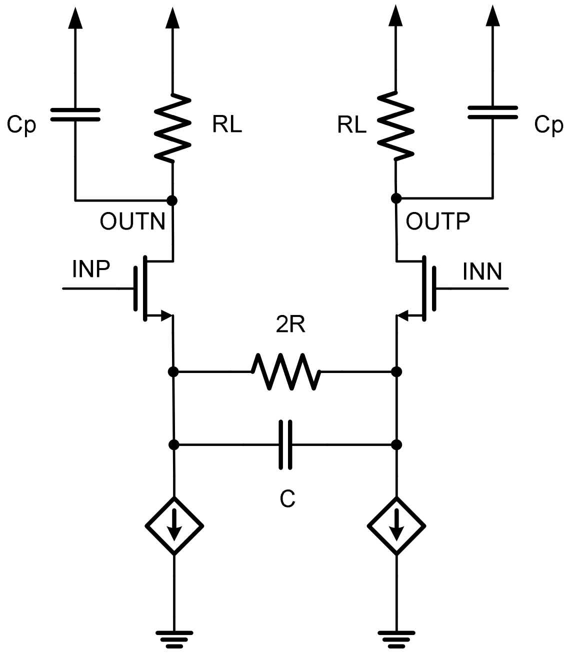


Figure 21: Linear equalizer.

The RC source degeneration introduces a zero-pole couple with the pole at a frequency (1+gmR) higher than the zero. The load resistance RL and the output parasitic capacitance Cp introduce an additional high frequency pole. To obtain the desired boost, the RC value must be chosen as a trade of between the DC gain and the high frequency boost. This corresponds to a compromise between sensitivity, dynamic range, noise and offset tolerance versus the capability to match the channel. In order to increase the boost a chain of degenerated differential pairs can be used.

1.6.3 Decision Feedback equalizer (DFE)

A band limited channel can be described in the time domain by a Finite Impulse Response (FIR) filter with this transfer function, [9]:

$$\textbf{1.12} \quad S_n = \sum_{i=-N_{pre}}^{N_{post}} C_i B_{n-i}$$

Where S_n is the analog value of the distorted symbol after the channel at the time n, B_x is the x-th digital bit value, C_x is the x-th pre or post cursor value, N_{pre} and N_{post} are the number of pre and post cursors of the impulse response.

The digital DFE then uses information about previously received bits to cancel out their ISI contributions from the current decision, as shown in Figure 22. This figure shows the block diagram of a 3 taps digital DFE. The shift register clock is a clock at the data rate frequency (full-rate).

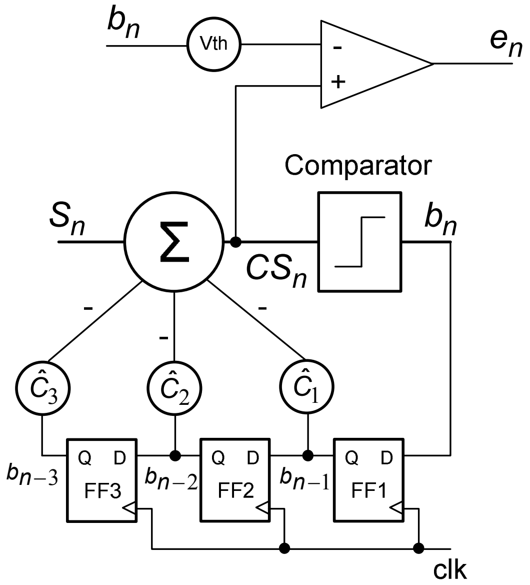


Figure 22: 3 taps DFE block diagram.

If the values of the estimated post-cursors $\hat{C}_1$, $\hat{C}_2$ and $\hat{C}_3$ are known the circuit removes the ISI of the three previous bits. The DFE can only remove post-cursor ISI. The values of these coefficients are computed through a least mean square algorithm (LMS) that uses the error e(n) signal.

A drawback of DFE implementations is that at high data rates, the latency of the feedback loop in the standard implementation can present a serious bottleneck. The corrections, in fact must be applied in less than one bit time to be effective. The most critical path in term of timing is the one that start at the flip-flops outputs, pass through the post-cursor multiplier, the summer and the comparator and is applied at the first flip-flop input.

The LMS algorithm works is now explained.

The CS_n signal is the signal after the ISI correction and can be written in this way:

1.13
$$CS_n = \sum_{i=-N_{pre}}^{N_{post}} C_i b_{n-i} - \sum_{i=1}^{3} \hat{C}_i b_{n-i}$$
$$CS_n = \sum_{i=-N_{pre}}^{0} C_i b_{n-i} + \sum_{i=4}^{N_{post}} C_i b_{n-i} + \sum_{i=1}^{3} (C_i - \hat{C}_i) b_{n-i}$$

Assuming that the distortion due to the post-cursors after third and to the pre-cursors is negligible, the ISI is corrected.

The signal e_n is defined as the difference between the equalized signal and the signal called Vth. The Vth must also be computed through the LMS algorithm. In first approximation the Vth will converge to the cursor $\hat{C}_0$ and for the moment we suppose it is equal to this value.

1.14 $$e_n = CS_n - \hat{C}_0 b_n = \sum_{i=-N_{pre}}^{-1} C_i b_{n-i} + \sum_{i=4}^{N_{post}} C_i b_{n-i} + \sum_{i=0}^{3} (C_i - \hat{C}_i) b_{n-i}$$

Correlating e_n with the previous bits remains only one term different by zero in terms of mean value.

1.15 $$e_n \cdot b_{n-i} = C_i - \hat{C}_i$$

This is true when the input stream is formed by random data, without correlation between them. The post-cursors estimated values are updated in this way:

$$\textbf{1.16} \quad \hat{C}\_new_i = \hat{C}\_old_i + \mu \cdot e_n \cdot b_{n-i}$$

μ is the integral gain of the coefficients update. When the coefficients converge they usually toggle between two values.
The Vth value is also adapted correlating the error with the actual received bit

$$\textbf{1.17} \quad Vth\_new = Vth\_old \ + \mu \cdot e_n \cdot b_n$$

In a digital implementation of this algorithm it is important to underline that to adapt the coefficients is enough the sign of the error.

1.6.4 Loop-Unrolled digital DFE

One or more taps of feedback equalization can be achieved by using loop unrolling to avoid the bottleneck in the latency of the feedback loop [10].
Since we cannot run the feedback loop fast enough, we can unroll it once and make two decisions each cycle. One comparator decides the input as if the previous output was a 1, and the other comparator decides the input as if the previous bit was a 0. Once we know the previous bit, we select the correct comparator output.
Instead of just one data sampler, for binary signaling the receiver has two samplers that are offset by ±α, anticipating the impact of trailing ISI α from a previously sent symbol of value of ±1, Figure 23.
This method can be applied to two or more taps of feedback. To remove N taps of ISI 2^N samplers are required, and their offset are all the possible combination of ISI.
Usually, only a small amount of unrolling is needed to bridge the latency gap.

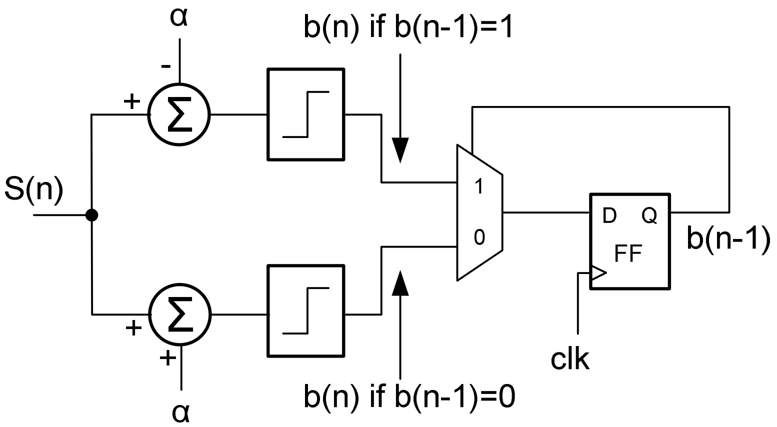


Figure 23: One tap DFE using loop-unrolling.

1.6.5 Half-rate DFE with one tap of loop unrolling

An elegant and efficient DFE architecture that relaxes the timing requirements by a factor of two with respect to the full-rate topology, while still achieving equalization at the maximum data rate, is shown in Figure 24 [11].

This particular DFE employs two techniques to achieve this feat: loop unrolling and half rate clocking. In loop-unrolling decisions are made for both cases in which the previous bit was a “1” and a “0”; this is accomplished by having two analog summers, two regenerative flip-flops, and a mux. The C1 tap now functions as a DC offset and is not dynamically switched. A second flip-flop at the output then drives the mux select, and effectively picks the “correct” decision, thus ignoring the “wrong” decision.

In order to allow 2UI for the critical timing path to settle, a second technique of half rate clocking is employed.

Here, a half-rate clock drives two duplicate paths at opposite clock phases. Decisions “ping-pong” back and forth between the two paths, generates even and odd bit sequences.

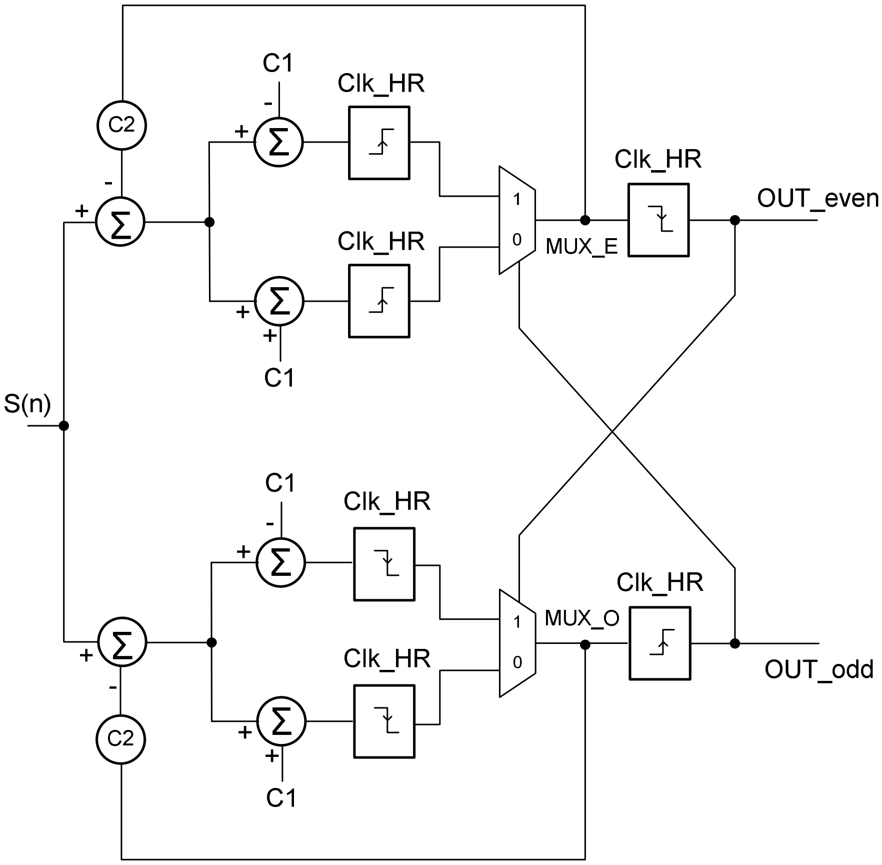


Figure 24: DFE architecture employing half-rate clocking and one tap of loop-unrolling.

Figure 25 shows the time diagram of this DFE: S(n) is the input signal: the transmitted bits plus the ISI of the successive and previous ones; MUX_E signal is the input signal sampled and corrected on the rising clock edge (only the even bits); MUX_O signal is the input signal sampled and corrected on the falling clock edge (only the odd bits); OUT_even and OUT_odd are the re-sampled even and odd bits.

After the sampling instant, shown in figure, the MUX_E signal is b(n-2) while before this instant was b(n-4), this value is multiplied by the second post-cursor C2 and subtracted at the input signal. This path, starting at the flip-flops output before the multiplexer, passing through the mux, the C2 multiplier and the two summers, is the critical one but it has 2UI of time to settle.

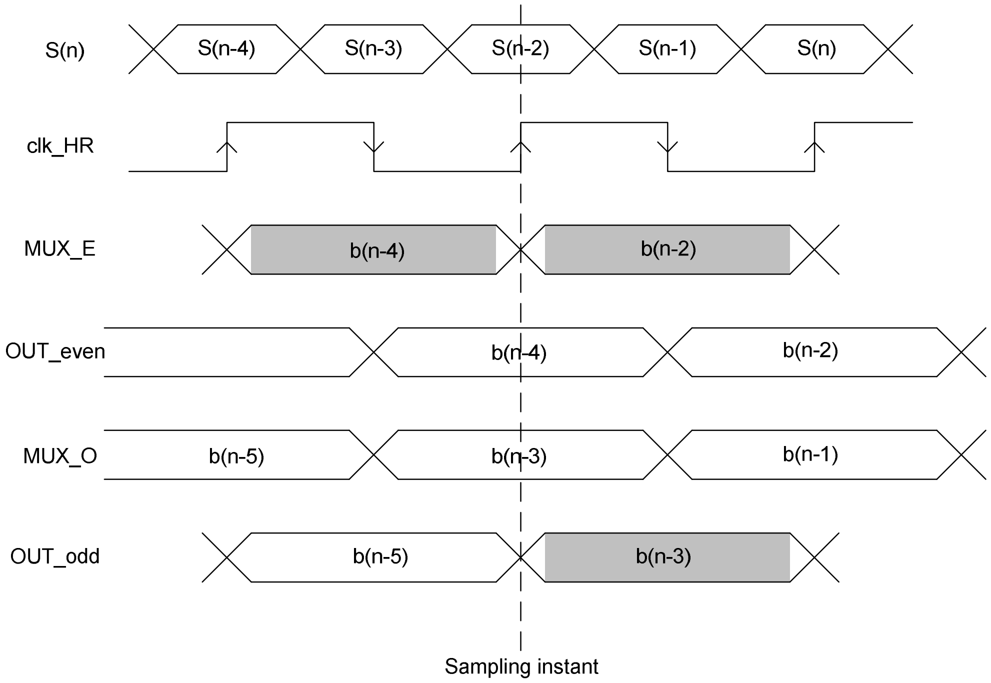


Figure 25: Half-rate DFE timing diagram.

The even mux selection signal, at the current sampling instant, is instead the odd path output b(n-3).

The same described processing is performed to equalize the odd bits.

In this way both the first and second post-cursor of ISI are correctly subtracted to the input signal.

1.6.6 Continuous time DFE (CTDFE)

Continuous time or un-clocked architecture is an efficient way to overcome the first feedback loop latency bottleneck exhibited by the conventional digital DFE architecture.

As displayed in Figure 22, digital DFE feeds back binary decisions through a FIR filter, while continuous time decisions are fed back through variable delays approximately equal to multiple of the bit time, [12], [13].

Figure 26 shows an example of a two taps continuous time DFE.

Delays values can be controlled dynamically so that there is an additional freedom degree to maximize the equalization efficiency. The tunable feature also enables dynamic adjustment for changes to the design goal caused by process, voltage and temperature variations.

In the next chapter, we explain an efficient algorithm used to auto-adapt the delay value.

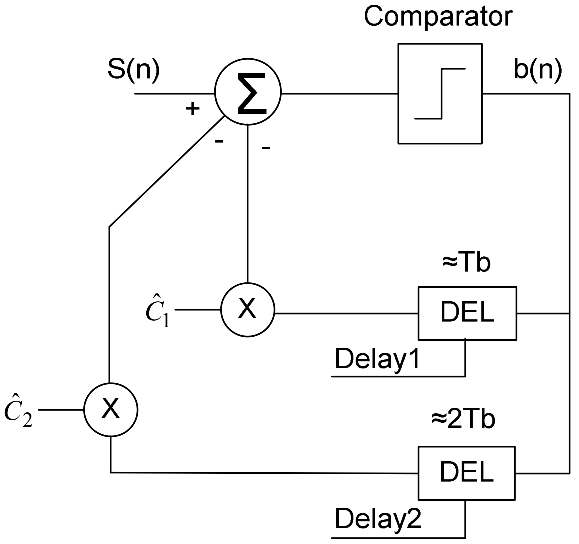


Figure 26: Second order continuous time DFE.

1.7 DIGITAL EQUALIZATION TECHNIQUES

1.7.1 Introduction

A digital equalization approach is based on the digital conversion through an ADC of the input signal, the transmitted one distorted by the channel, and a successive elaboration focused in the reconstruction of the transmitted bit

The big bottleneck is the power consumption required by the ADC. This converter must be able to sample the incoming signal at the data-rate and convert it with the resolution required by the employed equalization technique.

The digital approach enables power/area scaling with process, simplifies production testing, allows integration of a FFE, and provides a flexible design with a configurable number of filter taps.

The considered digital equalization techniques are two: a digital feed-forward equalizer (FFE) filter and a Viterbi algorithm which is a maximum-likelihood sequence estimator.

1.7.2 Digital Feed-forward equalizer (FFE)

A programmable feed-forward equalizer is usually used as first equalization circuit after the ADC to invert the effect of the channel. The FFE is basically a finite impulse response (FIR) auto-adaptive filter. A FIR filter is a type of a discrete-time filter. The impulse response is finite because it settles to zero in a finite number of sample intervals. This is in contrast to infinite impulse response (IIR) filters, which have internal feedback and may continue to respond indefinitely.

As reported in Figure 27, a FIR filter is formed by a repetition of delay cells, multiplication blocks and adder. Each section, formed by a delay and multiplication pair, is called tap. The output results the sum of the current input with the Ntap previous bits weighted by corresponding coefficients Wi, called tap weight.

1.18 $$OUT_n = \sum_{i=0}^{N_{TAP}} W_i \cdot b_i$$

The impulse response of these filters is defined by the set of tap weights Wi.

Usually an LMS algorithm is used to auto adapt this digital filter setting the taps coefficients in order to have the optimum impulse response that maximizes the eye opening.

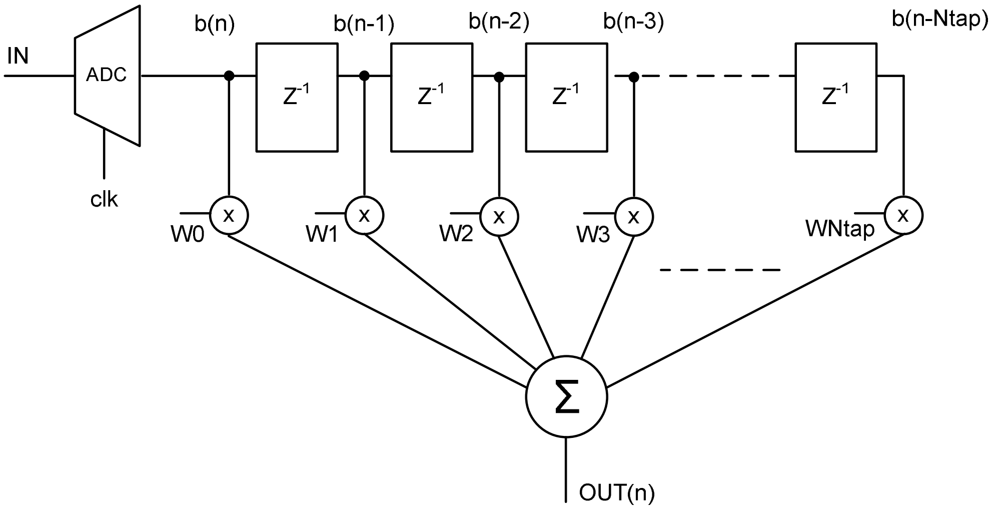


Figure 27: FIR filter.

The frequency response of this filter must be of high pass type and perform a boost to match the magnitude of the channel transfer function with reasonable accuracy

However, with high losses channels, it is not convenient to boost the signal to much because the ADC quantization noise is also amplified.

For this reason usually the equalized signal, after the FFE, is left with some residual ISI.

In order to equalize further the signal a digital DFE or a Viterbi decoding algorithm is used.

1.7.3 VDA (Viterbi decoding algorithm)

In 1967 Andrew Viterbi first presented his now famous algorithm for the decoding of convolutional codes [14]. A few years later, what is now known as the Viterbi decoding algorithm (VDA) was applied to the detection of data signals distorted ISI [15].

In this paragraph we introduce the basic functionality of the VDA without going into details.

A NRZ signal, distorted by a band limited transmission medium, does not show only two possible values (one and zero), but is spread due to ISI over a range of values depending on the number of impulse pre-cursor and post-cursor and on the preceding and subsequent transmitted bits. If the ISI is large it is impossible to recognise the correct transmitted bit by the signal sign. Moreover if the transmitted medium is an optical link and its impulse response presents, for example, a big precursor like the impulses defined in the IEEE 802.3aq 10GBASE- LRM standard [6], a DFE is not adequate to equalize the signal. VDA, instead, is adequate to equalize data link affected by big amount of ISI and also to equalize multimodal fiber data link.

VDA works in this way: we consider a set of possible signal values depending on the number of precursor and post-cursor taken into account. For example if we take into account one pre-cursor and one post-cursor, there are $2^3=8$ considered values.

The received signal value is not equal to one in the considered set, because of the additional ISI due to non-considered post-cursors and pre-cursors, but it is possible to find the more similar one. We can define a distance metric between the received value and the values in the considered set; the value in the considered set with the smaller distance respect to the received one is the more similar. The knowledge of this more similar value is not enough to recognize the correct transmitted bit because there are more than one bit sequences that generates that value.

If we know a sequence of at least K received value (where K-1 is the link ISI), it is possible to find the correct value of one bit.

A maximum likelihood algorithm computes the cumulative distance of the received sequence of N values (where $N \geq K$) with all the possible values corresponding to all the possible bit sequences, and select the more similar one.

Usually the VDA is preceded by an FFE filter that shapes the impulse response to obtain the desired values of pre and post cursors.

The VDA requires an ADC to convert the input signal. The signal value is necessary to compute the distances.

1.8 CDR TECHNIQUES

1.8.1 Introduction

The data stream received is both asynchronous and noisy. For subsequent processing, timing information a clock must be extracted from the data so as to allow synchronous operation. Furthermore the data must be retimed such that the jitter accumulated during transmission is removed. This task of clock extraction and retiming is called clock and data recovery.

The clock generated must bear a certain phase relationship with respect to data, allowing optimum sampling of the bits by the clock. If the sampling instants coincides with the midpoint of each bit, than the sampling occurs farthest from the preceding and following data transition, providing maximum margin for jitter and other timing uncertainties.

There are two main categories of clock extraction circuits: open loop filters and closed loop synchronizers.

The first solution doesn’t suffer from instability and nonlinear problems, but requires high selective external filters and additional calibration to ensure the correct alignment. Conversely, the closed loop CDR is fully integrable and thanks to the loop can self-compensate for changes in the environment.

A closed loop CDR is implemented as shown in Figure 28. A phase detector first recognizes the phase error between the sampling clock and the incoming data, it incorporates also the sampling circuit. After which the error is integrated and this value, passed through a loop filter, is used to generate the sampling clock.

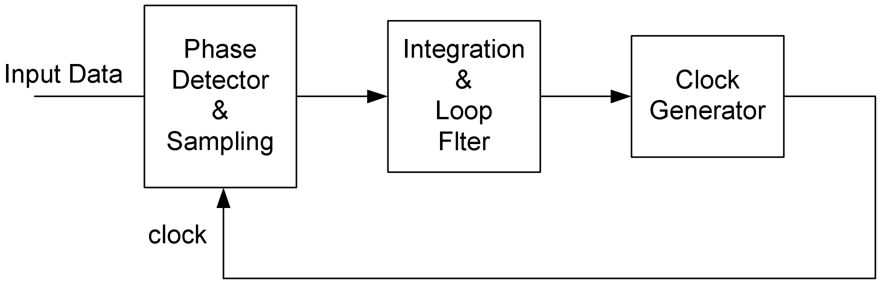


Figure 28: Closed loop CDR.

1.8.2 Linear PD (Hogge)

The recovered clock must sample the data in the middle of the eye. In a phase tracking system this is accomplished by measuring the phase difference between the data and the clock and driving it toward the wished value.

The Hogge phase detector is shown in Figure 29.

Since sample B changes only on the clk edges X = Din XOR B contains pulses whose width represents the phase difference between Din and clk.

It is important to note that the circuit produces a pulse for each data transition, thereby providing edge detection, and the width of the output pulses varies linearity with the input phase difference, suggesting that the circuit can operate as a linear PD. This output is called "proportional pulses".

The X output can't be used alone as phase detector because its average value is a function of the transition density and fails to uniquely represent the phase difference for various data patterns.

To overcome the above ambiguity, the proportional pulses must be accompanied by a reference pulses, the Y output. The latter are impulses that appear on data edges but exhibit a constant width, thus eliminating the pattern dependency.

The difference between the areas under X and Y can be viewed as a PD output, eliminating the ambiguity due to transition density.

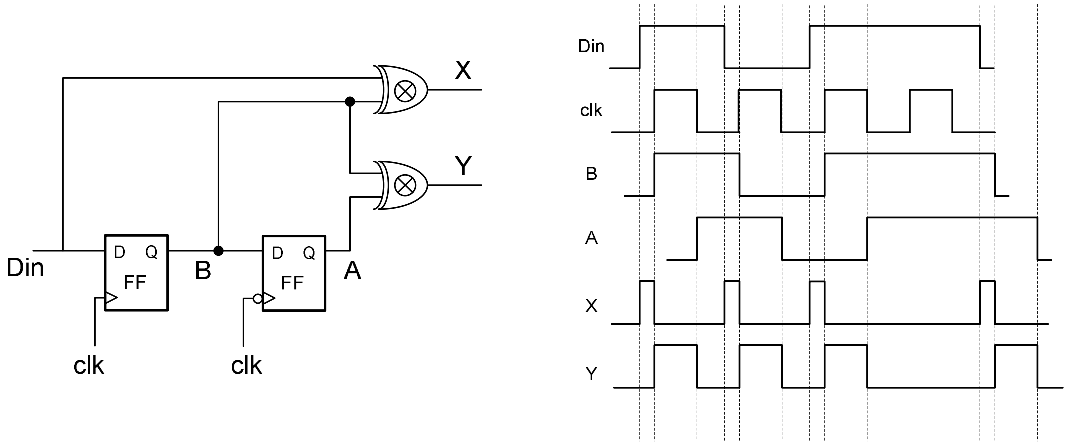


Figure 29: Hogge phase detector.

The Hogge topology is a true linear phase detector, generating a vanishing average as the phase difference approaches zero.

Usually the Hogge PD output is integrated using a charge pump and the clock is generated through a voltage controlled oscillator (VCO).

1.8.3 Bang bang PD (Alexander)

Figure 30 illustrates the Alexander phase detector principle, also known as "early-late" detection method [16]. Using three data samples S1-S3 taken by three consecutive clock edges, the PD can determine whether a data transition is present and whether the clock leads or lag the data. In the absence of data transition, all three samples are equal and no action is taken.

If the clock is early respected to the data, than the last sample S3 is unequal to the first two S1 and S2. Conversely if the clock is late, the last two samples S2 and S3 are equal but unequal to the first sample S1.

Thus S1 XOR S2 and S2 XOR S3 provide the early-late information:

$$\textbf{1.19}\quad \begin{Bmatrix} S1 \otimes S2 = 0 \; AND \; S2 \otimes S3 = 1 & clock\, EARLY \\ S1 \otimes S2 = 1 \; AND \; S2 \otimes S3 = 0 & clock\, LATE \\ S1 \otimes S2 = 0 \; AND \; S2 \otimes S3 = 0 & no\, Transition \end{Bmatrix}$$

The foregoing observations lead to the circuit topology shown in Figure 31. Flip-flop FF1 samples S1 and S3 on the rising edge of clk and FF2 delays the result by one clock cycle. Flip_flop FF3 samples S2 on the falling edge of clk and FF4 delays this sample by half a clock cycle.

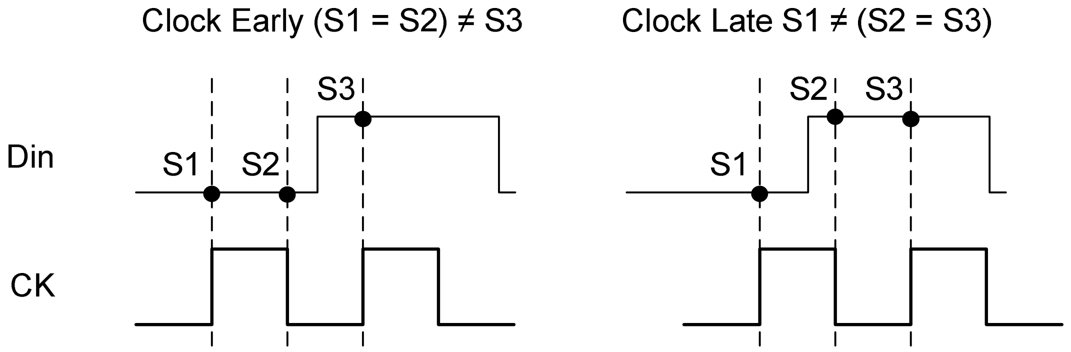


Figure 30: Alexander PD functioning.

As depicted in Figure 31, the first rising edge of clk samples a high data level. The second rising edge of clk than accomplishes two tasks: it produces a delayed version of the first sample at the output of FF2, and it samples the low level on the input data. On the first falling edge of clk, FF3 samples a high level on the input data and on the next rising edge, FF4 reproduces this level. The values of S1, S2, S3 and S4 are therefore valid for comparison at t=t1. As a result the XOR gates generate valid outputs simultaneously.

The Alexander phase detector is a bang-bang system and the relationship between phase difference and output signals is strongly not linear: the output phase information are only two EARLY or LATE. The CDR loop locks such that S2 coincides with the data zero crossing.

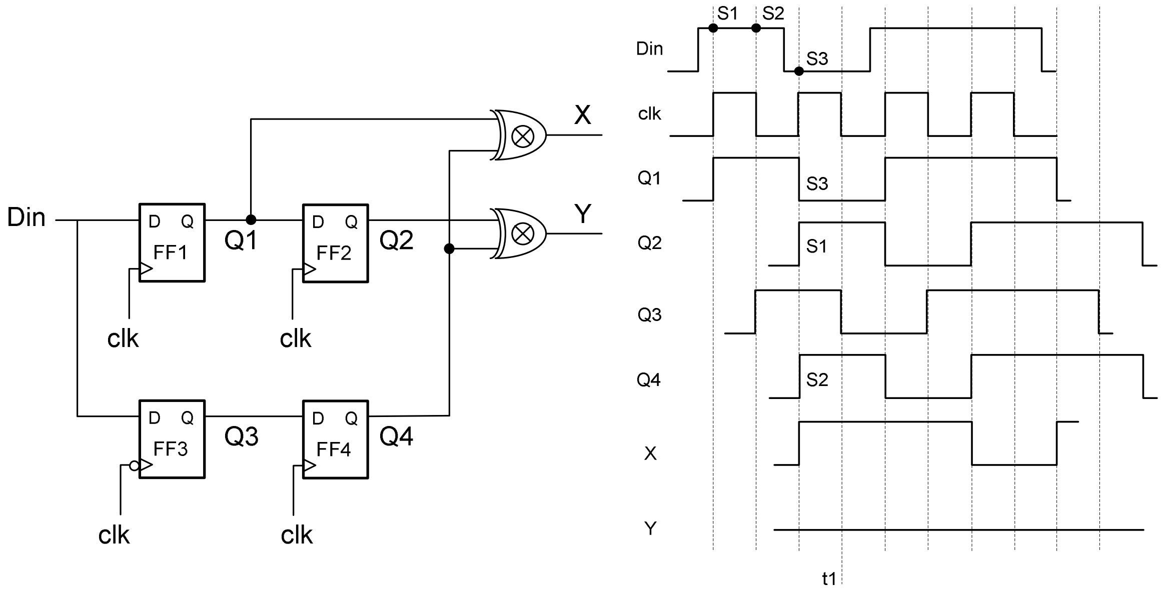


Figure 31: Alexander phase detector.

1.8.4 Mueller Muller PD

Mueller Muller is an example of a baud rate CDR method and is usually used in a digital receiver where the signal is A/D converted with a clock frequency equal to the symbol rate.

The two phase detecting methods seen before are based on the zero threshold crossing of the received signal, and on the comparison, of this point, with the value sampled on the center of the eye. For this reason, these PD methods are not usually used in digital receiver, where the ADC samples and converts only the signal on the center of the eye.

Mueller Muller instead uses the signal derivative at the sampling instant; this derivative, or at least its sign, is usually correlated with the estimated data to produce the updating information required for the timing control loop. The resulting sampling phase is such that the mean square error between the signal and the appropriate reference level is minimized, or, with slight changes, such that the sampling will occur at the peak of the impulse response.

A possible implementation of this algorithm is shown in Figure 32, where s(n) represent the digital converted value at the instant n, and b(n) is its sign.

Correlating s(n) with b(n-1) and subtracting the correlation between s(n-1) and b(n), a phase information is obtained. This information is proportional to the difference between the post-cursor and the pre-cursor of the system impulse response, [17].

1.20 $s(n)\times b(n-1)-s(n-1)\times b(n)\propto(C_1-C_{-1})$

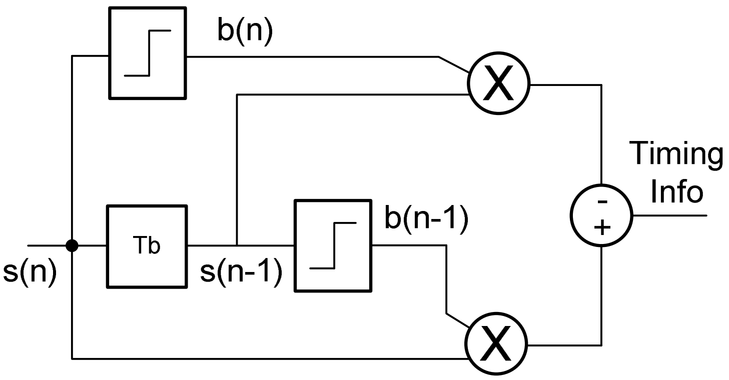


Figure 32: Mueller Muller algorithm block diagram.

Mueller Muller can be equivalently seen as an algorithm that sets the sampling clock in the midpoint between two time values. These time values are the ones at which the impulse response amplitude is equal to the estimated first post-cursor and pre-cursor. Figure 33 shows an example of the early-late information generated by supposing a symmetrical impulse response with equal post-cursor and pre-cursor values.

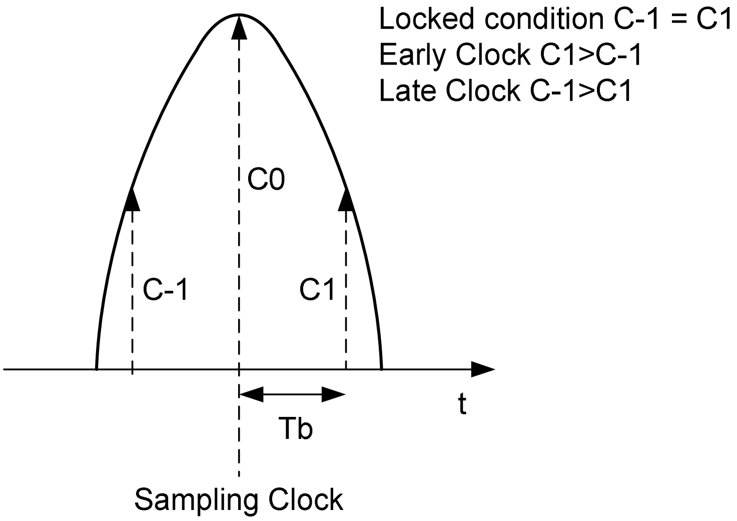


Figure 33: Early-late information in a Mueller Muller algorithm.

1.8.5 Clock generation (charge pump & VCO)

The integrated phase error information must be used to generate the sampling clock. When the error signal is provided from a charge pump followed by an analog filter the most common option is to use a Voltage Controlled Ring Oscillator or LC based VCO. Figure 34 shows a CDR loop incorporating a Hogge PD. The XOR outputs drive a charge pump and a loop filter, the output of the loop filter is the VCO input. The need for a charge pump in linear CDR loops poses serious speed limitations.

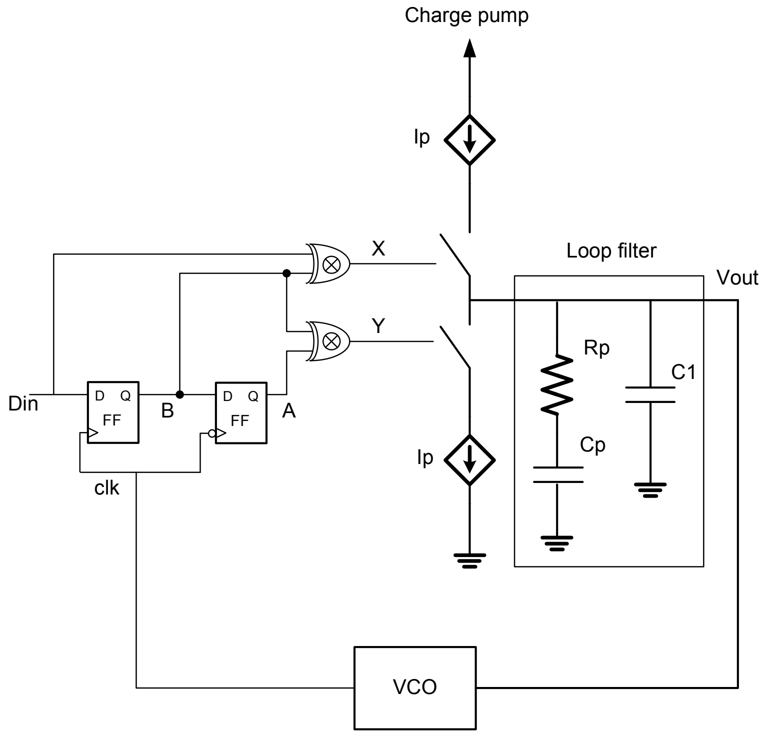


Figure 34: CDR based on a Hogge PD, a charge pump and an analog loop filter.

1.8.6 *Clock generation (phase interpolator)*

When the integrated error is a digital word, coming from a digital filter implemented inside the CDR's loop, solutions based on the UI quantization in time and phase domains, become more practical. This digital approach is used, for example, with Alexander or Mueller Muller PD.

The clock Unit interval is divided in a fixed number of phases and the loop filter output is a digital number representing one phase. Figure 35 shows an example of UI quantization.

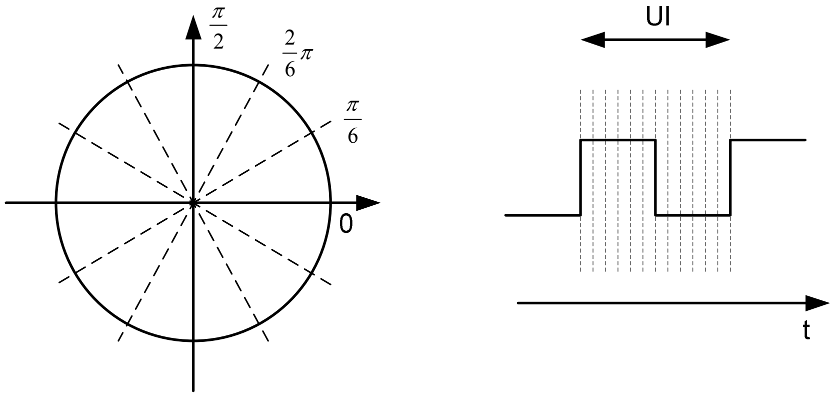


Figure 35: Clock unit interval quantization.

A widely used approach, to substitute VCO in digital CDR loop, uses phase interpolator. Given two sine wave, the linear combination of them is again a sine wave, with phase intermediate between the initial ones phases.

Figure 36 shows the block diagram of a digital CDR and a phase interpolator.

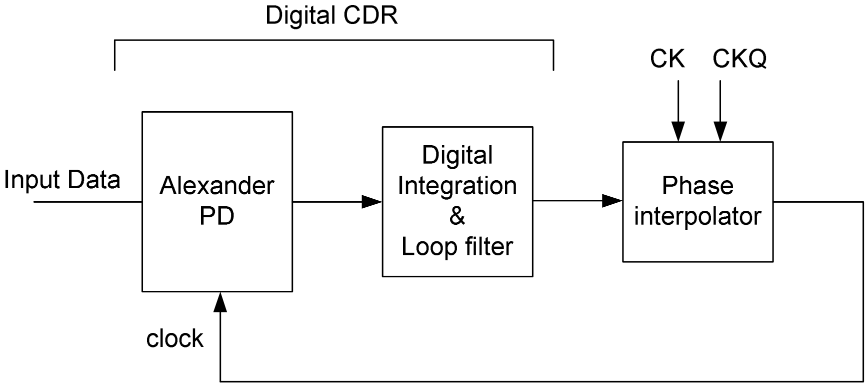


Figure 36: Digital CDR and PI.

Basically a phase interpolator can be made of two or more differential pairs with programmable tail current I1 and I2, Figure 37. The inputs are two differential reference clocks CK and CKQ. The ratio between the current represents the weight of the two sine waves. Different ratio implies different phase's offset: this ratio is the CDR output.

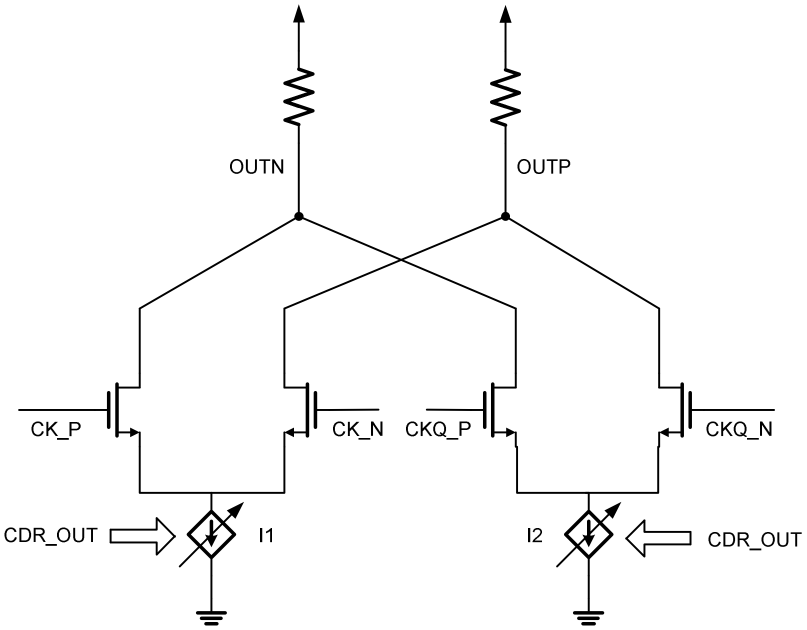


Figure 37: Phase interpolator.

1.9 SUMMARY

In this chapter we have introduced some concepts about modern communication systems. The most used modulation code (NRZ) and the main data communication standard (SATA, SAS and Fiber Channel) have been analyzed.

We have also described the most important system characterization metrics (eye diagram, BER, bathtub curve and jitter tolerance) and the sources of performance limits (ISI, noise and jitter).

We have analyzed the most important equalization techniques, both analog and digital. The analog ones are: linear equalizer (LE) and different topology of decision feedback equalizer (DFE). The digital ones are: FIR filter (FFE), and the Viterbi decoding algorithm (VDA).

Finally we have described different clock and data recovery (CDR) topologies.

The next chapter deals with a 12 Gb/s receiver and in particular the designed digital circuits.

Chapter 2

DIGITAL CIRCUITS FOR A 12GB/S RECEIVER

2.1 INTRODUCTION

This chapter focuses on a 12 Gb/s analog receiver for the standards described in the previous chapter. The architecture is based on a continuous-time DFE and on a bang-bang CDR.

The implemented digital circuits are described: a bang-bang CDR for the timing recovery, the circuits implementing the auto-adaptive algorithms for horizontal and vertical eye opening maximization and some auxiliary circuits. Finally the die micrograph, measurements results and literature comparison are shown.

2.2 RECEIVER ARCHITECTURE

The implemented receiver for a 12 Gb/s maximum data-rate is shown in Figure 38. The NRZ data is provided to a programmable gain amplifier (PGA), that correctly sets the signal amplitude, to a linear equalizer (LE), that partially compensates for the channel induced ISI, and to a continuous time DFE.

The LE pre-shapes the channel pulse-response into a pulse-response with a single dominant post-cursor and this residual post-cursor is removed by the single tap un-clocked DFE.

The DFE feedback is composed by a limiting amplifier, a variable delay (Δ) and a multiplier for $\hat{C}_1$ that represents the residual post-cursor value.

Both the post cursor value $\hat{C}_1$ and the delay value (Δ) are self-adapted. $\hat{C}_1$ is computed through an LMS algorithm that maximizes the vertical eye opening; Δ is computed through an algorithm capable to concentrate the transitions at the data edge, that optimize the horizontal eye opening.

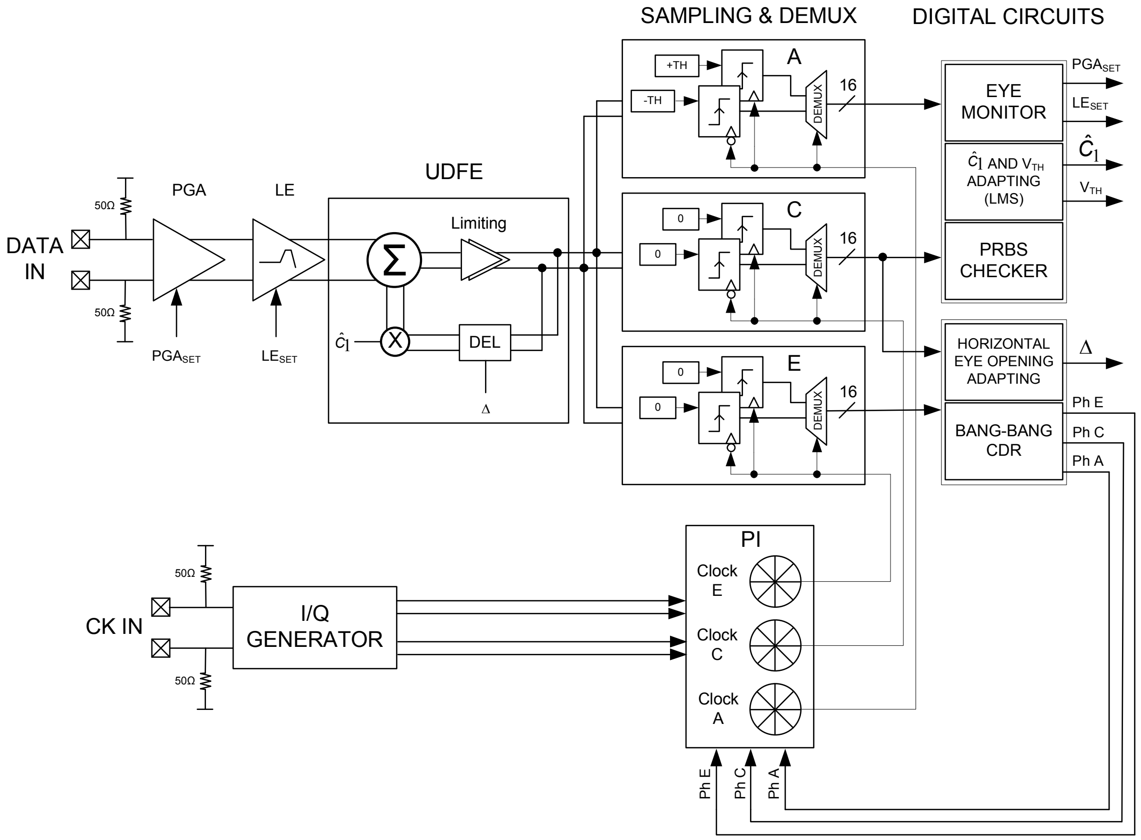


Figure 38: Receiver architecture.

The linear path, made of PGA, LE and unclocked DFE (UDFE), feeds three half-rate sampling paths: center (C), edge (E) and auxiliary (A).

Data through path C are sampled at the center of the eye; data through path E are sampled at data edge. Both of them are demuxed to the bang-bang CDR and to the DFE delay control engine that optimizes the horizontal eye opening.

The auxiliary path A can sample data in variable phase position and respected to programmable thresholds (+/-TH).

Path A can compute the error value of the LMS algorithm to adapt the DFE tap $\hat{C}_1$. In this way the LMS algorithm computes also the threshold TH that, in first approximation, is equal to the cursor C0.

Path A can also work as an internal Eye Monitor; this feature is employed to plot the eye diagram at the system start-up and to optimize the PGA and the LE.

The CDR drives three phase interpolators (PI) generating the three sampling clocks.

2.3 BANG-BANG CDR

2.3.1 *Digital CDR*

The main issue into designing a digital CDR is to minimizing the latency between input and output and so to minimize the number of pipe-line stage. A digital bang-bang CDR is composed by the phase detector, the loop filter and a digital circuit which generates the representation of the sampling phase, the phase interpolator input. Minimizing the input-output latency, the jitter tolerance is maximized.

We have designed the CDR in only three pipeline stage: the first stage is composed by an Alexander type phase detector, the second stage encloses the proportional-integral loop filter and the third one encloses the phase generation circuit, Figure 39.

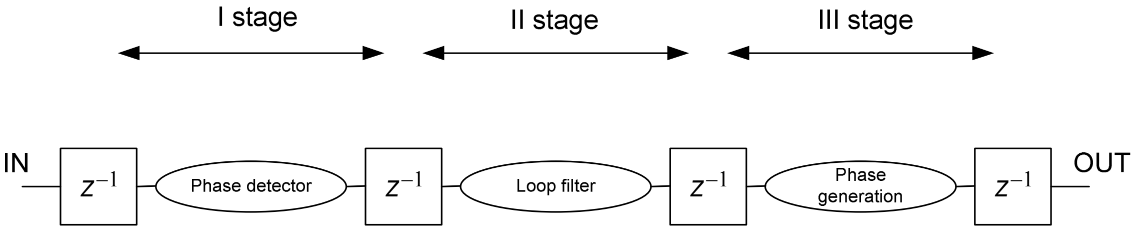


Figure 39: CDR pipe-line stages.

The CDR inputs are 16 bits digital words, and its clock has a frequency of 750 MHz. The input output latency is (1/750MHz)*3 = 4ns. This delay impacts on the CDR bandwidth.

2.3.2 *Phase detector*

The inputs of the bang-bang CDR are two digital words of 16 bits corresponding to the two demultiplexed sampling paths C and E. The CDR clock frequency is the ratio between 12Ghz (the maximum data-rate) and 16; so the clock frequency is 750Mhz. If not specified, the flip-flops work on the clock rising edge.

The phase detection is performed through an Alexander type phase detector [16] that generates early-late information in parallel.

Figure 40 shows the data, the two clocks used for sampling the data center and the edge, and the notation of all digital circuits. This notation is the "MSB first" one, this means that the most significant bit, the 15th one, is the older.

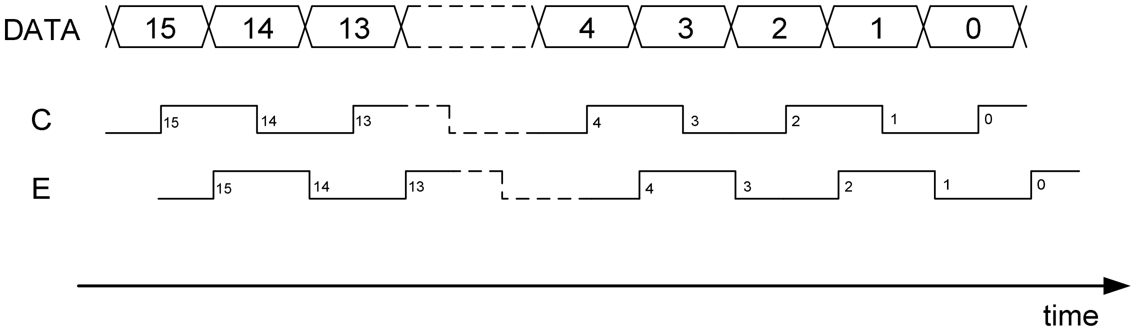


Figure 40: Data and clocks for 12 Gb/s mode.

A digital logic computes early-late information in the following way:

$$\left\{ \begin{array}{l} clock\,EARLY\_12G < n-1 >= \overline{C < n > \oplus E < n >}\;\; AND\;\; C < n-1 > \oplus E < n > \\ clock\,LATE\_12G < n-1 >= C < n > \oplus E < n >\;\; AND\;\; \overline{C < n-1 > \oplus E < n >} \end{array} \right\}$$

$$1 \leq n \leq 15$$

The two vectors contain 15 bits. In both vectors each bit is 1 if there is an early information, zero otherwise.

The system is designed for working in the 12Gb/s mode and in the 6Gb/s mode. In the 6Gb/s mode one of the two half-rate paths, that sample data in C, E, and A position, is turned off and clock E is aligned with clock C, Figure 41. In 6Gb/s mode we don't use a 3GHz clock because the phase interpolators are designed to work with 6GHz clock frequency.

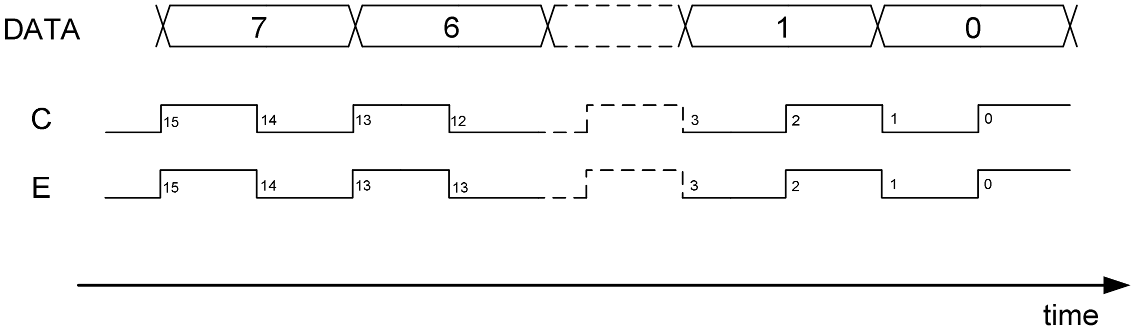


Figure 41: Data and clocks for 6 Gb/s mode.

The following logic generates early-late information:

$$\left\{\begin{array}{l} clock\,EARLY\_6G<n>=\overline{C<2n+2>\oplus E<2n+1>}\;\;AND\;\;C<2n>\oplus E<2n+1> \\ clock\,LATE\_6G<n>=C<2n+2>\oplus E<2n+1>\;\;AND\;\;\overline{C<2n>\oplus E<2n+1>} \end{array}\right\}$$

$$0 \le n \le 6$$

The two vectors clockEARLY and clockLATE for the two modes (6 and 12 Gb/s) contain the number of early and late information. In order to make these information suitable for a loop filter, they must be converted in a binary format and subtracted, Figure 42. Moreover the programming bit “bit16 selects the 6 or 12 Gb/s mode.

The signal EPLN is the binary representation of the phase information: positive if the clock is early, negative otherwise.

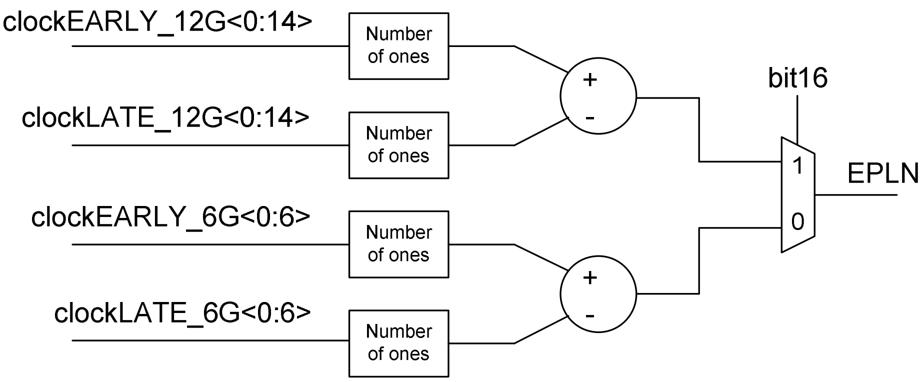


Figure 42: Early late binary conversion.

The combinatorial logic between the CDR input and the EPLN signal is enclosed in the first pipe-line stage.

2.3.3 Loop Filter

The second pipe-line stage encloses the CDR loop filter, that has a proportional and an integral part, each with a programmable gain Prop_GAIN and Int_GAIN, Figure 43.

The loop filter need an integral part to track a huge frequency difference between the received signal and the re-sampling clock. The proportional part tracks the jitter.

In the digital domain an integrator is simply an accumulator; this accumulator can assume a maximum value of 256.

There is also the possibility to reset the accumulator setting the programming bit “int_rst” and pre-charge the integral variable “int_rst_val” with the accumulator value.

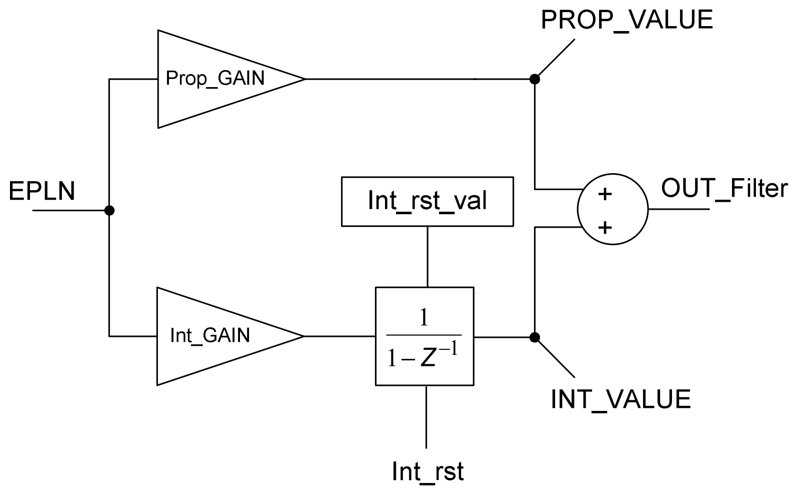


Figure 43: Loop filter.

2.3.4 Signals of phase increment and decrement

The third pipeline stage encloses the circuit described in this paragraph. The output of the proportional-integral loop filter is stored in an accumulator. The accumulator generates the signals (sig_bit_P, SEL1 and SEL2) of increment or decrement of the clock phase; these signals can represent variation of one step or two consecutive steps.

If the phase must be decremented, the signal sig_bit_P is 1; if the phase must be incremented, the signal sig_bit_P is 0.

The signals SEL1 and SEL2 are set in this way:

- One step (SEL1 = 1 and SEL2 = 0);
- Two steps (SEL1 = 1 and SEL2 = 1);
- No phase variation (SEL1 = 0 and SEL2 = 0);

Figure 44 shows the block diagram of the digital circuits that generate these signals.

The OUT_filter signal is in a two's complement format and its MSB is the sign bit. This signal is also accumulated and when the value of the accumulator exceeds appropriate thresholds the signal SEL1 and SEL2 are activated.

SEL1 is set if the 3 current and preceding MSB of the accumulator are unequal; SEL2 is set if these 3 bit differ more than ±1 and simultaneously SEL1 signal is set.

Figure 45 shows an example with a 10 bits accumulator: when the accumulator overcomes a single threshold of value 128, the sampling clock performs a single phase step. If more than one threshold is overcome by the accumulator value, the sampling clock performs two phase steps.

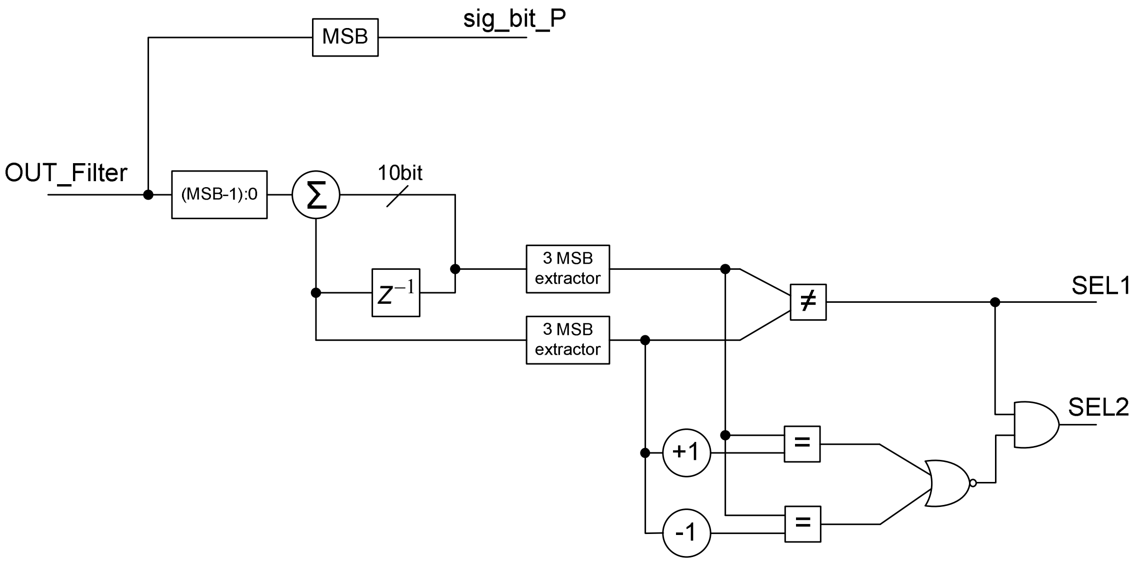


Figure 44: Generation of signals of phase increment and decrement.

	3 MSB	
1023	111	1111111
		⋮
512	100	0000000
511	011	1111111
		⋮
384	011	0000000
383	010	1111111
		⋮
256	010	0000000
255	001	1111111
		⋮
128	001	0000000
127	000	1111111
		⋮
0	000	0000000

Figure 45: Accumulator thresholds.

2.3.5 Clock phases generation

Figure 46 shows the circuit that increment or decrement the clock phase. PH represents the current clock phase and PH-2, PH+2, PH-1 and PH+1 are the incremented or decremented phase signals that are computed in a look-ahead configuration.

The signals sig_bit_P, SEL1 and SEL2, described in the previous paragraph, correctly select the new clock phase. Out1, that is a partially phase update, can be modified of ±1 respect to the old value or be unchanged, out2 can be equal to out1 or modified of ±2.

There are 64 phases [0:63] in a clock period; when PH reach the value 64 it restarts from 0.

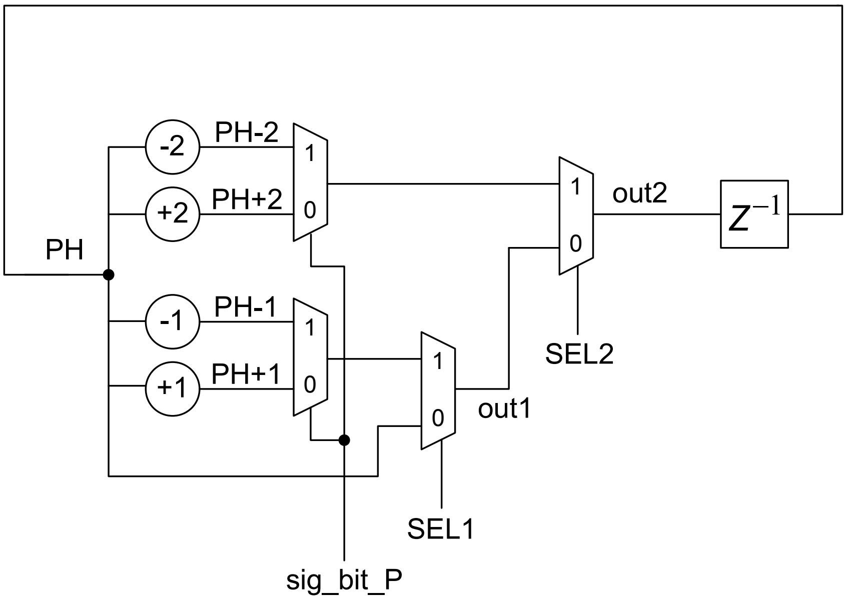


Figure 46: Clock phase update.

The signals out1 and out2 are the binary representation of the phases of the clock that samples the center of the data. These signals must be converted in the correct format to drive the phase interpolator. The phase interpolator inputs are two digital words: a word of two bit in Gray code represents the quadrant of the Cartesian plane and a word of 15 bits in thermometer code represents one of the 16 phases in one quadrant. Out1 and out2 are converted in this representation.

The clock phases for E and A paths must also be generated: these phases are obtained adding a programmable offset to the PH phase. For example delaying the PH signal of 32, we obtain the E clock phase. Figure 47 shows this signals generation for out1.

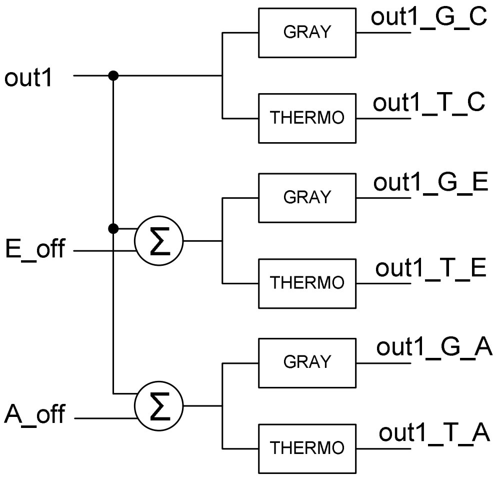


Figure 47: Gray and Thermo conversion.

The clock phase can be incremented or decremented by one or two steps . The clock phase can't be modified of two steps at the same time because the phase rotators are not designed to do that. For this reason, a "two step variation" is performed as two consecutive "one step variation". One step is made on the falling edge of the CDR clock and the successive on the rising edge. Figure 48 shows the circuit that performs this task for the thermometer code of the C path clock. Out_T_C is equal to out2_T_C when the CDR clock (clk_CDR) is low, while Out_T_C is equal to out1_T_C, when clk_CDR is high. In this way the first step is performed on the falling edge and the second step on the rising edge. This circuit introduces also a latency of half clk_CDR period in addiction to the three pipeline stage. Each CDR outputs employ this circuit: these outputs are the thermometric and Gray representation of the clock phase for paths C, E and A

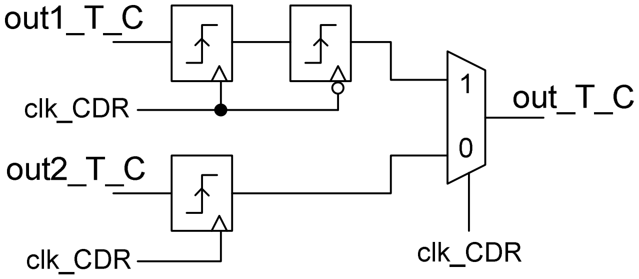


Figure 48: Output circuit.

2.3.6 Tracking of frequency difference between clock and data

The frequency difference between data and clock is mainly due to two reasons:

- The unavoidable small difference between the transmitter and receiver clock frequency.
- A frequency data modulation employed to reduce electro magnetic interference (EMI)

This frequency difference between clock and data is quantified in parts per million (PPM).

In order to track this frequency difference the CDR uses mainly the integral part; the integrator moves to a specific value. This specific value is the accumulator input and its thresholds are overcame at constant time intervals; this lead to a constant increment or decrement of the phase value.

The accumulator generates a "change phase information" when its value overcomes a threshold of value 128.

A phase variation occurs when the number of CDR clock cycles reach the following value:

2.1 $$\left(\frac{INT\_VALUE}{128}\right)^{-1}$$

A phase variation is equivalent to 1/32 UI and the CDR clock has a period equal to 16 UI. The following equation is equivalent to the mean value of the phase changing normalized to the UI value:

2.2 $$\frac{INT\_VALUE}{128}\cdot\frac{1}{16}\cdot\frac{1}{32}$$

We can also use the following relation:

2.3 $$\frac{INT\_VALUE}{128}\cdot\frac{1}{16}\cdot\frac{1}{32}=\frac{PPM}{10^{6}}$$

PPM/10^6 is the difference between the UI and the sampling clock period divided by two. It is also possible write the formula as following:

2.4 $$INT\_VALUE=\frac{65536\cdot PPM}{10^{6}}$$

The integrator value can be in the range [-256:256]; we can calculate the maximum number of PPM that the CDR is able to track. This value is approximately equal to ±3900 PPM.

2.3.7 Tracking of sinusoidal jitter

In order to test the correct function of the CDR and the receiver we can use data with superposed sinusoidal phase jitter. The jitter tolerance simulation evaluates the system capability to track sinusoidal jitter at various frequencies and with different amplitude.

Figure 49 shows an example of data with sinusoidal jitter. The dashed line represents the nominal data transition; the distance between them is T, which is the bit time (UI). The vertical arrows are the data transitions with the sinusoidal phase jitter. The horizontal arrows follow a sinusoidal function in the time domain.

The jitter amplitude is usually expressed in UI and is the peak to peak distance between the maximum and minimum of the sinusoid.

The CDR can be able to tolerate sinusoidal jitter without error at various frequencies and amplitude, as specified by the jitter tolerance mask for the specific standard.

If the frequency of the jitter is included in the CDR bandwidth, the system is able to correct sample data despite the jitter amplitude is many UI wide. Otherwise if the jitter frequency is over the CDR bandwidth, the sampling clock does not track the jitter. In this second event the jitter tolerance value is equivalent to the horizontal eye opening.

The CDR bandwidth is mainly due to the demultiplexing factor of the input data and to the number of pipeline stages, or equivalently to the input output latency.

In order to maximize the jitter tolerance capability the proportional and integral controller gains are set with appropriate simulations.

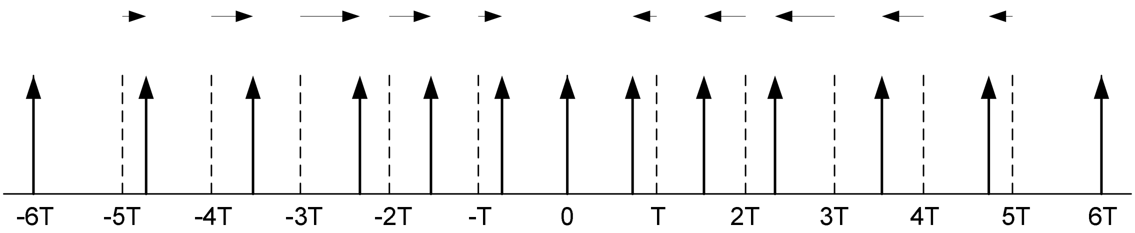


Figure 49: Sinusoidal jitter.

2.4 HORIZONTAL EYE OPENING MAXIMIZATION

2.4.1 Un-clocked DFE

In an un-clocked DFE, a delayed replica of the transmitted bit, the DFE output, multiplied by C1, is subtracted to the DFE input, that is the signal distorted ISI.

C1 value must be equal to the first post-cursor to maximize the signal to noise ratio in the sampling instant, that is the center of the data eye. The delay value (Δ) is adapted to maximize the horizontal eye opening.

Figure 50 shows the UDFE block diagram: bit16 signal selects between the two possible data rate 6 Gb/s or 12 Gb/s. If bit16 is equal to one, the data rate is 12 Gb/s and the DFE output is delayed by only one programmable delay block (PD). If bit16 is 0, the data rate is 6 Gb/s and the delay is equal to two PD.

The PD block delays the signal of a value dependent on Δ, that is the output of the digital circuit for auto-adapting the feedback delay: this value is in the range [0:22].

If Δ is equal to zero the PD delays its input of a value of 62ps; if Δ is equal to 22 the delay is equal to 106ps. The cell delay can be finely tuned with a 2ps resolution.

In the 12 Gb/s mode, the UI is equal to 83.3ps and the delay range is ±22ps. In the 6 Gb/s data rate, two PD delay the feedback signal and the delay step variation is twice the preceding one, equal to 4ps.

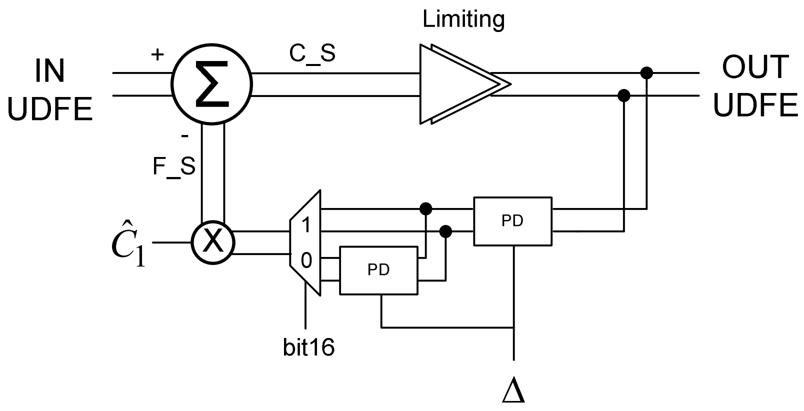


Figure 50: UDFE block diagram.

2.4.2 Algorithm works

When a data transition follows two identical bits (pattern XXY) the UDFE feedback delay does not affect the timing occurrence of the data transition at the UDFE output. On the other hand, when there are two consecutive data transitions (pattern YXY), the timing occurrence of the second transition will result advanced or delayed as a function of the UDFE feedback delay. Figure 51 shows an example with two data pattern 110 and 010.

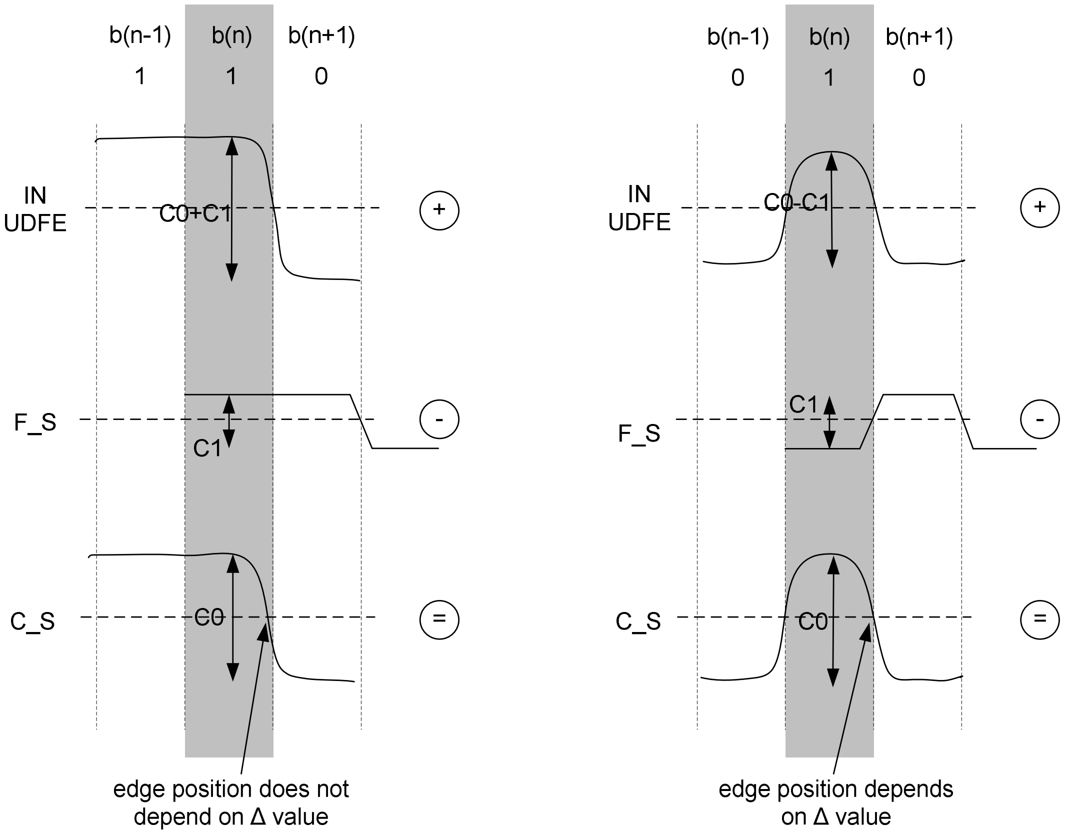


Figure 51: Delay adapting base principle.

The feedback signal (F_S) is a squared and delayed replica of the DFE output multiplied by C1. This signal is subtracted to the DFE input. It is clear that, with 110 data pattern, the delay value does not influence the edge position because the F_S signal is constant when the delay changes. With 010 data pattern, the delay value influences the edge position.

The horizontal eye opening is maximized if the transitions of the two patterns (XXY and YXY) are concentrated in an instant time if we represent them on an eye diagram. To perform this task, early late information for these two different data pattern can be used.

Figure 52 shows an example of early late information generated by 110 and 010 patterns. 110 pattern generates a late information, while 010 pattern generates an early information; in this event the delay, and consequently the Δ value, must be decreased to better concentrate the transitions of these two patterns.

The sampling clocks are in the shown positions because the CDR correctly aligns the edge sampling clock with the mean value of the data transitions and consequently the center clock is aligned with the center of the data.

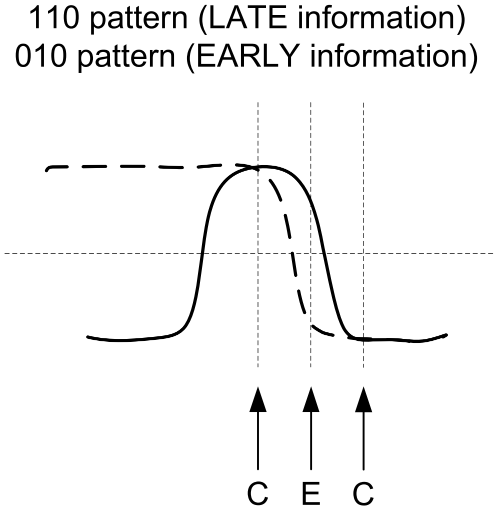

110 pattern (LATE information)
010 pattern (EARLY information)
C
E
C

Figure 52: Early late information of 110 and 010 patterns.

2.4.3 Pattern-selective early late detector

Early late information are generated in parallel, like in the CDR circuit. Moreover these phase information must be independently generated for XXY and YXY patterns. In order to perform this task an additional masking is employed.

Early-late information are computed and memorized in two vectors, in which 1 corresponds to a phase information. Simultaneously two mask vectors are generated for the two data patterns.

XXY mask vector contains 1 if the two bits used for early-late information follows two identical bits, zero otherwise. YXY mask vector is the complement of the XXY one. Figure 53 shows the data and the two clocks employed for sampling the data center and the edge in12 Gb/s modality.

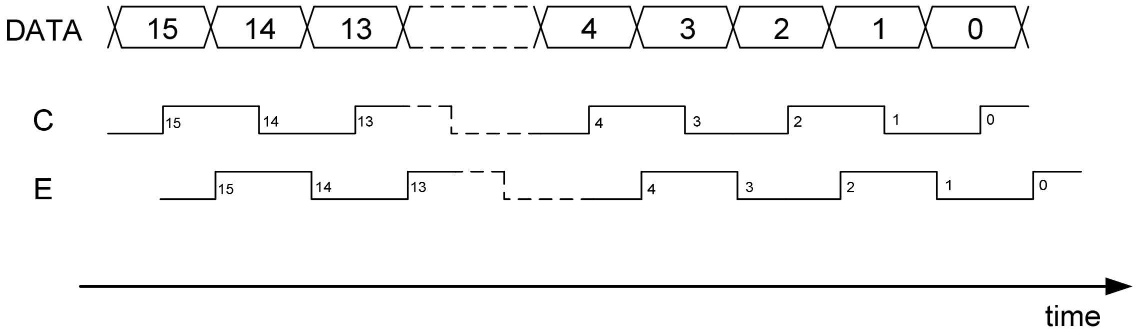


Figure 53: Data and clocks for 12 Gb/s mode.

Early late vectors for 12 Gb/s mode are computed with the following logic operation:

$$\begin{cases} clock\,EARLY\_12G<n-1>=\overline{C<n>\oplus E<n>}\quad AND\quad C<n-1>\oplus E<n> \\ clock\,LATE\_12G<n-1>=C<n>\oplus E<n>\quad AND\quad \overline{C<n-1>\oplus E<n>} \end{cases}$$

$$1\leq n\leq 14$$

Mask vectors for YYX and XYX data patterns are generated in this way:

$$\begin{cases} MASK\_XXY\_12G<n-1>=\overline{C<n+1>\oplus C<n>} \\ MASK\_YXY\_12G<n-1>=\overline{MASK\_YYX\_12G<n-1>} \end{cases}$$

$$1\leq n\leq 14$$

A similar task is performed for the 6 Gb/s data rate mode. Figure 54 shows the data and the two clocks employed for sampling the data center and the edge in 6 Gb/s mode.

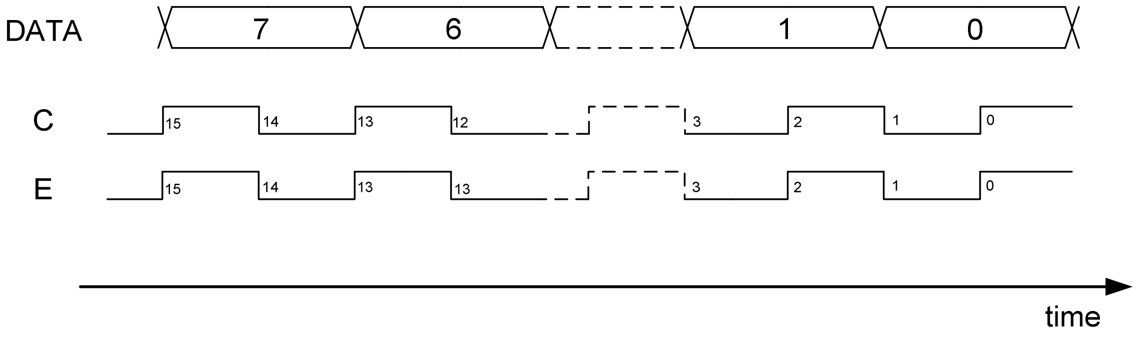


Figure 54: Data and clocks for 6 Gb/s mode.

Early late vectors for 6 Gb/s mode are computed with the following logic operation:

$$\begin{cases} clock\,EARLY\_6G<n>=\overline{C<2n+2>\oplus E<2n+1>}\ \ AND\ \ C<2n>\oplus E<2n+1> \\ clock\,LATE\_6G<n>=C<2n+2>\oplus E<2n+1>\ \ AND\ \ \overline{C<2n>\oplus E<2n+1>} \end{cases}$$

$$0\le n\le 5$$

Mask vectors for XXY and YXY data patterns are generated in this way:

$$\begin{cases} MASK\_XXY\_6G<n>=\overline{C<2n+4>\oplus C<2n+2>} \\ MASK\_YXY\_6G<n>=\overline{MASK\_YYX\_6G<n>} \end{cases}$$

$$0\le n\le 5$$

The vectors, containing the number of transitions relative to the two pattern type, are also generated. A logical OR between early and late vectors gives a new vector containing 1 if there is a transition.

First circuit outputs, “the pattern selective early late detector”, are four: two signals EPLN_XYY and EPLN_YXY represent the difference between early and late information related to the two data patterns; two signals TNUM_XXY and TNUM _YXY represent the number of transition. Figure 55 shows the generation of these signals for XXY data pattern. A logical AND is employed to mask only the XXY pattern.

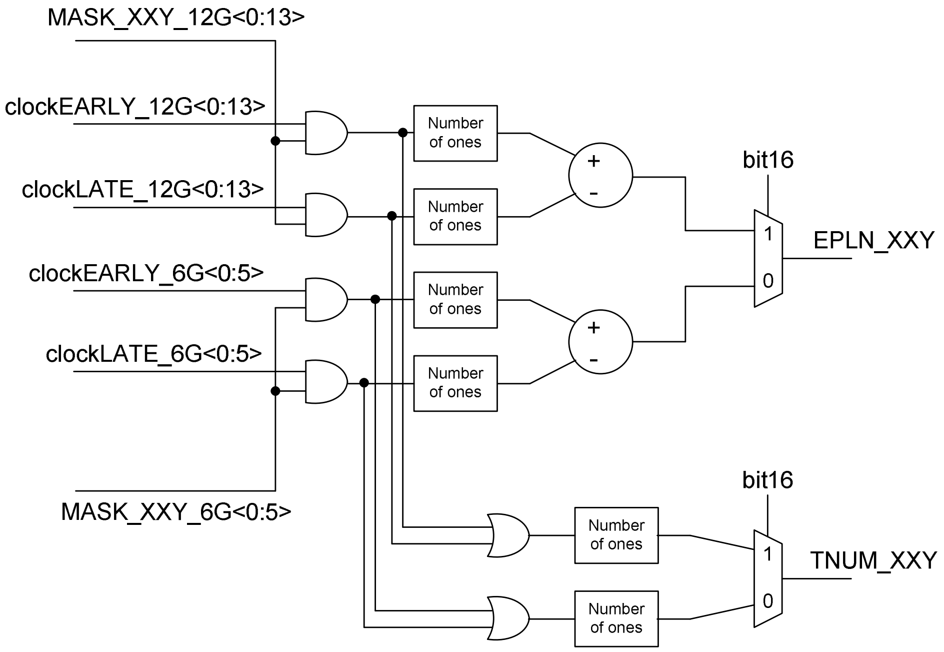


Figure 55: EPLN and TNUM generation for YYX data pattern.

The first pipeline stage encloses the EPLN and TNUM signals generation.

2.4.4 Feedback delay modification

EPLN signals of the two patterns XXY and YXY are accumulated with opposite sign into the early late accumulator (ELA). The sign of this accumulated value (SIGN_DEL) indicates if the feedback delay must be increased or decreased. This signal is used only if the UPDATE signal is set.

The number of XXY and YXY pattern occurrences (TNUM_XXY and TNUM_YXY) are counted by a second couple of counters and accumulated into the additional accumulators (XXY_A and YXY_A). The UPDATE signal is set if both the TNUM accumulators have reached a programmable threshold (ACC_TH) and contemporarily the absolute value of ELA accumulator has reached a minimum programmable value.

An “accumulators monitor” is employed to reset or stop the accumulators. If both the TNUM accumulators have reached the ACC_TH threshold, all the accumulators are reset. Moreover if one of the two patterns is less represented, the monitor temporarily stops the EPLN and TNUM accumulation of the more represented pattern. This assures the equal occurrence of the two patterns on average.

The second pipeline stage encloses SIGN_DEL and UPDATE signals generation. The third pipeline stage encloses the logic which generates the Δ signal. SIGN_DEL and UPDATE signals are first provided to a post-scaler that select, through the “DEL_postscaler” value, the number of UPDATE signal before modify Δ. The post-scaler outputs settle if Δ must be incremented, decremented or left unchanged. If

"DEL_preset" is set the algorithm stops and Δ is set equal to DEL_preset_val, Figure 57.

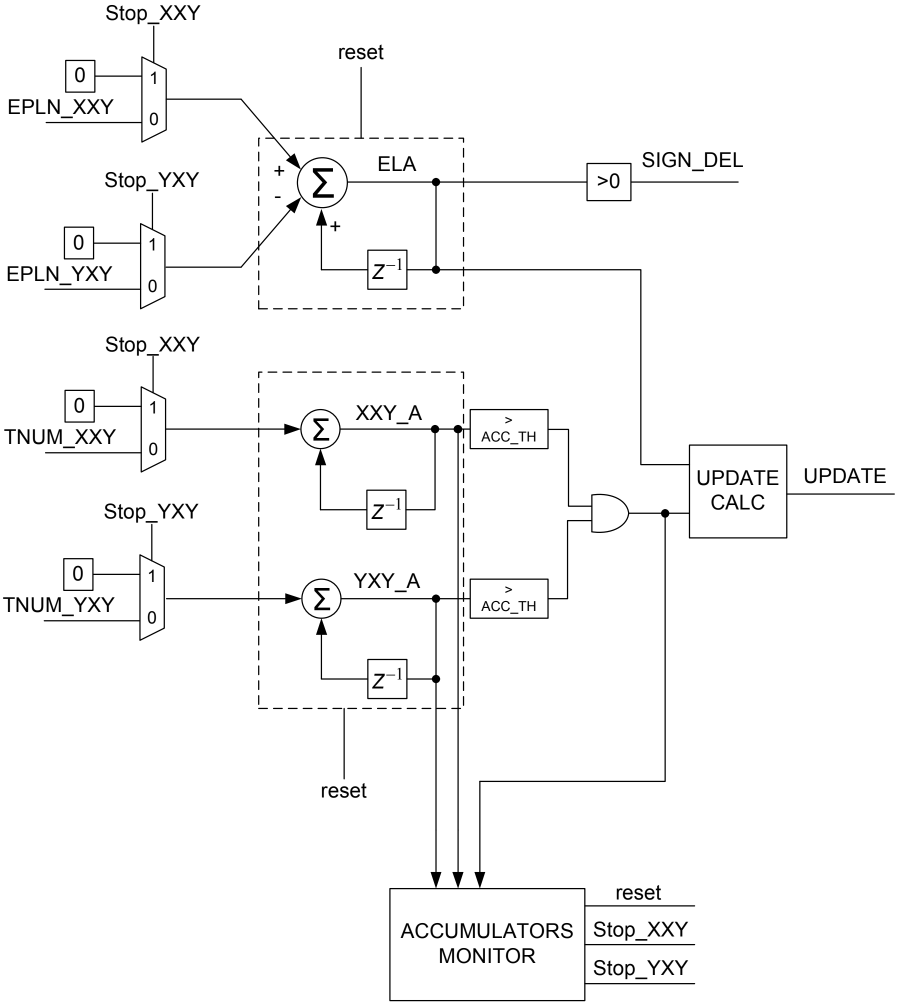


Figure 56: Delay signal update.

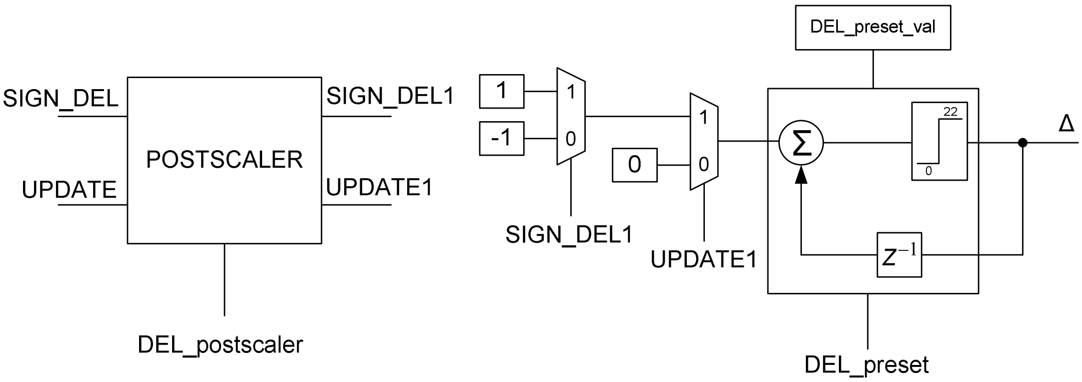


Figure 57: Δ signal generation.

2.5 VERTICAL EYE OPENING MAXIMIZATION

2.5.1 Sign-sign LMS algorithm

In order to maximize the vertical eye opening, the sign-sign LMS algorithm must correctly set the $\hat{C}_1$ value. This algorithm uses the center data and the error data sampling to auto-adapt this value. The center data sampling arise from C path, while the error data sampling (ERR) arise from A path. Figure 58 shows the samples position for the two sampling paths C and A, and the sampling values (±1).

The samples coming from C path are read in the sampling instant respect to threshold 0, A path samples are read in the sampling instant respect to thresholds ±TH. The sampling with threshold +TH is correct if the received data is positive, the sampling with threshold –TH is correct otherwise. TH threshold is the mean amplitude of the analog signal, after the UDFE, and this value is auto-adapted like $\hat{C}_1$.

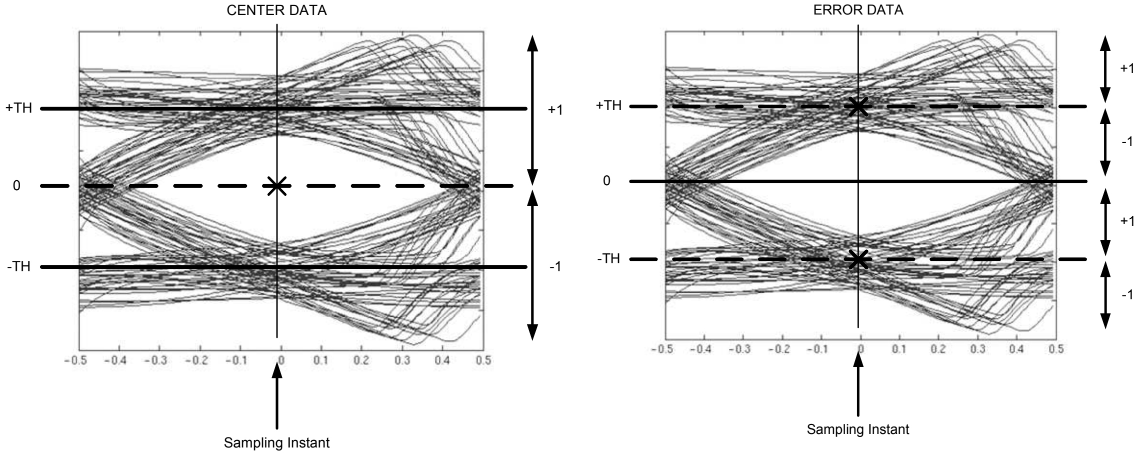


Figure 58: C and A path samples.

TH threshold is simply auto-adapt correlating the error value with the center data signal value:

- If the center data signal and the error have the same sign, TH is decreased.
- If the center data signal and the error have different sign, TH is increased.

In this way the TH value converges to the mean amplitude of the analogue signal. This value is equal to C0. In order to auto-adapt $\hat{C}_1$ the data error, computed in a given instant (n), is correlated with the preceding bit, received at the instant (n-1).

Figure 59 shows the base work of a 1 tap DFE: the goal is to remove, at the input (Vin), an amount of signal which corresponds to the ISI due to the preceding bit that is b(n-1) multiplied by $\hat{C}_1$.

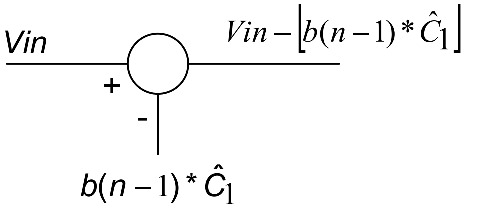


Figure 59: DFE base work.

The information related to the $\hat{C}_1$ increment or decrement is obtained correlating the error signal, computed in the time instant n, and the center data signal, computed in the time instant (n-1).

Figure 60 shows two couples of examples: one in which $\hat{C}_1$ must be increased and one in which $\hat{C}_1$ must be decreased. $\hat{C}_1$ is increased if the error and the bit b(n-1) present the same sign, otherwise is decreased.

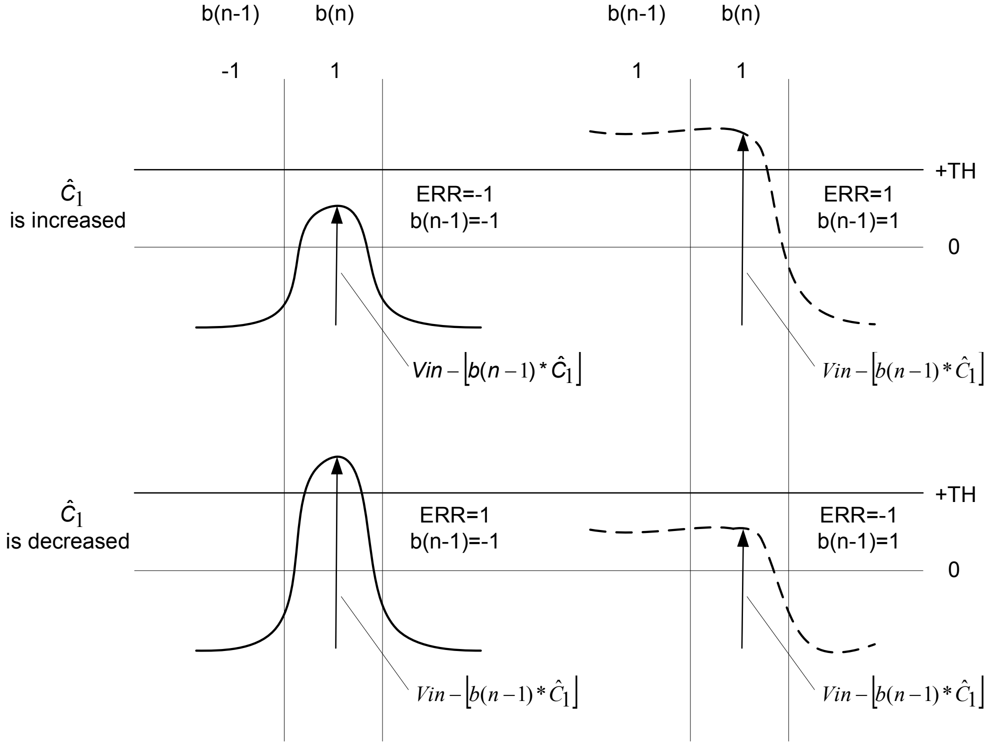

b(n-1)
b(n)
b(n-1)
b(n)
-1
1
1
1
$\hat{C}_1$ is increased
ERR=-1
b(n-1)=-1
ERR=1
b(n-1)=1
+TH
0
$Vin - \lfloor b(n-1) * \hat{C}_1 \rfloor$
$Vin - \lfloor b(n-1) * \hat{C}_1 \rfloor$
$\hat{C}_1$ is decreased
ERR=1
b(n-1)=-1
ERR=-1
b(n-1)=1
+TH
0
$Vin - \lfloor b(n-1) * \hat{C}_1 \rfloor$
$Vin - \lfloor b(n-1) * \hat{C}_1 \rfloor$

Figure 60: C1 adapting base principle.

An offset, in the input signal, can also be corrected subtracting it at DFE input. The offset value (OFF) can be auto-adapted like $\hat{C}_1$: the increment or decrement information is simply the error sign. The increment or decrement signals, related to $\hat{C}_1$, TH and OFF are integrated. The integral values related to $\hat{C}_1$ and OFF are subtracted at the DFE input: $\hat{C}_1$ value is first multiplied by b(n-1). The TH integral value is employed as TH threshold.

2.5.2 Signals correlation

The center data sampling and the error signal are provided to the digital circuit in a parallel form. The inputs are two words of 16 bits. Figure 61 shows the timing relation between the data and the clock that sample C and ERR signals relate to the 12 Gb/s mode. The figure shows also if the TH is positive or negative: One half-rate path samples the error signal respect to positive TH and the other samples the error signal respect to negative TH. Even samples are valid only if the data are positive; odd samples are valid if the data are negative.

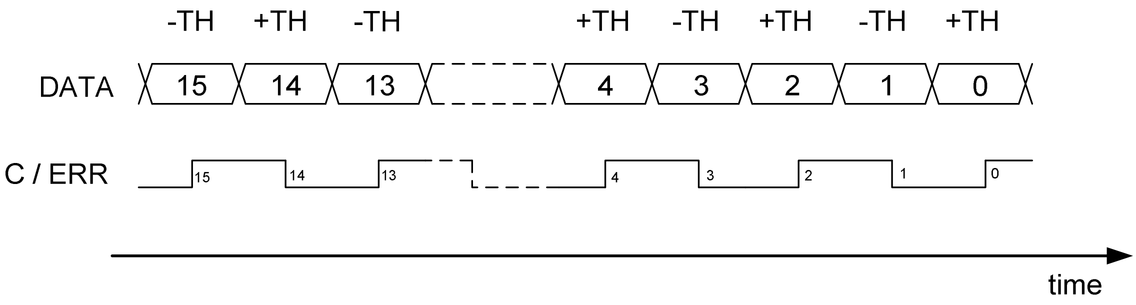


Figure 61: 12 Gb/s mode: data and clock.

Figure 62 shows the timing relation between the data and the clock that sample C and ERR signals relate to the 6 Gb/s mode.

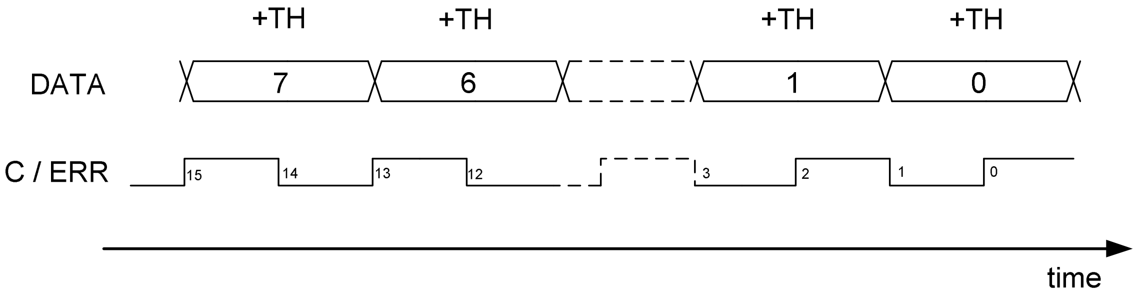


Figure 62: 6 Gb/s mode: data and clock.

Figure 63 shows the generation of the correlation signals.

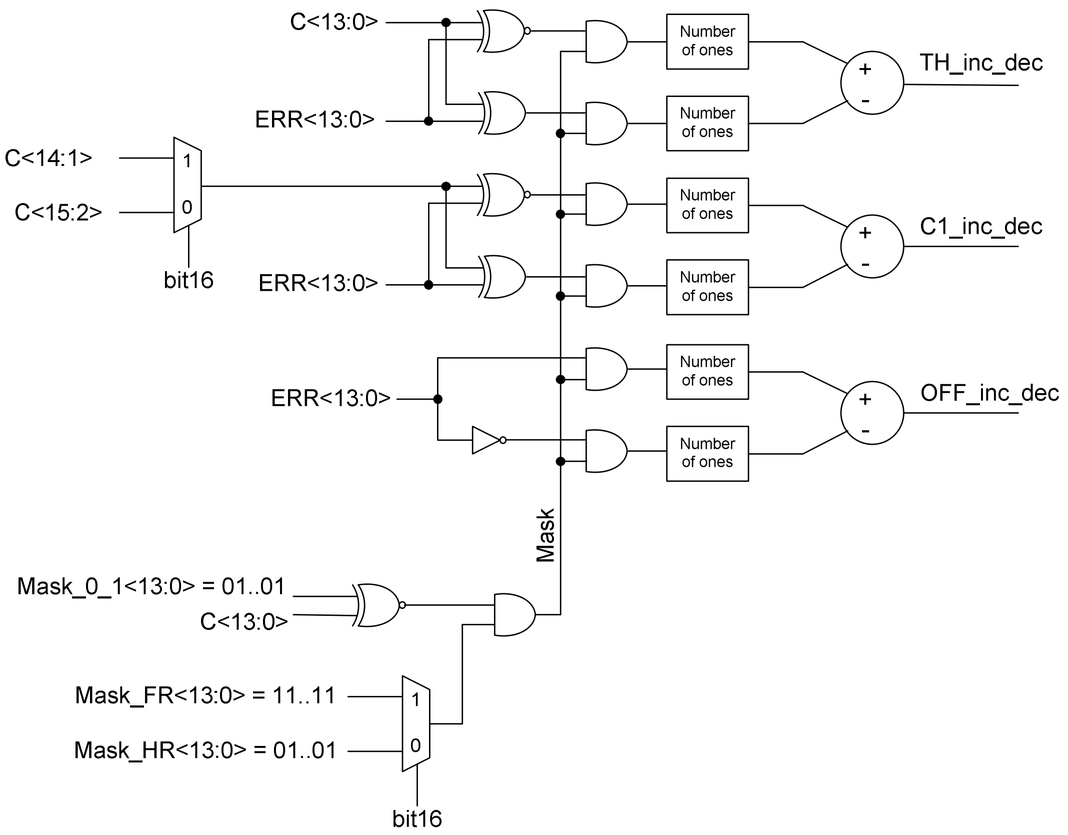


Figure 63: Signals correlation.

The circuit generates TH_inc_dec signal, related to the TH threshold variation, subtracting the number of C(n) and ERR(n) signals, with equal polarity, with the number of C(n) and ERR(n) signals with different polarity. A logical XNOR generate a 1 if the inputs are equal, while a logical XOR generate a 1 if the inputs are unequal.

C1_inc_dec signal is generated in a similar way, correlating the ERR(n) signal with the C(n-1) signal, related to the older bit. The older bit is different for the 12Gb/s and 6Gb/s modes; bit16 selects between the two modes. OFF_inc_dec signal is simply the sign of the ERR signal.

The Mask signal is employed to mask the unused or wrong signals. If we are in 6Gb/s mode, only one half-rate path, that samples the even bits, is employed; the odd bits are set to zero. Moreover, the ERR signal, related to the even bits, is valid only for positive bits: this additional masking is performed through a logical XNOR between Mask_0_1 signal and C signal.

2.5.3 Signals integration

The correlation signals must be integrated: Figure 64 shows the integration circuit related to the TH_inc_dec signal. This signal is multiplied by an appropriate gain and sent to an accumulator; the crossing of an appropriate threshold of the accumulator implies the setting of UPDATE and SIGN signals. After there is a post-scaler, that generate SIGN1 and UPDATE1. These two signals command the increment or decrement of the TH threshold. A similar circuit is employed for the integration of C1_inc_dec and OFF_inc_dec.

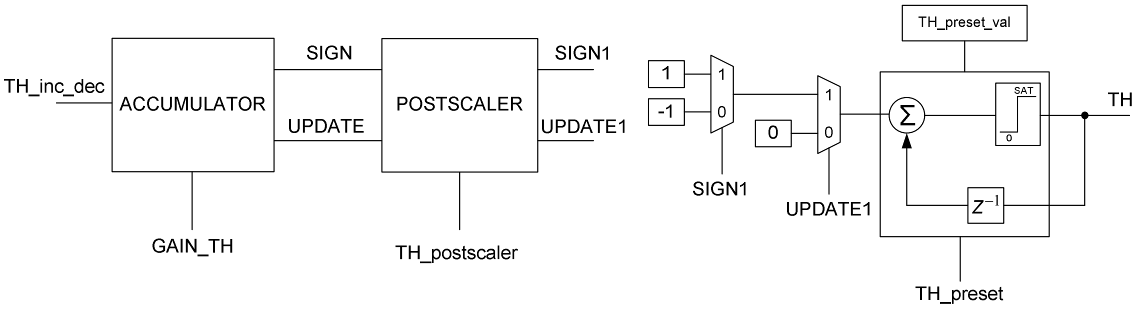


Figure 64: TH_inc_dec signal integration.

2.6 PRBS CHECKER

In order to test the correct functionality of the receiver, we use data, generated through a PRBS. These data are distorted by the transmission media, and the receiver must be able to correct recognise them. The equalized data are re-sampled and provided, in a parallel form, to the "PRBS checker" circuit that detect if there is an error.

The input is a 16 bits parallel word; these bits arise from the C path and are sampled in the center of the eye.

Figure 65 shows data and C clock related to the 12 Gb/s mode: this figure shows also the connection between data if them are generated with a PRBS7. In a PRBS a shift register generates the pattern; the input of the first flip-flop is the logical XOR between two flip-flop outputs of the shift register. In a PRBS7 the XOR is between the flip-flop outputs 4 and 7.

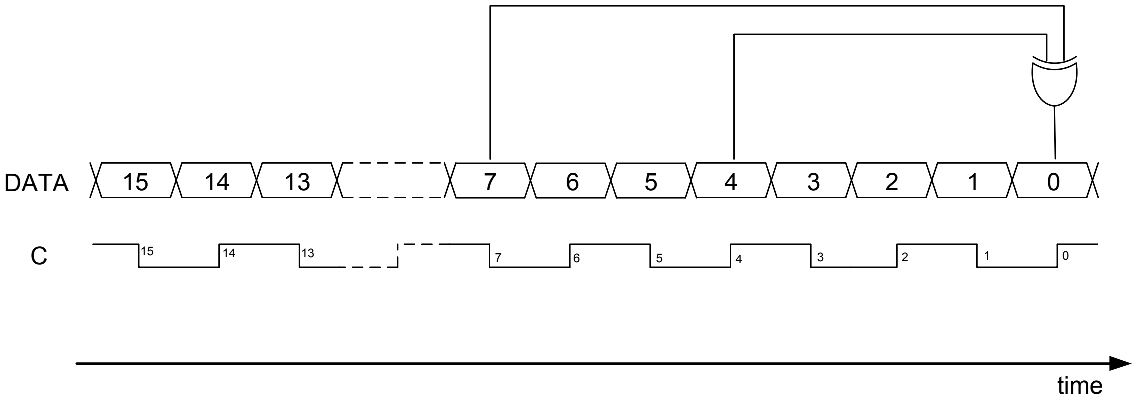


Figure 65: PRBS7 data, 12 Gb/s.

Figure 66 shows data and C clock related to the 6 Gb/s mode: this figure shows also the relation between data if them are generated with a PRBS7.

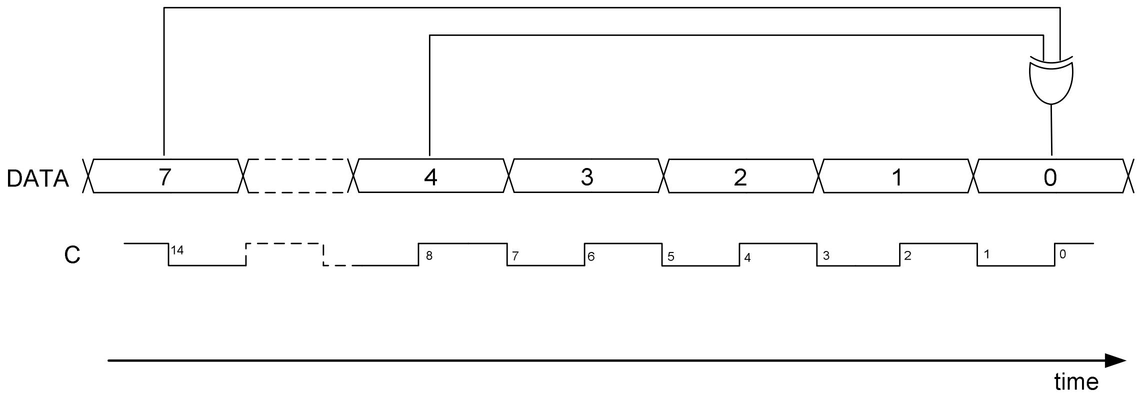


Figure 66: PRBS7 data, 6 Gb/s.

If the received data are in the following relation, there are not errors. Figure 67 shows the circuit employed to test a PRBS7 data path. The 16 bits input word is first delayed by a clock cycle to generate a word of 32 bits in parallel (C_new<31:0>). The circuit tests the data connection shown before for both 12Gb/s and 6Gb/s modes. The two signals out_4_7_12G and out_4_7_6G are vectors containing a 1 if there is an error. Bit16 select between the two data rate modes.

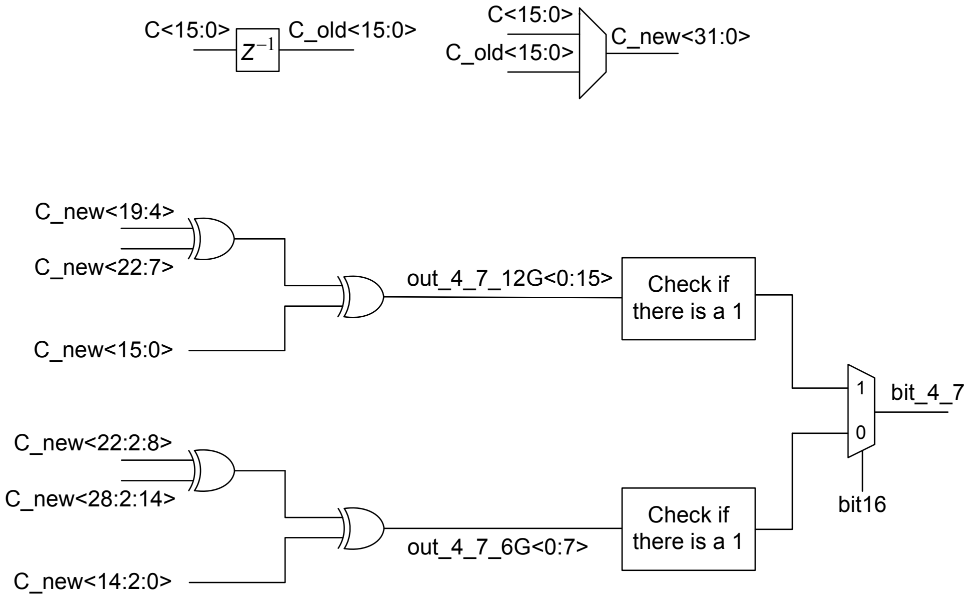


Figure 67: PRBS7 checker.

The output of the PRBS checker is a signal that assumes a high logical value is there is an error in the received data. This signal must also stays high until it is cleared by the signal Clear_PRBS_N, active low. Figure 68 shows the circuit employed.

PRBS7 is not the only bit sequence employed to test the receiver: for example PRBS11 and PRBS23 are used. In the others PRBS path, the data connection is similar to the PRBS7 one, but the logical XOR is between different bits.

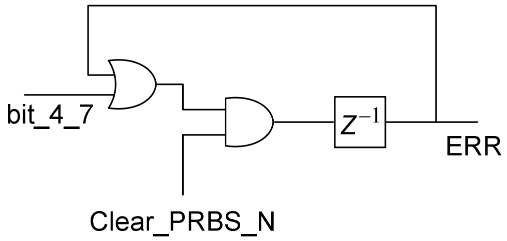


Figure 68: PRBS checker output.

2.7 INTERNAL EYE MONITOR

The internal eye monitor algorithm employs data sampled by path C and path A. Path C samples data in the center data eye and path A samples data in various positions.

To compute the horizontal eye opening, the phase of clock A is first set equal to the clock C one and TH threshold is set to zero: the two path read the same data value. The phase of clock A is so decremented to compute the left horizontal eye opening: when the data sampled by C path differs from the A path one, the eye is closed. To compute the right horizontal eye opening, the phase of clock A is incremented.

We find the vertical eye opening in a similar way: the phase of clock A is equal to the clock C one and the value of TH is changed. When the C and A samples differ the eye is closed, Figure 69.

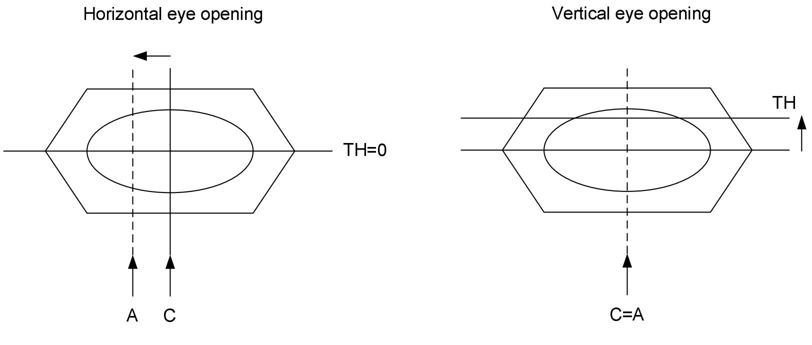


Figure 69: Internal eye monitor.

2.8 DIE MICROGRAPH

Figure 70 shows the die micrograph of the receiver. We have shown: the Rx Core, composed by the PGA, the LE, the UDFE, the demuxes and the PI; the generation of the clock used by the phase interpolators (I/Q gen) and the buffers employed to feeds the LE and first flip-flop (the UDFE) outputs to pins for measurements. The digital part is not included in the photograph because it is hidden by a big filter capacitance.

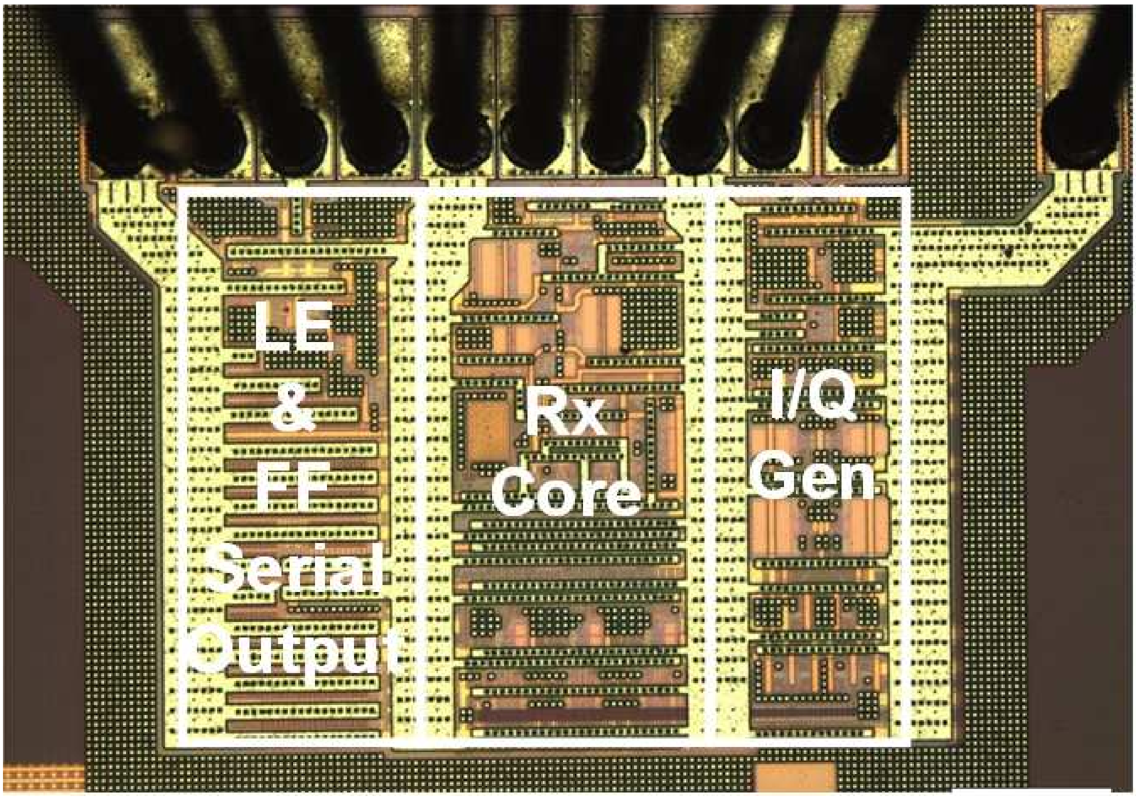


Figure 70: Die micrograph.

The receiver's core, realized in 45nm CMOS, occupies 580×170μm^2 and consumes 130mW from a 1.1V supply at 12Gb/s. The digital part occupies 290×170 μm^2 and consumes 10mW.

Figure 71 shows the power consumption distribution.

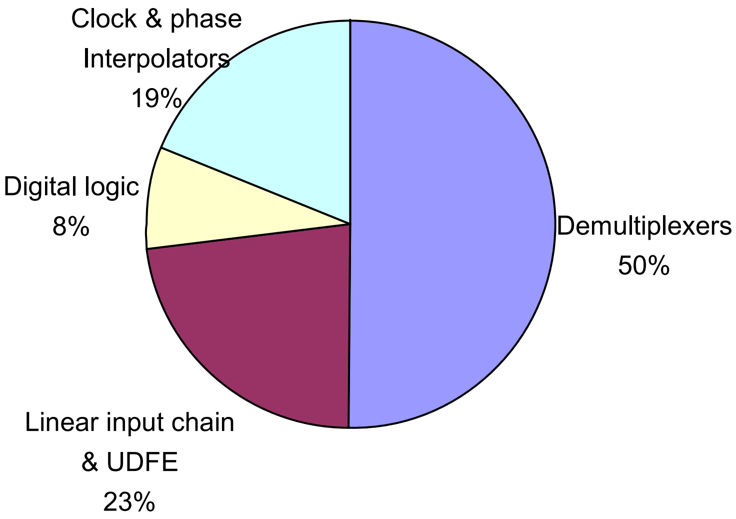


Figure 71: Power consumption distribution.

2.9 MEASUREMENTS RESULTS

2.9.1 Measurements set-up

The block diagram of the measurements set-up [18] is shown in Figure 72. A Pulse/pattern generator instrument (Jbert) generates a 12 Gb/s data stream, coming from a PRBS7; this data stream is sent to a backplane and the distorted signal is sent to the receiver. A personal computer (PC) is employed to set the receiver programmable variables. The variables related to the digital parts are described in the previous paragraphs.

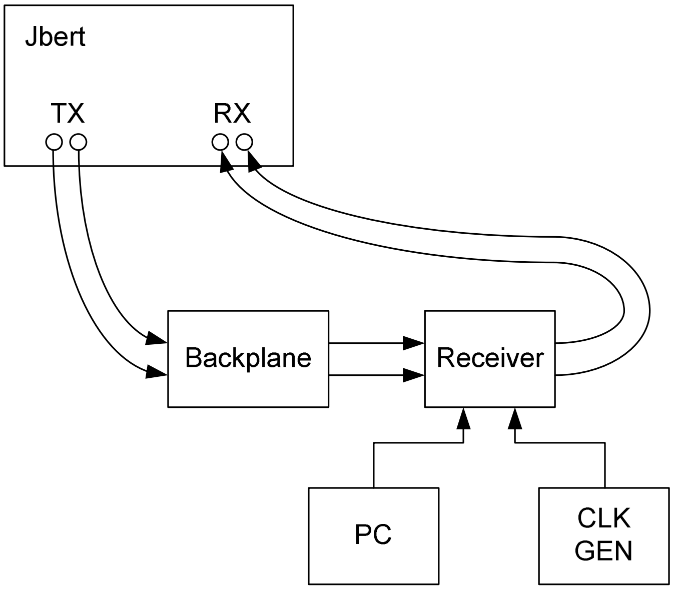


Figure 72: Block-diagram of the measurements set-up

In order to check the correctness of the received sampled data, the internal PRBS checker can be used. This checker tests the received bit after the UDFE. Alternatively, the received serial data after the LE or the UDFE, are sent to the Jbert serially for testing them correctness. The Jbert incorporates a PRBS checker. We can select, through a configuration bit, if the receiver output is after the LE or after the UDFE.

The receiver is characterized through a jitter tolerance measure. Sinusoidal jitter, at different frequency, is added to the transmitted data. The amplitude of the jitter is increased until the PRBS checker detects an error.

Depending on the required BER, a different number of bits are sent. For example, if the required BER is 10^{-12}, 10^{12} bits must be correctly recognized by the receiver.

2.9.2 Jitter tolerance (DFE on and off)

A 40” BERT backplane [19], with 18 dB loss at the Nyquist frequency, is used to test the jitter tolerance, with a BER<$10^{-12,}$ with and without the UDFE. Before the jitter tolerance measurement, the auto-adaptive DFE coefficients, C1 and the delay value, correctly set to maximize the eye opening.

Figure 73 shows the measured jitter tolerance with data-rate of 12 Gb/s. The figure shows two plots: with and without the UDFE. The UDFE increase of 0.1 UI the high frequency jitter tolerance even if the transmission medium is a low loss channel.

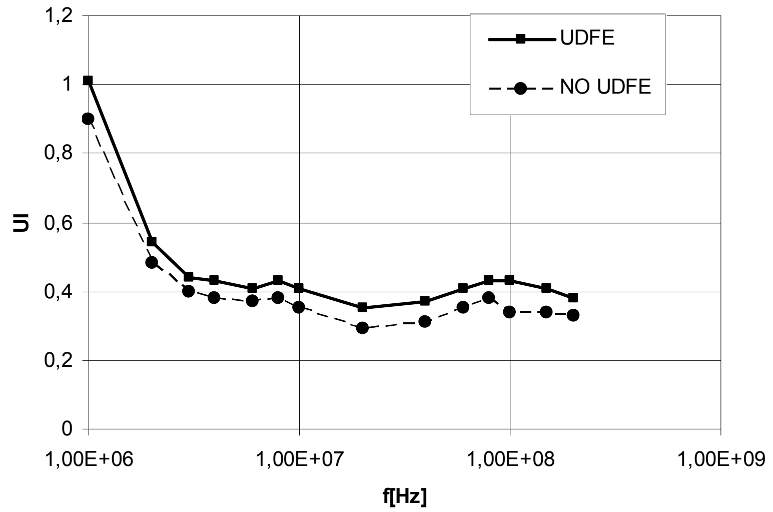


Figure 73: Jitter tolerance with and without UDFE.

2.9.3 Jitter tolerance (worst case)

Figure 74 shows the transfer function of one of the worst BERT backplanes, employed for testing the receiver. This channel has 36 dB of loss at the Nyquist frequency (6GHz).

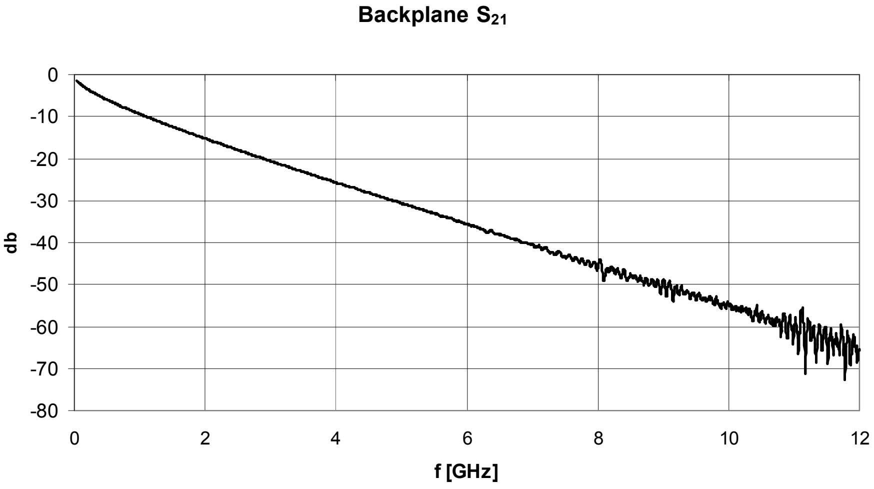


Figure 74: Backplane transfer function.

Figure 75 shows the 12 Gb/s jitter tolerance. Transmitting data are generated through a PRBS7 and there is 50 PPM of difference between receiver clock and data frequency. The receiver tolerate more than 0.27 UI additional jitter with a BER<10^{-12}.

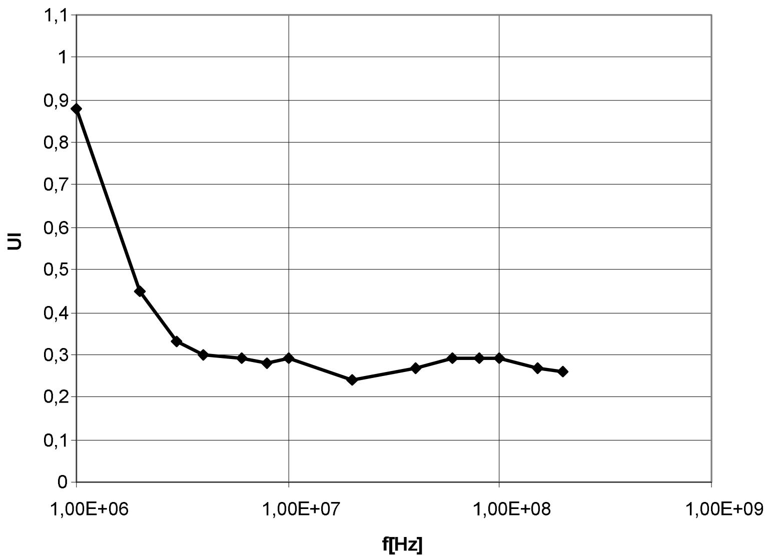


Figure 75: Sinusoidal jitter tolerance.

2.9.4 Jitter tolerance (high frequency)

Figure 76 shows the measured high frequency jitter tolerance versus backplane losses. 30.6 dB and 39.3 dB are the losses of a Tyco 30'' backplane at 10Gb/s and 12Gb/s; the others channel losses refers to BERT backplane.

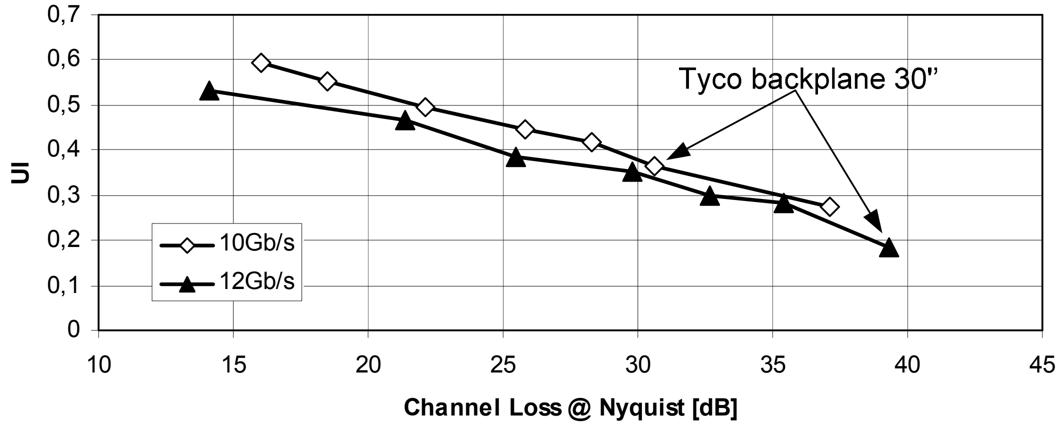


Figure 76: High frequency sinusoidal jitter tolerance vs. channel loss.

The jitter frequency is 200MHz. This value is out of the CDR bandwidth and so the CDR is un-able to tracks the data. For this reason the sampling clock moves on two values. The high frequency jitter tolerance is an indirect measure of the horizontal eye opening.

2.10 LITERATURE COMPARISON

We show a comparison between this work and the previously and contemporary published works. We consider four aspects to realize this comparison: operating frequency, power consumption, recovered channel lost and horizontal eye opening measured through a high frequency jitter tolerance.

We summarize in Table 1 the performances of the first in class serial interface receivers, published in the last three year at ISSCC.

	Operating Frequency	Rx power	Recovered Loss
Fukuda et al. ISSCC 08 [20]	8 Gb/s	131 mW	36.8dB* @ 8Gb/s (0.11UI eye opening) *RX+TX ffe
Bulzacchelli et al. ISSCC 09 [21]	11.1 Gb/s	78 mW	28dB* @ 10Gb/s (0.19UI eye opening) *RX+TX ffe
Sugita et al. ISSCC 10 [22]	16 Gb/s	69 mW	22dB* @ 16Gb/s (0.3UI eye opening) *RX only
This work ISSCC 10 [23]	12 Gb/s	130 mW	36dB* @ 12Gb/s (0.27UI eye opening) 39dB* @ 12Gb/s (0.19UI eye opening) *RX only

Table 1: Comparison with the state of the art.

2.11 SUMMARY

In this chapter we have described the digital circuits of a 12Gb/s analog receiver. A bang-bang CDR, the circuits necessary to auto-adapt the continuous time DFE and some auxiliary circuits. After that we have shown the measurements results and a literature comparison. Measured results prove the device capability of 12Gb/s error-free operation with 39dB backplane losses.
The next chapter deals with a 6 bit, 5 GS/s flash ADC.

Chapter 3

DESIGN OF A 6BIT, 5GS/S FLASH ADC

3.1 INTRODUCTION

In this chapter we describe the designed 6 bit 5GS/s flash ADC. This ADC is intended to be employed as the front end of a digital receiver.

The chapter starts with a brief introduction about ADCs and their main metrics. After, we study the circuit non-idealities in order to define the specifications for each ADC block. Then we report a comparison between two flash ADC architectures and we select the better one in terms of power consumption. After that we describe the designed digital offset-calibration algorithm and the digital encoder. Finally we present the simulations results and a literature comparison.

3.2 ADC BASICS

3.2.1 Basic operation

An ADC converts a continuous-amplitude continuous-time input to a discrete-amplitude discrete-time signal. Usually, an analog low pass filter first limits the input signal bandwidth, so that subsequent sampling does not alias any unwanted noise or signal component into the actual signal band. Next the filter output is sampled so as to produce a discrete-time signal. The amplitude of this waveform is than “quantized”, and approximated with a level from a set of fixed reference, thus generating a discrete-amplitude signal. Finally a digital representation of that level is established at the output [24], Figure 77.

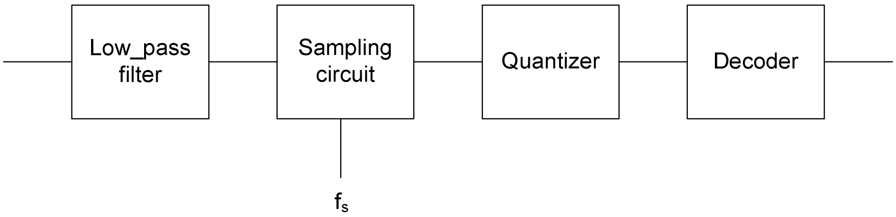


Figure 77: Analog to digital interface.

The ratio of the sampling rate f_s to the signal bandwidth distinguishes two classes of A/D converters. In “Nyquist rate” ADCs, the sampling frequency is slightly higher than twice the analog signal bandwidth to allow accurate reproduction of the original data. In “oversamplig” converter, on the other hand, the signal is sampled at many times the Nyquist rate and subsequent digital filtering is utilized to remove the noise outside the signal bandwidth. These two classes require vastly different architectures and design techniques. In this thesis, we do not consider the “oversamplig” converters.

3.2.2 Sampling

A sampler transforms a continuous-time signal into its sampled-data equivalent. Ideally, a sampler yields a sequence of delta functions whose amplitude equals the signal at the sampling times. For uniform sampling with period Ts the output of a sampler is given by:

3.1 $$\hat{X}(nT_S) = X(t) \cdot \sum \delta(t - nT_S)$$

Figure 78 shows the waveform of a continuous-time signal and the resulting sampled-data signal. Multipling a signal in the time domain by a sequence of delta, spaced of Ts=1/fs, is equivalent, in the frequency domain, to convolving the signal spectrum with a sequence of delta, spaced of fs, Figure 79. The sampling frequency (fs) must be at least twice the signal bandwidth (B) to prevent aliasing in the spectrum of the sampled signal.

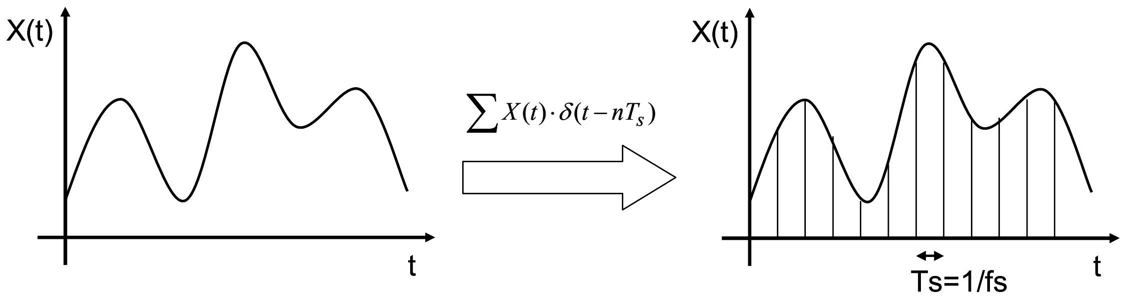


Figure 78: Sampled signal.

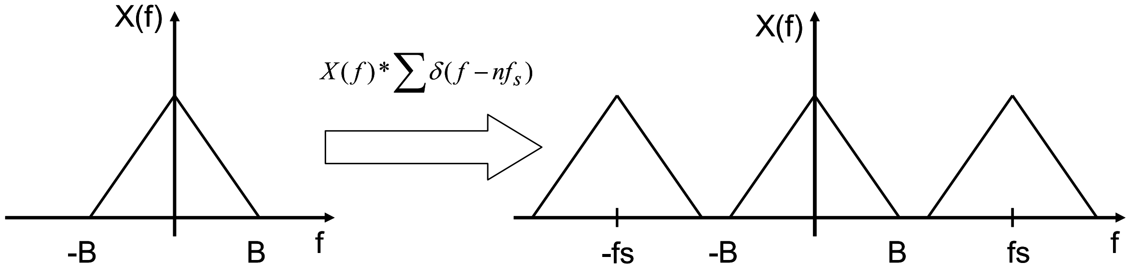


Figure 79: Spectrum of a sampled signal.

3.2.3 Amplitude quantization and ADC specifications

Amplitude quantization changes a sampled-data signal from continuous level to discrete-level. The dynamic range of the quantizer is divided into a number of equal quantization intervals, each of which is represented by a given analog amplitude. The quantizer modifies the input amplitude into a value that represents which quantization interval it resides in. Often the value representing a quantization interval is the mid-point of the interval [25].

If A is the input signal range and N_bit is the ADC number of bit, the quantization step, called also LSB, is equal to $A/2^{N\_bit}$. Figure 80 shows the input-output transfer function of a 3-bit ADC and its quantization error (the difference between an ideal transfer function and the ADC transfer function).

We can now calculate the signal to noise and distortion ratio (SNDR) of the ADC output in function of its quantization error, with a sinusoidal wave input signal.

3.2 $$SNDR = 10\log\left(\frac{P_{\sin}}{P_{Eq}}\right)$$

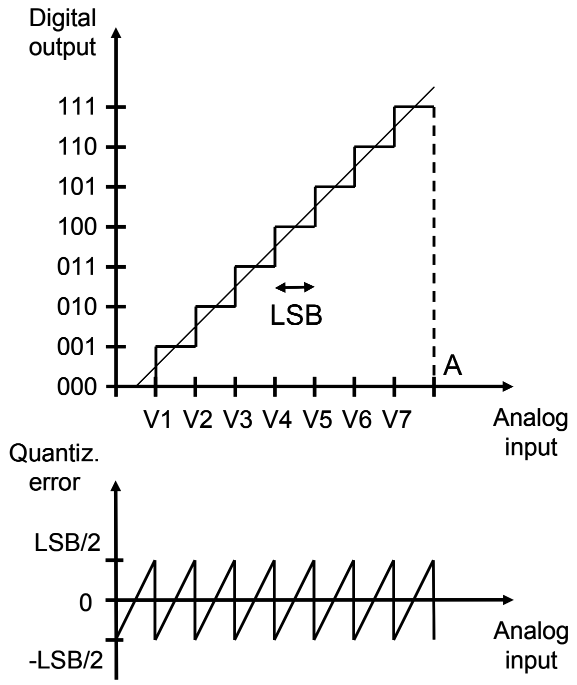


Figure 80: 3 bit ADC transfer function and quantization error.

Estimating the time average power of the quantization error assumes a constant probability distribution function p(Eq) in the range [−Δ/2 : +Δ/2]; and that outside this range p(Eq) is zero. Since the integral of the probability distribution function over the infinite range [−∞ : +∞] is equal to one, it results:

$$\textbf{3.3} \quad \begin{cases} p(Eq) = \dfrac{1}{LSB} \quad \text{for } Eq \in \left[-\dfrac{LSB}{2} : \dfrac{LSB}{2}\right] \\ p(Eq) = 0 \quad \text{otherwise} \end{cases}$$

The time average power of p(Eq) is given by

$$\textbf{3.4} \quad p(Eq) = \int_{-\infty}^{\infty} Eq^2 \cdot p(Eq)\partial Eq = \int_{-\frac{LSB}{2}}^{\frac{LSB}{2}} \frac{Eq^2}{LSB}\partial Eq = \frac{LSB^2}{12}$$

The average power of a sine wave with amplitude equal to the ADC dynamic range (A) is:

$$\textbf{3.5} \quad P_{\sin} = \frac{1}{T}\int_0^T \frac{A^2}{4}\sin^2(2\pi ft)\partial t = \frac{A^2}{8} = \frac{(LSB \cdot 2^{N-bit})^2}{8}$$

The signal SNDR, defined below, is:

3.6 $SNDR = 10\log\left(\frac{3}{2}\cdot 2^{N\_bit\cdot 2}\right) = (6.02\cdot N\_bit + 1.76)_{dB}$

This formula relates the ADC number of bit with the ADC output SNDR. The SNDR is computed through a fast Fourier transforms (FFT) of the sinusoidal converted signal and the equivalent number of bit (ENOB) is given by the following formula:

3.7 $ENOB = \frac{SNDR - 1.76}{6.02}$

There is another ADC specification that is important to know: the spurious free dynamic range (SFDR) that is the ratio of the root-mean square signal amplitude to the root-mean-square value of the highest spurious spectral component.

3.2.4 Differential and integral non linearity (DNL & INL)

The input-output transfer characteristic depicts the static behavior of a data converter. For an ideal case the input-output characteristic is a staircase with uniform steps over the entire dynamic range [25]. Deviations from the ideal transfer characteristic produce results like the ones shown in Figure 81.

In order to evaluate the non-linearity, there are two metrics: the differential non linearity (DNL) and the integral non linearity (INL). Both are defined for each digital output code.

DNL is the deviation of the step size of a real data converter (LSB') from the ideal width of the bins (LSB), divided by LSB. INL is a measure of the deviation of the transfer function from the ideal interpolating line and can be obtained from the cumulative sum (integral) of the DNL. In other worlds INL is the difference between the ideal and real transition points between successive codes.

We use a histogram or code density test to compute the DNL and INL. A large number of output samples are collected and their frequency of occurrence is plotted as a histogram versus possible output code.

For example, if an ideal, full-scale ramp is applied to an ideal ADC, the histogram consists of equal-sized bins for all output codes because the input distribution is uniform and the ADC exhibits equal probability of generating any of the codes. On the other hand, if the ADC is not ideal, the bins may not have equal height because some codes occur more often than others.

From this histogram we can compute DNL and consequently INL. DNL is the difference between the real occurrence of each output codes and the ideal occurrence divided by the ideal occurrence. This test reveal also offset and gain error [26].

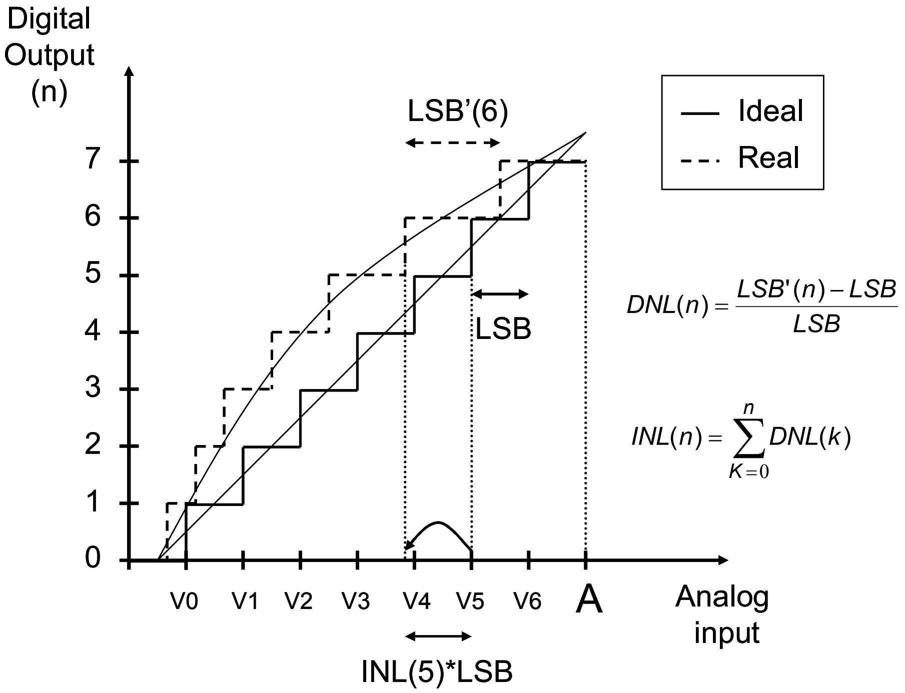


Figure 81: DNL and INL.

3.3 FLASH ARCHITECTURE & CIRCUITS NON-IDEALITIES

3.3.1 Flash architecture

The goal of this work is to design a full-flash ADC converter in a 32nm tecnology with a resolution of 6 bits and a sampling rate of 5GS/s with the minimal input-output latency. The digital ADC output must be in two's complement.

In a flash ADC converter, with a resolution of N_bit, 2^{N_bit} -1 reference voltages and comparator stages (called slices) are usually employed to convert the analog input signal into a thermometer digital output code that can be converted into a binary representation.

In order to define the thresholds position it's useful to consider a flash ADC with a resolution of only 4 bits. In a 4 bit ADC the quantization step (LSB) has an amplitude of 2A/ 2^{N_bit} = 2A/16, where [-A:A] is the input signal dynamic range and there are 15 slices to detect the crossing of the same number of thresholds, Figure 82.

If we want a symmetrical transfer function there isn't a digital representation of the zero input signal. The digital output presents an offset of LSB/2 respect to the input signal.

If we are interested in a two's complement digital output representation with the zero digital output code, corresponding to the zero input signal, the thresholds positions must be the one shown in Figure 83. If we want a symmetric output between -7 and 7, the ADC uses one less threshold.

Actually the input signal offset usually is not corrected with a resolution better than LSB/2 and so the two representations are roughly equivalent.

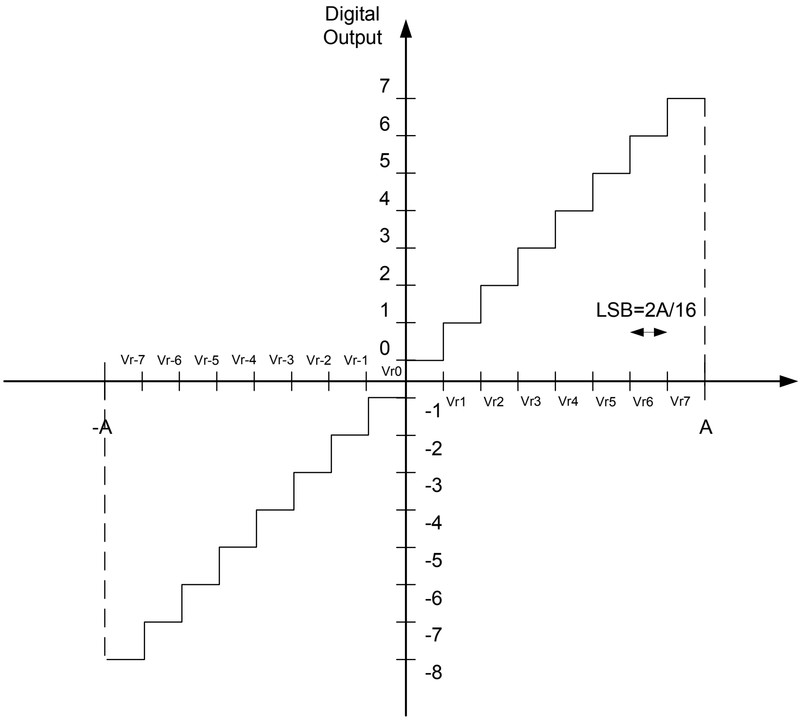


Figure 82: Input-output transfer function without the representation of the zero input signal.

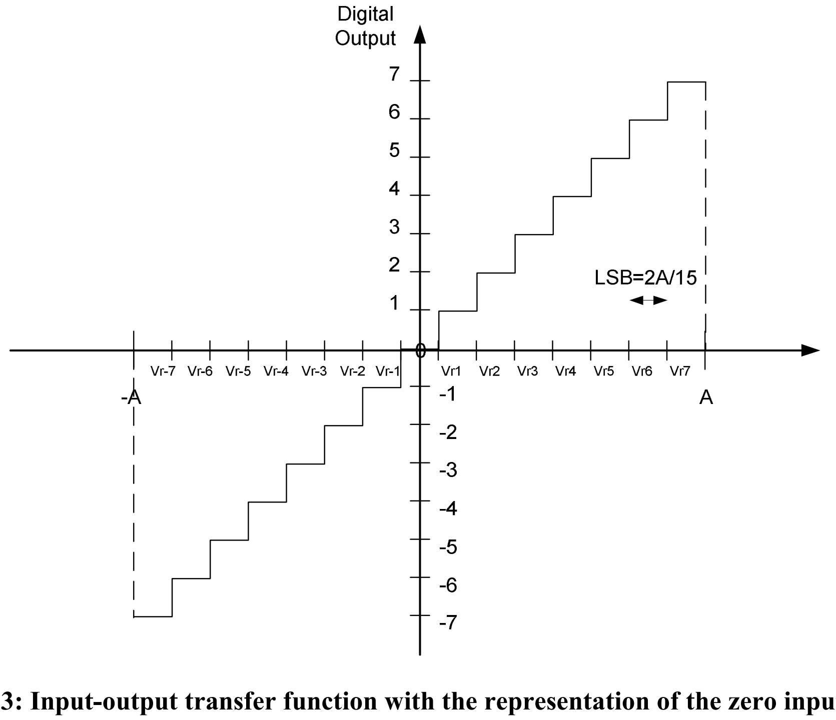


Figure 83: Input-output transfer function with the representation of the zero input signal.

The designed flash ADC has 62 reference voltages and comparator stages and the output is in two's complement between [-31:31]. The differential input signal has a peak to peak differential amplitude of 500mV; the LSB amplitude is approximately 8mV.

This signal is first provided to a buffer and track&hold: the buffer decouples the input from the ADC input capacitance, coming from all the slices; the track&hold circuit samples the input signal and reduces the problem of the skew in the comparators clocks. The thermometric code is binary converted by a digital encoder that provides a two's complement binary output in the range [-31:31].

In order to compensate the offset in the slices, an offset calibration procedure has been implemented. The offset calibration algorithm works in the following way: instead of having 62 slices there are 64 slices, two of which are simultaneously offset calibrated through an auto-zero procedure; the slices in calibration are periodically chosen among the 64 slices. In this way the offset calibration is running while ADC converts the input signal and it is possible to remain offset calibrated even if there is offset drift due, for example, to temperature variation.

Figure 84 shows the designed flash ADC architecture. The dashed lines, originating from the offset calibration block, means that the algorithm selects both the slices in calibration and the ADC outputs employed for the conversion.

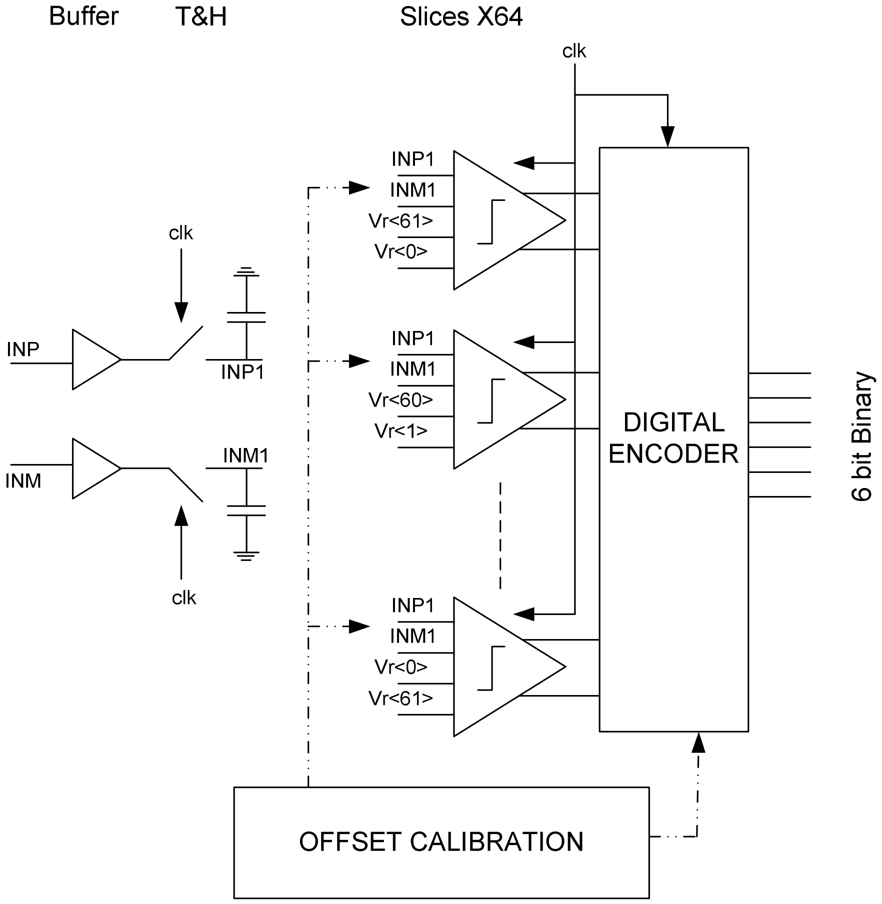


Figure 84: Flash ADC block diagram.

3.3.2 Buffer and Track & Hold attenuation and non-linearity

Buffer and track&hold attenuation and non-linearity are the first source of non-ideality in the ADC input-output transfer function and so they are the first source of ENOB reduction.

The attenuation can be evaluated as dB loss at the Nyquist frequency (2.5GHz).

In order to evaluate the linearity requirement we calculate the total SNDR due to the quantization noise of an ideal 6 bit ADC, with full-scale input signal, and the harmonics power.

3.8 $$SNDR\big|_{dB} = 10 \cdot \log\left(\frac{<S^2>}{<E^2>+<M^2>}\right)$$

$<S^2>$ is the signal power, $<E^2>$ is the quantization noise power of a 6 bit ADC and $<M^2>$ is the harmonics power.

Moreover total SNDR can be rewritten in function of THD and SNRQ (signal to noise ratio due to the quantization error).

3.9 $$SNDR\big|_{dB} = 10 \cdot \log\left(\frac{1}{\left(\frac{<S^2>}{<E^2>}\right)^{-1} + \left(\frac{<S^2>}{<M^2>}\right)^{-1}}\right) = 10 \cdot \log\left(\frac{1}{\left(10^{\frac{SNRQ}{10}}\right)^{-1} + \left(10^{\frac{THD}{10}}\right)^{-1}}\right)$$

The formula that relates the SNDR to the equivalent number of bit is:

3.10 $$SNDR = 6.02 * N\_bit + 1.76dB$$

Plotting the overall ENOB reduction, respect to the nominal 6 bits, as function of the buffer and track&hold THD, Figure 85, we can see that a better linearity than 50 dB is not required if we are interested to design a 6 bit ADC. With a 50dB THD the ENOB reduction is less than 0.1 bit.

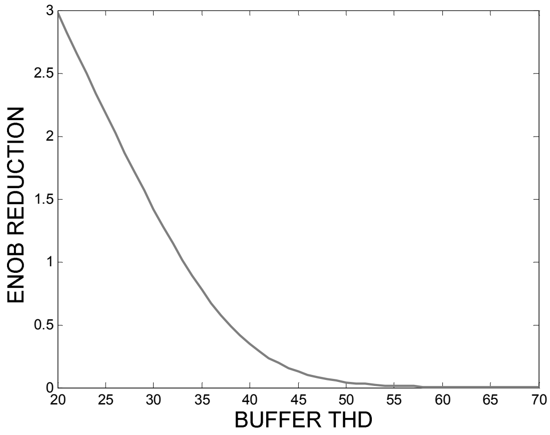


Figure 85: ENOB reduction in function of different buffer THD.

3.3.3 Thermal noise

We have to take into account two thermal noise sources: the Track & Hold kT/C noise and the slice input-referred noise. In first approximation, we can sum these two sources with the quantization error power:

3.11 $$SNDR = 10 \cdot \log\left(\frac{<S^2>}{<E^2> + <T^2>}\right)$$

$<S^2>$ is the signal power, $<E^2>$ is the quantization noise power of a 6 bit ADC and $<T^2>$ is the power of the thermal noise.

The kT/C noise is due to the total input capacitance (sampling capacitance plus the ADC input capacitance). We can vary the input capacitance calculating the ENOB reduction respect to the nominal 6 bits with an LSB equal to 8mV, Figure 86.

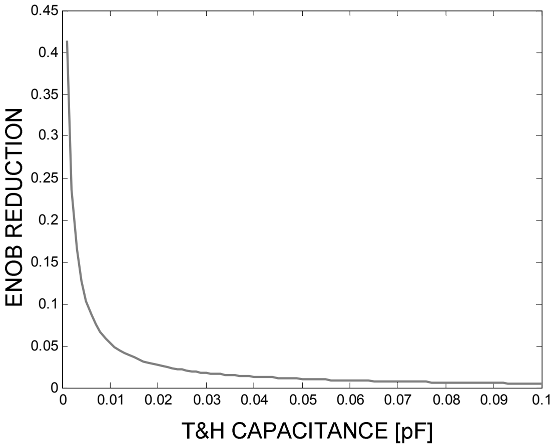


Figure 86: ENOB reduction in function of different input capacitance.

The total input capacitance must be grater then 0.05pF in order to avoid substantial ENOB reduction.

Analog considerations can be made considering the slice input-referred noise. Figure 87 shows the ENOB reduction as a function of the standard deviation values of slice input - referred noises, expressed in LSB.

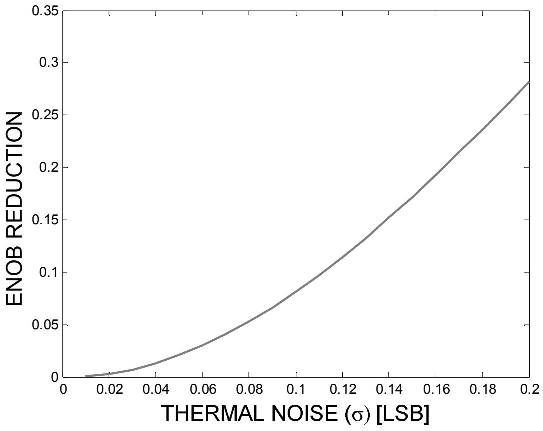


Figure 87: ENOB reduction in function of different noise values.

A noise standard deviation of 0.1 quantization step doesn’t introduce substantial ENOB reduction. The two noise sources are uncorrelated and so can be summed quadratically.

The slice input-referred noise can be approximately evaluated in this way: a DC voltage generator of Vx value is applied at the slice input and a high number N of conversion cycles are simulated with a “noise tran” simulation (a transient simulation where the circuit thermal noise is also taken into account).

Figure 88 shows this situation: the x axis represents the input of the slice, Vx is the DC voltage and the thermal noise PDF is centred around Vx. Without noise the slice output is always of the same polarity (+1) but, due to the thermal noise, some sampled values will have the wrong polarity (-1).

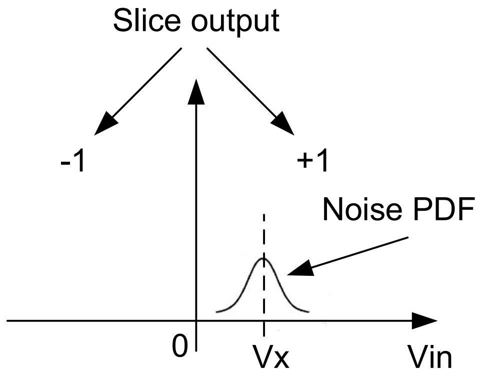


Figure 88: Slice input-referred noise evaluation.

We can use the number of samples with wrong polarity to evaluate the approximate noise standard deviation when its PDF is of Gaussian type.

We call N_sample the number of conversion cycle and N_error the number of sample with a wrong polarity. N_error/N_sample is the error probability with a given Vx value. We can find the correspondence, between Vx and the standard deviation of a Gaussian distribution, through the Q function.

For example if the DC voltage Vx is equal to 1mV, the number of simulated conversion cycles N is 1000 and there are 160 samples with a wrong polarity, the noise standard deviation is, in first approximation, 1mV.

3.3.4 Slice resolution

Another source of non-ideality in a flash ADC is the finite resolution of the slices. A comparator stage must be able to detect the sign of the difference between the differential input signal (VINP-VINN) and the differential thresholds (VRP-VRN).

If the differential input signal overcomes the considered differential threshold by the resolution, the corresponding slice is able to detect the correct sign; otherwise the sign remains the preceding sampled one. This behavior results in a hysteresis in the input output transfer function. If the slice resolution is not negligible respected to the ADC quantization step, there is ENOB reduction due to odd harmonics.

In order to quantify this problem a Matlab model has been developed that takes into account this hysteresis. Figure 89 shows the ENOB reduction as a function of the slice resolution with a sinusoidal Nyquist frequency input signal (2.5GHz).

When the slice resolution is lower than 0.1-0.2 quantization steps the ENOB reduction is negligible.

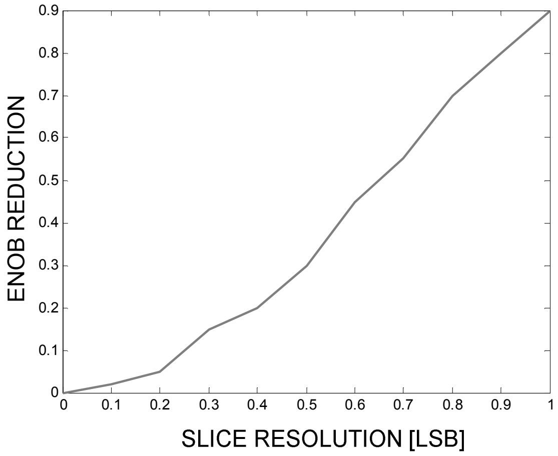


Figure 89: ENOB reduction in function of different slice resolution.

An over-drive test is employed to evaluate the slice resolution. In this test the slice input signal has four different values, in four subsequent clock periods, when the clock is at the maximum sampling frequency as follows:

- The input signal is first completely unbalanced (Full scale) in one direction;
- After a clock period, the signal is unbalanced in the same direction but with an amplitude equal to the resolution we want to test;
- At the next clock period it is completely un-balanced (Full scale) in the same direction;
- Finally the signal is unbalanced in the opposite direction of an amplitude equal to the desired resolution.

If the slice is able to correctly recognize the signs of the over-drive test signal, the hysteresis is smaller that the tested resolution. Figure 90 shows the overdrive test relative to the slice that detects the crossing of the zero differential thresholds. The slice must be able to pass the over-drive test relative to each threshold.

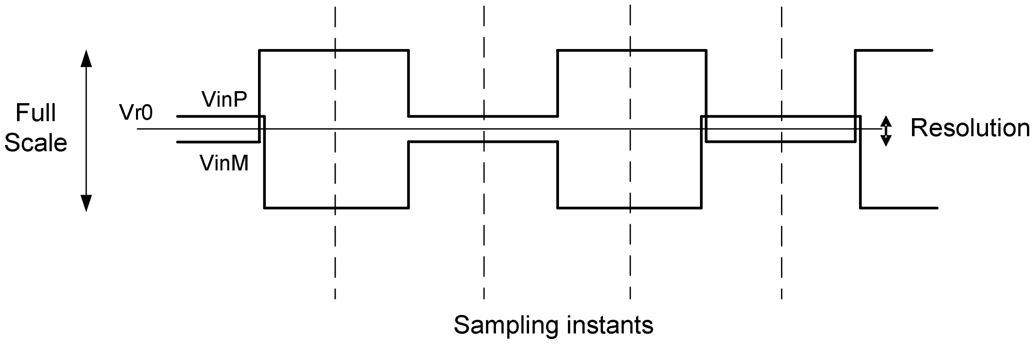


Figure 90: Over-drive test.

3.3.5 Residual offset

Another non-ideality source is the residual offset, referred to the slice input, which remains after the offset calibration. This residual offset can be summed to the offset that affects the thresholds and must be less than 0.1-0.2 quantization steps to avoid significant ENOB reduction. This value is chosen through simulations of the ADC Matlab model explained in the next paragraph.

3.3.6 Expected results

We have separately seen the main sources of ENOB reduction. A Matlab model is employed to evaluate the expected ENOB; this model takes into account almost all these non-idealities.

Table 2 shows two examples with realistic values of the described non-ideality: an optimistic one and another with pessimistic performances.

	Optimistic performances	Pessimistic performances
Buffer and T&H THD	50dB	42dB
Buffer and T&H attenuation (at Nyquist)	1dB	1.5dB
Thermal noise (σ)	0.1LSB (0.8mV)	0.15LSB(1.2mV)
Slice resolution	0.1LSB (0.8mV)	0.15LSB(1.2mV)
Residual offset	0.1LSB (0.8mV)	0.15LSB(1.2mV)

Table 2: Examples of non-idealities.

The ENOB is so calculated with this model using as input a Nyquist frequency signal; Figure 91 shows the FFT.

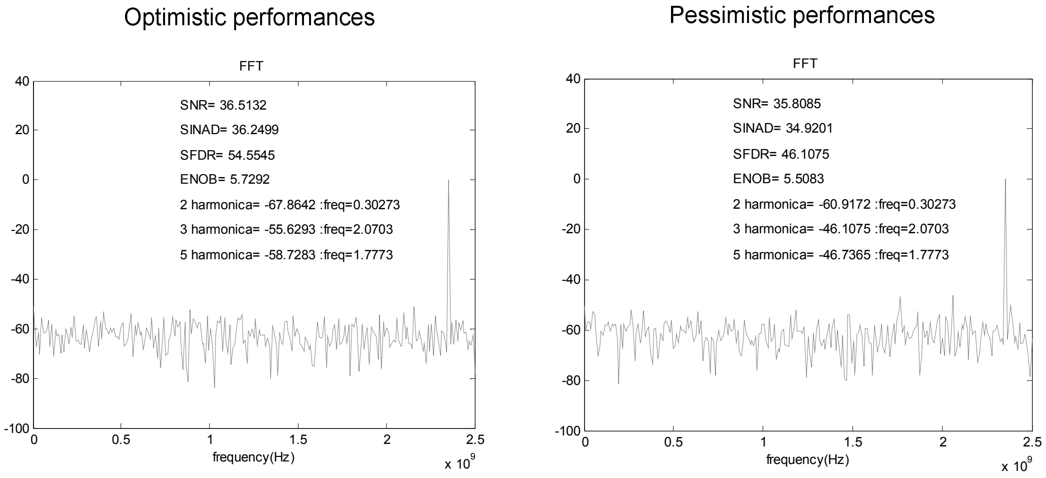


Figure 91: FFT of the converted signal using an ADC model.

This model doesn’t take into account the KT/C noise due to the track&hold capacitance. KT/C noise is totally negligible because of the big ADC input capacitance.

3.4 FULL-RATE VS. HALF-RATE ARCHITECTURE

3.4.1 Introduction

We will compare two possible implementation of the flash ADC in order to evaluate which is the best, especially in terms of power consumption.

The first architecture is called full-rate because each circuit of the slice uses a full-rate clock at 5 GHz. The building blocks of the ADC core are the buffer, the track&hold and 62 slices. The two slices in offset calibration are not considered. Each slice contains a preamplifier, two reset latches and a dynamic CMOS flip-flop, Figure 92. The track&hold and the preamplifier sample the signal and the latches chain regenerates it.

The preamplifier is a circuit that detects the crossing of differential thresholds: its differential output is the difference between the differential input signal and the differential thresholds: (INP1-INN1) – (VrP-VrN). This value is multiplied by its voltage gain.

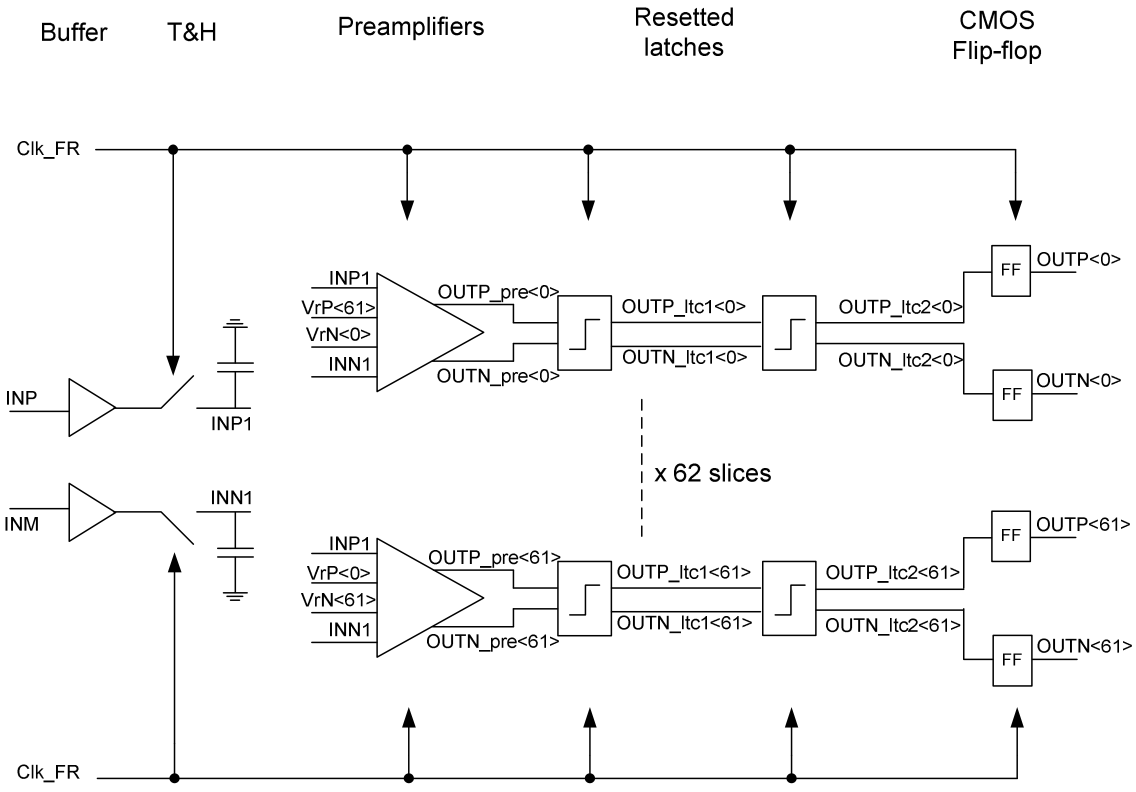


Figure 92: Full-rate architecture.

The first reset latch regenerates the preamplifier output. This latch has also a reset phase that cancels the memory of the preceding sampled signal. At this clock frequency a single latch is not enough to fully regenerate the signal at CMOS levels, for this reason there is a second reset latch followed by a CMOS flip-flop.

The second architecture is called half-rate; in this architecture the ADC works with two clock domains: the first part, made of the track&hold and the preamplifiers, is full-rate; the second part, the regenerative one, is half-rate (2.5 GHz) in order to give to the latches more time to regenerate the sampled signal. The regeneration phase is long twice the full-rate one; for this reason a single latch is enough to translate the signal at CMOS levels. Obviously the half-.rate part is made of two different paths that regenerate alternately the full-rate samplings (path A and path B).

The buffer, the track&hold and the preamplifiers are the same for the two architectures. Both the two slices are designed to have a resolution of 0.15 LSB (1.2mV) and the noise standard deviation value referred to the slice input is 0.15 LSB (1.2mV). These values have been calculated in the worst corner and temperature, according to the specification defined in the previous paragraph.

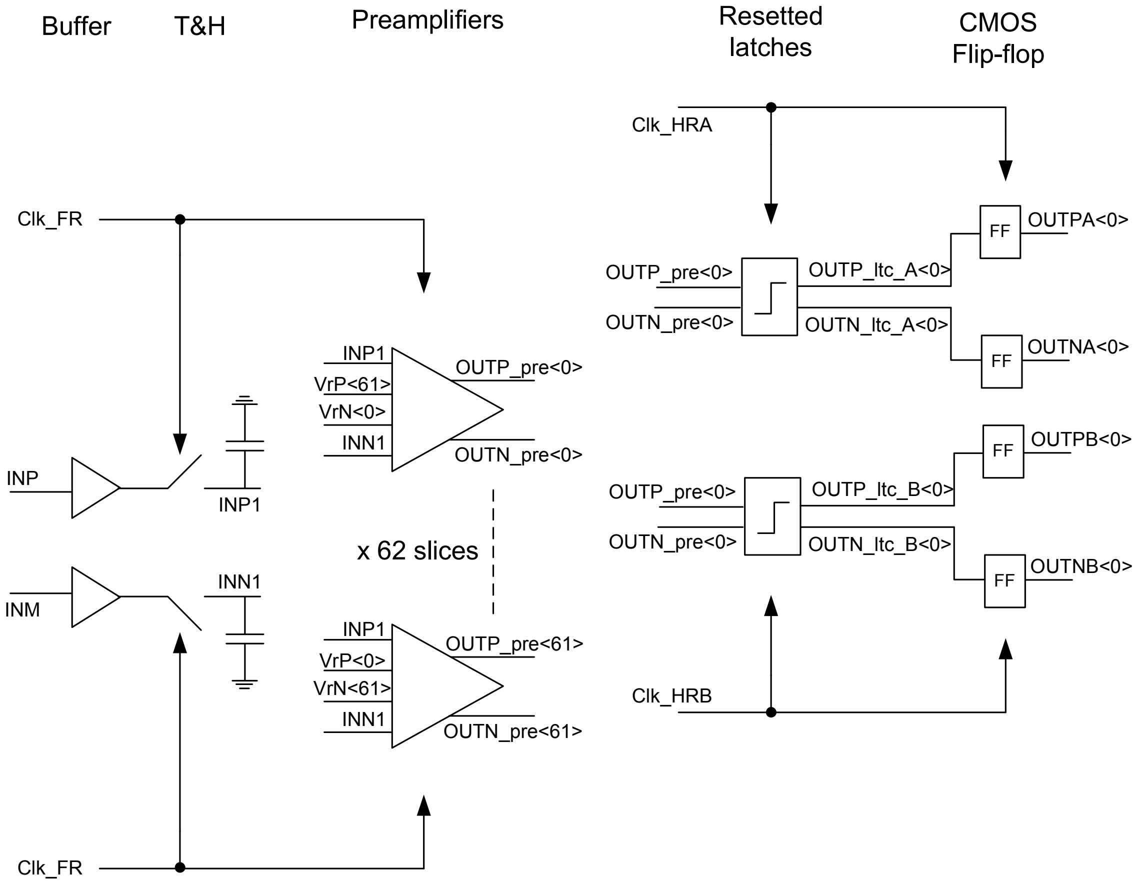


Figure 93: Half-rate architecture.

3.4.2 *Buffer and track & hold*

The buffer is a simple source follower followed by a complementary n-p MOS switch for the Track&Hold function; this complementary switch requires a clock and its complemented version. When the CLK signal is low the switch are conductive and the output is a replica of the input signal. When the clock goes high the signal is sampled and held by the sampling capacitance consisting of the parasitic capacitance of all the slices (approximately 1pF). With this big value of the sampling capacitance the standard deviation of the KT/C noise is 64.5uV, which is totally negligible respected to the ADC quantization step.

The circuit shown in Figure 94 is pseudo differential and this partially rejects the even harmonics distortion, but there is linearity degradation due to odd harmonics, in particular the third one. This circuit is designed to have a THD better than 50dB in the typical corner with a signal at Nyquist frequency, full scale amplitude and 5GHz sample clock. In the worst corner and temperature the circuit has a THD of 42 dB. Moreover there is the bandwidth limitation due to the big ADC input capacitance: the attenuation at Nyquist frequency (2.5GHz) is approximately 1 dB in the typical corner and 1.5 dB in the worst corner and temperature.

The output common mode is 650mV and the buffer power consumption is approximately 15 mW.

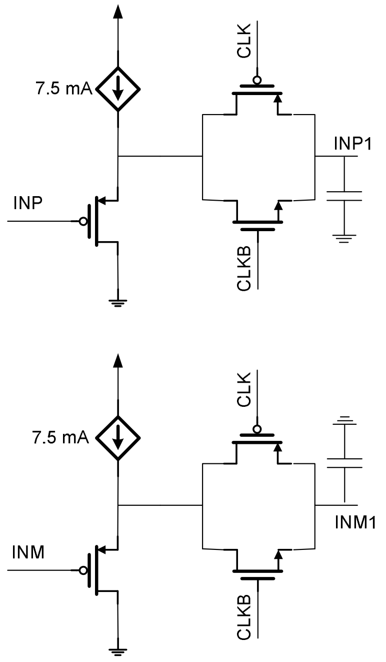


Figure 94: Buffer and Track&Hold.

3.4.3 Preamplifier

The preamplifier is the first building block of a slice and must detect the crossing of differential thresholds.

This circuit is made of two differential pairs, biased with a constant gm current and connected as in Figure 95.

This circuit works in two phases: the first phase is a reset and coincides with the track phase of the track&hold. The reset shorts the outputs and in this way the memory of the previous sample is deleted, mitigating the bandwidth requirements. The second step is the amplification phase and the logical function is to compute the difference between the differential input signals and the differential thresholds, (INP-INN) – (VrP-VrN), and amplify as much as possible this value to improve the successive regeneration.

The two steps are selected by the RSTB_pre signal that coincides with the track&hold CLK signal. If RSTB_pre is low the preamplifier outputs are shorted. The preamplifier gain must be enough to guarantee an adequate slice resolution to prevent ENOB reduction due to odd harmonics. This gain must be suitable for each threshold crossing of interest.

This preamplifier has an effective gain at the end of the amplification phase of about 2, with an input common mode of 650mV and an output common mode of 700mV. Its power consumption is approximately of 600uW.

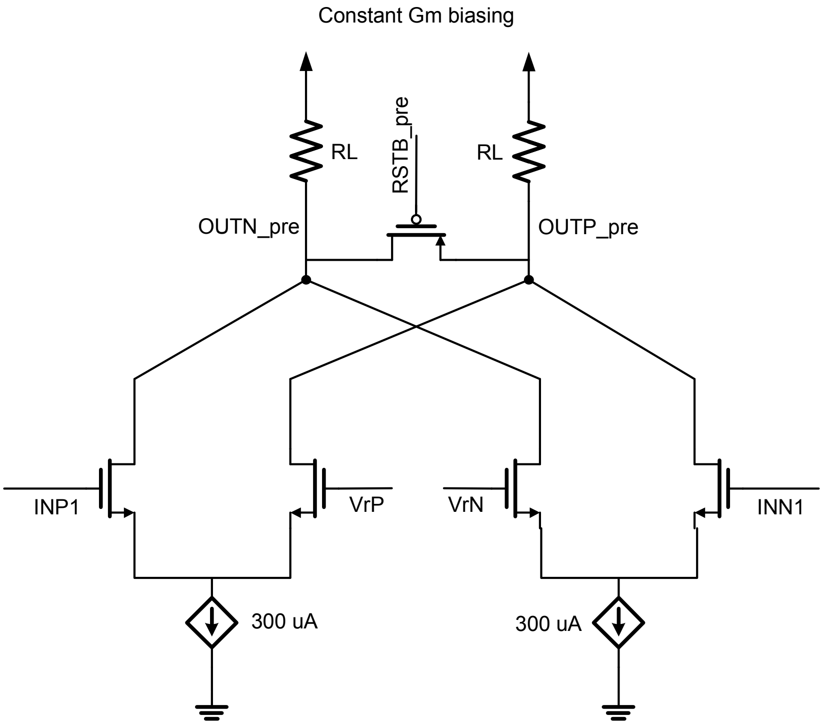


Figure 95: Preamplifier.

The slice must be able to correctly recognize the signs of an over-drive test signal.

Figure 96 shows the overdrive test waveforms related to the full-rate architecture. This figure shows also the four step of the overdrive test:

1. The input signal is completely unbalanced (Full scale) in one direction;
2. The signal is unbalanced in the opposite direction with an amplitude equal to the resolution we want to test;
3. It is completely un-balanced (Full scale) in the direction of the first step (1);
4. The signal is unbalanced in the same direction of an amplitude equal to the desired resolution.

In the half-rate topology, the overdrive test signal is quite different because there are two paths that alternatively regenerate the input signal. For this reason the length of each phase of the overdrive test pattern must be twice the full-rate topology. In this way each path regenerates the correct signal sequence, Figure 97.

This figure shows also the four steps of the overdrive test described before.

The input signal for both full-rate and half-rate is an overdrive test for testing a resolution of 0.1LSB (0.8mV). The preamplifier gain is approximately 2 and so the output at the end of the amplification phase is equal to 1.6mV.

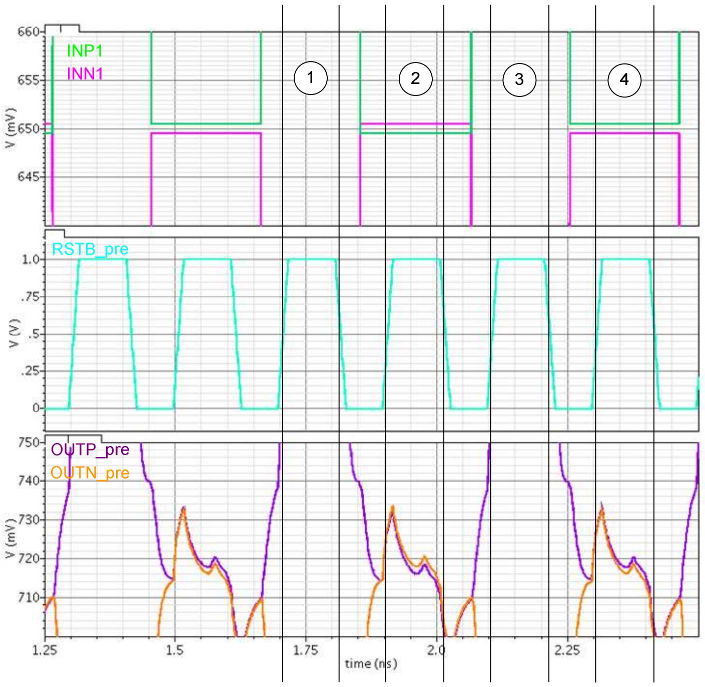


Figure 96: Input, clock and output of the preamplifier (Full-rate overdrive test).

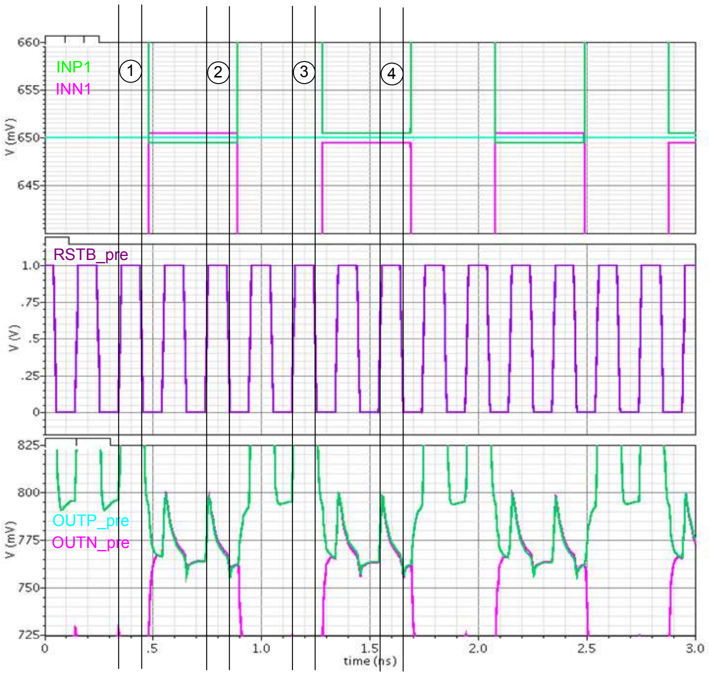


Figure 97: Input, clock and output of the preamplifier (Half-rate overdrive test).

3.4.4 Full-rate latches

In the full-rate architecture the two reset latches are equals and have the topology shown in Figure 98. This latch has three phases: a reset phase (RSTB), a sense phase (SNS) and a regeneration phase. During the reset phase, active low, the transistors with the command signal called RSTB are conductive; the transistors with the command signal called SNS and REG are in high impedance. The output signals are shorted and their voltages are reset to the supply voltage, in this way the sampled signal memory is set to zero.

After the reset there is the sense phase, active high: the transistors with the command signal called SNS are conductive; the transistors with the command signal called RSTB and REG are in high impedance. In this phase the preamplifier output is further amplified by a differential pair on the latch outputs nodes. After the sense, the output latch signal must be regenerated.

In this phase the transistors with the command signal called REG are conductive and the OUTP_ltc, OUTN_ltc signals are regenerated through the two cross-coupled inverters. The latch reset is necessary due to the high frequency sampling rate.

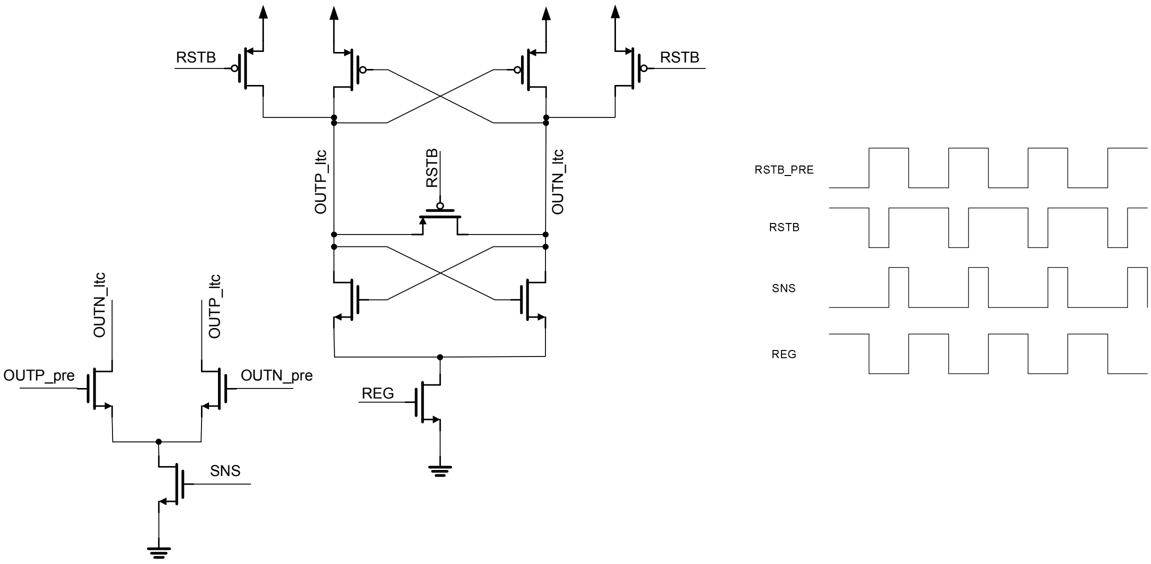


Figure 98: Full-rate latch.

Figure 99 shows the preamplifier output (the first latch input), the three latch sampling phases and the latch outputs. The reset phase coincides with the first interval of the preamplifier amplification, while the sense phase coincides with the second interval, when the preamplifier output has the maximum value. During the reset we can see that the latch outputs are both set to the supply voltage and the differential value is zero. The regenerative latch phase coincides with the preamplifier reset.

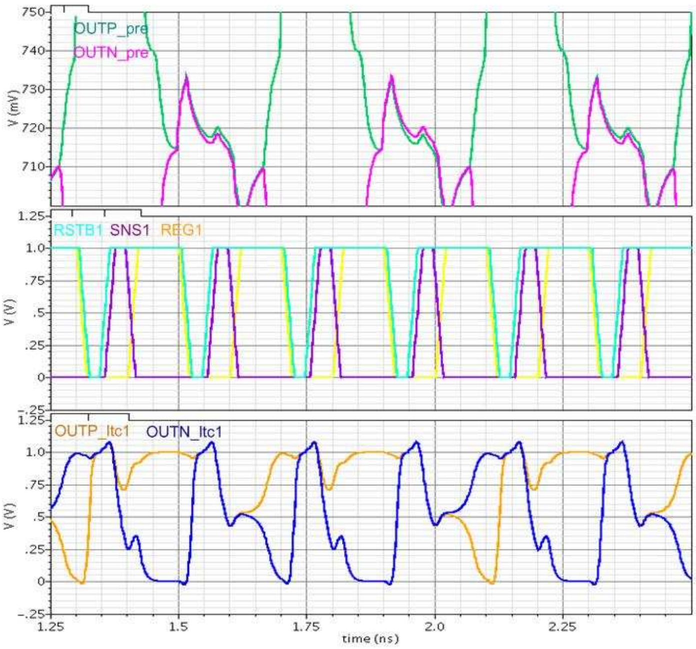


Figure 99: Input, clock and output of the first reset latch.

The first latch output is not a good CMOS signal. A second reset latch, equal to the first one, is employed to further regenerate the signal. Figure 100 shows the first latch output (the second latch input), the three second latch sampling phases and the second latch outputs. The reset phase coincides with the first interval of the first latch regeneration, while the sense phase coincides with the second interval, when the first latch output has the maximum regenerated value.

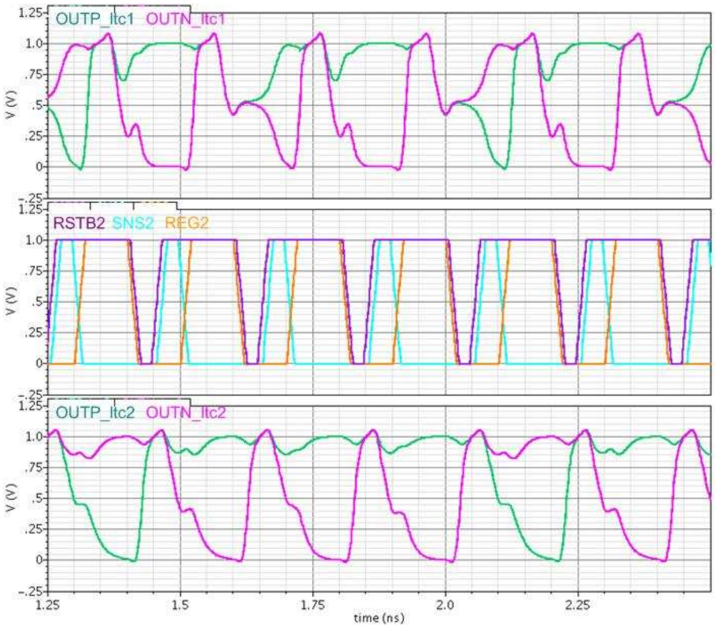


Figure 100: Input, clock and output of the second reset latch.

The second reset latch drives two CMOS dynamic flip-flop, one for the positive output and one for the negative, Figure 101.

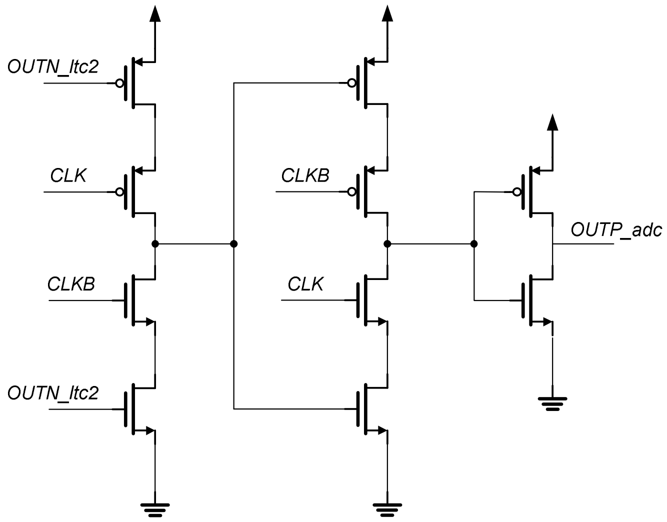


Figure 101: Dynamic CMOS flip-flop.

Figure 102 shows the outputs of the second reset latch, the clock that samples the CMOS flip-flop inputs on the rising edge and the outputs of the two flip-flops.

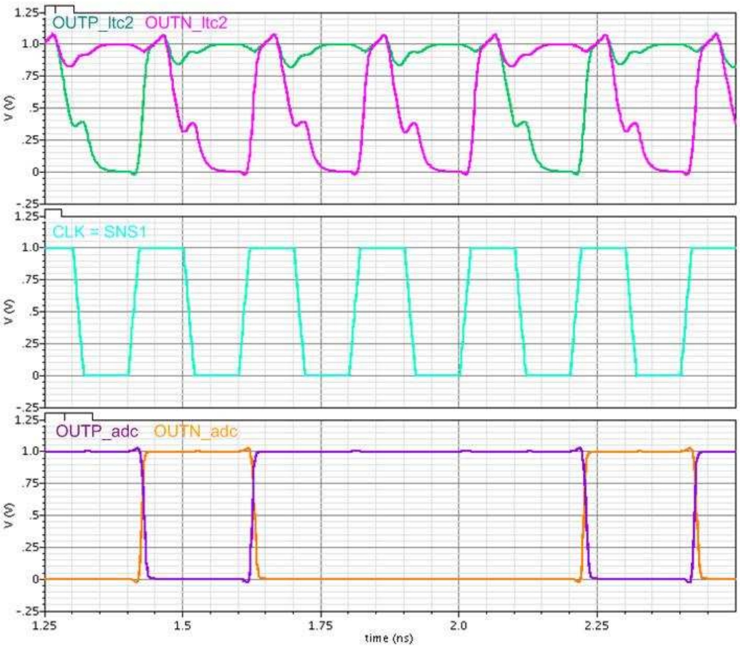


Figure 102: Input, clock and output of the CMOS flip-flop.

The slice regenerative part, made of two reset latches plus two CMOS flip-flop, consumes 400uW.

3.4.5 Full-rate clocks signals generation

The reset latches shown in the previous paragraph require the generation of signals with duty cycle different from 50%.

These signals, the sense and reset, are high or low for a time equal to 50ps, a quarter of the clock period, and the rising and falling time becomes very critical.

It is better to generate these signals locally, inside the single slice, starting from clock signals with duty cycle of 50% in order to not have clock buffers too big.

Figure 103 shows the signals generation starting from two full-rate clocks CLKP and its negative version CLKN locally generated. The signal generation uses only inverters and NAND gates.

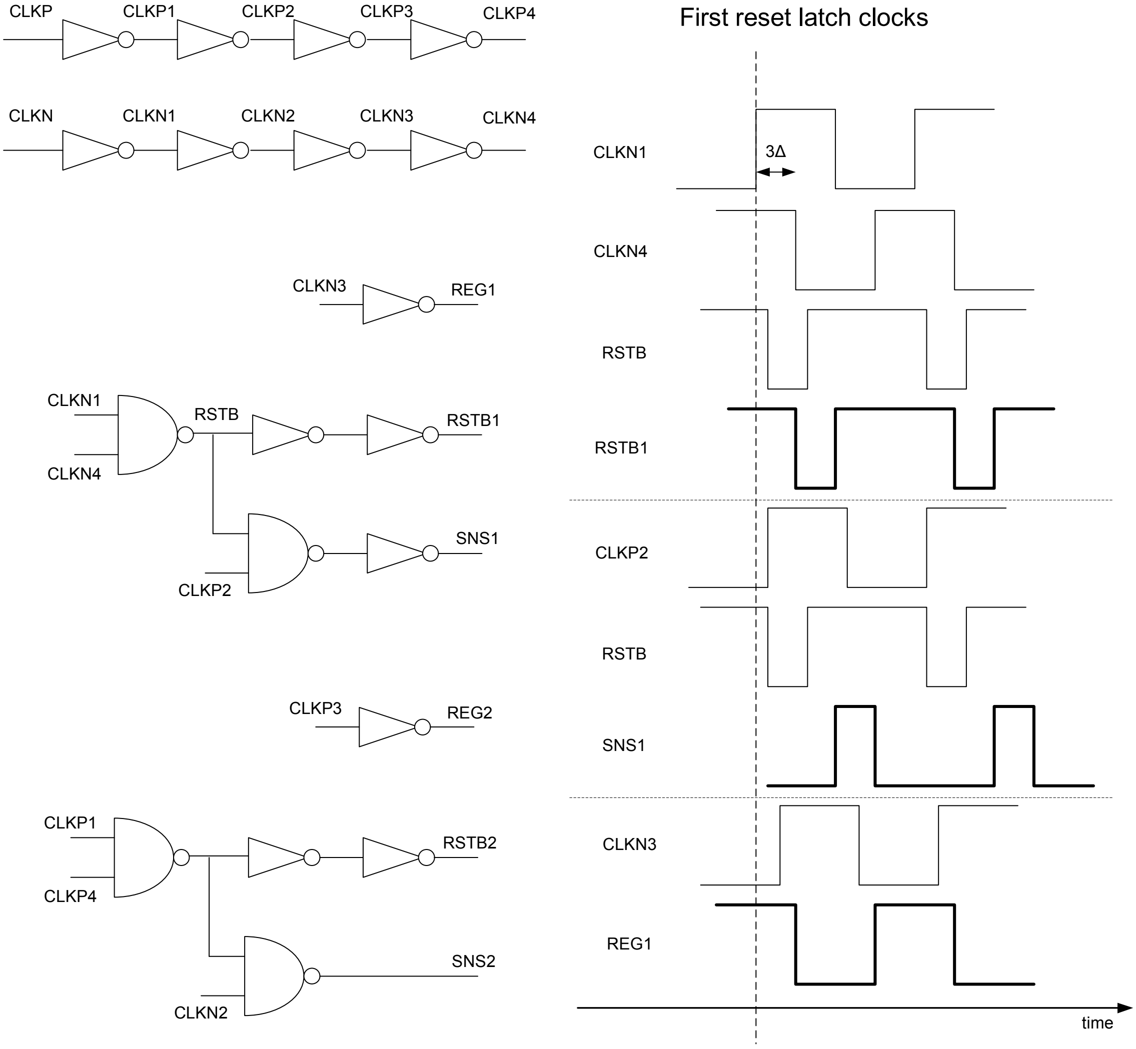


Figure 103: Full-rate local clock signals.

The waveforms related to the first latch signals generation is drown supposing a gate delay of Δ for inverters and NAND. 3 gate delay (3Δ) is equal to 50ps. REG signals are also employed as clock signal for the CMOS latch.

The power consumption for the full-rate clocks signals generation is approximately 1.1mW for each slice.

3.4.6 Full-rate timing diagram

Figure 104 shows the timing diagram of the full-rate ADC architecture. A latency of 1.5 full-rate clock period is required to have a fully regenerated CMOS signal.

The track&hold circuit uses a clock signal with a duty cycle of 50% and a period of 200ps. In the first phase the track&hold senses the input signal and in the successive phase the input signal is sampled and stored in the sampling capacitance (hold phase).

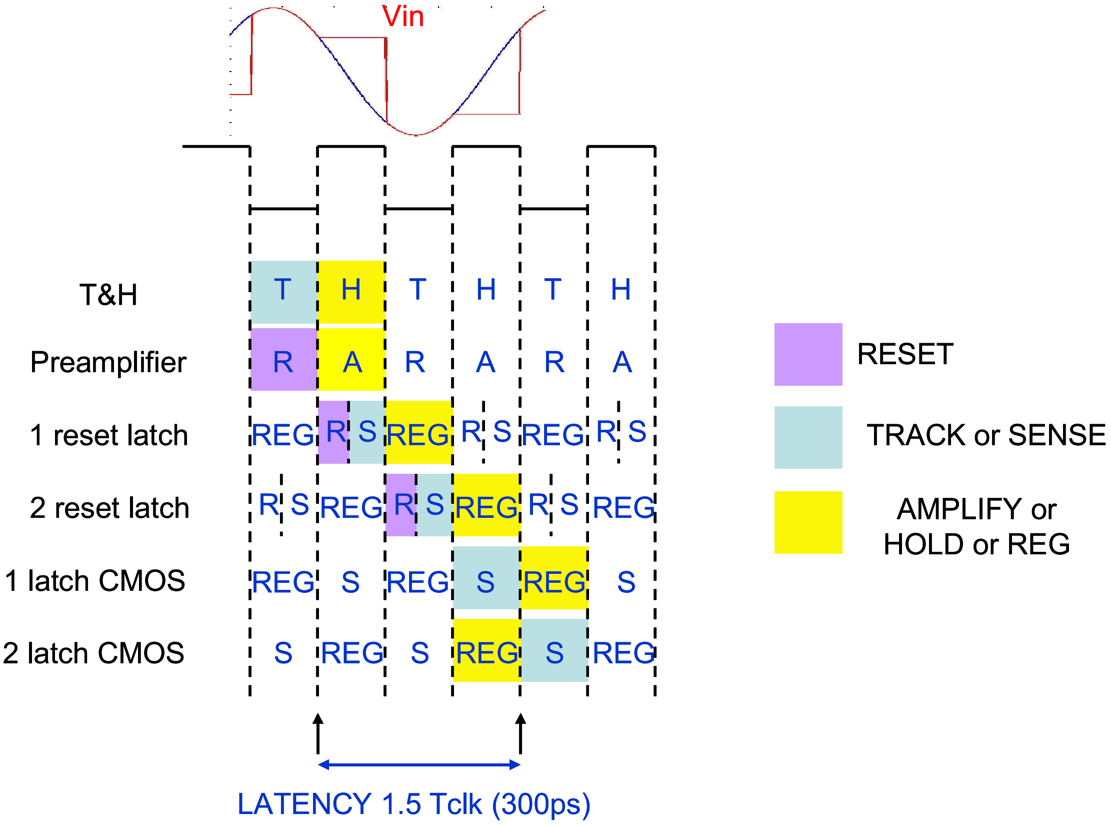

Vin
T&H T H T H T H
Preamplifier R A R A R A
1 reset latch REG R S REG R S REG R S
2 reset latch R S REG R S REG R S REG
1 latch CMOS REG S REG S REG S
2 latch CMOS S REG S REG S REG
RESET
TRACK or SENSE
AMPLIFY or HOLD or REG
LATENCY 1.5 Tclk (300ps)

Figure 104: Full-rate mode timing diagram.

The preamplifier also uses a clock signal with duty cycle of 50%: while the track&hold is sensing the signal, the preamplifier is reset; when the signal is held in the sampling capacitance the preamplifier amplifies this value. After the preamplifier, there are two reset latches and a CMOS flip-flop.

In order to evaluate the total latency of the chain, the initial time is set at the start of the hold phase of the track&hold; the preamplifier does not introduce additional latency and each latch introduce a latency of half clock period (100ps). At the end of the regenerative phase of the second latch the signal is sampled by a flip-flop, made of two CMOS latches, and the signal is available as ADC output. So the latency is of 300ps.

3.4.7 Half-rate latches

The half rate latches chain is made of one reset latch and a dynamic CMOS flip-flop. One latch is enough to fully regenerate the sampled signal because the regeneration time is double respected to the full-rate architecture.

The reset, sense and regenerate latch phases are long twice the full-rate latch ones. There is only one difference with the full-rate latches: in the half-rate latch there are two pass-gates, between the preamplifier output and the latch output, instead of a differential pair. In the half-rate latch an additional gain is not required in the sense phase due to the long regeneration time; the sense phase is also active low, Figure 105.

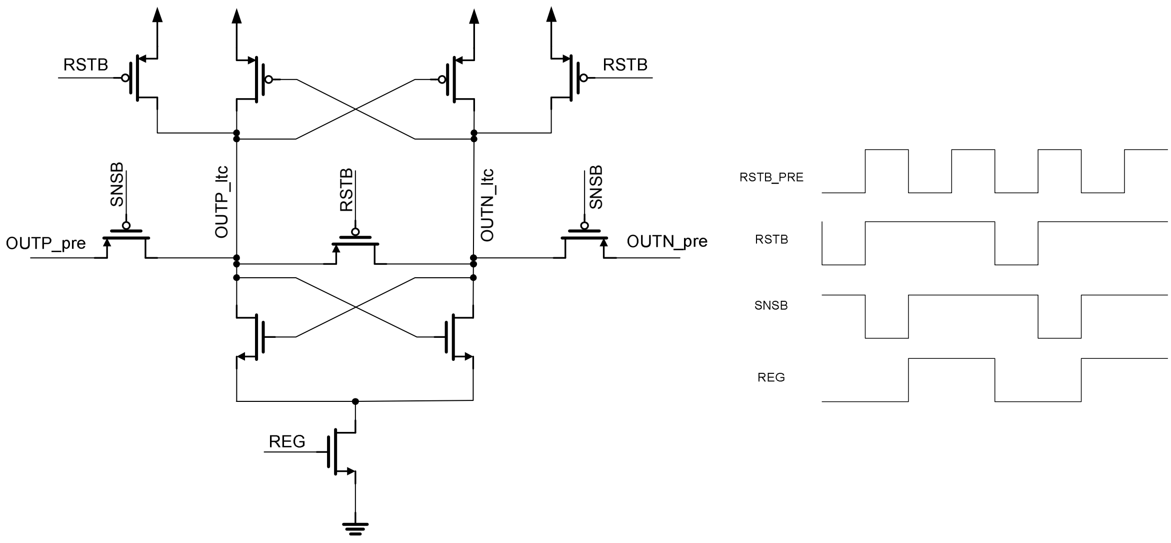


Figure 105: Half-rate latches.

Figure 106 shows the waveforms of the input, the clock signals and output of the half-rate latch topology corresponding to one of the two paths. The latch input is the preamplifier output with an overdrive test signal as input of the slice.

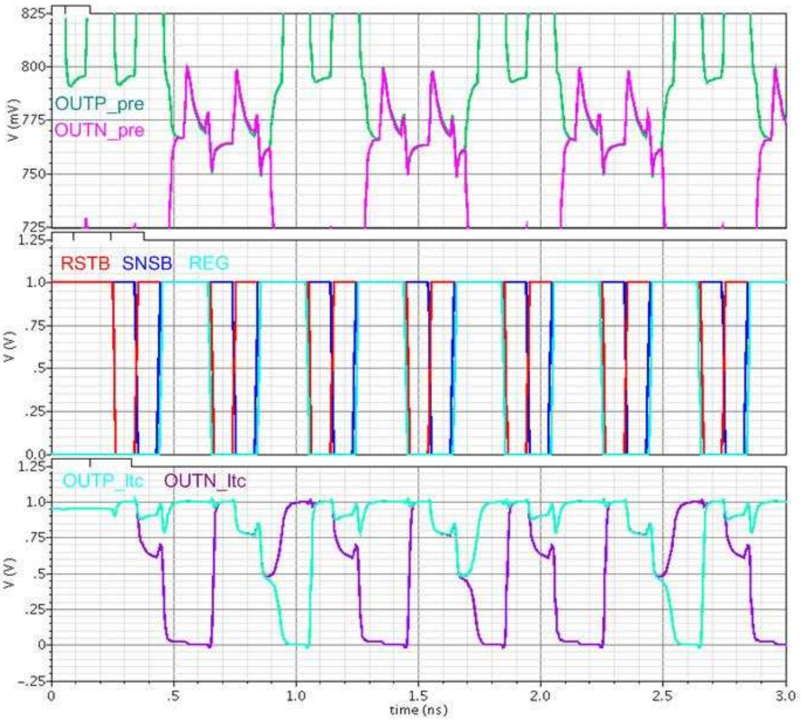


Figure 106: Input, clock and output of the half-rate reset latch.

The track & hold and the preamplifiers work with a full-rate clock, while the latches chain works in half-rate mode; for this reason there are two latches chains that alternatively work, Figure 107.

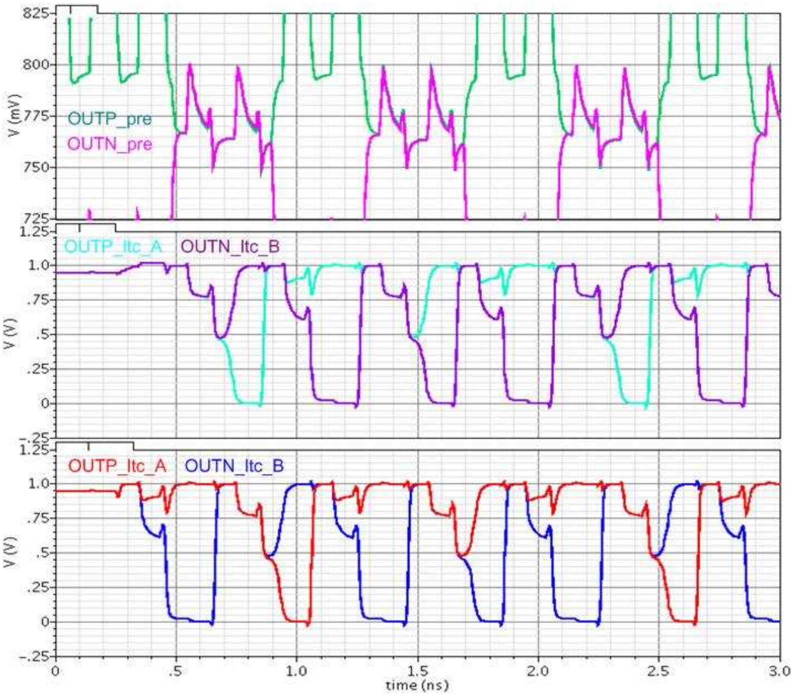


Figure 107: Output of the two half-rate paths.

After the reset latch there are two CMOS flip-flop, one for each output, positive and negative. This circuit is the same of the full-rate architecture one, but with a clock signal of doubled period.

3.4.8 Half-rate clock signals generation

The clock signals generation for the half-rate mode is similar to the full-rate one and is also locally generated in the slice. The slice receives a full-rate clock (200ps period) and a half-rate (400ps period) clock.

These clocks (CLK_FRP, CLK_FRN, CLK_HRP, CLK_HRN) are employed to generate the RSTB, SNSB and REG signals for the two paths A and B. The signal generation uses only inverter and NAND gates, Figure 108. The waveforms related to the signals generation of path A is drown, supposing a gate delay of Δ both for inverters and NAND.

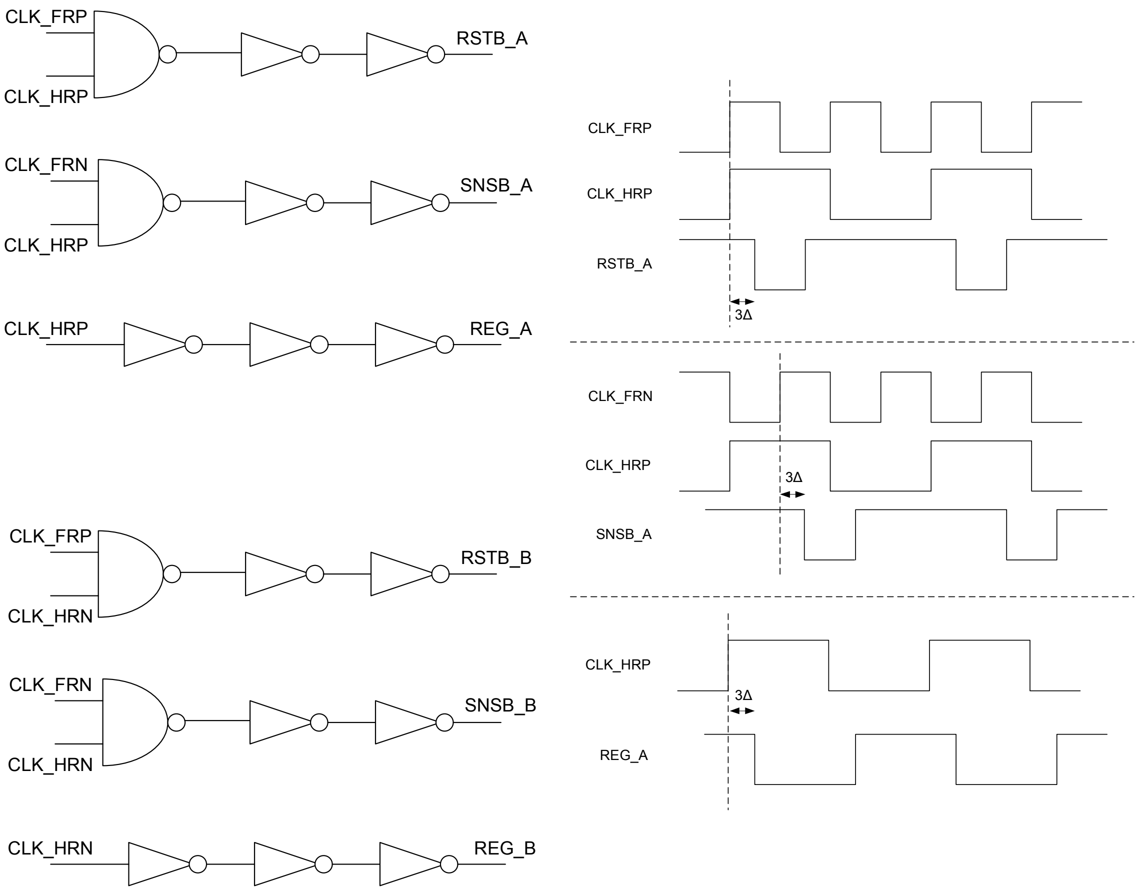


Figure 108: Half-rate local clock signal.

The power consumption for the half-rate clock signals generation is approximately 600uW.

3.4.9 Half-rate timing diagram

Figure 109 shows the timing diagram of the half-rate ADC architecture. The timing of the track & hold and the preamplifier is exactly the same of the full-rate architecture. The latch reset phase coincides with the preamplifier reset and the sense coincides with the preamplifier amplification phase. Both are 100ps wide. The regenerative phase is 200ps wide. The two paths A and B alternatively regenerate the preamplifier output.
CMOS signal is generated with a latency of 1.5 full-rate clock period, exactly equal to the full-rate mode.

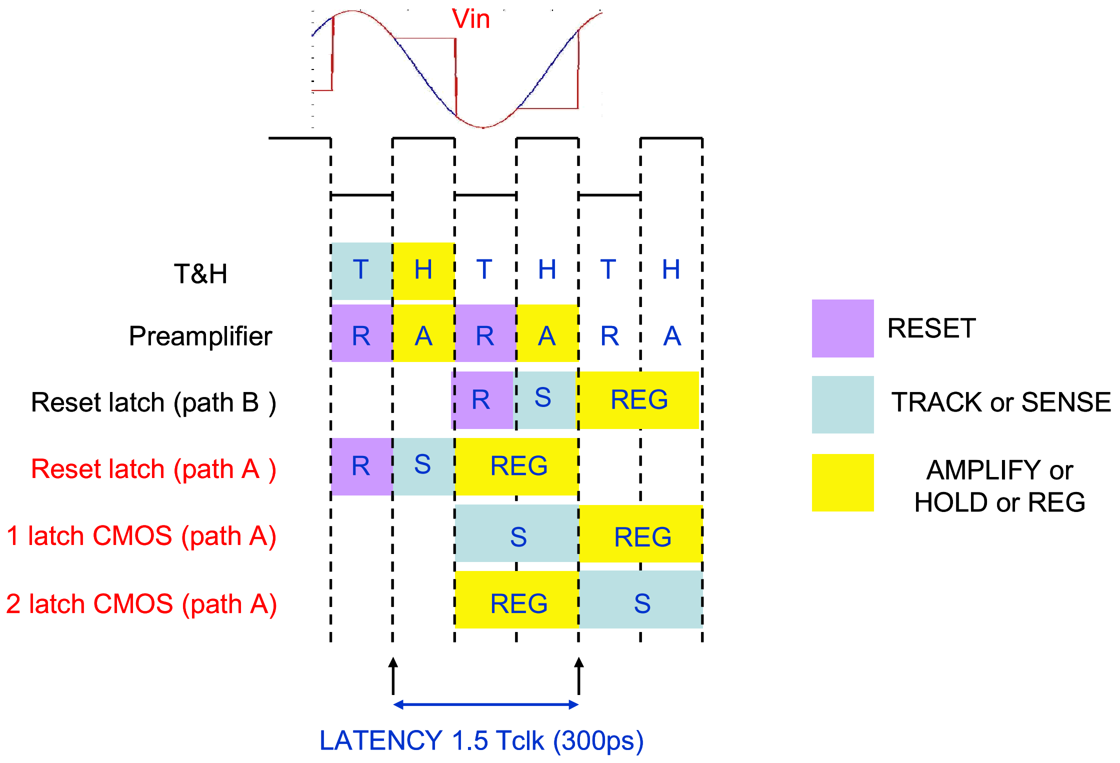


Figure 109: Half-rate mode timing diagram.

3.4.10 Power consumption comparison

Table 3 compares the power consumption of the full-rate and the half-rate architecture; the track and hold, the clock buffer and the digital part are not taken into account. We can see that the latches consume a little more power for the full-rate architecture respect to half-rate. The preamplifier is exactly the same.
The big difference is in the local clock signals generation.
For the full-rate architecture the generated clock signals are:

- the preamplifier reset (5GHz frequency);
- The sampling signals for the two cascaded reset latches at 5GHz frequency (sense, reset and regeneration) ;

- The clock for the two single ended CMOS flip-flops (5GHz frequency).

For the half-rate architecture the generated clock signals are:

- the preamplifier reset (5GHz frequency);
- The sampling signals for the two reset latches at 2.5GHz frequency (sense, reset and regeneration), one for each paths A and B;
- The clock for the four single ended CMOS flip-flops (2.5GHz frequency), two for path A and two for path B.

The power consumption of the clock signals generation is dominated by the generation of the sampling signals for the reset latches. Full-rate and half-rate employ the same number of reset latches, but the operating frequency is halved in half-rate respect to full-rate. For this reason the power consumption for half-rate clocks generation is approximately half respect to the full-rate one.

The difference in term of power consumption is approximately 38mW. Because of the lower power consumption, we have chosen the half-rate architecture.

	Full-rate	Half-rate	Full-rate x 64	Half-rate x64
Preamplifiers	600uW	600uW	38.4mW	38.4mW
Latches	400uW	300uW	25.6mW	19.2mW
Clock gen.	1.1mW	600uW	70.4mW	38.4mW
Total	2.1mW	1.5mW	134.4mW	96mW

Table 3: Power consumption (full-rate vs. half-rate).

3.5 OFFSET CALIBRATON

3.5.1 Introduction

We need an offset calibration due to the small dimension of the circuits and due to the preamplifier gain (approximately 2), which makes the latches offset non negligible. We assume the offset referred at the slice input has a Gaussian pdf distribution. The goal of this calibration is to correct a maximum value equal to +- 3σ with a resolution better than 015 quantization steps.

The 3σ input referred offset, evaluated from a "Montecarlo" simulation, is ±50mV. Furthermore the ADC calibration should track the offset drift with the temperature and therefore a background calibration has been implemented [27].

3.5.2 Auto-zero

The offset calibration is made by an auto-zero technique. The slice inputs are shorted to the respective thresholds and its output signs are integrated. These integral values are employed in feedback to control the amplitude of the differential currents, which calibrates the offset. These currents are injected to the reset latch output; in this way the preamplifier offset is also corrected.

Figure 110 shows the offset calibration for the path A; for the path B there is the same mechanism that acts contemporarily. When the offset is calibrated the integrators outputs toggles between two values. Figure 111 shows the reset latch with the transistors employed to convert the integrators voltage in a differential current, injected in its outputs.The latch has three working phases: reset, sense and regeneration.

The differential current, for the offset calibration, is injected in its outputs during the first part of the regeneration phase. The current generators for the offset calibration are active only in this phase. The transistor, driven by the OFF_CALB signal (active low), acts as switch and turns off the current injection.

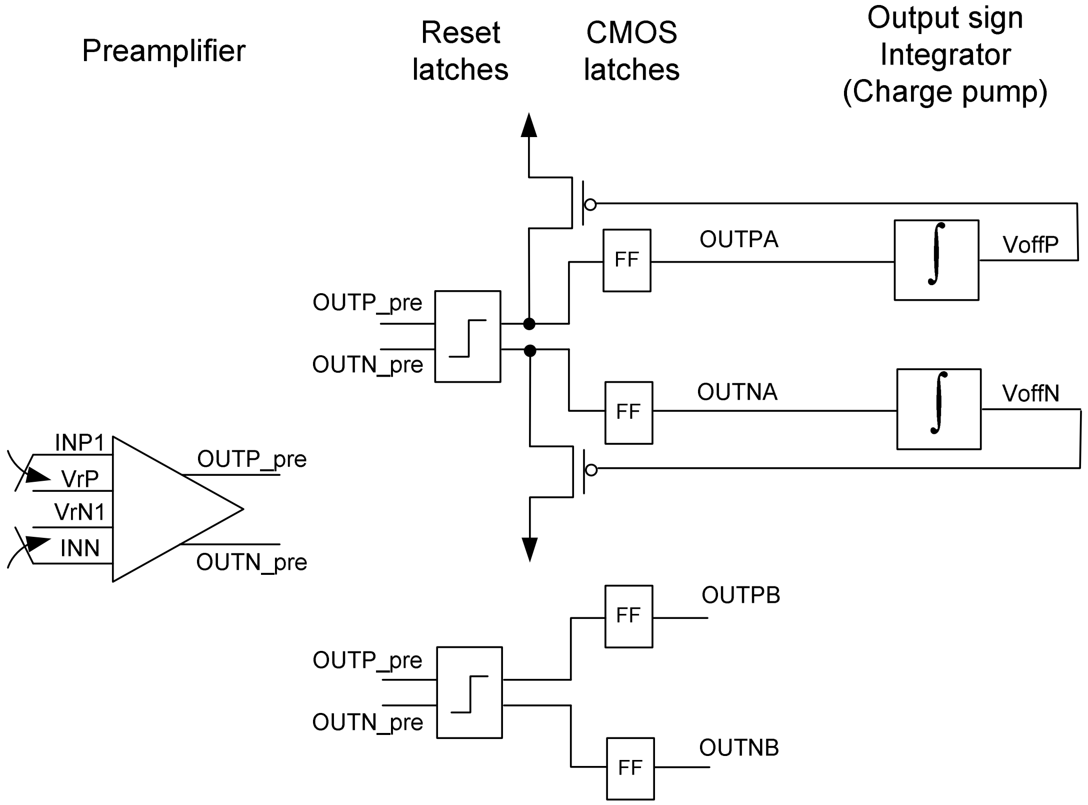


Figure 110: Offset calibration: auto-zero.

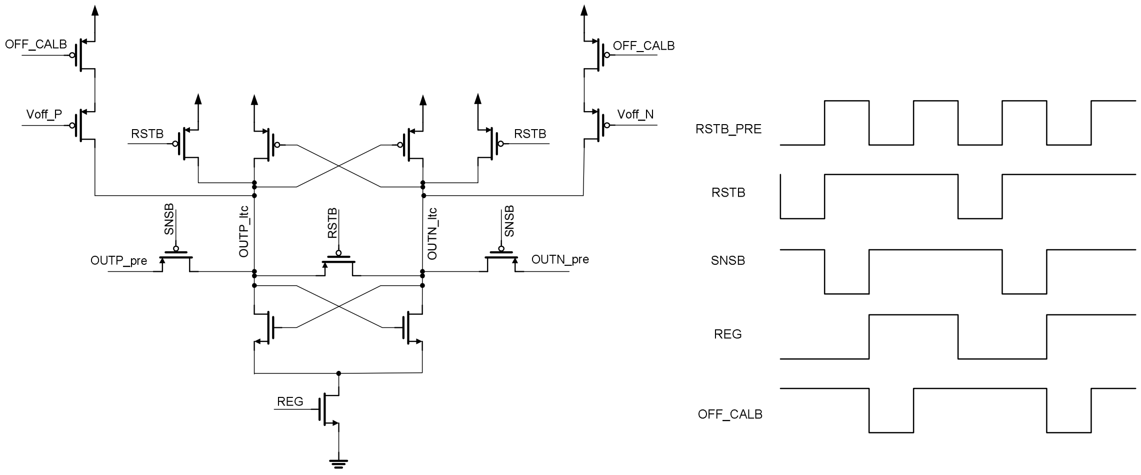


Figure 111: Latch with the offset calibration.

3.5.3 Charge pump

A charge pump is employed to integrate the sign of the output signal. Basically this circuit is the one shown in Figure 112.

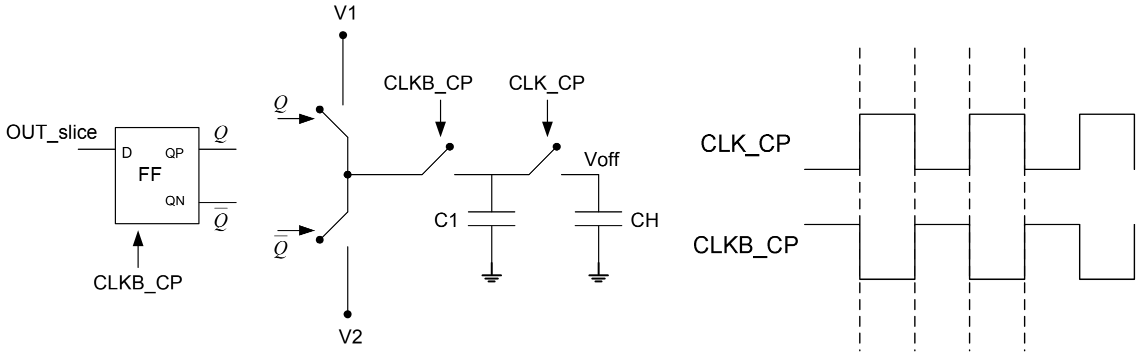


Figure 112: Charge pump.

The flip-flop samples the single ended output of the slice (positive or negative) with a clock divided by 16 respect to the full-rate one. The flip-flop outputs are employed to control two switches that select between the signals V1 and V2 (V1 is used to increments the integrated signals, V2 to decrement it). One of these values is pre-charged on the C1 capacitance during the first integration phase (CLKB_CP high); during the second phase (CLK_CP high) there is a charge sharing between C1 and CH. The charge pump output changes at the rising edge of CLK_CP. The integrated value is Voff. Voff can be calculated in this way, when V1 signal is selected:

3.12 $$V_{off} = \frac{C_1}{C_1 + C_H} \cdot V_1 + \frac{C_H}{C_1 + C_H} \cdot V_{off} Z^{-1}$$

We can see that Voff at a given sampling time depends on the Voff at the preceding sampling time, multiplied by a capacitive partition, plus the V1 signal multiplied by another capacitive partition. It is possible to write the integration step value in this way:

3.13 $$V_{off} - V_{off}Z^{-1} = \frac{C_1}{C_1 + C_H} \cdot V_1 - \frac{C_1}{C_1 + C_H} \cdot V_{off}Z^{-1}$$

The integration step is not constant but depends on the difference between V1 and Voff: the integration step value gets low when this difference decreases. The step value determines the resolution of the offset correction, which must be less than 0.2 LSB, referred to the slice input, to prevent ENOB reduction.

The integration step is chosen equal to 10mV corresponding to a resolution of 1mV, referred to the slice input. The reduction of this step affects only the required time for the convergence; so, since the calibration is running, this reduction affects the capability of compensate offset drift. In order to minimize the offset calibration time the integration step must be as constant as possible. In order to have a constant step, we can rewrite the formula in the following way:

3.14 $$V_{off} - V_{off}Z^{-1} = \frac{C_1}{C_1 + C_H} \cdot V_1 - \frac{C_1}{C_1 + C_H} \cdot V_{off}Z^{-1} = CONSTANT$$

3.15 $$V_1 = \frac{C_1 + C_H}{C_1} \cdot CONSTANT + V_{off}Z^{-1}$$

Figure 113 shows the employed charge pump architecture: the V1 and V2 values are shifted replica of Voff; two source followers are employed to shift them of a transistor threshold (Vth). In this way the integral step is equal to:

3.16 $$V_{off} - V_{off}Z^{-1} = \frac{C_1}{C_1 + C_H} \cdot V_{th}$$

In order to increase the integration step the C1/CH value must be increased.

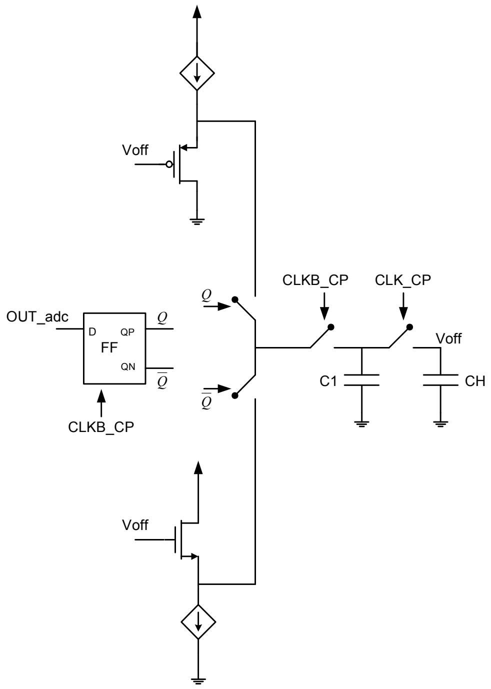


Figure 113: Charge pump with constant integration step.

3.5.4 Algorithm

The aim of the calibration algorithm is to assure the compensation of the offset of every slice during the normal working of the ADC. The offset compensation has to be transparent to the converter so that it can be always active tracking the slow offset drift. There are 64 slices logically divided in two groups of 32. Two of them, one for each group, are contemporarily offset calibrated through an auto-zero procedure, while the remaining 62 are employed to convert the signal. The slices in calibration are periodically chosen among the 64 slices.

The digital algorithm generates three types of signals, employed by the two groups of 32 slices:

- CAL<0:31>, are employed to select the slice under calibration (active high) and to activate the corresponding charge pump; only one CAL signal is active for each algorithm step;
- TH<0:29>, selects the threshold employed by each slices for the signal conversion or calibration. During the auto-zero the input of the slice under calibration is shorted to the threshold, which the slice will use for converting the input signal at the next algorithm step.
- SEL<0:30>, selects the outputs of the slices, which are employed for the conversion.

While the algorithm is working, it is essential to minimize the number of switching multiplexers, because the charge injection, associated with these switch, degrades the ENOB.

In order to minimize those switches, the slice in calibration is always adjacent to the preceding calibrated one. Each slice has a fixed dedicated time for the calibration, after that the next slice must be calibrated. This time is equivalent to 8 clock period of the charge pump and the calibration signals are aligned with the rising edge of this clock

The charge pump employs a clock divided by 16 respected to the full-rate clock and so each integral step requires a time of 3.2ns (200ps X 16). The calibration time for each slice is 25.6ns (3.2ns X 8).

Using a graphical representation, the evolution of the slice under calibration is represented as a triangular waveform with values between 0 and 31, Figure 114.

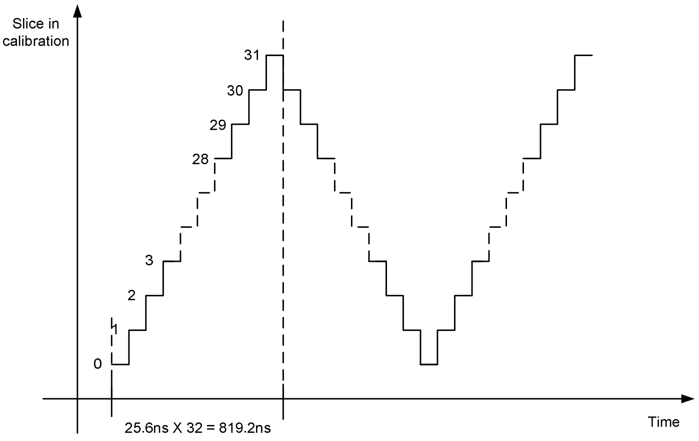


Figure 114: Time evolution of the slice in calibration.

Figure 115 shows four steps of the algorithm, which calibrates 32 slices. The slice under calibration is indicated in blue colour (CAL=1); the slice employed for the conversion is shown in red (CAL=0). After the last or first slice has been calibrated, the algorithm starts again.

The figure also shows the selected input of the thresholds multiplexers and ADC output multiplexers. The slice in calibration has the inputs shorted to the thresholds and the input signal path is switched off. The first and last slices are always employed to calibrate or to detect the crossing of the first and last thresholds.

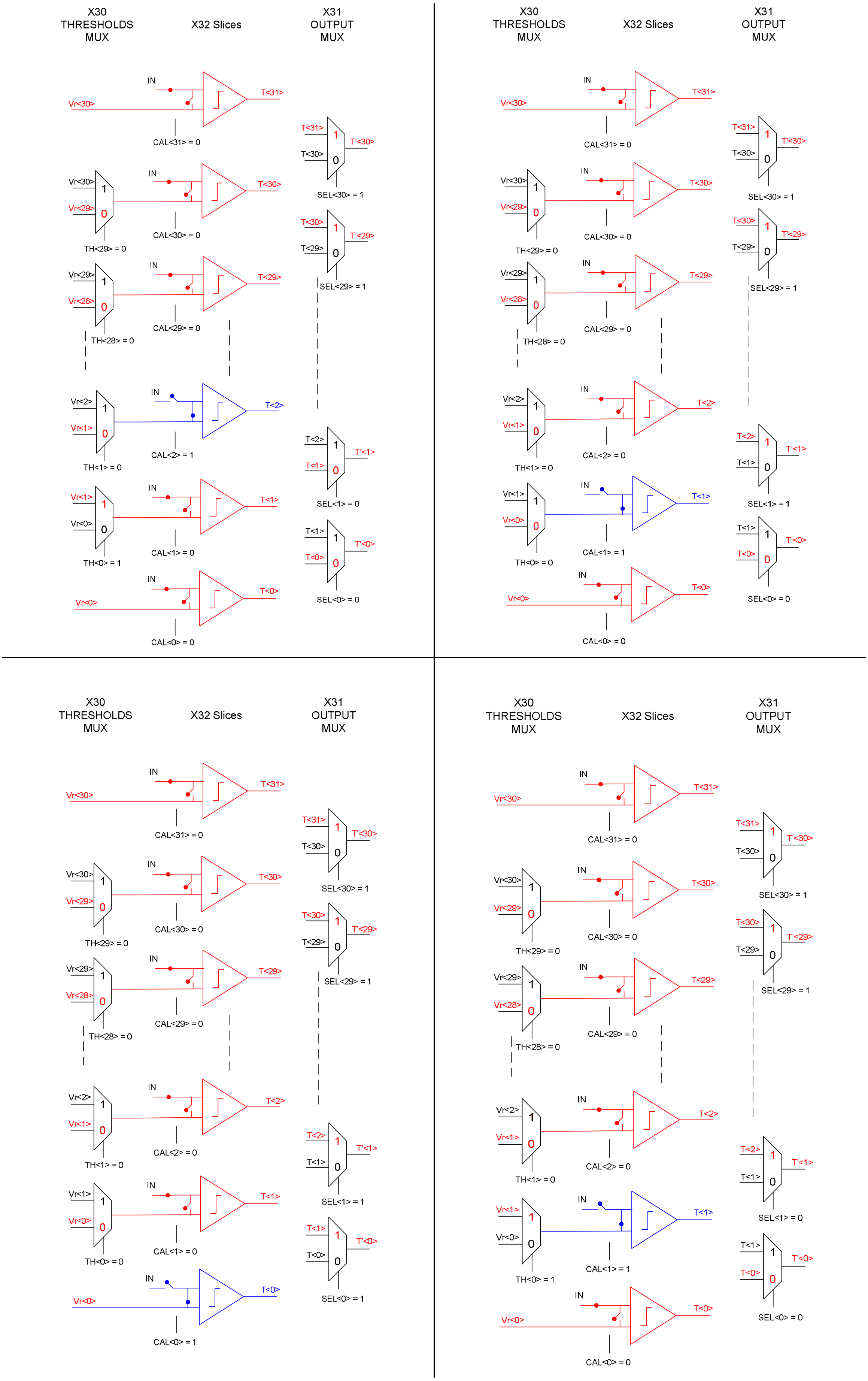


Figure 115: Offset calibration.

Actually the CAL signals are high for only 7 charge pump clocks period (22.4ns) and for one clock period none slice is under calibration. This ensures that when the SEL signals (that select the outputs) switches, the just calibrated slice has already been employed for the conversion and its output has settled, Figure 116.

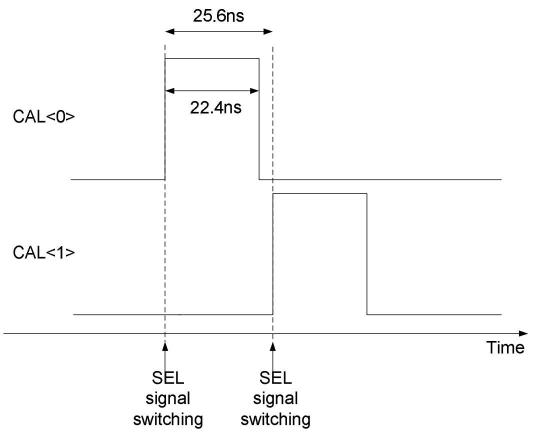


Figure 116: CAL signals.

Figure 117 shows waveforms of the offset calibration related to the thirtieth slice. The plotted waveforms are the CAL<30> signal, the charge pump clock CLK_CP and the difference between the output of the two charge pumps VoffP-VoffN. A 9mV input referred offset is present.

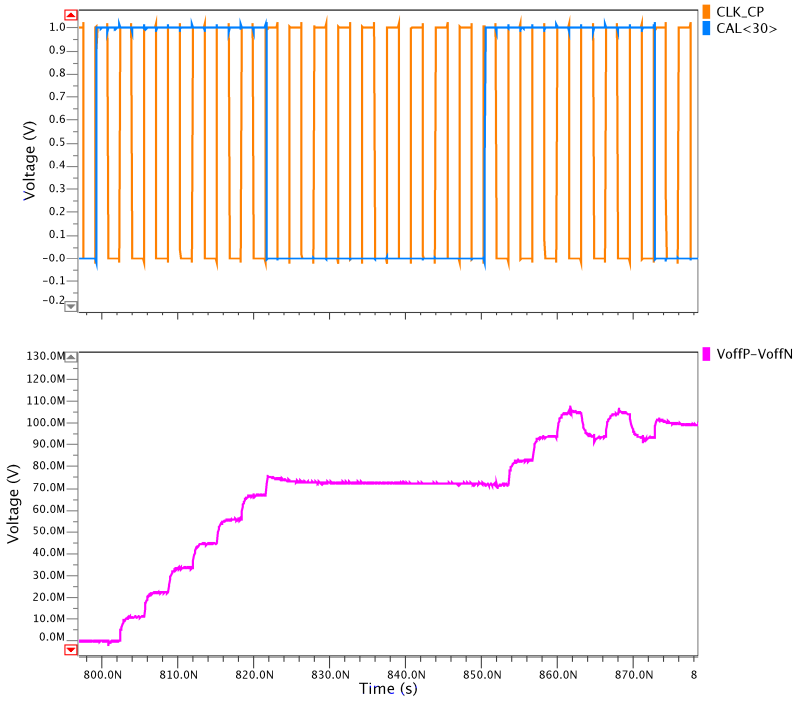


Figure 117: Running offset calibration algorithm.

The differential integrator step is 10 mV. If we refer it to the slice input, it corrects an offset value of approximately 1mV. 9 integrator steps compensate the offset and 2 time interval of the offset calibration algorithm are required. When the offset is calibrated the differential integrator output toggles between two values.

3.5.5 Digital signals generation

The digital control signals required by offset correction are three: CAL <0:31>, TH<0:29> and SEL<0:30>. These signals respectively select the slice under calibration (active high), the thresholds of each slice and the correct slices outputs.

There is a direct relation between CAL and SEL signals: the not employed slice output is always the slice under calibration one. On the other hand the TH signal depends on the next slice under calibration and so from the future CAL signal.

The first step to generate these signals is to design a 5 bit binary up-down counter that takes into account the slice under calibration. This counter must have a control signal indicating if the count must be incremented or decremented. When the count reaches its maximum or minimum value (31 or 0), this command signal must be inverted. Figure 118 shows the waveforms of a binary counter.

If the count is incremented the waveforms have the following behaviour: b0 commutes at each clock rising edge; b1 commutes if b0 is equal to a high logical value and there is a clock rising edge; b2 commutes if b1 and b0 are equal to a high logical value and there is a clock rising edge, and so on. If the count is decremented the waveforms have the following behaviour: b0 commutes at each clock rising edge, b1 commutes if b0 is equal to a low logical value and there is a clock rising edge, b2 commutes if b1 and b0 are equal to a low logical value and there is a clock rising edge, and so on.

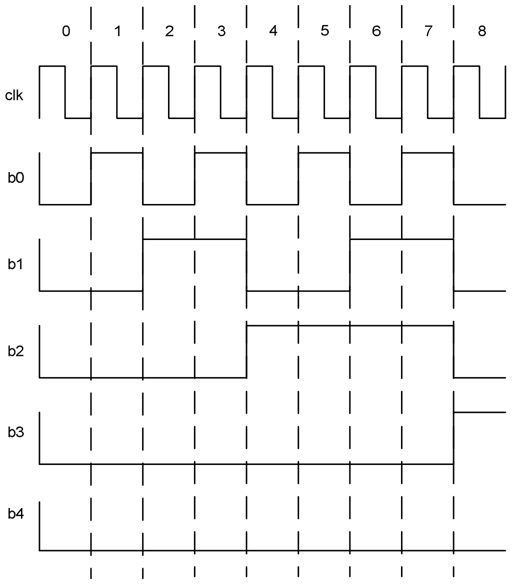


Figure 118: Binary counter waveform.

This behaviour is obtained by the synchronous counter shown in Figure 119. This figure doesn’t show the clock signal, which is a clock divided by 8 respect to the charge pump one.

The first flip-flop, that computes QP<0>, acts as a toggle and its output changes every clock cycle; the others flip-flop toggle only if the flip-flop outputs on the left are 1 or 0 depending from the INC signal. The INC signal (active high) selects if the count must be incremented or decremented. The flip-flop outputs QP<0:4> are the binary representation of the counter.

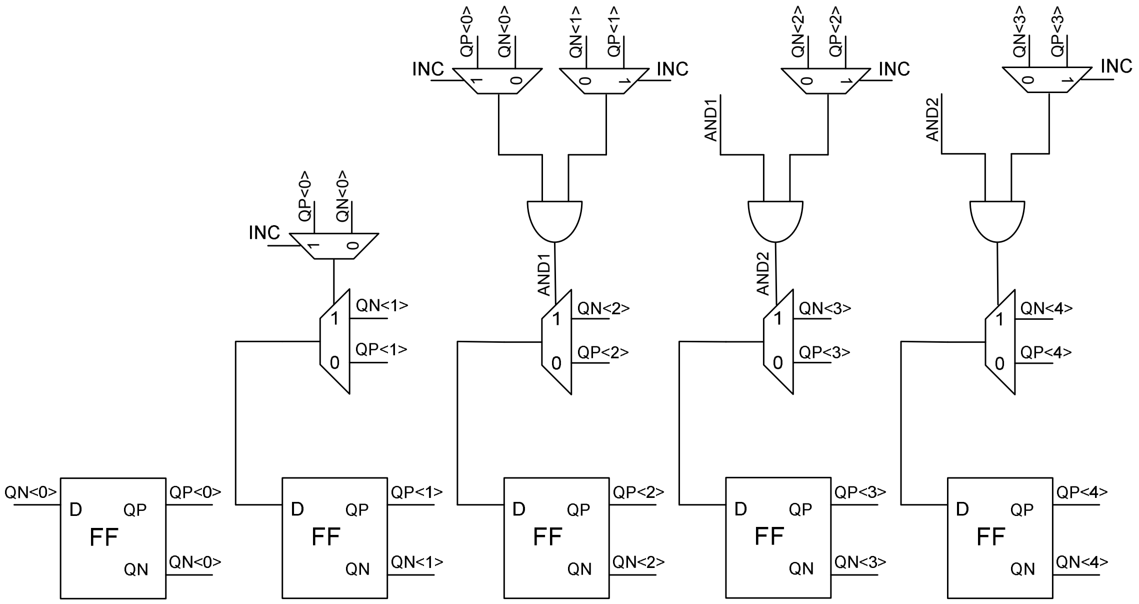


Figure 119: Up-down counter.

When the count reaches the maximum or minimum value (31 or 0) the INC signal must be inverted: we get this behaviour using a simple state machine.

The digital circuit, that implements this state machine, is made of a flip-flop and combinatorial logic. The detected count value are 30 and 1 instead of 31 and 0 because the INC signal changes at the clock edge; so there is a one clock cycle delay to invert the counter increment or decrement.

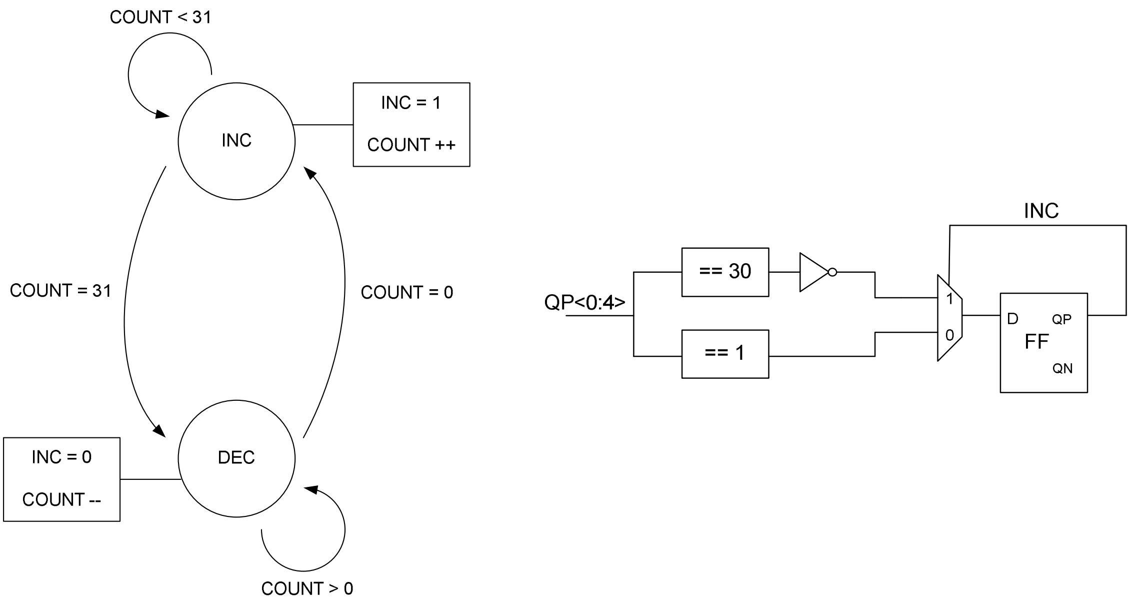


Figure 120: Inc-dec state machine.

The counter output must now be converted to generate the SEL, CAL and TH signals. The SEL<0:30> signal, that select the output slice, is simply a thermometric representation of a binary number in the range [0:31]. This signal is generated converting the counter output in a thermometric code. CAL<0:31> signal, that represents the slice under calibration, is generated by MUX<0:30> using a simple combinatorial logic:

$$CAL<0>=\overline{MUX<0>}$$
$$CAL<1>=\overline{MUX<1>}\cdot MUX<0>$$
$$CAL<2>=\overline{MUX<2>}\cdot MUX<1>$$
$$CAL<3>=\overline{MUX<3>}\cdot MUX<2>$$
|
|
$$CAL<31>=MUX<30>$$

Signal TH<0:29>, selecting the thresholds employed by each slice, is generated from MUX<0:30> and INC signals:

- If INC = 1: $TH<0:29>=\overline{MUX<29:0>}$;
- If INC = 0: $TH<0:29>=\overline{MUX<30:1>}$.

This is because the employed thresholds depend on whether the slice in calibration at the successive step of the algorithm is the preceding or the successive, respect to the actual one.

3.6 DIGITAL ENCODING

3.6.1 Introduction

The thermometer code must be converted into a binary format and in particular the two's complement one. The goal of the decoder is to minimize the conversion error, due to metastability and errors in the thermometer code.

We study two conversion methods: the first one employs a ROM based decoder and the second one employs a direct coding. Each of the two methods has some limitations; we use a theoretical evaluation to choose the best one [28].

3.6.2 ROM based encoder

In a ROM based encoder the thermometer code, composed by 2^n-1 bits (where n is the bit resolution of the ADC), is first provided to a decoder which produces 2^n output lines. With an ideal thermometer code only one line, corresponding to the 1 to 0 transition, is active [24].

Usually it is employed a decoder made of 3 input AND gate, with 1 inverted input, to detect a 11 to 0 transition; in this way, if there is a 1 between two 0 in the position shown in Figure 121, only one line is active. If there are two 1 between two 0, and a 3 input AND decoder is employed, two lines are contemporary active.

The output of this first decoder is employed to address a ROM encoder; Figure 122 shows an OR ROM, the name is due to the fact that, when more than one input line is active, the output is the logical OR of the two addressed code. The error rejection of a ROM decoder is the capability to give an output as correct as possible at the activation of more than one line.

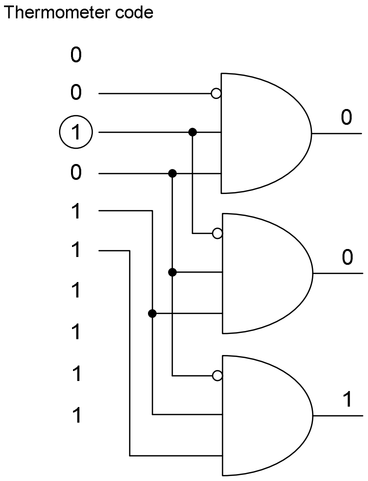


Figure 121: Bubble suppression.

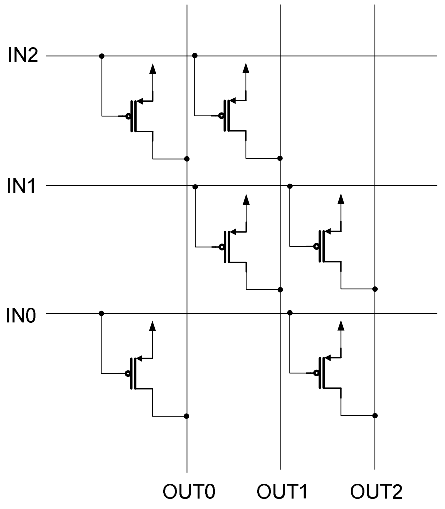


Figure 122: OR ROM.

A well known error reducing encoding method is the Gray code, which provides only one bit change in the encoder codes of adjacent addresses. Another common encoding method is the Half-Gray code [29]; this code divides the binary word in two parts, and each part is codified in Gray code. Table 4 shows an example of a 4 bit Gray and Half-Gray coding.

Dec	Binary				Gray				Half-Gray			
	B3	B2	B1	B0	G3	G2	G1	G0	Hg3	Hg2	Hg1	Hg0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1	0	0	1	1
3	0	0	1	1	0	0	1	0	0	0	1	0
4	0	1	0	0	0	1	1	0	0	1	0	0
5	0	1	0	1	0	1	1	1	0	1	0	1
6	0	1	1	0	0	1	0	1	0	1	1	1
7	0	1	1	1	0	1	0	0	0	1	1	0
8	1	0	0	0	1	1	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1	1	1	1	1
11	1	0	1	1	1	1	1	0	1	1	1	0
12	1	1	0	0	1	0	1	0	1	0	0	0
13	1	1	0	1	1	0	1	1	1	0	0	1
14	1	1	1	0	1	0	0	1	1	0	1	1
15	1	1	1	1	1	0	0	0	1	0	1	0

Table 4: 4 bit Gray and Half-Gray code.

In Gray or Half-Gray codes, the logical OR, between more than one codes from near addresses, doesn't degrade too much the output, while, with a binary coding, the error may be very big. For example the logical OR between the binary codes corresponding to the digital numbers 7 and 8, produce the number 15.

The main problem of a ROM based decoder is the metastability: in fact, if there is a metastable bit in the thermometer code, it is possible that different AND gates recognize the bit in different ways. So no lines could be active, with a possible big error in the ROM output respect to the input, Figure 123.

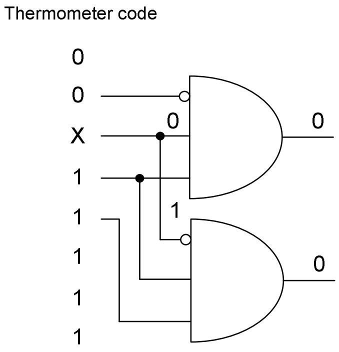


Figure 123: Metastable bit, none active line.

The output ROM code must be converted into a binary format.

We need a chain of n-1 XOR gates to convert a Gray code into an unsigned binary one; for example, with a four bit Gray code the conversion is (B3 is the MSB):

$$B3 = G3$$
$$B2 = B3 \otimes G2$$
$$B1 = B2 \otimes G1$$
$$B0 = B1 \otimes G0$$

3.6.3 Gray Direct Coding

Instead of using a ROM based encoder to convert the thermometer code into a binary one through a Gray or Half-Gray coding, we can use a direct thermo to Gray coding to reject errors. Moreover the metastability problem doesn't affect this method, unlike the ROM based encoder, because each thermometer bit is applied to only one gate, and the conversion is not based on the addressing [24].

To make the explanation easy, we start with the example of a 3 bit thermo to Gray direct coding. From Table 5 we can see that:

$$G2 = T3$$
$$G1 = T1 \otimes T5$$
$$G0 = T0 \otimes T2 + T4 \otimes T6$$

A logical XOR can be written:

$A \otimes B = \overline{A} \cdot B + A \cdot \overline{B}$ and so $T1 \otimes T5 = \overline{T1} \cdot T5 + T1 \cdot \overline{T5}$

In an ideal thermometer code is impossible that T1=0 and T5=1, so $T1 \otimes T5 = T1 \cdot \overline{T5}$. The same applies to T0,T2,T4,T6.

T6	T5	T4	T3	T2	T1	T0		G2	G1	G0		B2	B1	B0
0	0	0	0	0	0	0		0	0	0		0	0	0
0	0	0	0	0	0	1		0	0	1		0	0	1
0	0	0	0	0	1	1		0	1	1		0	1	0
0	0	0	0	1	1	1		0	1	0		0	1	1
0	0	0	1	1	1	1		1	1	0		1	0	0
0	0	1	1	1	1	1		1	1	1		1	0	1
0	1	1	1	1	1	1		1	0	1		1	1	0
1	1	1	1	1	1	1		1	0	0		1	1	1

Table 5: Thermo, Gray and binary code.

An easier code is:

$$G2 = T3$$
$$G1 = T1 \cdot \overline{T5}$$
$$G0 = T0 \cdot \overline{T2} + T4 \cdot \overline{T6}$$

It’s possible to further simplify applying the De Morgan law: $A + B = \overline{\overline{A} \cdot \overline{B}}$

$$G2 = T3$$
$$G1 = T1 \cdot \overline{T5}$$
$$G0 = \overline{\overline{T0 \cdot \overline{T2}} \cdot \overline{T4 \cdot \overline{T6}}}$$

In this way the encoding uses only AND and NAND gates. Thermometric to gray encoding for a 6 bit code is:

$$G5 = T31$$
$$G4 = T15\cdot\overline{T47}$$
$$G3 = T7\cdot\overline{T23} + T39\cdot\overline{T55}$$
$$G2 = T3\cdot\overline{T11} + T19\cdot\overline{T27} + T35\cdot\overline{T43} + T51\cdot\overline{T59}$$
$$G1 = T1\cdot\overline{T5} + T9\cdot\overline{T13} + T17\cdot\overline{T21} + T25\cdot\overline{T29} + T33\cdot\overline{T37} + T41\cdot\overline{T45} + T49\cdot\overline{T53} + T57\cdot\overline{T61}$$
$$G0 = T0\cdot\overline{T2} + T4\cdot\overline{T6} + T8\cdot\overline{T10} + T12\cdot\overline{T14} + T16\cdot\overline{T18} + T20\cdot\overline{T22} + T24\cdot\overline{T26} + T28\cdot\overline{T30} +$$
$$T32\cdot\overline{T34} + T36\cdot\overline{T38} + T40\cdot\overline{T42} + T44\cdot\overline{T46} + T48\cdot\overline{T50} + T52\cdot\overline{T54} + T56\cdot\overline{T58} + T60\cdot\overline{T62}$$

Applying De Morgan law:

$$G5 = T31$$
$$G4 = T15\cdot\overline{T47}$$
$$G3 = \overline{\overline{T7\cdot\overline{T23}}\cdot\overline{T39\cdot\overline{T55}}}$$
$$G2 = \overline{\overline{T3\cdot\overline{T11}}\cdot\overline{T19\cdot\overline{T27}}\cdot\overline{T35\cdot\overline{T43}}\cdot\overline{T51\cdot\overline{T59}}}$$
$$G1 = \overline{\overline{T1\cdot\overline{T5}}\cdot\overline{T9\cdot\overline{T13}}\cdot\overline{T17\cdot\overline{T21}}\cdot\overline{T25\cdot\overline{T29}}\cdot\overline{T33\cdot\overline{T37}}\cdot\overline{T41\cdot\overline{T45}}\cdot\overline{T49\cdot\overline{T53}}\cdot\overline{T57\cdot\overline{T61}}}$$
$$G0 = \overline{\overline{T0\cdot\overline{T2}}\cdot\overline{T4\cdot\overline{T6}}\cdot\overline{T8\cdot\overline{T10}}\cdot\overline{T12\cdot\overline{T14}}\cdot\overline{T16\cdot\overline{T18}}\cdot\overline{T20\cdot\overline{T22}}\cdot\overline{T24\cdot\overline{T26}}\cdot\overline{T28\cdot\overline{T30}}}$$
$$\overline{\overline{T32\cdot\overline{T34}}\cdot\overline{T36\cdot\overline{T38}}\cdot\overline{T40\cdot\overline{T42}}\cdot\overline{T44\cdot\overline{T46}}\cdot\overline{T48\cdot\overline{T50}}\cdot\overline{T52\cdot\overline{T54}}\cdot\overline{T56\cdot\overline{T58}}\cdot\overline{T60\cdot\overline{T62}}}$$

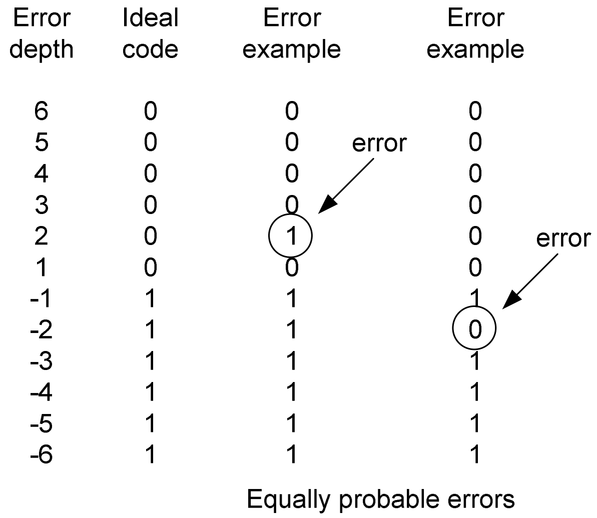
In order to binary convert the Gray code we can use the logic seen before:

$$B5 = G5$$
$$B4 = B5 \otimes G4$$
$$B3 = B4 \otimes G3$$
$$B2 = B3 \otimes G2$$
$$B1 = B2 \otimes G1$$
$$B0 = B1 \otimes G0$$

3.6.4 Errors metric

The first thing to take into account in a decoder study is the capability to reject input error and so we have to define an error metric. An error is defined as an inversion in a bit of the thermometer code respect to the ideal one. We must also evaluate the position and the number of these errors.

We define the error depth as shown in Figure 124. The depth is positive when the ideal code is 0 and negative when the ideal code is 1. In this figure we can see a single error at depth 2 and one at depth -2, respect to the ideal code; the two errors are also equally probable. It is also possible to have multiples errors; in this event the error topology is identified by the number of errors and their depths.



Error depth	Ideal code	Error example	Error example
6	0	0	0
5	0	0	0
4	0	0	0
3	0	0	0
2	0	1	0
1	0	0	0
-1	1	1	1
-2	1	1	0
-3	1	1	1
-4	1	1	1
-5	1	1	1
-6	1	1	1

Figure 124: Error examples.

In this study four errors topologies are considered: single error, two separated errors, two consecutive errors and three consecutive errors. These errors topologies are shown in Table 6, the errors examples are in equally probable pairs.

Depth	Ideal	Single error		2 separated errors		2 consecutive errors		3 consecutive errors	
6	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	1	0
3	0	0	0	0	0	1	0	1	0
2	0	1	0	1	0	1	0	1	0
1	0	0	0	0	0	0	0	0	0
-1	1	1	1	1	1	1	1	1	1
-2	1	1	0	1	0	1	0	1	0
-3	1	1	1	1	1	1	0	1	0
-4	1	1	1	1	0	1	1	1	0
-5	1	1	1	1	1	1	1	1	1
-6	1	1	1	1	1	1	1	1	1

Table 6: Errors Topologies.

Obviously the more probable errors are the single ones at small depth

3.6.5 Single error remover

A simple circuit that removes single error at any depth is employed to improve the error rejection. This circuit is a re-mapping of the input thermometric code. The single re-mapped thermometric bit is 1 if there are at least two 1 in the three adjacent bits, zero otherwise, as shown in Figure 125. The first and last thermometer bit are re-mapped with a simple OR and AND. In this way, both errors of 1 and of 0 are removed.

Applying the De-Morgan's law the logical function AB+BC+AC can be rewrite as: $\overline{\overline{AB}\cdot\overline{BC}\cdot\overline{AC}}$.

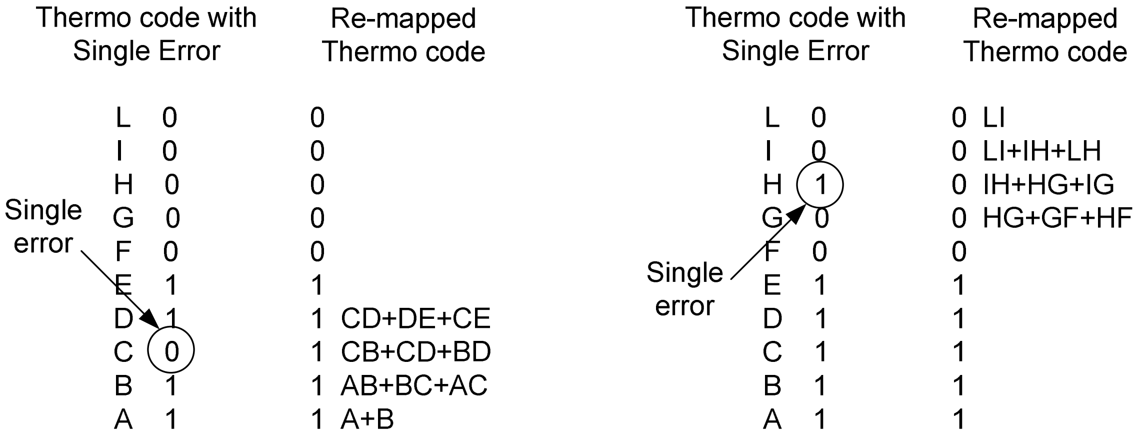


Figure 125: Re-mapped thermometric code.

3.6.6 Errors rejection evaluation

A Matlab program is employed to evaluate the error rejection performance of the considered thermometer to binary decoders. We perform this comparison through decoder models, developed for a 6 binary bit thermometric code (63 thermometric bits).

For each considered error topology and depth, the program generates all the thermometric codes corresponding to all the decimal values from 0 to 63, with the errors of the previous paragraph.

These thermometer codes are employed as inputs for the decoder models and the "Matlab program" computes the absolute value of the differences between the decoder model output and the decimal values, corresponding to the error free thermometer code. We call this difference "output error". For each depth and error topology we have 63 output errors.

There are two significant metrics to evaluate a decoder: the maximum output error and the mean output error. We perform a comparison between the considered decoding methods for different errors topologies: max error Figure 126 and mean error Figure 127. All the shown decoding methods are preceded by a single error remover.

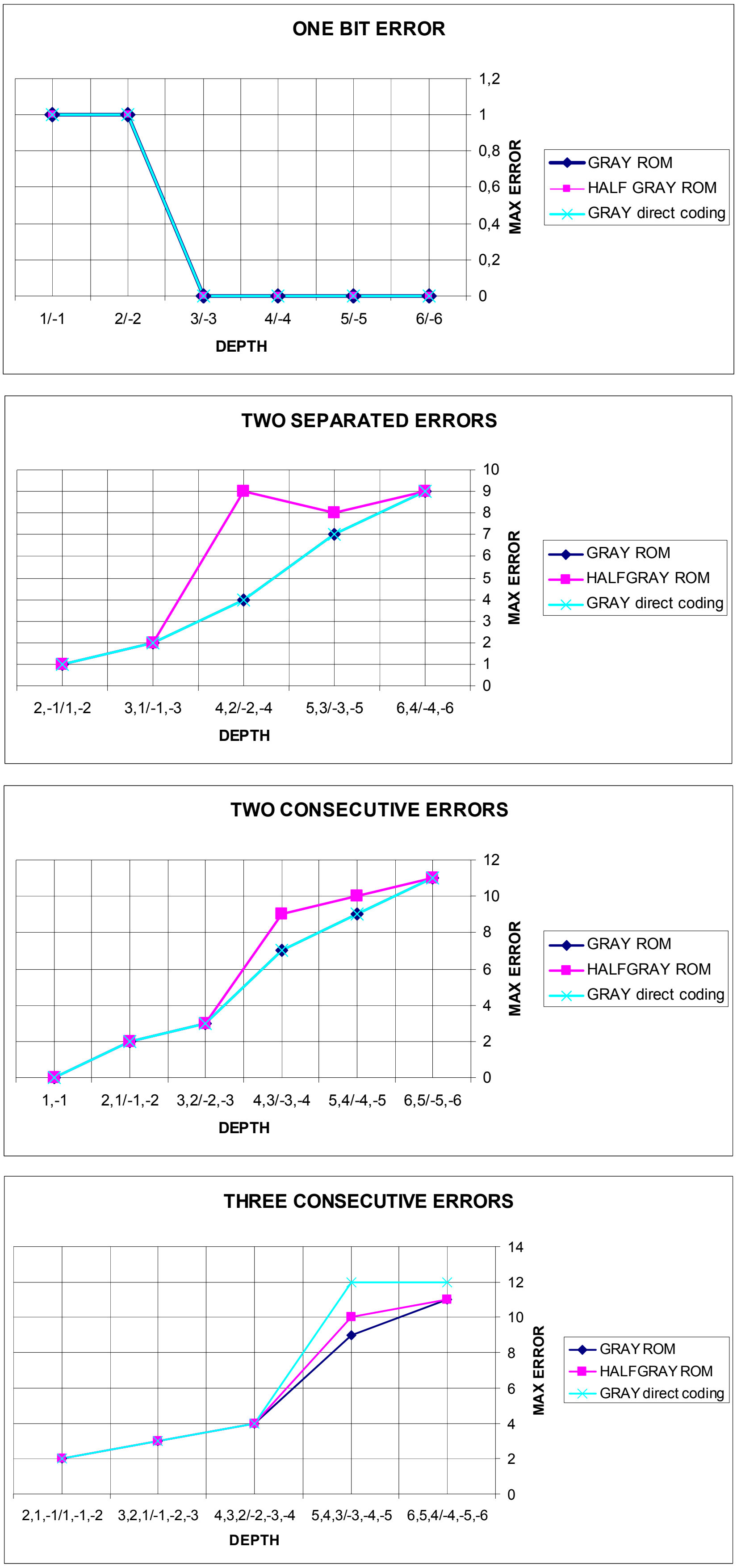


Figure 126: Maximum output error.

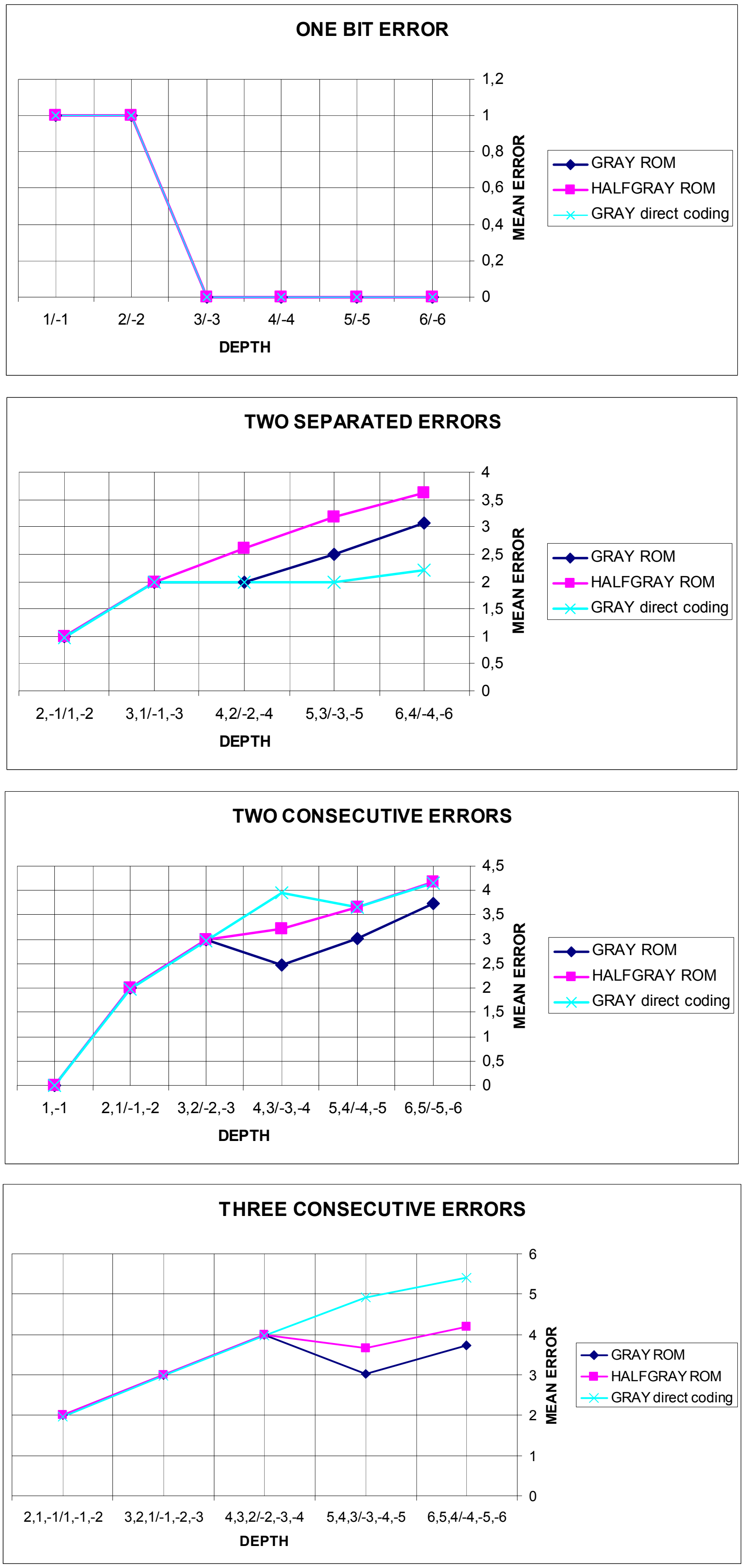
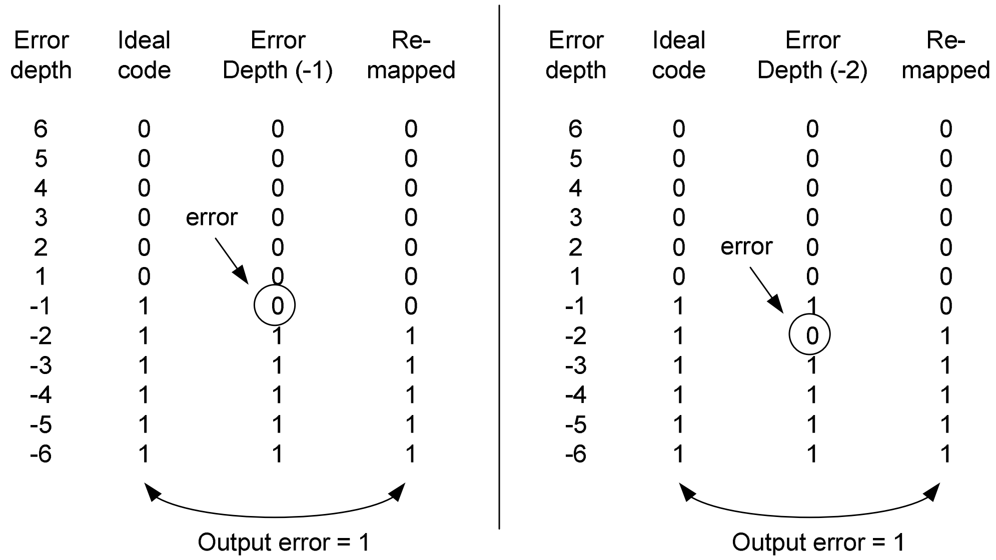


Figure 127: Mean output errors.

The charts show on the x axis the error depth, and on the y axis the maximum or mean error. The figures show that the three considered coding topologies have the same error rejection capability for “one bit error” because all the decoders are preceded by a single error remover. With “two separated errors” and “two consecutive errors”, GRAY ROM and direct coding are better than HALF GRAY ROM respect to maximum error. “Three consecutive errors” are very improbable and the error rejection capability is better with GRAY ROM decoder. The charts show that there is not a big difference between the considered codes.

The single error remover totally rejects single errors at any depth, except the ones at depth +-1 and +-2. These errors give an output error of 1, Figure 128.



Error depth	Ideal code	Error Depth (-1)	Re-mapped
6	0	0	0
5	0	0	0
4	0	0	0
3	0	0	0
2	0	0	0
1	0	0	0
-1	1	0	0
-2	1	1	1
-3	1	1	1
-4	1	1	1
-5	1	1	1
-6	1	1	1

Error depth	Ideal code	Error Depth (-2)	Re-mapped
6	0	0	0
5	0	0	0
4	0	0	0
3	0	0	0
2	0	0	0
1	0	0	0
-1	1	1	0
-2	1	0	1
-3	1	1	1
-4	1	1	1
-5	1	1	1
-6	1	1	1

Figure 128: Single error remover with single error at depth -1 or -2.

3.6.7 Digital encoder design

After the theoretical evaluations it was settled to implement the digital decoder using a single error remover, a thermo to GRAY direct coding circuit, followed by a Gray to binary converter. We have chosen GRAY direct coding mainly for its metastability insensitivity.

The flash ADC is made of 64 slices: two of these slices are simultaneously offset calibrated and 62 convert the input signal. The 64 slices are arranged in the layout like in Figure 129. The first 31 thermometer bits, after the multiplexer that select the outputs, are on one side and the other 31 on the other side. It is better to separately convert the two outputs group in a 5 bit code and subsequently recombine them using the sign bit. In this way only few wires must be brought around the chip. The theoretical evaluation of the previous paragraph is also valid for a 5 bit converter. The digital output must be in a two’s complemented one format and so the number is in the range [-31:31].

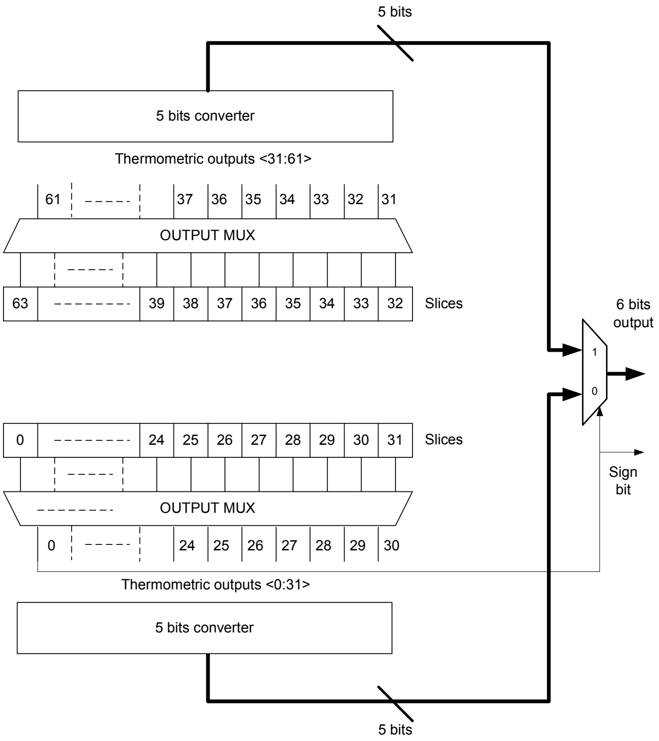
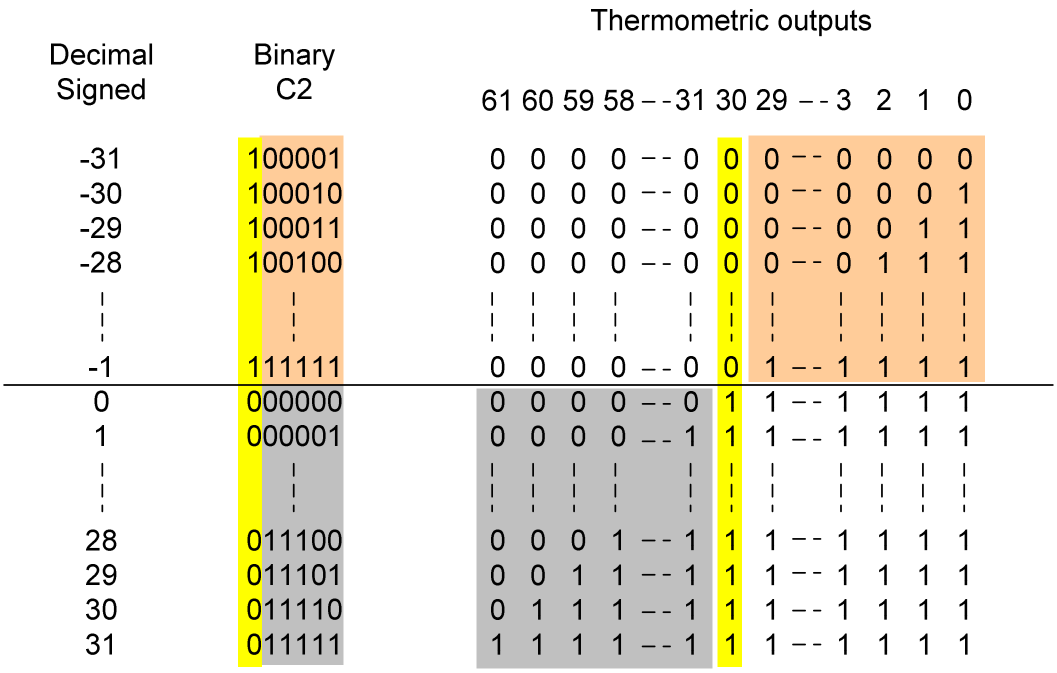


Figure 129: Layout slices arrangement.

Figure 130 shows the correspondence between the decimal signed representation, the binary two’s complement and the thermometer code.



Decimal Signed	Binary C2	61	60	59	58	--	31	30	29	--	3	2	1	0
-31	100001	0	0	0	0	--	0	0	0	--	0	0	0	0
-30	100010	0	0	0	0	--	0	0	0	--	0	0	0	1
-29	100011	0	0	0	0	--	0	0	0	--	0	0	1	1
-28	100100	0	0	0	0	--	0	0	0	--	0	1	1	1
¦	¦	¦	¦	¦	¦		¦	¦	¦		¦	¦	¦	¦
-1	111111	0	0	0	0	--	0	0	1	--	1	1	1	1
0	000000	0	0	0	0	--	0	1	1	--	1	1	1	1
1	000001	0	0	0	0	--	1	1	1	--	1	1	1	1
¦	¦	¦	¦	¦	¦		¦	¦	¦		¦	¦	¦	¦
28	011100	0	0	0	1	--	1	1	1	--	1	1	1	1
29	011101	0	0	1	1	--	1	1	1	--	1	1	1	1
30	011110	0	1	1	1	--	1	1	1	--	1	1	1	1
31	011111	1	1	1	1	--	1	1	1	--	1	1	1	1

Thermometric outputs

Figure 130: Correspondence between decimal, binary two's complement and thermometer code.

In order to convert the thermometer code into a two's complement binary representation, it's enough to convert the first 30 thermo bit [0:29] to 5-bit unsigned binary with an offset of 1, and those from 31 to 61 to normal 5-bit unsigned binary. Moreover the thirtieth bit is the complemented sign bit and is also employed to select which of the two 5-bit binary conversion must be employed.

Using the "Matlab model" to evaluate the error rejection of this new decoder, we can see that its capability is nearly equal to the 6-bit Gray direct coding with "single error remover" logic studied before. There are two single error remover, one for a half thermometric code and one for the other. The single error remover is also employed for the sign calculation.

Figure 131 shows the decoder block diagram. The circuit consists of two pipelines stage, working with a 2.5GHz clock. There are two equal decoders, one for each half-rate path. The mean power consumption is less then 5mW for the two paths.

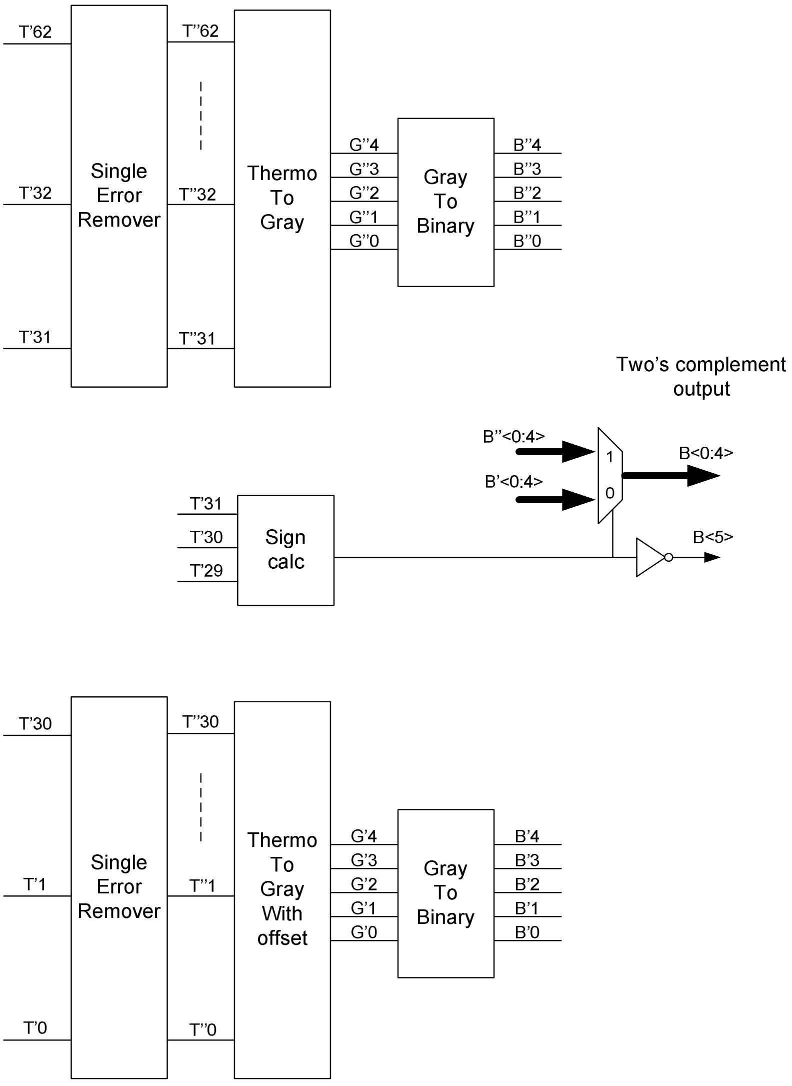


Figure 131: Decoder block diagram.

3.7 ADC SIMULATIONS RESULTS

This paragraph shows the simulation results coming from the schematic of the ADC. It is made of the buffer and track&hold, 64 slices, two of which are simultaneously offset calibrated, the digital encoding and the offset calibration algorithm.

In order to simulate the offset correction capability it is set an offset, referred at the slice input, of 15mV in opposite direction between adjacent slices. The offset calibration algorithm calibrates a slice among 32, and each slice is in calibration for a time of 7 integration steps. The resolution of the offset calibration, referred at the input of the slice, is approximately 1mV. In order to compensate an offset of 15 mV, 15 integration steps are required. Three calibration phases are required for each slice.

The slices, except the first and the last, are calibrated once every 32 calibration phases (25.6ns X 32 = 819.2ns), while the first and the last are calibrated every 64 (25.6ns X 64 = 1638.4ns). The calibration of all the slices requires approximately 5us.

3.7.1 Simulated DNL and INL

In order to measure the DNL and the INL before and after the offset correction, a periodic ramp is set as ADC input.

The amplitude of the ramp is the ADC full-scale minus one quantization step. The theoretical occurrence of each digital number is 20 and so the period of the ramp is 20*61*400ps =488ns, where 61 are the numbers of different digital coding and 400ps the clock period of each half rate path.

At the start time, DNL and INL are big because the offset is not calibrated; after approximately 5us, the offset is corrected and the DNL is less than 0.2 LSB, the INL is less than 0.35 LSB, Figure 132. This figure shows that there is, after the calibration, a little third order distortion in the input output transfer function and in the INL plot. This distortion arises from the buffer non linearity.

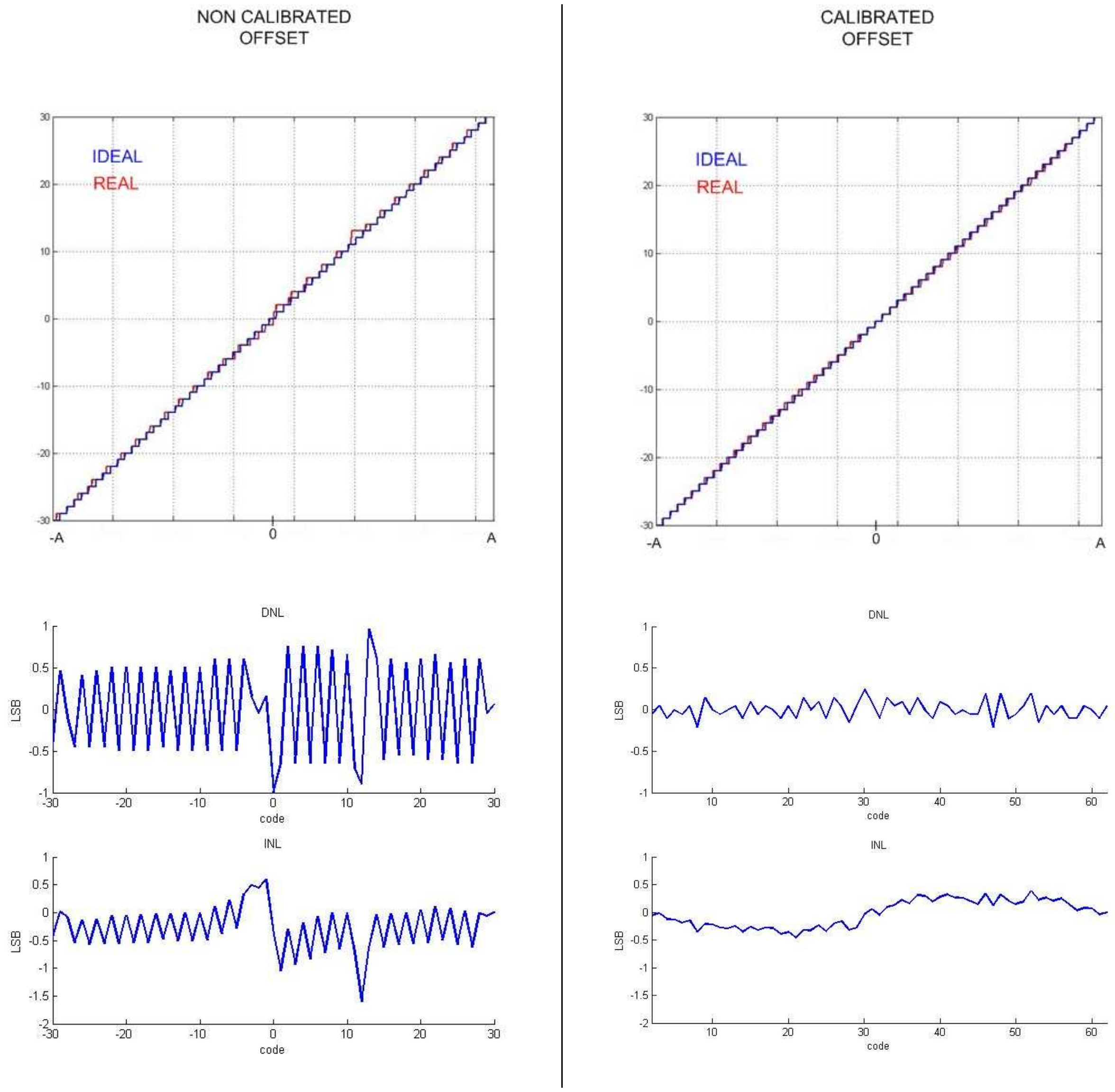


Figure 132: Input-output transfer function and DNL-INL plot before and after the offset calibration.

3.7.2 *Simulated ENOB*

Figure 133 shows the FFT of one path of the ADC output at time zero and after 5us with a signal at frequency close to the Nyquist one (2.33GHz). Each ADC path works with a half rate clock and so the input signal is under-sampled and the tone is folded at a frequency equal to 166MHz.

At the start time the ENOB is 4.19 bits and after 5us, when the offset is calibrated, the ENOB is 5.86 bits.

In these simulations the clock non-idealities (jitter and skew) and the thermal noise are not included.

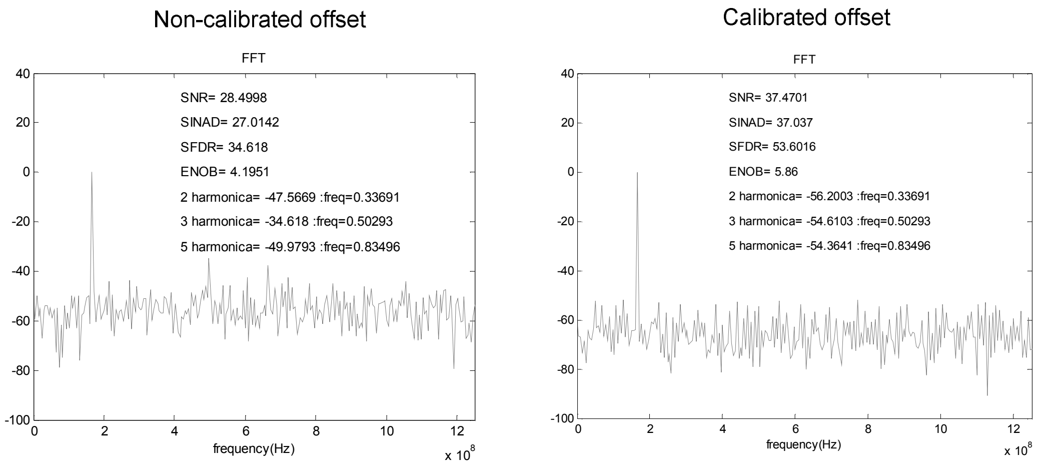


Figure 133: FFT before and after the offset calibration.

3.8 ADC LAYOUT

Figure 134 shows the flash ADC layout. The circuits described in this thesis are indicated. The layout occupies 530×500μm^2.

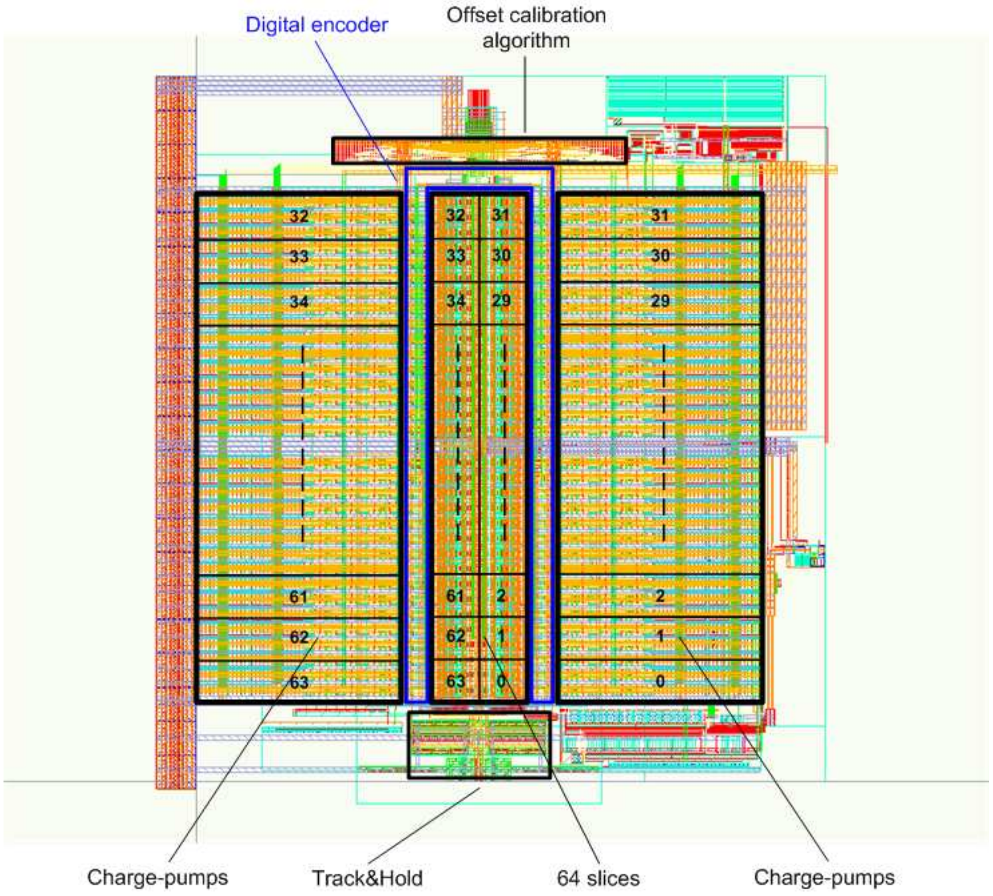


Figure 134: ADC layout.

3.9 LITERATURE COMPARISON

The designed flash ADC is not yet integrated and so the results are not measured but simulated. The total power consumption is approximately 130mW: the core, composed by the slices and the local clock signals generation, consumes 96mW, the digital logic consumes 5mW, the track&hold consumes 15mW and the clock buffer, which is not shown in this thesis, consumes 15mW.

The ENOB, simulated with a post layout simulation, is 5.6 with a Nyquist frequency signal; the DNL is less than 0.5LSB and the INL is less then 0.4LSB. ENOB is so high because there are two sources of ENOB reduction that, in simulation, are not considered: the thermal noise and the clock non-idealities (jitter and skew).

A possible figure of merit (FOM) used in many articles to compare ADCs with different sampling frequency and different number of bit is the following [27]:

3.17 $$FOM = \frac{Power}{2^{ENOB} \cdot f_s}$$

In order to correctly compare this work with the flash ADC proposed in literature we define a figure of merit not based on ENOB but based only on the power consumption and on the sampling rate. This FOM is defined as:

3.18 $$FOM = \frac{Power}{f_s} \quad [Energy\ per\ conversion\ step]$$

With this FOM is possible to compare only ADCs with the same number of bit. Table 7 shows a comparison with the state of the art of the flash ADCs with a sampling frequency ≥ 3GHz and a number of bit equal to 6.

Evaluating the flash ADC performances with the shown FOM, the presented work is comparable with the best in classes high frequency 6bit flash ADC, within author’s knowledge.

The ADC described in the article of Michael Choi et al. has a very high FOM compared to the ADC in the others article. The reason is that this flash ADC hasn’t an offset calibration procedure and, to prevent ENOB reduction, the circuits are designed to have negligible offset. High power consumption is necessary to achive this task.

	Sampling Frequency [fs]	Number of bit	ENOB (input signal frequency)	DNL / INL [LSB]	Power consumption	FOM [Joules]
Michael Choi et al. 2008 [30]	5GS/s	6	≈5 @ Nyquist	0.6 / 0.7	320mW	64 $E10^{-12}$
Kazuaki Deguchi et al. 2008 [31]	3.5GS/s	6	4.89 @ Nyquist	0.5 / 1	98mW	28 $E10^{-12}$
Masashi Kijima et al. 2009 [27]	3 GS/s	6	5.8 @ 10MHz	0.2 / 0.2	90mW	30 $E10^{-12}$
Chun-Cheng Huang, et al. 2010 [32]	16 GS/s	6	4.36 @ 3GHz	-0.5 – 0.6 / -0.4 – 0.7	435mW	27 $E10^{-12}$
This work* *simulated results	5GS/s	6	5.6 @ Nyquist	0.5 / 0.4	130mW	26 $E10^{-12}$

Table 7: Comparison with the state of the art.

3.10 SUMMARY

In this chapter we have described the designed 6bit, 5 GS/s flash ADC. First we defined the specifications related to each ADC block; after that we have shown a comparison between two different architectures. The two architectures are called respectively full-rate and half-rate. The full-rate slice is composed by: a preamplifier, the cascade of two reset latches and finally CMOS flip-flops resampling the signal. All the slice building blocks work at the sampling frequency. The half-rate slice is composed by: a preamplifier working at the sampling frequency, two reset latches which alternatively regenerate the preamplifier output at half the sampling frequency and CMOS flip-flops. The half-rate architecture has been chosen because of its lower power consumption.

We have also shown the digital encoder, which converts the thermometric code into a binary one, and the digital logic implementing the offset calibration algorithm. Finally we have shown the simulation results and the literature comparison. The flash ADC tape-out is planned for mid-november 2010.

Conclusions

In this Ph. D. thesis, the digital circuits for an analog high speed SerDes receiver and a high frequency flash analog to digital converter (ADC), have been presented. The ADC can be used as front end of a digital high speed SerDes receiver.

The work starts from an overview of high speed serial communication. In particular we have described the most utilized data modulations, the serial link standards focussed both on copper and optical links, the main system characterization metrics and the sources of non-idealities limiting the system performances. Finally we have analyzed the equalization and clock-recovery techniques both for analog and digital receivers.

After this introductive chapter we have analyzed a 12Gb/s analog receiver. This receiver is based on a linear equalizer and a continuous time DFE (CTDFE) for the data equalization and on a bang-bang CDR for the clock recovery. This second chapter deals with the digital circuits employed in this receiver: the CDR, the circuits implementing the algorithm to auto-adapt the CTDFE and some auxiliary circuits.

Measurements on the integrated receiver prove 12Gb/s error-free operation with 39dB backplane losses. This is the best equalization capability among the best-in-class serial interface receivers published in the last three years at ISSCC.

In the last part of this thesis, the designed 6bit 5GS/s flash ADC has been presented. After the definition of the specifications related to each block, we have evaluated two architectures (full-rate and half-rate). We have implemented the half-rate architecture because of its lower power consumption. Finally the designed thermometric to binary decoder and the digital algorithm for the offset calibration have been described. The flash ADC tape-out is planned for mid-november 2010 but the simulation results have proven good performances in terms of energy per conversion step.

References

[1] B. Sklar, Digital Communications, New jersey: Prentice-Hall, 1988.

[2] T. Plakert, “A review of current standards activities for high speed physical layers”, IEEE International Database Engineering & Application Symposium, 2005.

[3] B. Razavi, ”Design of integrated Circuits for Optical Communications”, McGraw-Hill Professional, New York, 2002.

[4] J. F. Buckwalter and A. Hajimiri, “Cancellation of Crosstalk-Induced Jitter”, Journal of Solid-State Circuits, vol.41, n°3, March, 2006.

[5] Robert W. Kembel, “Fibre Channel, A comprehensive introduction”, Northwest Learning Associates, 1998.

[6] IEEE Std 802.3aq™-2006, Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-LRM.

[7] Gary Breed, “Analyzing Signals Using the Eye Diagram”, High Frequency Electronics, November 2005.

[8] Johnnie Hancock, “Jitter: Understanding it, Measuring It, Eliminating It”, High Frequency Electronics, April 2004.

[9] J. G. Proakis, Digital Communications, New York:McGraw-Hill, 1983.

[10] Vladimir Stojanović, CHANNEL-LIMITED HIGH-SPEED LINKS: MODELING, ANALYSIS AND DESIGN, dissertation for the degree of doctor of Philosophy, September 2004.

[11] Matt Park and Michael Perrott, Behavioral Simulation of Decision Feedback Equalizer Architectures Using CppSim, October 31, 2005.

[12] Cathy Ye Liu, Joe Caroselli, Comparison of Signaling and Equalization Schemes in High Speed SerDes (10-25Gbps), DesignCon 2007.

[13] Soumya Chandramouli, Franklin Bien, Hyoungsoo Kim, Edward Gebara, and Joy Laskar, “A 10-Gb/sec Unclocked Current-Mode Logic (CML) Analog Decision-Feedback Equalizer (ADFE) in 0.18-µm CMOS”, Georgia Electronic Design Center.

[14] A.J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Trans. Inform. Theory, vol. IT-13, pp. 260-269, Apr. 1967.

[15] J.F. Hayes, “The Viterbi algorithm applied to digital data transmission,” IEEE Commun. Mag., vol. 13, no. 2,pp. 15-20, Mar. 1975.

[16] J. D. H. Alexander, “Clock Recovery from Random Binary Signals”, Electronics Letters, vol. 11, pp. 541-542, Oct. 1975.

[17] Kurt H. Mueller and Markus Muller, “Timing recovery in digital synchronous data receivers”, IEEE TRANSACTION ON COMMUNICATIONS, VOL. COM-24, No 5, MAY 1976.

[18] Johnnie Hancock and Steve Draving, “Jitter: Understanding it, Measuring It, Eliminating It, Part 2: Jitter Measurements”, High Frequency Electronics, May 2004.

[19] The BERTScopeTM Differential ISI Board Precision ISI Test Accessory, 2008, SyntheSys Research.

[20] K. Fukuda1 et al., “An 8Gb/s Transceiver with 3X-Oversampling 2-Threshold Eye-Tracking CDR Circuit for -36.8dB-loss Backplane”, ISSCC 2008 / SESSION 5 / HIGH-SPEED TRANSCEIVERS / 5.1.

[21] John F. Bulzacchelli et al., “A 78mW 11.1Gb/s 5-Tap DFE Receiver with Digitally Calibrated Current-Integrating Summers in 65nm CMOS”, ISSCC 2009 / SESSION 21 / 10Gb/s-TO-40Gb/s TRANSMITTERS AND RECEIVERS/ 21.6.

[22] Hideyuki Sugita et al., “A 16Gb/s 1st-Tap FFE and 3-Tap DFE in 90nm CMOS”, ISSCC 2010 / SESSION 8 / HIGH-SPEED WIRELINE TRANSCEIVERS / 8.4.

[23] Massimo Pozzoni et al., “A 12Gb/s 39dB Loss-Recovery Unclocked-DFE Receiver with Bi-dimensional Equalization”, ISSCC 2010 / SESSION 8 / HIGH-SPEED WIRELINE TRANSCEIVERS / 8.5.

[24] B. Razavi, ”Principles of Data Conversion Systems Design”, IEEE Circuits and System Society, sponsor, 1994.

[25] Franco Maloberti, “Data Converters”, Springers, 2007.

[26] Rudy VanDePlassche, “CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters”, Hardcover, Feb 19, 2010.

[27] Masashi Kijima et al., “A 6b 3GS/s Flash ADC with Background Calibration”, IEEE 2009 Custom Intergrated Circuits Conference (CICC).

[28] Clemenz L. Portmann, “Power-Efficient etastability Error Reduction in CMOS Flash A/D Converters”, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 31, NO. 8, AUGUST 1996.

[29] Hooman Reyhani, “Half-gray digital encoding method and circuitry”, United States Patent, May 27, 1997.

[30] Michael Choi et al., “A 6-bit 5-GSample/s Nyquist A/D Converter in 65nm CMOS”, 2008 Symposium on VLSI Circuits Digest of Technical Papers.

[31] Kazuaki Deguchi et al., “A 6-bit 3.5-GS/s 0.9-V 98-mW Flash ADC in 90-nm CMOS”, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 43, NO. 10, OCTOBER 2008.

[32] Chun-Cheng Huang et al., “A CMOS 6-Bit 16-GS/s Time-Interleaved ADC with Digital Background Calibration”, 2010 Symposium on VLSI Circuits/Technical Digest of Technical Papers.