UNIVERSITÄT BONN
Physikalisches Institut

Development of high speed integrated circuit for very high resolution timing measurements

von
Christian Mester

A multi-channel high-precision low-power time-to-digital converter application specific integrated circuit for high energy physics applications has been designed and implemented in a 130 nm CMOS process. To reach a target resolution of 24.4 ps, a novel delay element has been conceived. This nominal resolution has been experimentally verified with a prototype, with a minimum resolution of 19 ps. To further improve the resolution, a new interpolation scheme has been described.
The ASIC has been designed to use a reference clock with the LHC bunch crossing frequency of 40 MHz and generate all required timing signals internally, to ease to use within the framework of an LHC upgrade. Special care has been taken to minimise the power consumption.

Post address:
Nussallee 12
53115 Bonn
Germany

BONN-IR-2009-09
Bonn University
October 2009
ISSN-0172-8741

UNIVERSITÄT BONN
Physikalisches Institut

Development of high speed integrated circuit for very high resolution timing measurements

von
Christian Mester

Dieser Forschungsbericht wurde als Dissertation von der Mathematisch - Naturwissenschaftlichen Fakultät der Universität Bonn angenommen.

Angenommen am:	22. Oktober 2009
Referent:	Dr. Paulo Rodrigues S. Moreira
Korreferent:	Prof. Dr. Norbert Wermes

Development of

high speed integrated circuit for

very high resolution timing measurements

Disseration

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Christian Mester

aus

Köln

Bonn 2009

Contents

1 Introduction

In this thesis, the development of a Time-to-Digital Converter (TDC) Application Specific Integrated Circuit (ASIC) is described. The work has been carried out at CERN[†] in Geneva, Switzerland. The device is targeted at applications in the domain of High Energy Physics (HEP) .

1.1 Structure of the thesis

The thesis is divided into 3 main parts:

- Introduction to TDCs in general and a brief description of a selected HEP application example (chapters 2 and 3),
- design of a new TDC and implementation of a prototype (chapters 4 to 7) and
- experimental results (chapter 8).

First, an example of an HEP experiment using TDCs is presented (chapter 2). It gives an overview of the requirements of a large scale high precision experiment. A very high resolution must be achieved on a large number of channels while the power consumption has to be minimised to keep the material budget low.

To compare different possible TDC architectures, a set of figures of merit is needed. As they have to be tailored to TDCs, they are developed in the course of an introduction to TDCs (chapter 3), including a comparison to Analogue-to-Digital Converters, which are more commonly known. A TDC performs a discretisation of signals, which by principle implies a loss of information. The most basic performance metrics are

[†]European Organization for Nuclear Research/Organisation européenne pour la recherche nucléaire

therefore the smallest and the biggest time difference that can be resolved unambiguously. For any implementation of this concept, imperfections degrade the performance of the TDC, causing non-linearity, jitter and data rate limitations. When TDCs are used in a system, further properties are important, such as the number of chips required to interface a given number of channels, the power dissipation introduced by the TDC. On the system-level, the time-to-digital conversion is only a part of the required functionality. A TDC ASIC can include additional functionality required on a system-level.

Given this set of metrics and the knowledge about present applications of TDCs in the HEP domain, it is possible to specify the requirements for the TDC to be developed. The performance of a TDC is also determined by the data flow architecture. The two main TDC architectures, clock driven and data driven, are introduced. Both architectures have advantages and drawbacks. The data driven architecture is chosen for the new TDC, as it is found to be most appropriate for HEP applications (chapter 4).

The precision of a TDC timing measurement is set by the time base (chapter 5). For low power operation of a multi-channel TDC, it is convenient to have one global time base per chip that provides the time information to all channels. In this way, the time base resources are shared minimising the global power consumption and consequently the power consumption per channel. Different time base architectures fulfil this requirement. In a comparison, a DLL based time base with a counter for dynamic range extension has been found most appropriate.

The choice of the time base (chapter 5) fixes the characteristics of the TDC that are directly related to the time-to-digital conversion within the limits given by the data flow architecture (chapter 4). Not of smaller importance, other parts of the chip decide which data rates can be handled and how other parts of a complete detector system can interface with the TDC chip.

A very powerful concept for measuring very high rate events when only little readout bandwidth is available is triggering. In HEP experiments, the total rate of events is generally orders of magnitude higher than the rate of events that are worth being analysed in detail. Discarding uninteresting events inside the TDC ASIC reduces the required readout bandwidth substantially (chapter 6). All these considerations are on a high level of abstraction. Most of them are conceptual, while some require simulations.

Other characteristics, especially the timing precision, are strongly affected by parasitics, which depend not only on the gate-level implementation of the circuit, but even on the actual layout, the process technology, temperature variations and supply voltage. As the delay of wires can be in the order of the TDC resolution, timing critical parts have to be designed in a full-custom way. Post-layout simulations can give an idea of the circuit properties. A prototype is required to fully characterise the timing performance. Disregarding side effects such as cross-talk, power supply and ground variations and increase of the ASIC temperature, synthesisable features such as large buffers don't influence the timing precision. They can be simulated with sufficient precision and can be generated using well established synthesis tools. The prototype contains the strict minimum of circuitry necessary for the evaluation of the timing part. It contains a Phase-Locked Loop (PLL) for multiplying the 40 MHz input reference clock up to 1.28 GHz as required by the following stage. A Delay-Locked Loop (DLL) provides 32 phase-shifted versions of the 1.28 GHz clock to the hit registers of the channels. The TDC bin size is defined by the delay corresponding to the phase shift between two successive DLL outputs. Using a newly developed delay element, a bin size of 24.4 ps has been achieved. Each channel consists of a bank of registers that store the state of all DLL outputs, called 'timestamp', at the arrival of a signal on the hit inputs. The timestamp stored in the hit registers is read out using a shift register. Another shift register is used for configuration purposes. Test outputs are used to analyse important signals. Possible sources of errors such as non-linearities, noise and jitter are identified (chapter 7).

Different tests for prototype evaluation are presented. Experiments carried out with the prototype confirm the simulations. A bin size of 19 ps has been reached, with an RMS integral non-linearity of 4 ps and an RMS integral non-linearity of 5 ps. A power dissipation of 8 mW per channel has been observed (chapter 8).

2 A Short Introduction to Time Measurements in HEP Experiments

2.1 Introduction

The HPTDC, the previous version of a CERN-developed TDC, has been used in many different applications, including the LHC experiments ALICE [ALI02, BGG+05], ATLAS [AEF+04], CMS [Par08] and LHCb [LHC01]. The ALICE Time of Flight detector is described as case example, focussing on the timing measurement. NA62, also known as P326 and NA48/3, is a fixed target experiment at CERN's SPS accelerator. It contains the GigaTracker, for which a new front-end chip containing on-chip TDCs based on the TDC130 design is under development [Cec07].

The LHC is the last accelerator of a chain (fig. 2.1) at CERN. Its centre of mass energy is 14 TeV for proton-proton collisions ([BCL+04], p. 21) and 1.15 PeV for $^{208}Pb^{82+}$-$^{208}Pb^{82+}$ collisions ([BCL+04], p. 531).

2.2 Case Study: ALICE Time of Flight Detector

ALICE† (fig. 2.2) is an experiment at LHC. It is designed to observe heavy-ion collisions such as Pb-Pb, and study the plasma of deconfined quarks and gluons which

†A Large Ion Collider Experiment

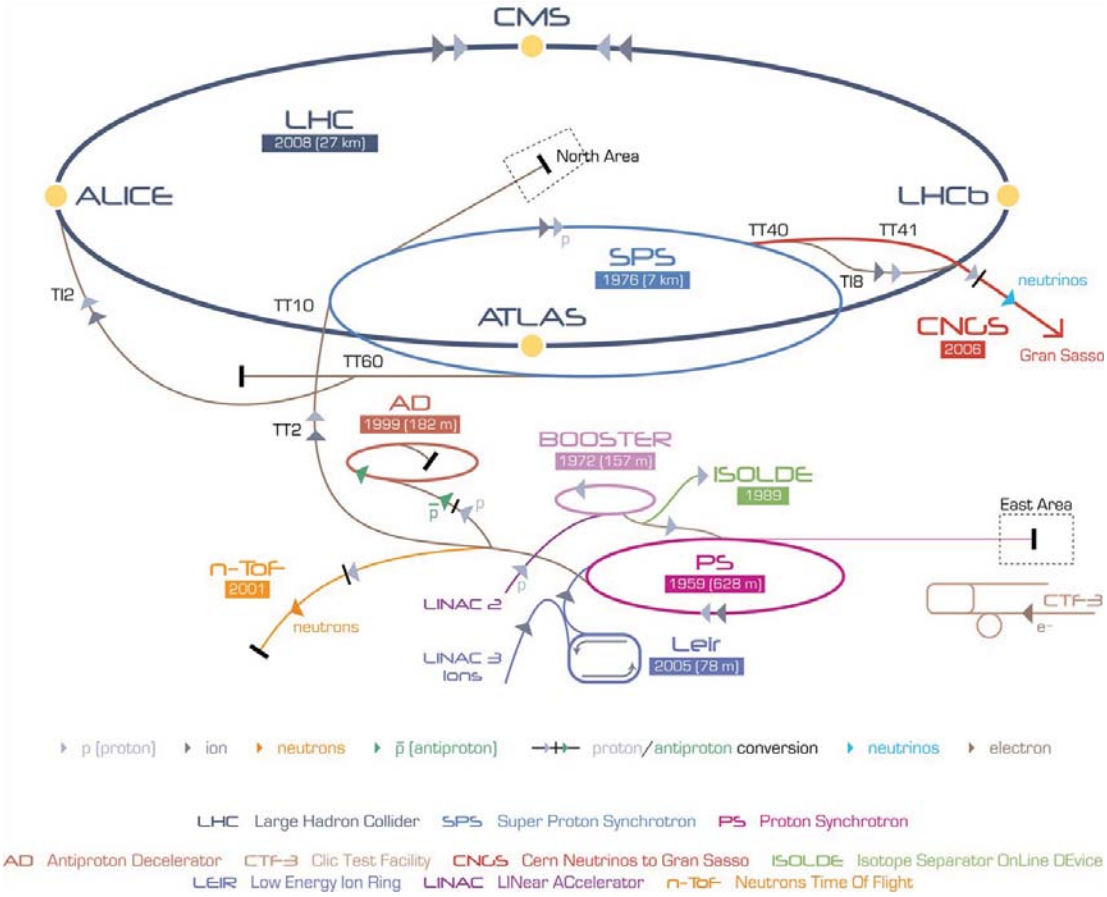


Figure 2.1: The CERN accelerator complex [Lef06]

is predicted by quantum chromodynamics [ALI95].

ALICE consists of different sub-detectors which are optimised for observing different particles emerging from the interaction point. The detectors are placed in a magnetic field that bends the trajectory of charged particles to simplify their identification. A Time Of Flight (TOF) detector for Particle Identification (PID) is installed in one of the outer layers. This detector uses a large number of HPTDCs in the 24.4 ps mode.

The TOF is used to distinguish pions, kaons and protons by measuring the time it takes them to reach the detector after emerging from the interaction point. For efficient

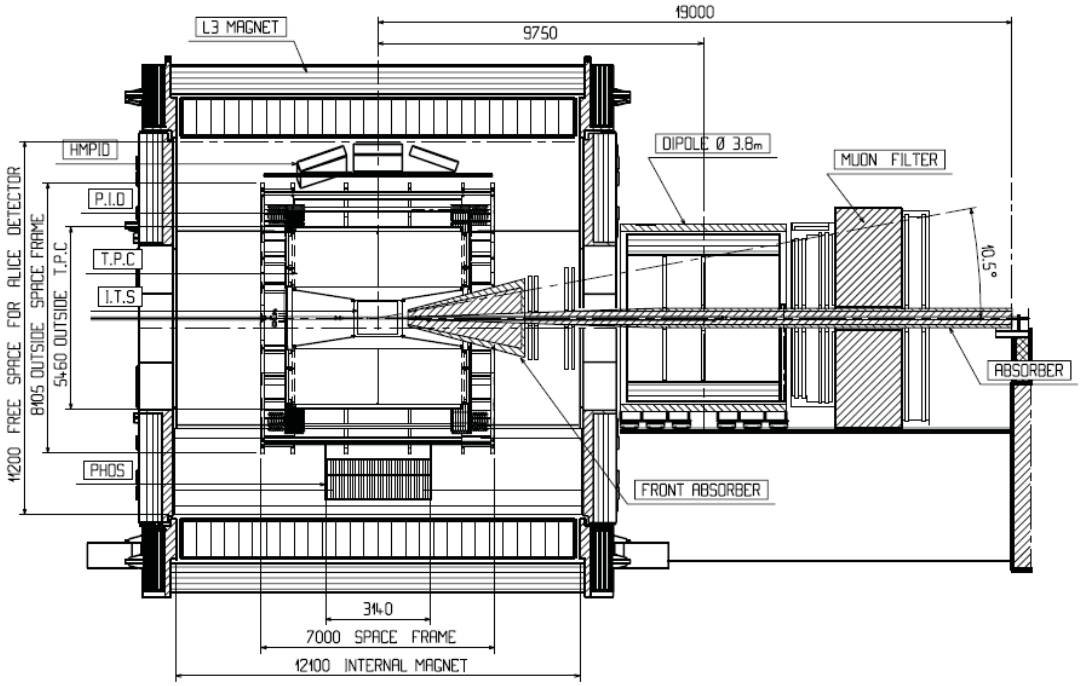


Figure 2.2: Longitudinal view of the ALICE detector [ALI95]

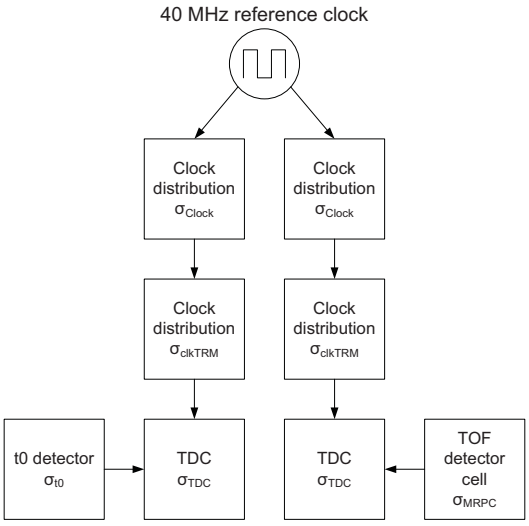


Figure 2.3: Signal paths of clock, TOF and t0 detector signal

separation of the signals of those particles, the time of flight needs to be measured with high precision to reveal the difference between the time a pion, kaon or proton takes. Studies have shown a relation between the efficiency of the identification and the resolution of the TOF time measurement [BKP+98]. In the higher momentum range of the expected particles, an overall resolution of 100 ps yields a considerably better result than e.g. 200 ps. An overall time resolution of 100 ps guarantees a 3σ separation of kaons and pions for momenta up to 2.1 $\frac{\text{GeV}}{\text{c}}$. With a resolution of 150 ps, momenta higher than 1.7 $\frac{\text{GeV}}{\text{c}}$ are not covered, for a resolution of 200 ps, the momentum limit is 1.4 $\frac{\text{GeV}}{\text{c}}$ [ALI95]. Several sub-systems of the detector contribute uncertainties to the overall resolution σ_{tot}. The particle's time of flight is calculated as the difference of the time the particle strikes the TOF detector and the time t_0 of the interaction, as captured by the t0 detector. The TOF detector consists of several modules, placed at the surface of a cylinder with a radius of 3.5 m and a length of 7 m. These dimensions render the traditional solution of routing the t_0 signal to all channels impractical. Therefore, the 40 MHz reference clock, which has to be distributed to all the system anyway, is used as a reference for the measurements. The time a particle arrives to a TOF detector channel is measured individually. Later on, after time to digital conversion, t_0 is subtracted. Fig. 2.3 shows the described signal paths. Each step introduces an additional uncertainty. Due to the large volume the system is spread across, they are considered uncorrelated and can be added:

$$\sigma_{\text{tot}}^2 = \sigma_{\text{t0}}^2 + \sigma_{\text{MRPC}}^2 + 2 \cdot \sigma_{\text{TDC}}^2 + 2 \cdot \sigma_{\text{clkTRM}}^2 + \sigma_{\text{Clock}}^2$$

assuming for simplicity that the uncertainties of the TDCs and the clock distribution network in the TDC readout module (TRM) are the same in both of the independent signal paths. σ_{MRPC} represents the uncertainty of the Multi Resistive Plate Chamber that has been chosen as detector for the TOF.

Table 2.1 shows the estimated time resolution of the whole ALICE TOF system [ALI02]. An RMS resolution of 25 ps translates into a bin size of 87 ps, 50 ps into 173 ps (see appendix A.1 for the calculation).

	Average time resolution in ps	Maximum in ps
σ_{t0}	50	50
σ_{MRPC}	50	80
σ_{TDC}	25	50
σ_{clkTRM}	15	15
σ_{Clock}	10	10
σ_{tot}	82	120

Table 2.1: Time resolution estimations for the ALICE TOF system [ALI02]. Note that the values given here are RMS values. The TDC130's nominal RMS resolution is 7 ps on 32 channels. The smallest RMS resolution the HPTDC can reach in 32 channel mode is 28 ps.

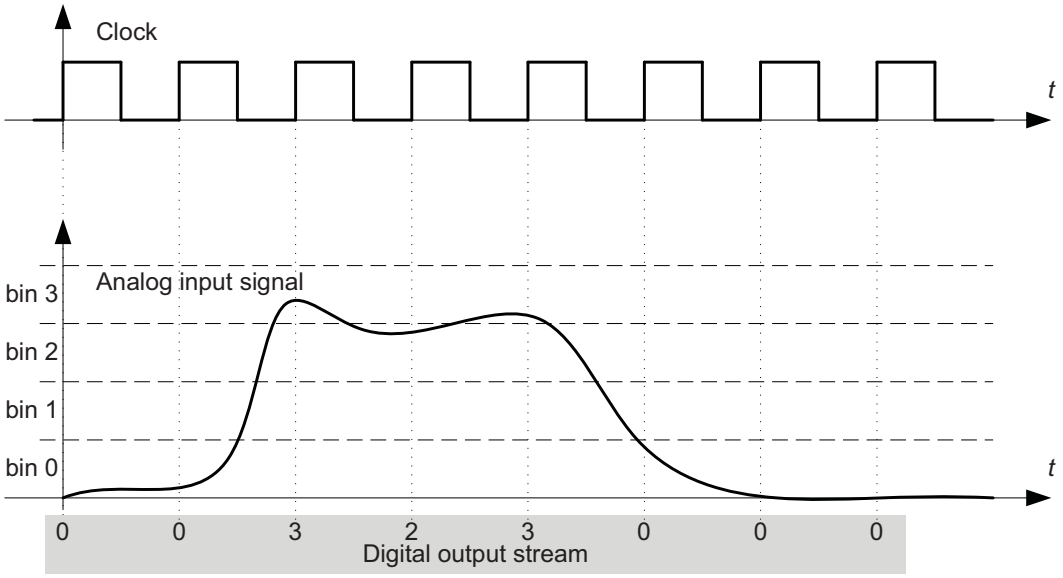
3 Basic Principles of a TDC

With increasing capabilities of digital data processing hardware such as FPGAs[†] and microprocessors, it has become common to analyse experimental data in the digital and discrete time domain. However, measurements of physical quantities usually take place in the analogue and continuous time domain. Conversion circuits are required to transform analogue signals such as a voltage into digital information and to perform a discretisation both of time and amplitude. Analogue-to-Digital Converters (ADCs) are used to sample analogue signals and convert the amplitude of the signal, most frequently an analogue voltage, into its discrete digital representation. Time-to-Digital Converters digitise the time their input signal crosses a given threshold (fig. 3.1). Since ADCs are more commonly known than TDCs, I will frequently use ADC related examples.

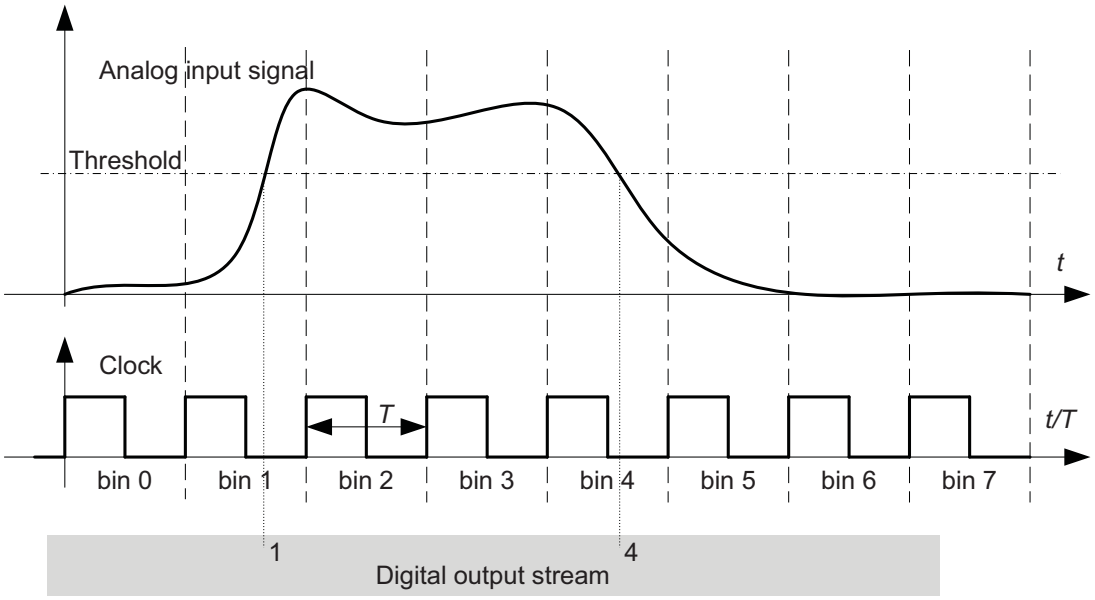
Unfortunately, conversion from continuous to discrete introduces a loss of information, even if an ideal converter is assumed. Imperfections of real converters introduce further errors. Therefore, only an approximation of the original signal can be reconstructed, based on some assumptions. An example is the Nyquist criterion: in order to perfectly reconstruct an analogue signal, the sampling frequency of an ADC has to be at least twice the bandwidth of the input signal. For a TDC, the spacing of two successive hits must not be smaller than a certain minimum, the double pulse resolution. If hits are closer, data loss occurs[‡].

[†]Field Programmable Gate Arrays

[‡]There is however a fundamental difference between the Nyquist criterion and the double pulse resolution. The Nyquist criterion is a fundamental limit, independent of the ADC circuit, while the double pulse resolution is limited by the circuit.



(a) Signal converted by an ADC



(b) Signal converted by a TDC

Figure 3.1: Comparison of ADC and TDC output data

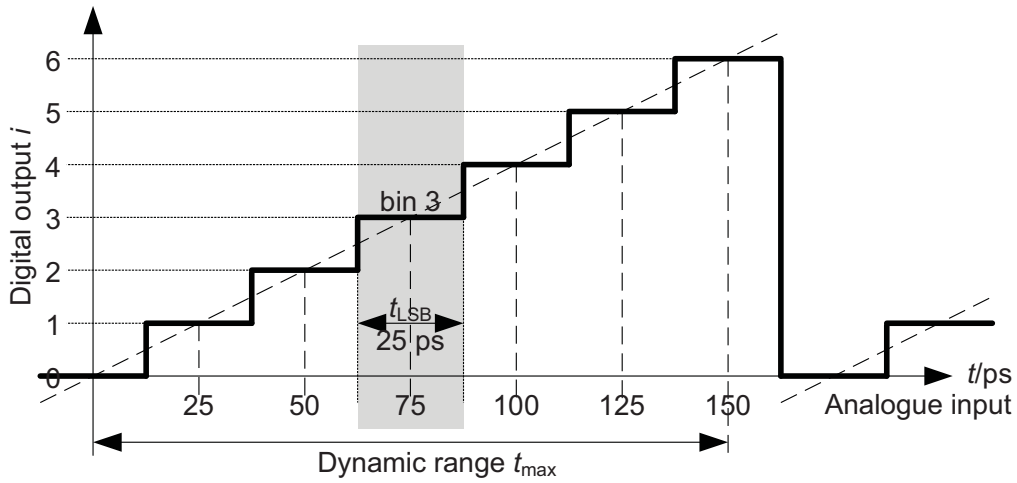


Figure 3.2: Transfer characteristic of an ideal TDC

3.1 Ideal TDC

A TDC converts a time interval into a digital number. The measured quantity can be a time relative to a reference, a delay with respect to another signal or between to edges of the same signal.

Fig. 3.2 shows the transfer function of an ideal TDC. The conversion consists of the discretisation of the continuous signal. It can be described by a piecewise linear equation:

$$i = (t \bmod t_{\mathrm{LSB}}) \ \mathrm{div} \ \frac{t_{\max}}{N}$$

mod being the modulo and div the integer division operator. The real number i is the digital output of the infinite resolution TDC, when $N \to \infty$. As real TDCs cannot have infinite resolution, i belongs to a discrete set of numbers generating the bold solid staircase function that is only an approximation to the ideal straight line. Signals within an interval $[t_i; t_{i+1})$ are rounded to the interval's mean value $\frac{t_i + t_{i+1}}{2}$ and represented by the number i. One key figure describing a TDC is therefore the bin size, the width of the discretisation interval, which is called Least Significant Bit (LSB). The LSB represents the smallest time difference that can be resolved with the TDC. In physics applications, the required RMS[†] resolution or the required standard deviation of the measurement is

[†]Root Mean Square

often specified, while in the domain of engineering, the bin size is frequently stated. These numbers can easily be converted:

$$\sigma_{\mathrm{LSB}} = t_{\mathrm{LSB,RMS}} = \frac{t_{\mathrm{LSB}}}{\sqrt{12}} \approx \frac{t_{\mathrm{LSB}}}{3.5}$$

as shown in appendix A.1.

Another key parameter is the dynamic range t_{max}, which gives the largest time difference that can be measured unambiguously. If the dynamic range is exceeded, an overflow occurs. The TDC output can represent either the measured time modulo the dynamic range, as assumed in this work if not otherwise stated and as shown in fig. 3.2, or, as it is the case for most ADCs, saturate to the maximum value.

Note that the number of digital output words, corresponding to the number of time bins, does not need to be a power of 2. Even though the output values are usually represented by a binary word, some words might not be used to represent a time. For example, the output words in fig. 3.2 can be encoded with 3 b. However, out of the $2^3 = 8$ available symbols, only $N = 7$ are used.

3.2 Performance Metrics for TDCs

Some performance metrics for TDCs have been established under the same name as for ADCs [IEE01]. However, the definition may differ. Therefore, the definitions used in this thesis will be explained. Other ADCs metrics are not applicable to TDCs (e.g. sampling frequency) and vice-versa (e.g. double pulse resolution).

The most basic performance metrics that describe the properties of an ideal TDC as explained in section 3.1 are:

- The Bin Size t_{LSB} and RMS Resolution σ_{LSB}
- The Dynamic Range t_{max}

Static deviations from the ideal behaviour (fig. 3.3) are characterised by:

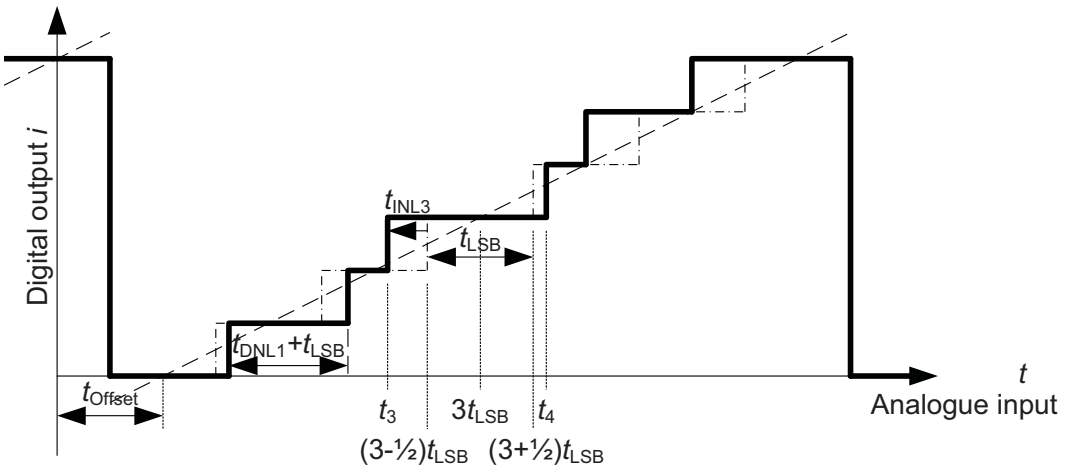


Figure 3.3: Transfer characteristic of a TDC. The thick solid line shows the characteristic of a real TDC, the dash-dotted line its approximation by an ideal TDC transfer function with an offset and the dashed line the piecewise-linear approximation.

- The Differential Non-Linearity (DNL)
  The differential non-linearity $t_{\mathrm{DNL},i}$ is the deviation of a particular bin size $\Delta t_i = t_{i+1} - t_i$ of bin i from its ideal value t_{LSB}:

  $$t_{\mathrm{DNL},i} = \Delta t_i - t_{\mathrm{LSB}} = t_{i+1} - t_i - t_{\mathrm{LSB}} \text{ with } i = 0 \ldots N-1 \text{ integer}$$

  Usually, the standard deviation σ_{DNL} is stated, sometimes with a graph showing the $t_{\mathrm{DNL},i}$ for all individual bins $i = 0 \ldots N-1$. The DNL is commonly measured in units of LSB.

- The Integral Non-Linearity (INL)
  The integral non-linearity $t_{\mathrm{INL},i}$ is the deviation of the total delay t_i, measured from the beginning of bin 0 to beginning of bin i, assuming the real offset t_{Offset} and ideal bin sizes t_{LSB}.

$$t_{\mathrm{INL},i} \quad = \quad t_i - i\,t_{\mathrm{LSB}} + \frac{1}{2}\,t_{\mathrm{LSB}} - t_{\mathrm{Offset}} \text{ with } i = 0 \ldots N-1 \text{ integer}$$

Usually, the standard deviation σ_{INL} is stated, sometimes with a graph showing the $t_{\mathrm{INL},i}$ for all individual bins $i = 0 \ldots N-1$. The INL is commonly measured in units of LSB, as it is the case for the DNL.

- The Gain Error
  The gain error is the deviation of the slope of a linear approximation to the TDC transfer function from its ideal value. As the TDC time base consists of two parts, the fine time base and the dynamic range extension, a gain error translates into a large nonlinearity between the last and the first bin of the fine time base. Thus, it is more meaningful to include gain error effects in the INL, as it is the case for the definition of the INL used in this work.

- The Offset
  For an event at time $t = 0$, a real TDC measures t_{Offset}. In every application, the offset of the TDC will add to differences in reference and signal paths. Practical applications have to be able to either cope with any offset, or include offset compensation. The offset does not depend on the time to be digitised. Any constant offset can be compensated for by adding a constant value to the measurement, regardless of the measurement itself. The transfer function of a real TDC can be approximated adding a constant to the approximation formula for ideal TDCs:

$$i = \left((t + t_{\mathrm{Offset}}) \bmod t_{\mathrm{LSB}} \right) \operatorname{div} t_{\mathrm{LSB}}$$

  The method of least squares is used to determine the offset, minimising the sum of the squared INLs for all bins:

$$\sum_{i=0}^{N-1} t_{\mathrm{INL},i}^2 = \sum_{i=0}^{N-1} \left(t_i - i\,t_{\mathrm{LSB}} + \frac{1}{2}\,t_{\mathrm{LSB}} - t_{\mathrm{Offset}} \right)^2 \longrightarrow \min$$

In practice, the offset of both the TDC and the input signal chain is expected to vary slowly during operation due to temperature changes. The offset can be minimised by careful layout, making sure the paths of both signal and reference are as similar as possible, the connected circuitry outside the chip will be very different in many applications. The off-chip contributions will be dominant and require periodic recalibrations at the system level.

- The Double Hit Resolution
  The double hit resolution is the minimum delay between two subsequent time measurements in the same channel. It is limited by the time it takes to process the data in the first stage and could be called maximum hit frequency, as it is similar to the maximum sampling frequency of an ADC. However, the term 'frequency' is rather used for periodic signals, while the hit signal is random by definition.

In addition, the conversion can be affected by random effects and by other signals:

- Jitter or Phase Noise
- Noise
- Crosstalk

Those effects depend strongly on the implementation of the chip and the system.

At the system level, further properties are important. They do not characterise the quality of the time to digital conversion, but reflect the requirements of the system.

- Number of channels per chip
  In HEP applications, a complete sub-detector can easily consist of many tens of thousands closely spaced channels. TDCs should have as many channels as possible, in order to minimise the number of chips required. In most other applications such as RADAR[†], e.g. in cars, LIDAR[‡] and mass-spectroscopy, only a few channels are needed [RRRK00, PMK02] and the sensors are often meters

[†]RAdio Detection And Ranging
[‡]LIght Detection And Ranging

apart from each other. Sharing a common TDC requires all timing sensitive signals to be brought to the chip, while in a one-chip-per-channel approach, only digital signals need to be distributed.

- Calibration requirements
  The implementation of the TDC time base also determines the calibration required to achieve precise measurements. While automatic calibration e.g. to the reference clock period is intrinsic to some time base architectures, others require special calibration procedures to be followed in regular intervals. If the signals to be measured are random, they can be used for a calibration using a code density test (section 8.2.2, p. 103). Otherwise, dedicated calibration signal generation can be required.

- Integrated System-level functionality
  A TDC digitises a time, and all measurement data is expected to be read out. In HEP applications, the amount of data generated is far too high to be stored for off-line processing. Data reduction is required. The TDC ASIC can include a first level of data reduction. Avoiding the readout of data to be discarded as early as possible reduces also the power required to transfer those data.
  The readout bandwidth is often limited by the availability of the data link or the receiver. In HEP applications, many events arrive within a small time window on different channels, followed by phases of no or low activity. In those cases, it is essential to derandomise the data. Storing data in a buffer memory until the next processing stage is ready converts data available at highly random times into a data flow with rather constant data rate [CAA57,Jai91]. If the link is shared among multiple TDCs, a readout buffer can be included on-chip to store any data that is ready for readout until the readout link is attributed to the specific chip.

- Maximum hit rate
  The maximum hit rate is the maximum rate at which the TDC can accept hits, averaged over a given time. Once this rate is exceeded, data losses can occur due to buffer overflows. It must not be confused with the double pulse resolution, which specifies the minimum time between any two successive hits.

- Power dissipation

  Power dissipation translates into supply current and heat dissipation. In HEP applications, active cooling is often required [CMS00]. The larger the heat dissipation, the more material is needed for efficient cooling. In the ALICE inner tracking system, cooling and mechanics figure accounts for a large fraktion of the material budget [ALI95]. As little extra material as possible should be added inside a detector, as it can interact with the particles that are to be measured [Hal09]. Larger supply currents lead to thicker supply cables or larger resistive losses in the supply cables, causing heat dissipation. Thicker cables increase the material budget both by themselves and by their support structures.

  Therefore, it is essential to minimise the power dissipation of all chips inside a detector. A chip with multiple channels can be designed such that some parts of the chip are used for all channels while only a few parts are duplicated for each channel. From a global detector point of view, it is advantageous to share one chip by all channels that are nearby and require only little signal routing.

4 TDC Data Flow Architecture

4.1 Specifications

The new TDC needs to have 32 channels with a resolution of 25 ps or better. The time of arrival of a leading or a trailing edge will be measured depending on the configuration. To save readout bandwidth, in case both leading and trailing edge have to be digitised, the data can be converted into time of arrival of the leading edge and pulse width, as the latter requires a smaller dynamic range. In Time Over Threshold (TOT) measurements, the relevant information is the pulse width, not the time of arrival of the trailing edge. Often, the pulse width has to be known with a higher resolution than the time of arrival and the precision of the leading edge timestamp can be reduced prior to readout. The minimum dynamic range to be covered is in the order of 150 ns. One option for the CLIC RF bunch train length is 58.4 ns [BCdR$^+$06]. A trigger feature allows to select hits for readout. Typically, the majority of data is discarded before leaving the TDC chip. This requires a buffer large enough to store all data during the trigger latency, i.e. until the trigger signal is available to the TDC chip. Overlapping triggers should be accepted. In triggered applications, it is often convenient to have the time scale relative to the trigger timestamp. The double pulse resolution, the time between to successive signals that can still be distinguished, has to be about 2 ns or smaller. For CLIC, bunch trains of 58 ns duration with a repetition rate of $150\,\mathrm{Hz} = (7\,\mathrm{ms})^{-1}$ are being discussed [BCdR$^+$06]. To save readout resources, multiple TDCs will share the readout link. Due to the trigger latency, readout will only take place after the high activity bursts. The last TDC to be read out will need to buffer the data during about 7 ms. Events may be lost, e.g. due to buffer overflows or single-event upsets, but the loss has to be signalled to the following data processing stage.

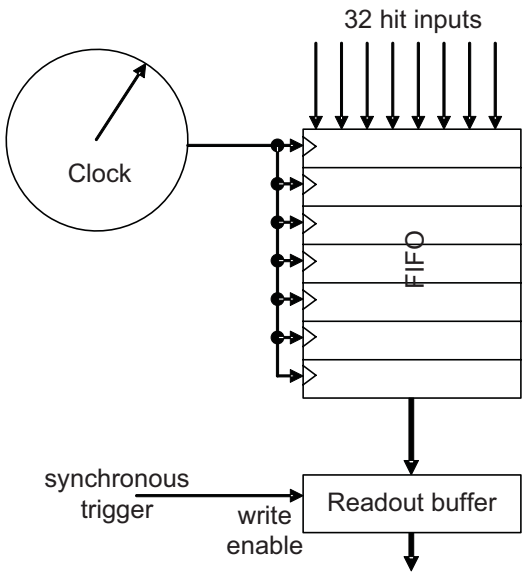


Figure 4.1: A clock driven TDC

The TDC130 has to be a highly configurable and multi-purpose, so that it can be used for most HEP applications without additional research, development and design effort, in order to reach production quantities that justify the development.

4.2 Clock driven architecture

In clock driven TDCs, the state of the hit inputs is written into a FIFO[†] once every clock cycle (fig. 4.1). Therefore, the bin size and the double pulse resolution are equal to the clock period. The width of the FIFO is equal to the number of channels, 32. A bin size of 25 ps requires a clock frequency of 40 GHz. The FIFO has a fixed latency, which is equal to the trigger latency. As one data word is generated per clock cycle, the FIFO occupancy is independent of the activity of the channels. In ALICE TOF, this latency is about 1.2 µs [AAea09], requiring a FIFO of 1.2 µs $\times$ 40 GHz $\times$ 32 b $\approx$ 1.5 Mb operating at 40 GHz. This cannot be implemented in the chosen 130 nm-technology and, moreover, consumes a considerable amount of power. At the arrival of the trigger signal, the corresponding hit input state word is leaving the FIFO and transferred to

[†]First In-First Out memory

the data input of the readout buffer. The readout buffer is write enabled if the TDC is triggered and disabled otherwise. Note that the trigger is not sensitive to the data in the FIFO, but to its position. Therefore, corrupted data in the FIFO doesn't influence the operation of the chip. The operating frequency of the data processing puts further limits on the double pulse resolution. In addition to the state of the hit inputs, a timestamp has to be stored in the readout buffer. The dynamic range of the time reference must be at least equal to the trigger latency if no timestamp is added in the FIFO. In HEP applications, very often a short period of activity is followed by a long break, as e.g. in the proposed CLIC system [BCdR+06]. In those cases, it is very inefficient that the input state has to be written into the FIFO even if there is no signal expected, in order to keep synchronisation.

4.3 Data driven architecture

In data driven architectures (fig. 4.2), the first memory, called hit register, is clocked by the hit inputs – whenever a hit signal arrives, and only then, the timestamp provided by a reference is stored. The operating frequency of the hit register is equal to the hit rate, the amount of data to be processed in a channel depends on the channel's activity. The maximum operation frequency fixes the double pulse resolution and does not influence the resolution. Each channel has its own hit register. The hit timestamps are transferred from the hit registers to a level-1 buffer. This buffer stores the timestamps during the trigger latency. The required buffer size depends on the number of hits during this period. Typically, in HEP experiments, the hit rate is in the order of 10 kHz to 1 MHz, which is considerably less than the inverse bin size of, in the hypothetical case being discussed, 40 GHz. Therefore, the level-1 buffer can be much smaller and consumes less power than the FIFO in clock driven architectures. However, if the actual hit rate exceeds the design value, buffer overflows can occur. Storing the data during the long breaks between the high activity periods, like in CLIC applications, has no implication on the buffer sizes.

Once the trigger signal arrives, a trigger timestamp is generated which takes the trigger latency into consideration. The data in the level-1 buffer is compared with the trigger timestamp. If the hit timestamp is within a configurable neighbourhood, called

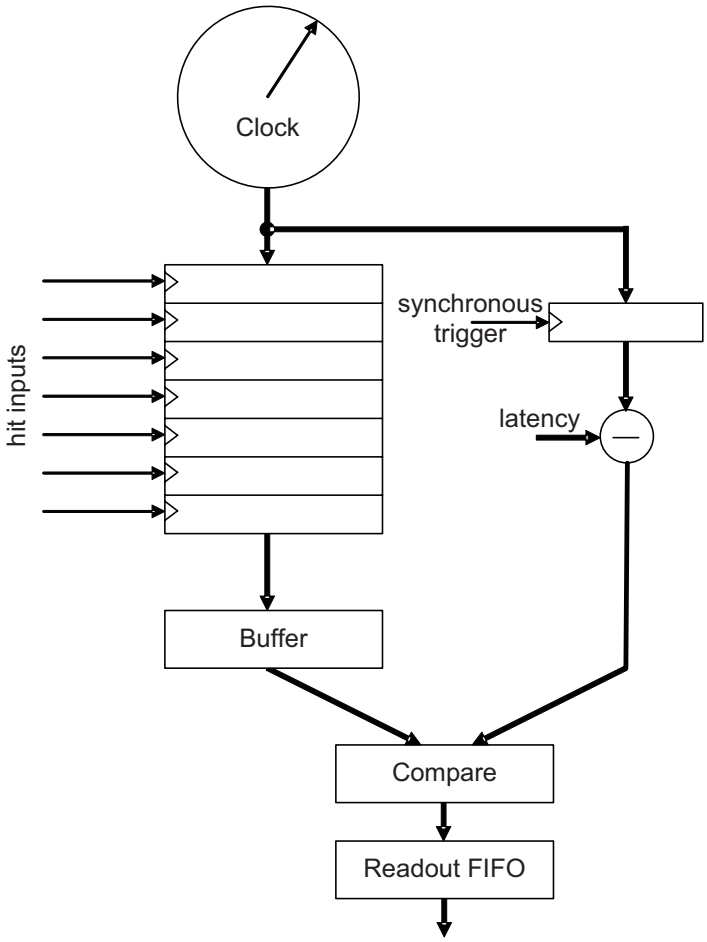


Figure 4.2: A data driven TDC

trigger window, of the trigger timestamp, it is transferred into the readout FIFO. While the trigger mechanism in clock driven architectures does not depend on the value stored in the buffer, in data driven architectures it does. Data corruption can consequently influence the operation of the trigger mechanism and thus, seen from the user's perspective, the complete TDC. In HEP applications, the loss of a single timestamp is often not a problem. It is sufficient to implement an error detection scheme and discard all corrupted data. The readout path is common for all channels. Therefore, the data coming from different channels need to be merged at some point between the hit registers and the readout buffer. This point can be chosen by the designer.

	Clock Driven Architecture		Data Driven Architecture
–	bin size equals clock period	+	bin size and clock period independent bin size can be much smaller than clock period
	double pulse resolution equals bin size		bin size and double pulse resolution independent bin size can be much smaller than double pulse resolution
–	cannot easily adapt to varying hit rates		
+	no buffer overflows	–	buffer overflow possible occupancy depends on hit rate
+	very simple trigger mechanism		
+	trigger does not depend on data in buffers	–	corrupted data in buffers can require reset of TDC

Table 4.1: Comparison of Data Flow Architectures

4.4 Choice of Data Flow Architecture

Tab. 4.1 summarises the comparison. Overall, the data driven TDCs can fulfil the requirements for HEP experiments much better than clock driven TDCs at the price of a more complicated trigger mechanism. The main disadvantage, the sensitivity of the control to corrupted data, can be compensated for using error detection or even error correction schemes. Therefore, the TDC presented in this thesis is data driven. The following sections describe the chosen architecture in more detail.

5 Time Base Architectures Overview

The time base provides a timing reference to the channels. Two different groups of TDC time bases are established. In time-tagging TDCs, the time is measured relative to a global reference, for example the 40 MHz LHC clock. This is similar to normal watches. In start-stop TDCs, there is typically one start channel and one or multiple stop channels. A signal on the start channel starts a circuit generating a time information, and a signal on a stop channel stops this circuit for the respective channel. This is comparable to stop watches in sport competitions.

Due to the high number of channels needed in an experiment and the availability of a very well defined reference clock that serves as a global reference for most systems in HEP experiments such as LHC, the TDC130 is a time-tagging TDC. Whenever a start-stop measurement in required, the time difference of two channels, which do not necessarily need to be connected to the same chip, can be calculated off-chip, e.g. in an FPGA.

There are many different options for the implementation of a time base, as shown in this chapter. To understand the global architecture, it is sufficient to know they all provide a digital timestamp that is distributed to the channels.

5.1 Time Base Architectures

Different methods have been proposed in the past for accurately measuring time [Por73, MRT+07, Kal]. To reduce power consumption, the TDC130 will have a single time base that provides the time information simultaneously to all channels. In this way, the power consumption of the time base is not duplicated for each channel.

The current integration method, for example, requires one time base per channel. In the simplest case, a constant current source is feeding a capacitor between the arrival of the start and the stop signal. The voltage across the capacitor is a measure for the time difference and can be converted using a standard ADC. This time base cannot be shared by all channels, which makes it unsuitable for the TDC130. Therefore, such techniques are not described in detail in the following sections.

5.1.1 Counter based Architecture

A common technique for timestamp measurements is based on a counter that is driven by a clock source. Whenever a hit arrives, the counter value is stored in the hit register of the corresponding channel while the counter continues running (fig. 5.1). The resolution is equal to the period of the reference clock, T_{clk}. The accuracy is given by the stability of the reference clock and the hit registers. The dynamic range is limited by the counting range and can easily be extended according to the application's needs. Counters are easy to implement in a digital design and could even be synthesised automatically, if no particular constraints, especially on speed, are to be respected.

A potential problem of the counter technique is the sensitivity to metastability [HEC89, Has97, Cyp97] in the hit registers. If a hit arrives while the counter is toggling, the timestamp stored in the hit registers is unpredictable. In a Gray code counter, only one bit toggles at a time. If this bit gets corrupted, the value of the timestamp can only shift by one bin size. Another method is to have two counters running on opposite phases of the same reference clock and sample both together at the arrival of a hit. While one counter might have toggled, the other counter is guaranteed to be stable. In a next step, one of the two timestamps will be discarded. The problem consists in determining which timestamp might have been affected by metastability.

A counter with a resolution of 25 ps requires a clock of $(25\ \mathrm{ps})^{-1} = 40\ \mathrm{GHz}$. In the chosen 130 nm-technology, such a clock cannot be provided, and neither can a counter running at that speed be built. In the following sections, other techniques are described. They all can give a higher resolution than a counter. Combining them with a counter extends their dynamic range to that of the counter while keeping the high resolution.

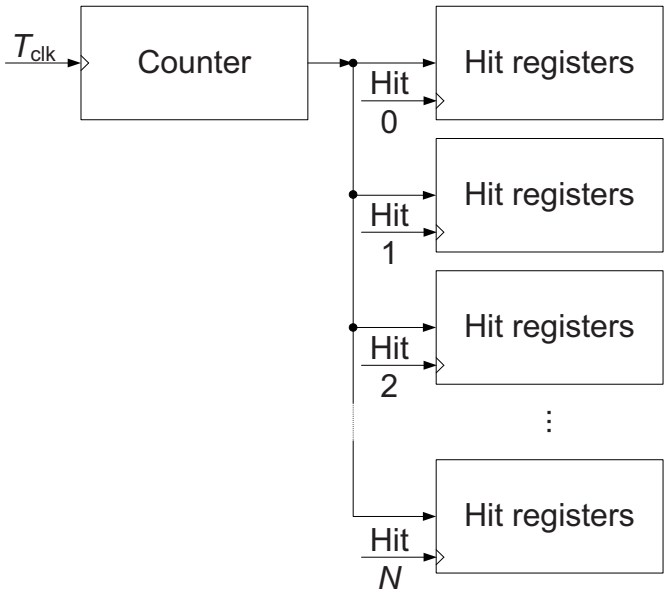


Figure 5.1: Counter based TDC
The hit registers are clocked by the respective channel's hit input signal.

They can also provide the information on which of the two counters has toggled more recently.

5.1.2 Delay line based Architecture

A conceptually simple way to obtain high resolution is phase interpolation using delay lines with multiple, equally spaced tap-outs (fig. 5.2). In this case, a low frequency clock signal is delayed in small steps. Let T_{clk} be the period of the reference clock and N the number of tap-outs. Assume the delays of all elements are equal and sum up to T_{clk}. The bin size is then $t_{\mathrm{N}} = \frac{1}{N} T_{\mathrm{clk}}$. The phase shifted versions $\varphi_0 \ldots \varphi_{N-1}$ of the clock signal are distributed to the hit registers. The position of the clock edge within the delay line at the time of the hit's arrival and thus within the hit registers gives the hit timestamp. It is irrelevant which clock edge is taken for this measurement. A hit may arrive at any time during a clock period. Therefore, the delay line must cover the full clock period. As the clock signal is periodic, even if the line delay was longer than

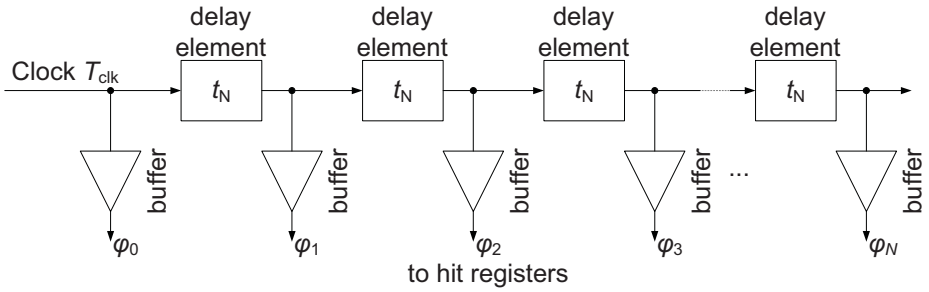


Figure 5.2: Delay line

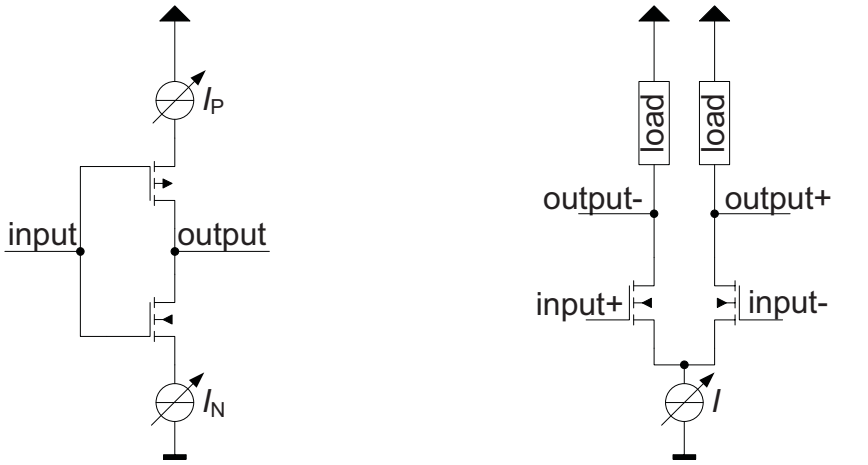
a clock period, the dynamic range is limited to the clock period. However, it can be extended using a counter clocked by the same clock as the delay line.

Passive delay elements

Delay elements, RCs, LCs or transmission lines, can be implemented in sub-micron CMOS technologies. Resistors are easy to implement in a digital CMOS process, but the signal is attenuated along the line. Inductors can be built using standard metal wires and generate less attenuation, but take considerable space [SRGR03, DDTGG05]. In addition, coupling between multiple, closely spaced inductors is expected. Transmission lines are basically straight, shielded wires, as opposed to circular, not necessarily shielded inductors. Compensation of the process variations is necessary at least once per chip, if not once per start-up [Mot00] to obtain optimum linearity. Resistors that are not implemented using metal interconnects are subject to temperature variations and may require temperature compensation. Additional capacitors can be connected to increase the delay in all implementations. For LCs and transmission lines, the signal can be tapped out from different places within the inductors or the transmission lines to reduce the delay. Synthesisable digital logic can perform the calibration based on a code density test (ch. 8.2.2).

Gate delays

In 130 nm-technologies, the gate delay is in the order of tens of picoseconds. As the desired resolution is in the same order of magnitude, using gate delays as delay ele-



(a) Single-ended inverter (b) Differential buffer/inverter

Figure 5.3: Adjustable delay gates

ments instead of the above-mentioned passive delay elements is an option. The gate delay can be adjusted limiting the current passing through its transistors (fig. 5.3). While this is trivial in differential buffers, single-ended inverters need two additional transistors for current limitation. The delay can be adjusted using a control voltage or a control current. The delay line is called Voltage Controlled Delay Line (VCDL) or Current Controlled Delay Line (CCDL). Alternatively, multiple current limiting transistors, typically with binary weighted sizes, can be used. The delay is adjusted by enabling or disabling current limiting transistors [DGLN95, WWCL05]. The adjustment can be performed by digital logic and the delay line is called a digitally controlled delay line.

The delay of a gate depends not only on the process, but also on temperature and even supply voltage variations. Therefore, the calibration needs to be performed more frequently.

Adjustable delay gates are slower than non-adjustable gates as the current control introduces additional parasitics and results in reduced current driving.

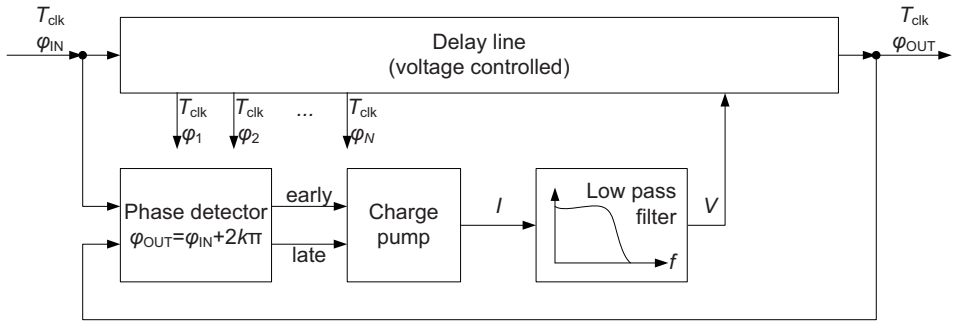


Figure 5.4: DLL (k=1 in nominal operation)

5.1.3 Delay Locked Loop based Architecture

A VCDL or CCDL can be part of a control loop that automatically adjusts the delay of the full line to be equal to one clock cycle. Such a device (fig. 5.4) is called Delay Locked Loop (DLL).

An N-element DLL provides N phase shifted tap outputs to the hit registers, just as a delay line (5.1.2). It is always tuned so that the sum of all delays equals one reference clock cycle, T_{clk}. This cancels process, temperature and supply voltage variations, as long as they affect all elements equally.

The control loop of a DLL consists of a phase detector, a charge pump, a loop filter and the delay line. The phase detector compares the phase at the delay line input to that at the output. The target phase difference between input and output is 2π[†]. Depending on whether the phase difference is smaller or larger than the target value, it makes the charge pump sink or source a current for one clock cycle. The current is low-pass filtered by the loop filter, which consists of a capacitor. Every phase detector decision thus adds or removes a constant amount of charge to or from the loop filter capacitor, increasing or decreasing the voltage across it by a constant amount. This voltage is called control voltage, as it is used to control the current sources in fig. 5.3 and thus the delay of the delay line.

[†]This is the nominal case. In principle, the phase detector can make the loop lock to a phase difference of $2\pi k$ with $k \geq 0$ integer. Additional circuitry assures that it always locks to 2π (section 7.2.7, p. 71)

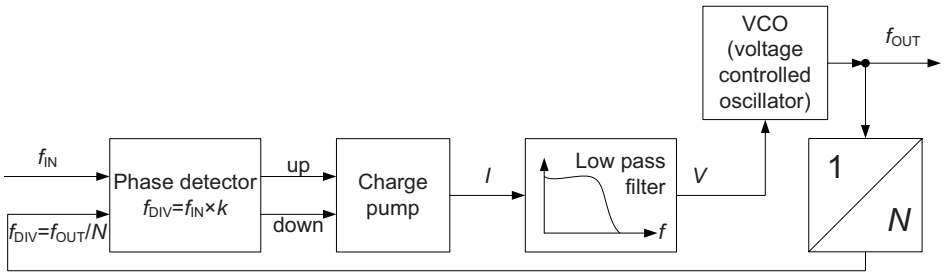


Figure 5.5: PLL (k=1 in nominal operation, N is the clock multiplication factor)

5.1.4 Phase Locked Loop based Architecture

Alternatively, a Phase Locked Loop (PLL) can be used as time base. A PLL (fig. 5.5) consists of a Voltage Controlled Oscillator (VCO) that generates an output signal with a frequency f_{out} which depends on a control voltage. The loop assures that the output frequency is N-times higher than the frequency of a reference clock signal. Let's first consider the case with $N = 1$, when both reference input and VCO output have the same frequency.

Both the reference clock signal and the VCO output signal are fed to the phase detector. The phase detector compares these two signals. Depending on whether the VCO output phase is leading or lagging the reference clock phase[†], it makes the charge pump sink or source a current for a duration of one reference clock cycle. If the frequencies are equal, the charge pump neither sinks nor sources any current. This current is converted into the VCO control voltage by a low-pass filter.

A major difference between PLL and DLL is that the PLL loop changes the frequency, not the phase, of the output signal in order to reach zero phase difference between reference input and VCO output. As the phase is the integral of the frequency, the phase detector introduces a new pole into the transfer function. The PLL is a second order system, which is not intrinsically stable.

If a frequency divided version of the VCO clock is fed to the phase detector, the phase detector compares the reference frequency to the divided VCO output frequency.

[†]This is the nominal case. In general, a phase detector cannot distinguish phase differences that are integer multiples of 2π. Depending on the phase detector implementation, additional circuitry may required to assure that the PLL always locks to the reference clock frequency.

The loop makes sure that the reference input and divided VCO output have the same frequency. The VCO frequency is N-times higher than the reference clock frequency.

Ring oscillators are commonly used as VCO. They consist of a chain of delay elements[†], where the output of the last element is connected to the input of the first. The element delay is adjusted using a control voltage. The output of a delay element is phase-shifted, i.e. delayed, with respect to the preceding element. As for the delay line and the DLL, the delay elements of the PLL oscillator can provide phase-shifted signals that are sampled by the hit registers.

A fundamental difference between DLL and PLL is that the PLL generates its own signal, while the DLL delays the input signal without changing its frequency. A PLL can reduce the jitter of the reference clock signal if the VCO jitter is lower than the reference jitter. In this case, the jitter of the TDC timing measurement is lower than it would be if a DLL was to be used.

5.1.5 Array of DLLs based Architecture

As shown in chapter 5.1.3, DLLs can be used to generate multiple phase-shifted versions of the same, rather low frequency signal. The bin size is equal to the phase shift. The minimum delay of a single element is limited by the technology, but it is possible to have multiple DLLs with slightly shifted input phases. The shifted input phases can be generated using another DLL. Long delay lines lead to mismatch and consequently linearity problems, but in an array of DLLs, the individual lines are short and independently controlled. For the encoding and following data processing, it is convenient, but not necessary, to have a total bin number which is a power of 2. Fig. 5.6 shows the arrangement of DLLs, fig. 5.7 the bins.

Note that in fig. 5.6, the M-element DLL's phase detector takes the signals from the end of the delay line, as usual, and from the output of the m^{th} delay element of the first

[†]To oscillate, the signal at the end of the chain needs to be inverted with respect to the input. If an odd number of inverters is used, this condition is fulfilled, but the number of bins is odd, in particular not a power of 2, and the delay of the chain covers only half a clock cycle. However, techniques to build ring oscillators with an even number of bins have been developed [AI96].

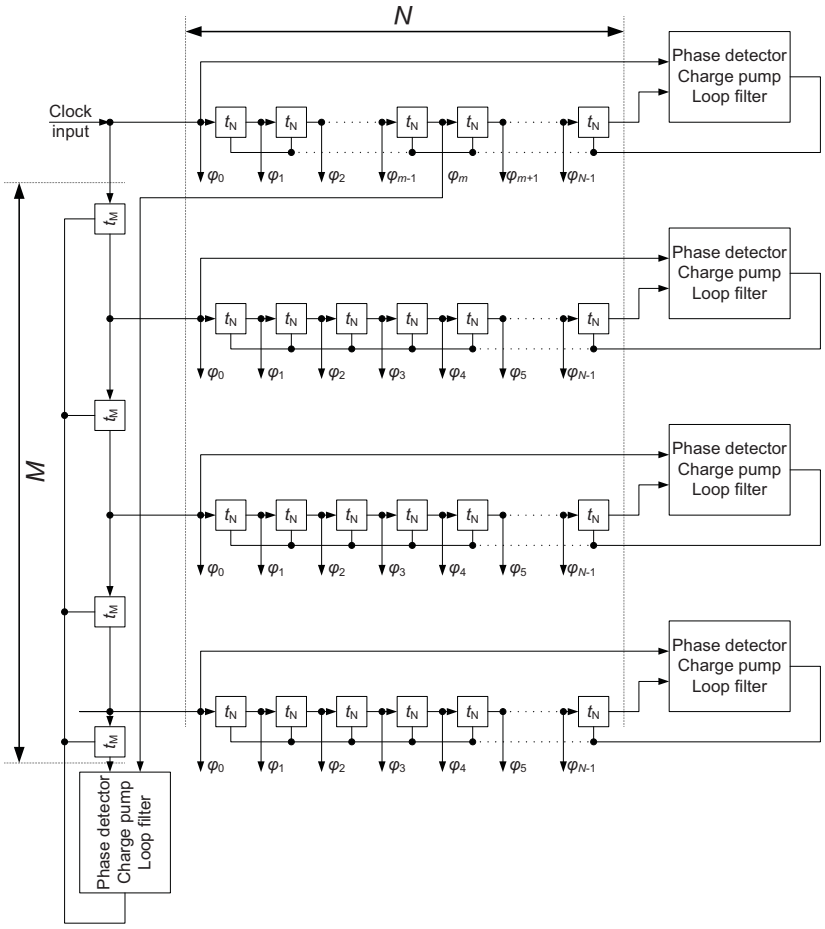


Figure 5.6: Array of DLLs

N-element DLL. Thus, in lock, the delay of the full M-element chain is equal to the delay of m elements in the N-element DLLs.

$$t_{\mathrm{M}} = \frac{m}{M} t_{\mathrm{N}}$$

The number of bins of the array into which a bin of an N-element DLL is divided into is called 'interpolation factor'. An interpolation factor $F = 4$ can be achieved with $m = 5$ and $M = 4$. The phase shift between two subsequent N-element DLLs is then $t_{\mathrm{M}} = \frac{5}{4} t_{\mathrm{N}} = \left(1 + \frac{1}{4}\right) t_{\mathrm{N}}$ instead of $\frac{1}{4} t_{\mathrm{N}}$. This is equivalent, as the subtraction of integer numbers of t_{N} corresponds to fixed rotations of the corresponding part of the timestamp.

As example, let's look at the outputs of the 4th element of each N-element DLL. For simplicity, let $t = 0$ be the time that the input clock signal toggles. In the DLL whose input is fed by the same signal as the M-element DLL, this signal toggles at $t = 4\,t_\mathrm{N}$. In the second DLL, the same element toggles with an additional delay of $t_\mathrm{M} = \frac{5}{4}\,t_\mathrm{N}$, thus at $t = 5\frac{1}{4}\,t_\mathrm{N}$, not at $4\frac{1}{4}\,t_\mathrm{N}$ as required. The difference is exactly $1\,t_\mathrm{N}$, so a signal with the required timing can be found at the output of the 3rd element's output. The 2nd element of the third DLL toggles at $t = 4\frac{2}{4}\,t_\mathrm{N}$, the 1st element of the fourth DLL at $t = 4\frac{3}{4}\,t_\mathrm{N}$.

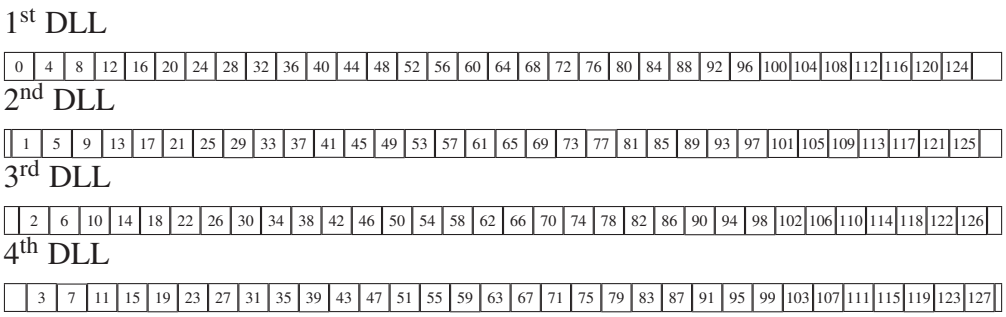


Figure 5.7: Bins in a 4×32 array of DLLs
The bins without number are bins $129 \ldots 131$ of the preceding clock cycle or bin $0 \ldots 3$ of the following cycle respectively.

This technique requires M DLLs with a constantly propagating clock signal, and one DLL with a clock signal that propagates only at the beginning of each clock cycle. The power dissipation is larger than that of a single DLL by a factor between 4 and 5. As the DLL array is shared among all channels, all MN tap-out signals need to be routed to all channels – the routing effort is non-negligible.

5.2 Fine Time Interpolation Techniques

The resolution of both DLL and PLL based techniques are limited by the delay of the gates used as delay elements, which is given be the technology. To further increase the resolution, different techniques can be used to provide sub-gate delay interpolation.

5.2.1 Dual-Slope Analogue Time Expansion

In the domain of voltage ADCs, the dual-slope conversion has been used for a long time. The input signal, a voltage, is converted into a time using an integrator, a reference, a comparator and some control logic (fig. 5.8a) [TS02].

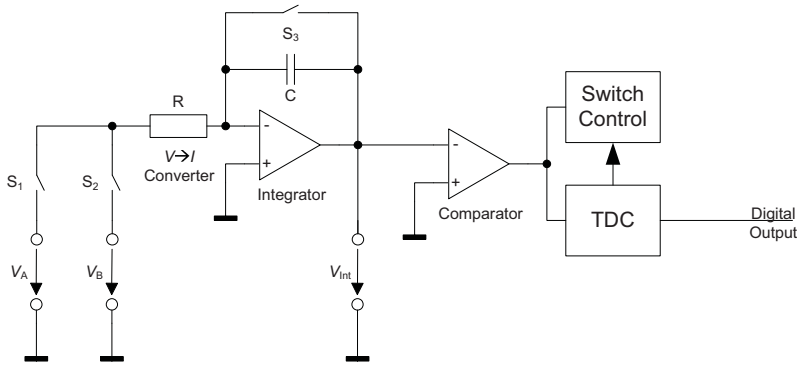
Fig. 5.8b shows a voltage measurement cycle. The input voltage $V_\mathrm{A} > 0$ is converted into a current, which is integrated over a fixed time $t_2 - t_1$. Afterwards, the switch control replaces the input voltage by a reference voltage $V_\mathrm{B} < 0$. When the integrator's output voltage V_Int reaches 0 at t_3, the switches are put into reset configuration, ready to take a new sample. The charge of the capacitor is zero at t_1, then decreased until t_2 and increased to zero at t_3:

$$\begin{aligned} & \frac{1}{R} V_\mathrm{A} \, (t_2 - t_1) + \frac{1}{R} V_\mathrm{B} \, (t_3 - t_2) = 0 \\ \iff \quad & V_\mathrm{A} = -V_\mathrm{B} \frac{t_3 - t_2}{t_2 - t_1} \end{aligned}$$

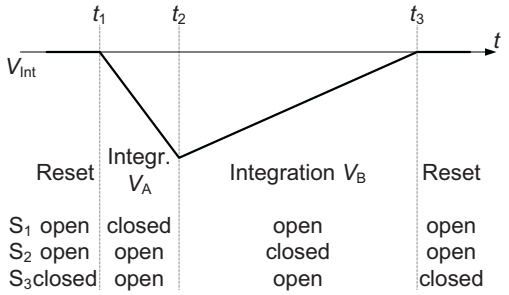
The durations of the charge and discharge cycle are measured and represent the input voltage as a function of the reference voltage.

This principle can be easily adapted to time measurements. The conversion circuitry may mainly remain unchanged; the only difference is the control of the current sources. Both V_A and V_B serve as reference voltages. Their sign is opposite, in fig. 5.8c $V_\mathrm{A} > 0$ and $V_\mathrm{B} < 0$. The dual-slope TDC is used to provide only the fine part $t_1 - t_0$ of a time stamp with high resolution and low dynamic range, while other circuitry gives the coarse part t_0 with low resolution and high dynamic range. Therefore, the presence of a reference clock with a period T_clk provided for the coarse measurement must be assumed. Without loss of generality, assume the TDC is set to be sensitive to leading edges on the hit input. At the arrival of a hit, the current $\frac{V_\mathrm{A}}{R}$ is integrated until the e.g. next leading edge of the reference clock. Subsequently, the switches are changed and $\frac{V_\mathrm{B}}{R}$ is integrated until the integrator output voltage V_Int equals 0. A coarse timestamp at the moments t_1, t_2 and t_3 is available. $t_0 = t_2 - T_\mathrm{clk}$ due to the implementation of the switch control. As before,

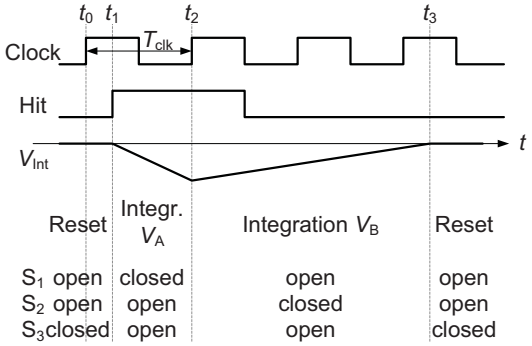
$$\frac{1}{R} V_\mathrm{A} \, (t_2 - t_1) + \frac{1}{R} V_\mathrm{B} \, (t_3 - t_2) = 0$$



(a) Dual-Slope converter schematic



(b) Dual-Slope ADC conversion cycle



(c) Dual-slope TDC conversion cycle

Figure 5.8: Dual-slope converter

Now, the quantity to be measured is the fine timestamp, $t_1 - t_0$:

$$\begin{aligned} \Longleftrightarrow t_2 - t_1 &= \frac{-V_\mathrm{B}}{V_\mathrm{A}} (t_3 - t_2) \\ \Longleftrightarrow t_1 - t_0 &= t_1 - (t_2 - T_\mathrm{clk}) = \frac{V_\mathrm{B}}{V_\mathrm{A}} (t_3 - t_2) + T_\mathrm{clk} \end{aligned}$$

Instead of $t_1 - t_0$, $t_2 - t_1$ can be used as fine part of the timestamp – in this case it is not referenced to the preceding rising edge of the reference clock, but to the following. As this difference is too small to be measured directly, it is stretched by a constant factor k and the larger value $t_3 - t_2$ is measured instead. k is called stretch factor [Mot00]:

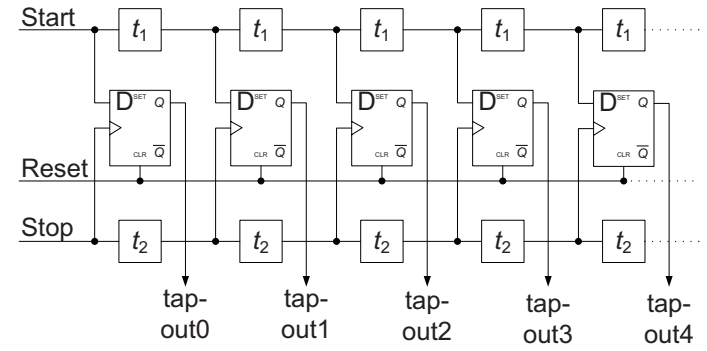
$$k = \frac{t_3 - t_2}{t_2 - t_1} = -\frac{V_\mathrm{A}}{V_\mathrm{B}}$$

To increase the resolution, k needs to be greater than 1, thus $|V_\mathrm{A}| > |V_\mathrm{B}|$.

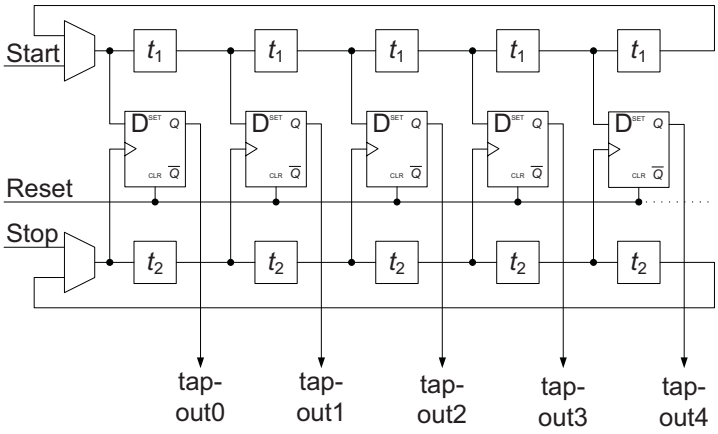
This technique introduces a dead time that is increasing with the stretch factor, thus the desired resolution. k needs to be stable and known with precision. Therefore, good matching of the reference voltage sources and the switches S_1 and S_2 is required. Noise coupling into the integrator, capacitor non-linearities and the comparator stability further limit the accuracy.

5.2.2 Vernier Techniques

Vernier techniques are very much similar to a vernier calliper. They are based on D flip flops and delay elements with two different delays and perform start-stop measurement [Mot00]. In the simplest case (fig. 5.9a), the D flip flops are clocked with the stop signal and sample the start signal. Assume the hit is the rising edge of both signals. Between one flip flop and the next, the start signal is delayed by a time t_1 and the stop signal by $t_2 < t_1$. Hence, the further the signals propagate through the lines, the smaller the difference between the arrival of the start and the stop signal. Eventually, if the line is long enough, the stop signal will overtake the start. The output of the following flip flops will be 0 instead of 1. Let 1-to-0 transition occur after the n^{th} flip flop. Assuming ideal D flip flops, the difference between start and stop signal to be measured is $t_\mathrm{start} - t_\mathrm{stop} = nt_1 - nt_2 = n\,(t_1 - t_2)$. A large dynamic range requires



(a) Vernier interpolation



(b) Vernier interpolation with loop

Figure 5.9: Vernier interpolation schemes

long chains of delay elements and flip flops. It is possible to let the signal propagate though the same delay elements multiple times arranging the delay elements in a ring structure (fig. 5.9b). The dynamic range of such a circuit is infinite. However, already the straight line implementation is very sensitive to delay cell mismatch. In a ring, the signal propagates through the same elements multiple times, accumulating the error. Calibration has to be performed in absence of hits. The delay elements can be operated as a ring oscillator in order to measure their total delay. Modifications of this technique enable the use for timestamp measurements, but still suffer from very high sensitivity to delay cell mismatch.

5.2.3 Passive LC lines or RC lines

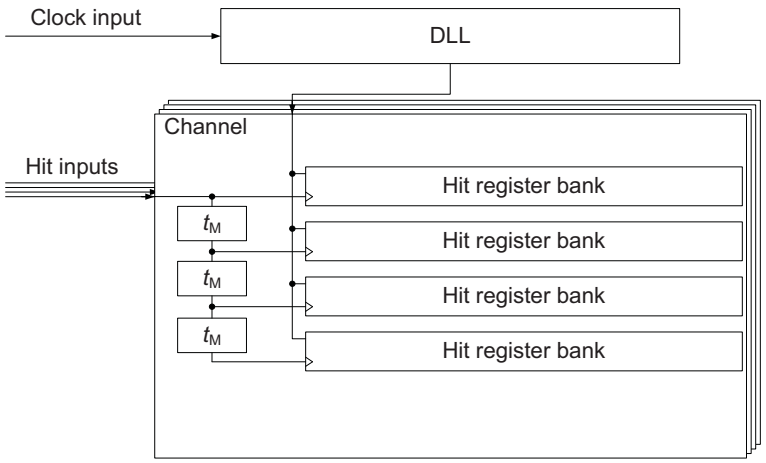


Figure 5.10: Passive delay line interpolation

Main register ($y = 0$)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

First interpolation register ($y = 1$): Delayed by $1 \times \frac{t_N}{4}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Second interpolation register ($y = 2$): Delayed by $2 \times \frac{t_N}{4}$

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Third interpolation register ($y = 3$): Delayed by $3 \times \frac{t_N}{4}$

			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Figure 5.11: Local interpolation: Shifted bins.
The bins without number are bin 31 of the preceding clock cycle or bin 0 of the following cycle respectively.

Another possibility for fine interpolation, i.e. interpolation which is performed using the hit signal and thus cannot be shared among channels, is based on passive delay lines. The timestamp coming from the main time base, e.g. a DLL, is distributed to all channels. To divide the DLL time bins of size t_N into M sub-bins, the hit registers are replicated M times. The hit signal propagates through an $M-1$-element delay line (fig. 5.10) with a unit delay $t_M = \frac{1}{M} t_N$. The tap-outs of this delay line are connected

to the clock signal input of the hit register. Fig. 5.11 shows the shifted bins in such an arrangement. As delay elements, RC or LC elements can be used as well as transmission lines. The advantages and drawbacks of different delay elements are discussed in section 5.1.2. In the HPTDC, local fine interpolation has been implemented with RC delay lines.

Fig. 5.11 shows the shifted time bins for $M = 4$ and $N = 32$. For a given hit, the timestamps in the hit register blocks are either all the same or differ by 1. After binary encoding of the timestamps, thus with only $\log_2(32) = 5$ b of data per hit register, the interpolation result can be computed. Let the transition from one timestamp value x to the next, $x+1$ occur between block y and $y+1$. The hit has then arrived in the $(M-y)^{\text{th}}$ quarter of the x^{th} main DLL time bin. If all timestamps in the hit register banks are equal, the transition has occurred after the block with $y = 3$, i.e. the hit has occurred in the 1^{st} quarter of the x^{th} main DLL time bin.

5.2.4 DLL adjusted delay lines

Resolutionwise, the array of DLLs is equivalent to one global N-element main DLL per chip and one local M-element secondary interpolation delay line per channel, reducing the amount of signals to be routed from the common time base to the individual channels. The secondary delay lines are fed by the hit input signal, not by the clock signal. This reduces the power consumption in idle mode considerably, as no signal is propagating along the line. The hit registers of each channel can be organised in M blocks of N elements. The first hit register block is triggered at the arrival of the hit, the second after a delay of t_{M}, the third after $2t_{\text{M}}$ and so on. The secondary delay lines are not part of a control loop, but they need to be calibrated frequently. They cannot be implemented as DLLs, as DLLs require a periodic signal to propagate constantly, and the hit signal is random by definition.

The calibration problem can be avoided with a global, secondary DLL with M elements, fed by the clock signal as it is the case in the DLL array. The delay line is replicated once per channel, and the control voltage of the secondary DLL distributed to all delay lines (fig. 5.12). As delay lines in the channels are not part of a closed loop, only $M-1$ elements are required. In this case, the local delay elements are auto-

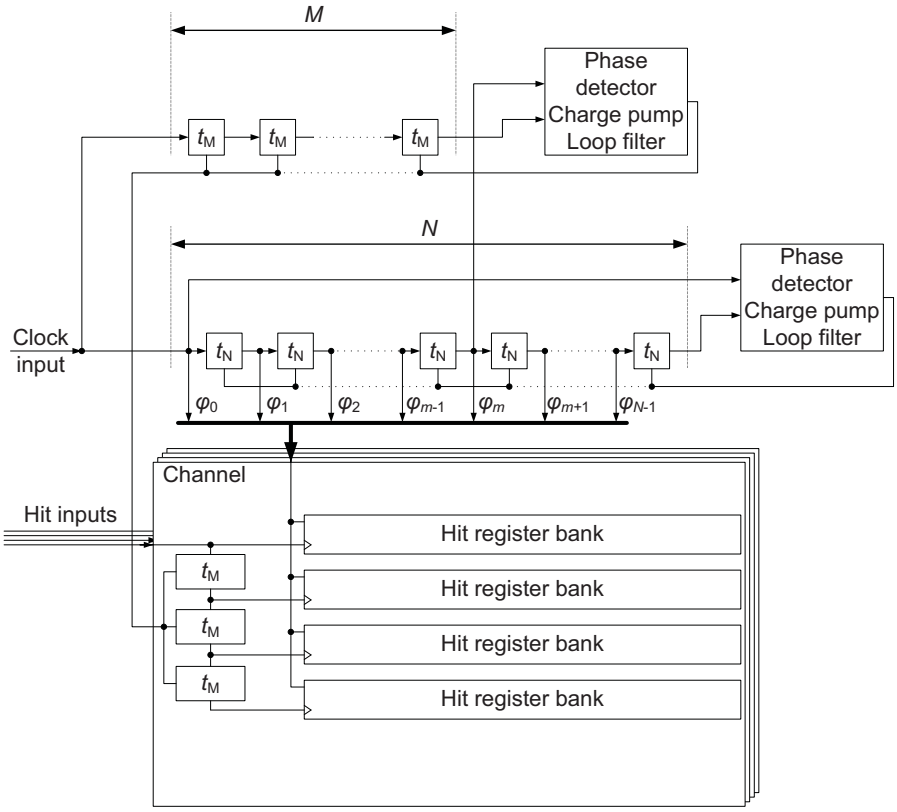


Figure 5.12: DLL adjusted delay lines for fine interpolation

Main register ($y = 0$)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

First interpolation register ($y = 1$): Delayed by $1 \times \frac{5t_N}{4}$

31	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	

Second interpolation register ($y = 2$): Delayed by $2 \times \frac{5t_N}{4}$

	30	*31*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	

Third interpolation register ($y = 3$): Delayed by $3 \times \frac{5t_N}{4}$

	29	*30*	*31*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	

Figure 5.13: Fine interpolation: Shifted bins.
The bins with numbers in italic belong to the preceding clock cycle.

calibrated, but do not consume power in absence of hits. The quality of the calibration depends on the mismatch between the delay line of the DLL and the delay lines in

the channels. Local fine adjustment needs to be implemented to limit the effects of mismatch. The control voltage distribution network is potentially prone to crosstalk.

Fig. 5.13 shows the shifted bins in the different hit register blocks for $M = 4$ and $N = 32$. The binary encoding of the individual blocks can be done independently, leading to the bin numbers given in the figure. It is advantageous for the decoding of the interpolation value if a rotation is performed before the encoding, resulting in bin numbers as given by fig. 5.11, because otherwise a subtraction has to be carried out. The interpolation decoding can be implemented as described in section 5.2.3.

5.3 Choice of Time Base Architecture

The most relevant characteristics of the different architectures are shown in table 5.1.

In the new TDC130, the main time base will consist of a DLL with a counter for dynamic range extension. Delay lines without feedback are discarded because they are not self-calibrating. Contrary to PLLs, DLLs are intrinsically first-order systems and thus stable. The advantage of the PLL, its capability of filtering jitter, would not be used, as a PLL is already needed to multiply the 40 MHz input clock up to 1.28 GHz, a value that leads to a useful bin size in a phase interpolating DLL or PLL. This clock-multiplying PLL can be used for jitter filtering.

The routing effort for an Array-DLL based TDC is judged to be excessive. Simulations show that the power required to distribute the DLL outputs to the hit registers is about the same as the power consumed by the DLL itself. In the prototype presented in chapter 7 the DLL output signals are routed at the pitch of the delay elements. Assuming the N-element DLLs of an array were placed in an $N \times M$ array, $M - 1$ additional wires were to be routed in the space between two neighbouring wires. This space however does not allow for routing of more signals. Using different layers of metal is not possible either, as their number is limited. Spreading the DLLs in a $1 \times (NM)$ row is not desirable. This would imply to spread the M-element secondary DLL across $N(M-1)$ delay elements of the main DLL, making it very vulnerable to process variations and crosstalk.

To further increase the resolution, either passive or DLL-adjusted delay lines will be used. Those techniques offer a low channel dead time. Their power dissipation de-

creases with decreasing activity on the channels, which is not the case for an Array-DLL.

Architecture	Resolution	Dynamic range	Dead time	Calibration	Power dissipation
Main time base architectures:					
Counter	T_{clk}	∞	0	none	high
Passive delay line	∞	T_{clk}	0	offline	low
Active delay line	gate delay	T_{clk}	0	offline	low
DLL	gate delay	T_{clk}	0	auto	medium
PLL	gate delay	T_{clk}	0	auto	medium
Array DLL	∞	T_{clk}	0	auto	very high
Fine interpolation techniques:					
Dual-slope	∞	OK	yes	offline	high
Vernier	∞	OK	yes	offline	low
Passive delay line	∞	OK	0	offline	low
DLL-adjusted delay line	∞	OK	0	auto	low

Table 5.1: Comparison of time base architectures
T_{clk} is a clock cycle of the reference clock.
'∞' and '0' mean the technique has no intrinsic limit. 'OK' means the dynamic range of the fine interpolation technique can reach the resolution of the main time base.
The power dissipation of an array of DLLs is high compared to other main time base architectures, but there is no need for a separate fine interpolation. However, the power consumption is higher than that of a DLL or PLL together with delay line base fine interpolation.

6 TDC130 Target Chip Architecture

6.1 Data Flow and Time Base Architecture

Based on the considerations stated above, it has been decided that the TDC130 (fig. 6.1) has a data driven architecture. The time base consists of a 32-element DLL with an element delay of 25 ps. The dynamic range is extended by a counter to comply with the ILC/CLIC requirements. Counter and DLL are clocked with a 1.28 GHz clock generated by an on-chip PLL. The external clock frequency is the LHC bunch crossing frequency, 40 MHz.

6.2 Channel Macro

For every channel, there is a bank of hit registers, which, at the occurrence of a hit, store the timestamp provided by the time base. A hit controller converts the incoming hit signal into a control signal for the hit register bank. For every edge whose timing is to be measured, the hit controller generates a pulse in the control signal. The controller is highly configurable and can make the hit registers capture leading edge, trailing edge or either edge of the input signal. If required, the channel hit controller can be disabled for each channel individually, so that regardless of the input signal, the hit registers will not store a timestamp. This is useful if a channel is not connected or the corresponding channel in the detector itself is known to be damaged. A disabled channel dissipates minimum power and does not require bandwidth at the readout. The hit register banks constitute a pipeline and buffer the data for a short period until the following stage is ready. Furthermore, the hit controller can operate the channel in a high precision mode or in a low precision, low power consumption mode. In the high precision mode, the data in the first hit register stage is always updated. This guarantees

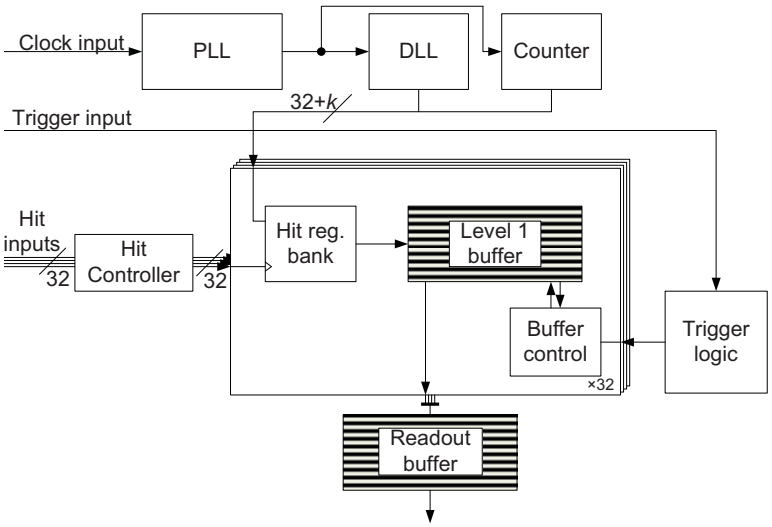


Figure 6.1: Target chip architecture

that the most recent timestamp is always available in the hit registers, at the price of additional power dissipation used to store data which might not be used. Once a hit arrives, the hit registers are switched from capture mode to store mode, the data is not updated anymore. If the hit arrives close to a change of timestamp, only those bits that have changed risk setup and hold problems. Only one bit of the data taken from the DLL changes at a time, as only one edge of the clock is used for the measurement. The resulting error is therefore limited to 25 ps. Once the stored hit timestamp is transferred to the next stage, the first hit register passes to capture mode again (fig. 6.2). This delay,

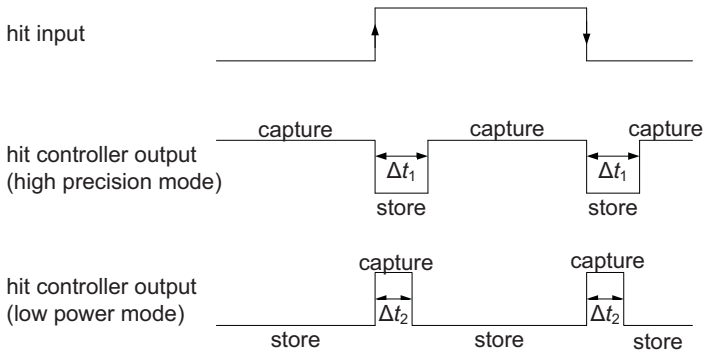


Figure 6.2: Hit controller output

Δt_1, does not influence the time measurement. Therefore, there is no need to keep it constant. In the low precision/low power mode, the hit registers are not updated when in idle mode, minimising the power dissipation of the first hit register stage. Once a hit arrives, they are switched into the capture mode for a fixed time Δt_2 before the data is stored. As it is the last timestamp captured which is stored, the hit seems delayed by Δt_2. Any variation of Δt_2 directly affects the time measurement.

6.3 Level-1 Buffer Organisation

After the data have been stored in the hit register banks, they are transferred into the level-1 buffer. While the hit registers are fully asynchronous, the level-1 buffer is synchronous to the logic clock. Therefore, synchronisation is required in the hit register pipeline. Some time bases provide data that is not binary encoded, as e.g. in a thermometer code. It is convenient to insert encoding logic before the level-1 buffer, as this minimises the width of words to be stored in this large buffer. If the data encoding is performed immediately before the level-1 buffer, the pipeline provides derandomisation that can also be used to cover the latency of the encoding logic. Another option is to insert the encoding logic, split into slices, between different pipeline stages. The encoding logic may also contain error detection or correction encoding such as a simple parity generation or hamming encoding to detect or prevent data losses within the level-1 buffer.

The level-1 buffer is a synthesisable standard memory with standard control logic. Tools generate a memory with the desired size using standard memory cells and control logic. It is therefore not necessary to deal with the detailed implementation of a basic memory cell. Two extreme cases of level-1 buffer organisation are possible, as well as a mixed architecture. One single buffer can be shared among all channels of a chip or each channel can have a dedicated level-1 buffer. Another option is to share i.e. 4 level-1 buffers among 8 channels each as in the HPTDC. The main criterion for this choice is the power consumption. If for the same power consumption, one of the options allows for a larger hit rate, this option will be preferred. The memory generator tools specify the power dissipation of a memory. For the target application, the geometrical size

is not of major importance. However, it should be remembered that the amount of connections to the 32 channels is considerable.

The data provided by the memory generator show that the synthesised buffers in 130 nm do not dissipate static power. One buffer of a given size S and activity f dissipates as much power as 32 buffers of the same size S, but with activity $f/32$. As the total amount of data to be stored is not influenced by the buffer organisation, the required size of 32 dedicated buffers for each channel will be $S/32$. However, in case of dedicated buffers per channel, no channel identifier is required in the level-1 buffer. Accessing a large memory takes more power than accessing a smaller memory, as the signal lines are longer and the parasitics increase. This leads to lower power dissipation in case of many small memories.

If one buffer is shared among multiple sources of data, it is a potential bottleneck. Especially in small detectors close to the interaction point, all data is likely to arrive virtually at the same time, while all channels will be idle between the bunches. This requires large derandomisation pipelines before the level-1 buffer if the operating frequency of the buffer and buffer control logic is to be kept low. In case of dedicated level-1 buffers per channel, the write access to the buffer is fully independent of the other channel's activity. The bottleneck is situated after the level-1 buffer that already provides large derandomisation and, in triggered mode, data reduction. Thus, one level-1 buffer per channel is advantageous from the hit rate point of view, too.

The memory control logic is geometrically small compared to the memory itself. Therefore, the geometrical size of one buffer is about the same as that of 32 buffers of $1/32$ size. For the total geometrical memory size, the penalty for having 32 small buffers is less than a factor of 4.

Table 6.1 shows key figures of the two implementations. One word corresponds to one timestamp of 27 b plus, in case of one buffer per chip, channel identifier of 5 b. One buffer per channel is preferred to one global level-1 buffer for all channels given the above-mentioned considerations. This is opposed to the implementation of the HPTDC in the ancient 250 nm-technology.

An additional benefit of a dedicated buffer per channel is that all hits are guaranteed to be in perfect time order in each level-1 buffer, easing the implementation of the trigger mechanism as shown in chapter 6.4.

	1 buffer per		Normalised value
	chip	channel	
Number of buffers	1	32	32
Total size	1024 words		1
Supply voltage	1.2 V		1
Hit rate	3 MHz/channel		1
Total power	5.6 mW	3.9 mW	0.70
Total geometrical size	0.110 mm^2	0.406 mm^2	3.69

Table 6.1: Comparison of different level-1 buffer organisation schemes.
The normalised value gives the ratio between the values for the 1 buffer per chip and the 1 buffer per channel scheme.

6.4 Trigger Mechanism

In many HEP applications, the amount of data read from the detectors is very high, but only a small fraction of the collisions lead to events that will be further analysed. In the LHC general-purpose detectors not more than 100 events per second can be stored for subsequent analysis. The event rate that is going to be observed at design luminosity is in the order 1 GHz, requiring a data reduction by seven orders of magnitude [Par08]. The principle of triggering has been largely adopted. A dedicated trigger system provides a trigger signal to the electronics. Hits for which no trigger has been received are discarded already inside the TDC.

The trigger detector, the generation and distribution of the trigger signal to the different sub-detectors introduce a delay, called trigger latency. Due to the finite speed of particles generated in a collision and then propagating through the detector, different detector delays and jitter, a trigger signal does not refer to a precise moment in time, but to a time interval called ‘trigger window’. Hits that arrived to the TDC within a configurable window a configurable time before the trigger signal has arrived to the same TDC are to be read out (fig. 6.3). This implies that the hit timestamps need to be stored in the level-1 buffers during the trigger latency. All hit timestamps are written into the level-1 buffer, but only those for which a trigger signal has been received are read out and passed to the following stages.

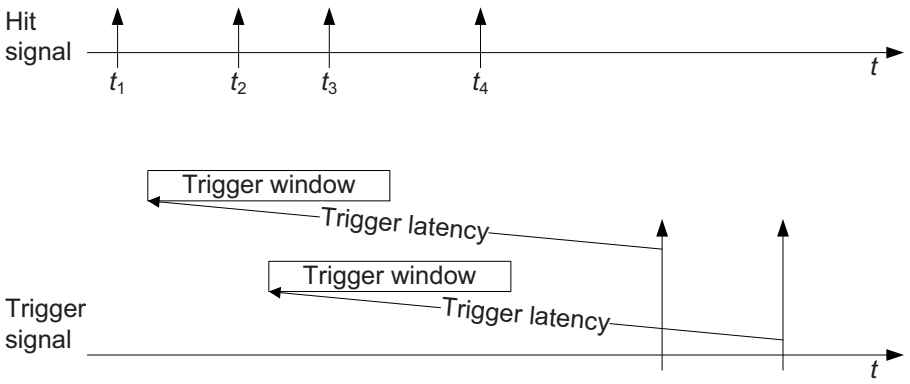


Figure 6.3: Triggering hits:
Hits arriving at t_2 and t_3 are selected by the first trigger signal and hits arriving at t_3 and t_4 by the second. The hit arriving at t_1 is not triggered and will be discarded.

The choice of the level-1 buffer organisation has implications on the trigger mechanism. As the level-1 buffers are dedicated to only one channel each, it is guaranteed that all timestamps are stored in perfect time order. The trigger logic needs to be instantiated once per level-1 buffer, thus once per channel. In triggered mode, a state machine receives the trigger timestamp, which does not need to have the same resolution as the hit timestamps. It controls the read pointer of the level-1 buffer. If the timestamp read out from the level-1 buffer is older than the trigger latency, a trigger for the corresponding hit would have been received already if the timestamp was to be read out. The timestamp is not passed to the following stage and the read pointer is incremented. If the timestamp is within the trigger window, it has to be transferred to the following stage and the state machine waits for the next stage to be ready to receive the timestamp before continuing. In pairing mode, the timestamp for leading and trailing edge of the hit input are read out as a pair before the timestamp of the trailing edge is eventually converted into a width information, i.e. the difference between leading and trailing edge is calculated. If the leading edge timestamp is triggered, the trailing edge timestamp is read out even if outside the trigger window. If the timestamp is that new that the trigger signal cannot have arrived at the TDC yet, the state machine waits until the trigger latency has elapsed before taking a decision. It is possible to have overlapping trigger windows. A timestamp can be triggered by multiple triggers

and consequently be read out multiple times. In all cases, if the buffer is empty, the advancement of the read pointer is suspended until a new timestamp has been written to the buffer. An ensemble of data belonging to the same trigger is called ‘event’.

It is worth noting that the trigger mechanism in a data driven architecture depends on the timestamps of hits read out from the level-1 buffer. Corrupted data can block the trigger mechanism. Error detection or correction in the level-1 buffer has to be implemented depending on the expected error rates. It is advisable to implement the trigger mechanism in a way that a single corrupted timestamp cannot block the state machine. Given that the time order is guaranteed, if a timestamp in the buffer is giving a time that is later than that of the following timestamp, which has been written more recently, the trigger state machine shall skip the concerned words rather than getting blocked. Note that without error detection or correction, it cannot be distinguished whether the first data word has been corrupted so that the timestamp is giving a bigger value or the second word is corrupted and giving a smaller value.

If the buffers were to be shared among multiple channels, the trigger mechanism would need to be more complicated as the timestamps in the buffers are not necessarily in time order: one channel could receive a hit before another, but be read out to the buffer later. Once the trigger mechanism would find a timestamp for which the trigger latency has not yet expired, it would need to check if this is true also for the next timestamps.

6.5 Channel Merging

Eventually, data from multiple channels have to be merged, as at least the readout is common for all channels. Merging data always presents a potential bottleneck. Only one source at a time can send data to the shared resource. This requires an arbitration system that passes an enable signal around all data sources that have data to send.

A convenient place for channel merging is right after the level-1 buffers. They provide sufficient storage capacity for derandomisation and the data read out has already been selected by the trigger mechanism, minimising the amount of data to be merged. Furthermore, a wait signal can easily be introduced to the trigger state machines.

If the bandwidth of the readout is smaller than the average bandwidth of all sources together, data loss is unavoidable. In this case, as much data belonging to the same event as necessary will be discarded and this data loss will be flagged by a special control word. This makes sure that a maximum number of events have all corresponding data while a minimum number of events have lost data and it is known which events are not complete. From a global detector point of view, it is unlikely that the same event is lost all over the detector, as the TDCs' buffer overflows are not synchronised.

6.6 Readout

Currently, the GBT[†], a set of new ASICs for high speed bidirectional optical links, is being designed at CERN. A macro that interfaces with the GBT will be available for integration in front-end ASIC such as the TDC130. The readout interface between TDC130 and this macro will be fixed once the macro is available.

The readout will contain a readout buffer that stores the data waiting to be read out. Prior to the buffer, data processing such as an INL correction look-up table or calculation of the pulse width base on leading and trailing edge timestamp can be included.

[†]GigaBit Transceiver

7 TDC130-0820 Prototype Chip Implementation

In this chapter, the detailed implementation of the TDC prototype is presented in a top-down hierarchical structure. Jitter and noise considerations for the whole chip are explained. Trade-offs across different blocks are frequently required.

7.1 Architecture

A prototype, called TDC130-0820, has been designed in order to evaluate new concepts and possible TDC improvements with respect to the predecessor. The main blocks of the TDC (fig. 7.1) are a phase interpolation DLL, a clock-multiplying PLL and hit registers for 44 channels with hit inputs, grouped into 8 different channel groups in order to reduce the number of pins and the silicon area, since the ASIC is pad limited. Readout and configuration is performed using shift registers. This chapter contains information on the detailed implementation of the prototype blocks.

7.2 DLL Implementation

The basic concept of a DLL as time base has been described in chapter 5.1.3. This chapter is focuses on the detailed implementation of the prototype in the IBM CMOS CMRF8SF 130 nm technology.

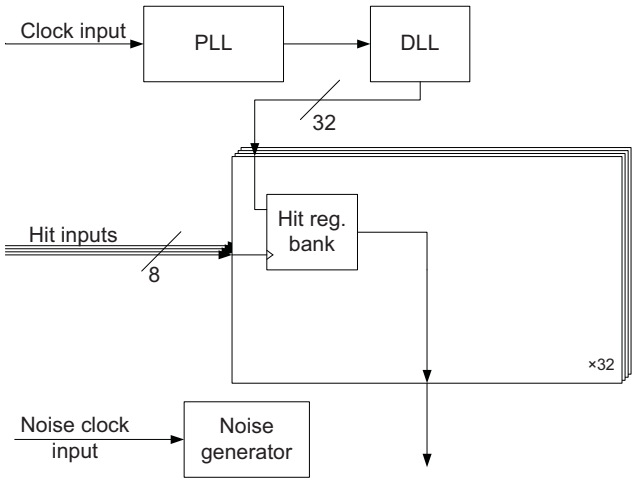
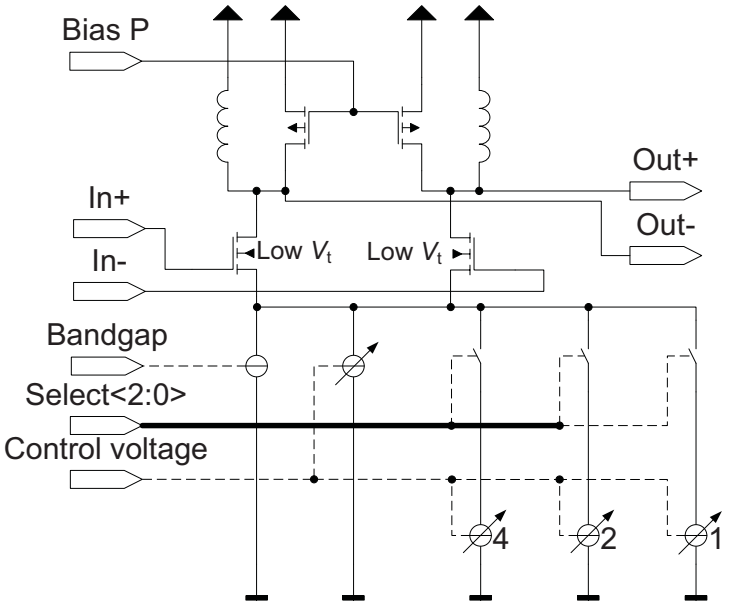


Figure 7.1: TDC130-0820 prototype architecture

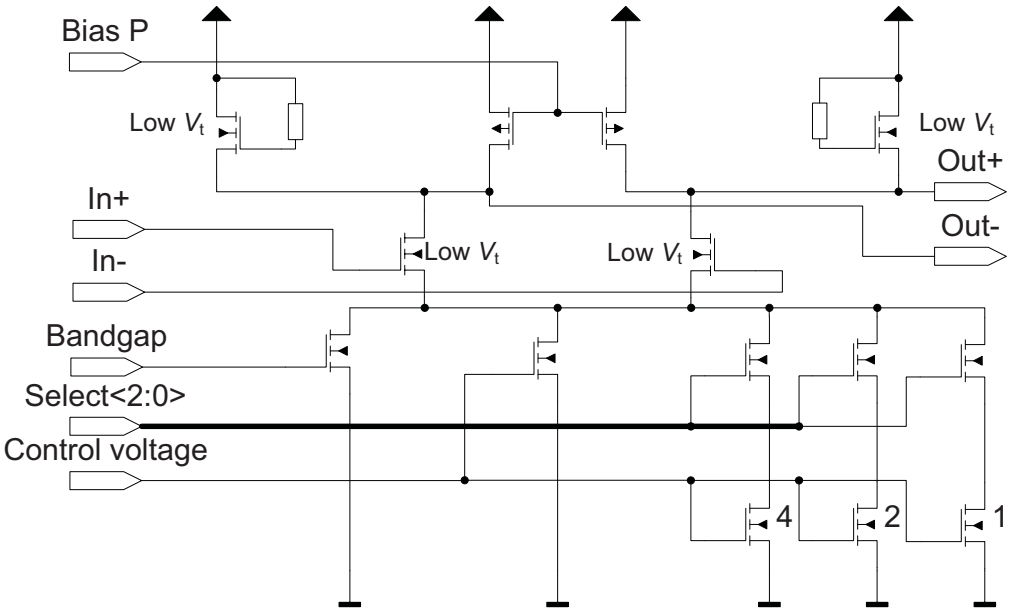
7.2.1 Delay Element

For the implementation of the delay line, two different kinds of delay elements are convenient to use, single-ended and differential buffers (fig. 5.3 in chapter 5.1.3) [Mor07]. As the gate delay of a technology depends strongly on parasitic effects, the precision of the TDC can be significantly degraded if related signals are subject to different mismatch effects. One example for such an effect is the polarity of the phase shifted VCDL output signals. If single inverters instead of buffers are used as delay element in a single-ended delay line, the bin size can be half as big. However, the sizes of two neighbouring bins are affected by independent mismatch mechanisms that cannot be compensated for easily. The rise time of the output of one element has to be equal to the fall time of the subsequent element. The same is true for the buffers that propagate the signal from the DLL to the hit registers, and while the threshold voltage of the NFETs is relevant for one signal, the other signal is sensitive to the threshold voltage of the PFETs.

These problems can be avoided using buffers as delay elements. In a single-ended implementation, buffers consist of two inverters in series. Their delay is to be compared to the delay of a single differential buffer. In differential implementations, a buffer is an elementary cell. When used in a voltage controlled delay line, the elements' delay must be adjustable. Differential buffers always contain a tail current source, and the



(a) Simplified narrow-band schematic, valid around the operating frequency



(b) Full schematic

Figure 7.2: Delay element

delay can be controlled adjusting the tail current. Standard single-ended inverters have a delay that is fixed by the dimensions of their transistors and their load. Additional series transistors have to be introduced to control the output current and thus their delay. This technique is called “current starving”. It further increases the delay.

Simulations of a single-ended buffer without current starving transistors show a delay of 45 ps when low V_t transistors are used. To accommodate process spread and supply variations, the nominal operating point must leave a margin both to higher and to lower delays, the nominal bin size exceeds 48.8 ps. To ease subsequent data processing, a constraint put on the TDC is that the bin size must be a binary fraction of the 40 MHz reference clock period, 25 ns. Therefore, if the nominal bin size exceeds 48.8 ps, the next bigger acceptable bin size is 97.8 ps, which has already been achieved in the HPTDC in a 250 nm technology [Chr04].

Differential buffers consisting of a differential pair, a tail current source and diode connected transistors as load elements show a delay of 32 ps under nominal conditions and less than 45 ps for worst case process, temperature and supply voltage. In order to reach a delay of 24.4 ps, a novel delay element (figs. 7.2, 7.3) is developed. The core of the delay element is a differential pair, with the gates connected to the input. The main tail current source is controlled by the DLL control voltage. To compensate for mismatch, three additional tail current sources are implemented and can be enabled after calibration. Their sizes are 25%, 12.5% and 6.25% of the main tail current source’s size. A constant tail current source reduces the slope of the delay vs. control voltage curve for low control voltages to reduce jitter. It assures that a minimum current is flowing through the differential pair even when to control voltage is lower than in nominal operation, so that a signal can propagate through the delay line even if the control voltage is unexpectedly low. It is controlled by an on-chip band gap voltage reference. The diode connected transistor load element is replaced by a source-follower transistor with series gate resistor. This circuit acts as an active inductor around the operating frequency and decreases the rise and fall time of the signal.

Fig. 7.3a shows the dependency of the delay on the control voltage for different simulation corners. In all cases except the pessimistic slow corner with 90% of the nominal supply voltage and maximum temperature, the delay element can reach a bin size of 24.4 ps. A typical chip with nominal supply voltage and operating temperature

can reach a delay of 19 ps. Fig. 7.3b shows the gain of a delay element for different clock frequencies. A clock frequency of 1.28 GHz corresponds to a bin size of 24.4 ps, 2.56 GHz to 12.2 ps. Simulations show that in this range, the frequency has virtually no effect on the gain of the delay element. For the control voltage corresponding to an element delay of 24.4 ps, the gain is 2.1. The larger the gain, the more the slew rate of the signal at the output is dominated by the delay element and not by the slew rate of the input signal. If the gain is smaller than 1, the signal is degrading whilst propagating along the delay line.

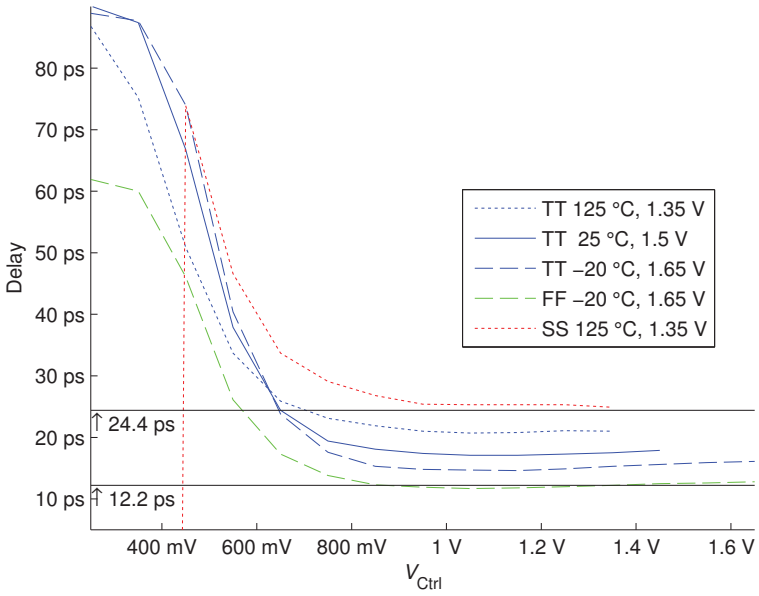
Active inductors

The circuit shown in fig. 7.4a can easily be implemented in a standard CMOS technology. The current source represents the remaining parts of a branch of a differential delay element. As the delay elements in a VCDL are closely spaced, area constraints and magnetic coupling between neighbouring elements prohibit the use of metal inductors. In fig. 7.4b, the transistor is replaced by its equivalent circuit.

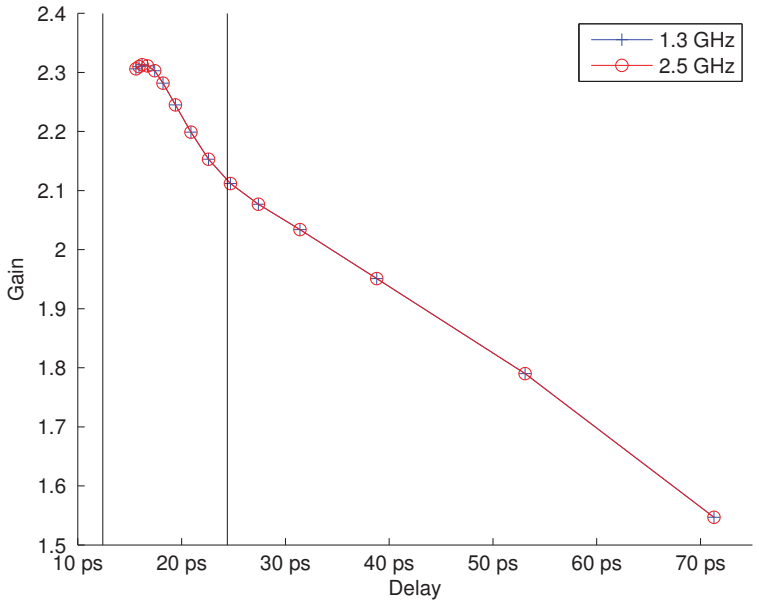
The impedance Z seen by the current source I is

$$\begin{aligned} Z &= \left(R+\frac{1}{\mathrm{j}\omega C_{\mathrm{GS}}}\right) \parallel \frac{1+\mathrm{j}\omega C_{\mathrm{GS}}R}{g_{\mathrm{m}}} \parallel R_{\mathrm{DS}} \\ &= \left(R+\frac{1}{\mathrm{j}\omega C_{\mathrm{GS}}}\right) \parallel \left(\frac{1}{g_{\mathrm{m}}}+\mathrm{j}\omega L_{\mathrm{eq}}\right) \parallel R_{\mathrm{DS}} \text{ with } L_{\mathrm{eq}} = \frac{C_{\mathrm{GS}}R}{g_{\mathrm{m}}} \end{aligned}$$

Therefore, the source-follower transistor with series gate resistor can be represented by the circuit in fig. 7.4c. This compares to the impedance $Z = \mathrm{j}\omega L$ of an ideal inductor, and the impedance $Z = R+\mathrm{j}\omega L$ of an inductor with resistive wires as it is available in CMOS technologies.

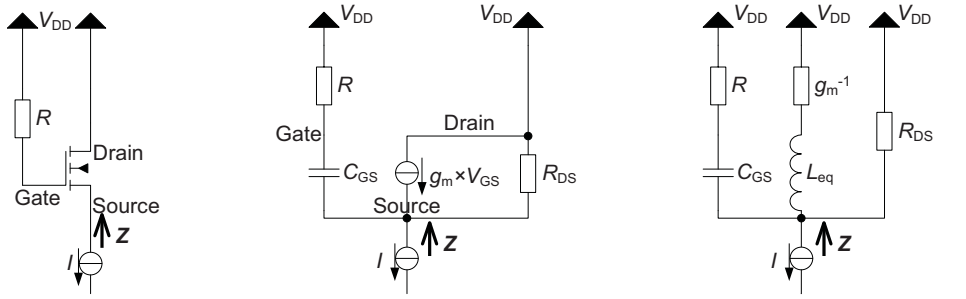


(a) Delay vs. control voltage



(b) Delay element gain (typical corner) for 1.28 GHz and for 2.56 GHz signals

Figure 7.3: Delay element simulation



(a) Implementation (b) Equivalent schematic (c) Equivalent schematic

Figure 7.4: Schematic of an active inductor (with $L_{\mathrm{eq}} = \dfrac{C_{\mathrm{GS}}R}{g_{\mathrm{m}}}$)

The impedance has one pole and one zero:

$$\begin{aligned}
\omega_{\mathrm{Zero}} &= \frac{1}{C_{\mathrm{GS}}R} \\
\omega_{\mathrm{Pole}} &= \frac{1+g_{\mathrm{m}}R_{\mathrm{DS}}}{C_{\mathrm{GS}}(R+R_{\mathrm{DS}})} \\
&\overset{R\gg R_{\mathrm{DS}}}{\approx} \frac{1+g_{\mathrm{m}}R_{\mathrm{DS}}}{C_{\mathrm{GS}}R} = \omega_{\mathrm{Zero}}\left(1+g_{\mathrm{m}}R_{\mathrm{DS}}\right) = \frac{1}{C_{\mathrm{GS}}R} + \frac{R_{\mathrm{DS}}}{L_{\mathrm{eq}}}
\end{aligned}$$

7.2.2 Phase Detector

The phase detector compares the phase difference between the beginning and the end of the delay line. It generates a control signal indicating whether the loop delay is too big or too small. Two different phase detector implementations are to be considered, the XOR phase detector, based on an XOR gate (appendix B.1, p. 115), and the bang-bang phase detector, based on a D flip-flop (fig. 7.5) [RCN03, Mor07].

Bang-Bang Phase Detector

A D flip flop (DFF) (fig. 7.5) can serve as a DLL phase detector. Using the VCDL output signal to sample the VCDL input signal, the output of the phase detector gives the sign of the phase difference. The edge on the VCDL output is either early or late with respect to the corresponding edge of the VCDL input signal (fig. 7.6(a) and (b)).

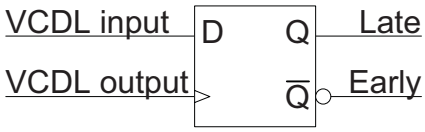


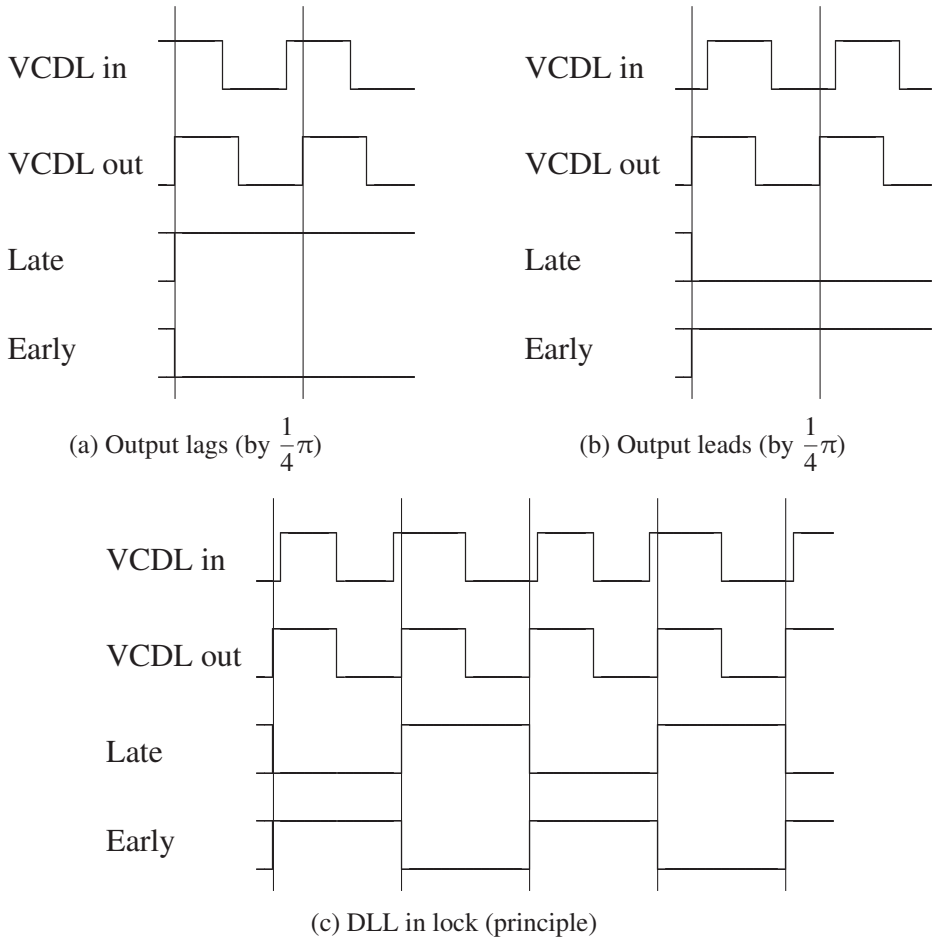
Figure 7.5: D Flip-Flop as Phase Detector

If the leading edge of the VCDL input is earlier than that of the preceding cycle of the VCDL input, the flip flop samples a 'low' level on the input. The pulse width of the VCDL input signal defines how much earlier the input leading edge may arrive before the phase detector wrongly signals that the input signal is late. For a duty cycle of 50%, the operating range of the bang-bang phase detector extends from $-\pi$ to $+\pi$ (fig. 7.7). For a duty cycle of $D = 25\%$, input signal is 'high' $D = 25\%$ of the period and 'low' the remaining $1 - D = 75\%$. The operating range is consequently shifted and spans from $-2\pi D = -\frac{1}{2}\pi$ to $+2\pi(1-D) = +\frac{3}{2}\pi$. The duty cycle of the VCDL output does not influence the phase detector behaviour, as the only the leading edge is used.

If the output edge is late, the loop will reduce the delay of the VCDL, and eventually the output edge will be early. In lock, the output of the phase detector changes virtually every clock cycle, so that the phase error becomes 0 in average (fig. 7.6c).

As the phase detector is clocked by the VCDL signal, its output can change only once per VCDL clock cycle. Thus, its maximum frequency is half the frequency of the VCDL signal and the minimum pulse length is the VCDL clock period. This is convenient if a charge pump based loop filter is used.

A D flip flop is a complex circuit – it contains multiple elementary circuits and has memory. The phase detector is supposed to generate a signal that depends on the sign of the phase difference of the input signals, but in fact, special care is required to avoid that it also depends on e.g. on the previous state and the magnitude of the phase difference. Moreover, the output signal has to be valid as quickly as possible after the rising edge of the VCDL output has arrived. If the delay to the output equals one clock cycle or more, the loop reaction is always at least one cycle late with respect to the input signals, leading to additional jitter. A D flip flop implementation with special consideration on load balancing is used (fig. 7.8). It contains 3 RS latches, an AND gate, implemented as a series of a NAND gate and an inverter, input inverters, and dummy circuitry to ensure that all delays of corresponding signals are as equal as possible. All corresponding



(a) Output lags (by $\frac{1}{4}\pi$)

(b) Output leads (by $\frac{1}{4}\pi$)

(c) DLL in lock (principle)

Figure 7.6: Bang-Bang Phase Detector: Pulse Diagrams

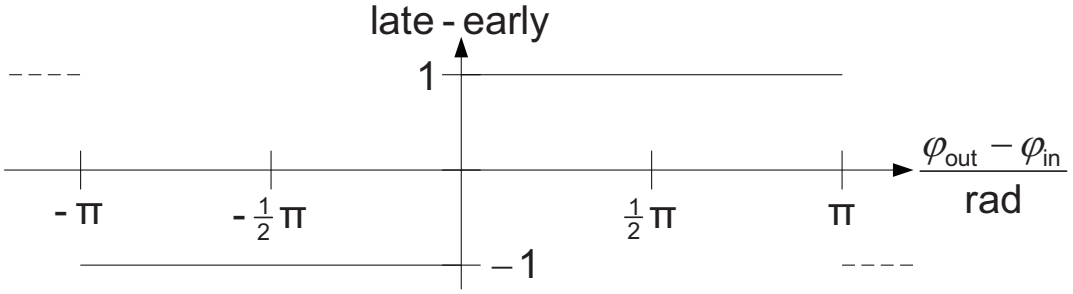


Figure 7.7: Bang-Bang Phase Detector: Transfer function for $D = 50\%$ duty cycle

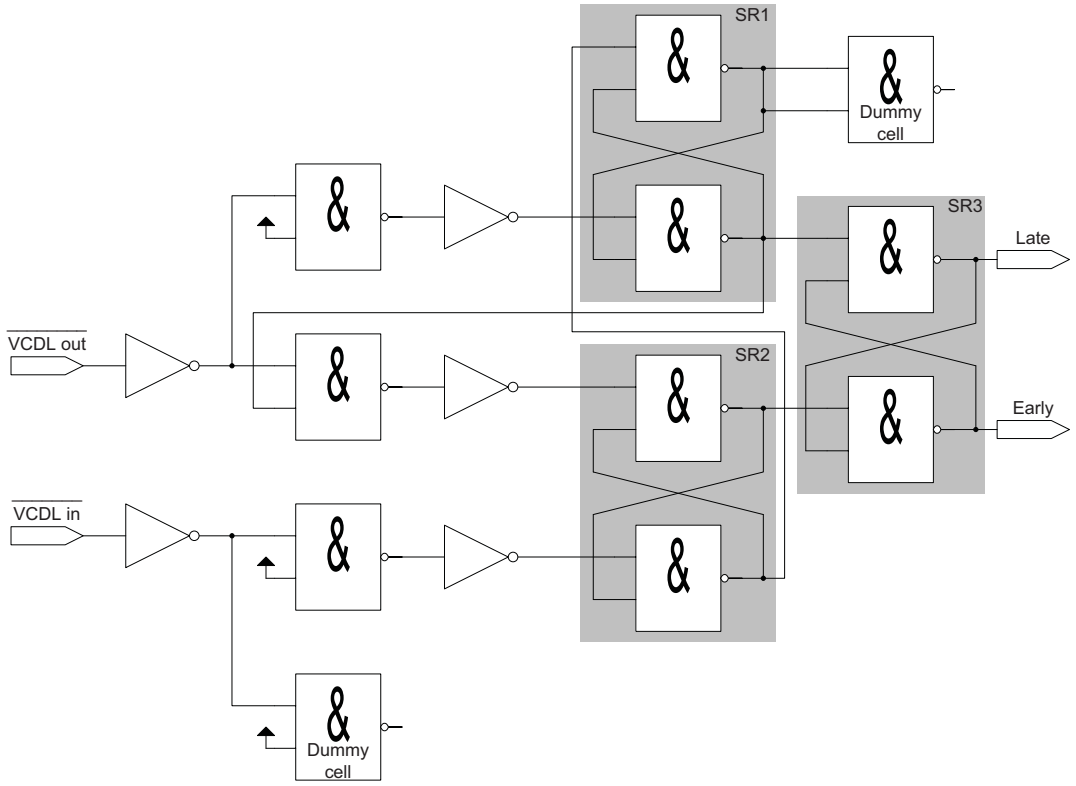


Figure 7.8: Balanced implementation of a D flip-flop

nodes have the same fan-out and fan-in. All gates are designed such that they have equal driving capability. All corresponding wires have the same physical surface, and thus the same parasitic capacitance. In addition, their wire lengths, thus the wire resistances, are as equal as possible, so that the *RC* delay of corresponding wires is virtually identical.

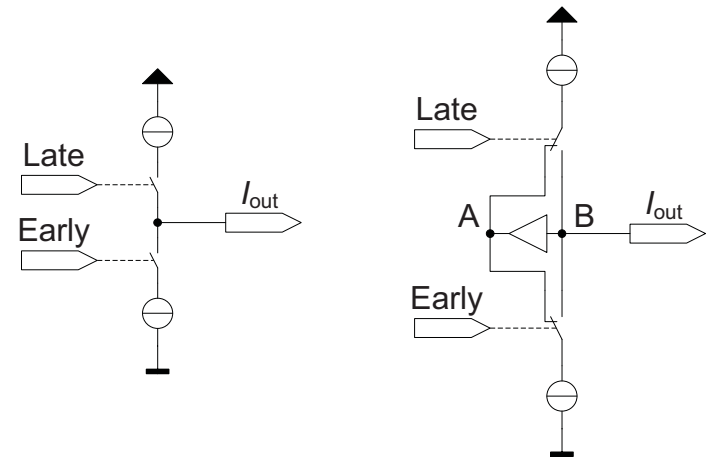
The bang-bang phase detector has a number of advantages compared to the XOR phase detector. However, the choice of phase detector should be made also considering the properties of the loop filter that is most appropriate for the corresponding phase detector.

7.2.3 Loop Filter

While both phase detector implementations have a digital output, the delay line is controlled by an analogue voltage. The conversion is performed by a low pass filter, called

loop filter. The two most promising options are a RC filter followed by and amplifier (appendix B.2, p. 117), and a charge pump with a filter capacitor [Mor07].

Charge Pump and Filter Capacitor



(a) Standard implementation

(b) Optimised implementation: Points A and B always at equal potentials

Figure 7.9: Charge Pump

A charge pump connected to a capacitor can be used as loop filter. Depending on the input signal, the charge pump sinks or sources a current I_{CP} (fig. 7.9a). The bang-bang phase detector provides the required control signals 'Late' and 'Early'. The voltage at a capacitor can be described by

$$V_C(t) = \frac{1}{C}\int_0^t I(t)\mathrm{d}t + V_C(t=0) = \frac{Q(t)}{C}$$

Together with the clocked nature of the phase detector outputs, this explains the name 'charge pump' instead of current source. The current sources provide a constant current, but as this current is provided during a constant time, the result of a phase detector decision is the injection or extraction of a constant charge to or from the capacitor.

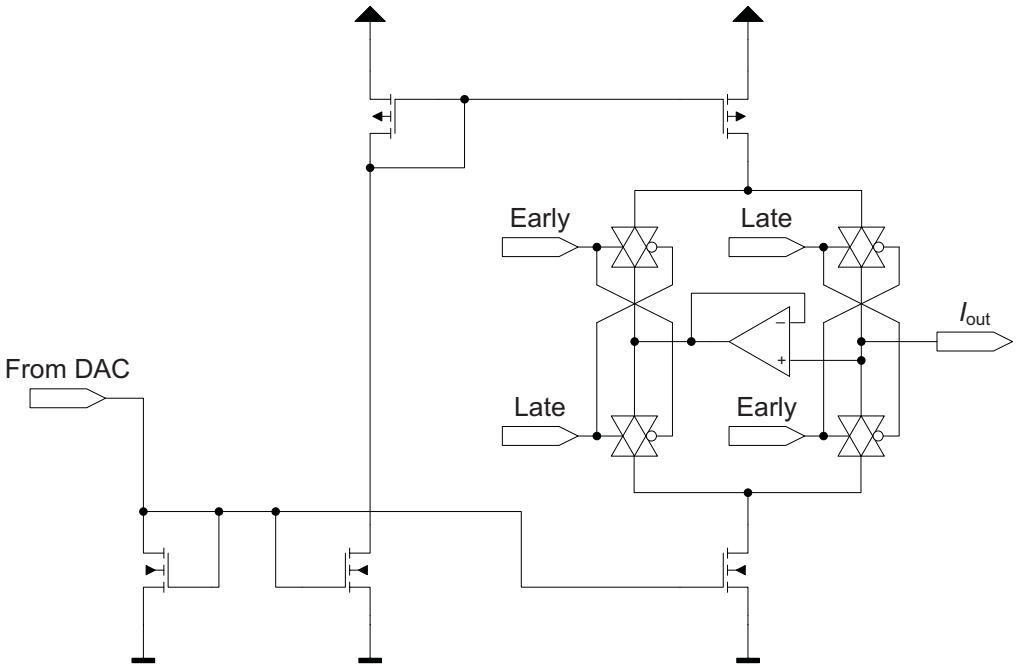
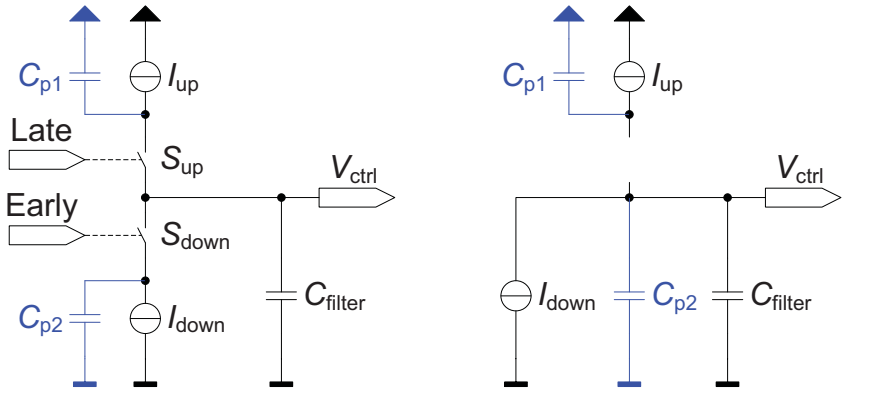


Figure 7.10: DLL charge pump: Schematic

Fig. 7.9a illustrates the intended behaviour of the charge pump. The disadvantage of this circuit is that when the charge pump switches the direction of the current, the output node gets disconnected from one current source and connected to the other one. As the current source outputs are fully independent, the potential of the outputs will be different, and the potential of the I_{out} output node will jump when the phase detector outputs change. This can be avoided using a voltage follower that keeps the output (point A in fig. 7.9b) of the unused current source always at the potential of the I_{out} output (point B), without loading the output itself. Fig. 7.10 shows a more detailed schematic of the charge pump.

The effect that causes voltage jumps at the output of the charge pump even when it is connected to the filter capacitor is called charge sharing. Fig. 7.11a shows the schematic of the simple charge pump (fig. 7.9a) with parasitic capacitors C_{p1} and C_{p2} at the outputs of the current sources and the filter capacitor C_{filter}. Assume that S_{up} is closed and S_{down} open. The parasitic capacitor C_{p2} is now being discharged by the current sink, while the current source charges C_{filter}. $C_{\text{p1}} \ll C_{\text{filter}}$ is negligible. Let



(a) General schematic

(b) Switch S_{down} closed to decrease the control voltage

Figure 7.11: Charge pump with parasitic capacitors C_{p1} and C_{p2} (in blue) and filter capacitor

$Q_{p2} = C_{p2} V_2$ be the charge at the parasitic capacitor C_{p2} and $Q_{filter,0} = C_{filter} V_{ctrl,0}$. Assume now that S_{up} opens and S_{down} closes (fig. 7.11b). The previously discharged C_{p2} is now parallel to C_{filter}, forcing the voltages of the parasitic capacitor C_{p2} and of the filter capacitor C_{filter} to equalise to a new control voltage $V_{ctrl,1}$. The charges on the capacitors move into equilibrium:

$$\begin{aligned}
\left(Q_{p2} + Q_{filter}\right) &= \left(C_{p2} + C_{filter}\right) V_{ctrl,1} \\
&= \left(C_{p2} V_2 + C_{filter} V_{ctrl,0}\right) \text{ with charge conservation} \\
\Longrightarrow V_{ctrl,1} &= \frac{C_{p2}}{C_{p2} + C_{filter}} V_2 + \frac{C_{filter}}{C_{p2} + C_{filter}} V_{ctrl,0} \\
\Longrightarrow V_{ctrl,1} &= V_{ctrl,0} \text{ if } V_2 = V_{ctrl,0}
\end{aligned}$$

Thus, charge sharing can be avoided keeping the outputs of both current sources at the same potential, e.g. using a voltage follower.

	XOR Phase Detector Passive RC filter		Bang-Bang Phase Detector Charge Pump Filter Capacitor
+	simple	–	more complicated
–	sensitive to duty cycle distortion, gain saturation	+	no gain saturation, duty cycle distortion at VCDL input creates an asymmetry in the transfer function duty cycle distortion at VCDL output has no effect
–	cannot be designed to work at 0 phase offset	+	operates at 0 phase offset by default
+	information about phase error magnitude	–	no information about phase error magnitude
–	no information about sign of phase error	+	information about sign of phase error
–	finite gain $\frac{V_{DD}}{\pi}$	+	infinite gain
–	small operating range $0 \ldots \pi$	+	large operating range $-\pi \ldots +\pi$
+	keeps control voltage constant at 0 phase error	–	always increases or decreases the control voltage, cannot keep it constant

Table 7.1: Comparison of DLL Phase Detector and Loop Filter combinations

7.2.4 Choice of Phase Detector and Loop Filter

The comparison of the two possible combinations of phase detector and a loop filter is summarised in tab. 7.1. As the bang-bang phase detector with a charge pump and a filter capacitor is basically insensitive to duty cycle distortion, one of the most frequent signal distortions, and has a much larger range of operation and infinite gain, it is implemented in the prototype.

7.2.5 Transfer function

The transfer function describes the DLL behaviour in the frequency domain and allows considering a DLL as a fundamental block, a black box. Let us recall the behaviour of the different functional blocks of a DLL:

- The VCDL translates the control voltage V_{ctrl} into a proportional delay[†]. The proportionality constant is the VCDL gain K_{VCDL}.
- The phase detector converts the phase error $\varphi_{\text{out}} - \varphi_{\text{in}}$ into an output setting the charge pump duty cycle to $\frac{\varphi_{\text{out}}-\varphi_{\text{in}}}{T_{\text{clk}}}$.
- The charge pump converts the phase detector output into a current with instantaneous value $\pm I_{\text{CP}}$ and average $\frac{\varphi_{\text{out}}-\varphi_{\text{in}}}{T_{\text{clk}}} I_{\text{CP}}$.
- The loop filter, a capacitor, integrates this current, thus accumulates a charge $\frac{\varphi_{\text{out}}-\varphi_{\text{in}}}{T_{\text{clk}}} I_{\text{CP}} \frac{1}{s}$. The control voltage $V_{\text{ctrl}} = \frac{\varphi_{\text{out}}-\varphi_{\text{in}}}{T_{\text{clk}}} I_{\text{CP}} \frac{1}{s} \frac{1}{C}$ is the voltage across this capacitor.

The output signal is the phase $\varphi_{\text{out}}(s)$, which equals the control voltage multiplied by the VCDL gain:

$$\varphi_{\text{out}}(s) = \frac{\varphi_{\text{out}}(s) - \varphi_{\text{in}}(s)}{T_{\text{clk}}} \frac{I_{\text{CP}}}{sC} K_{\text{VCDL}}$$

The transfer function $H(s)$ is defined as the ratio between output and input signal. Here, the signal is the phase:

$$H(s) = \frac{\varphi_{\text{out}}(s)}{\varphi_{\text{in}}(s)} = \frac{1}{1 + \frac{s}{\omega_{\text{n}}}}$$

where $\omega_{\text{n}} = \frac{I_{\text{CP}} K_{\text{VCDL}}}{TC}$ is the natural frequency of the only pole. A DLL is, as a first order system, intrinsically stable. However, parasitics introduce undesired higher order poles, which are irrelevant if their frequencies are much higher than ω_{n}.

Designing a DLL thus means choosing the parameters that fix ω_{n}:

[†]This simplification is only valid around the operating point, which is where the transfer function needs to be known.

- K_{VCDL} is fixed as soon as the technology is chosen and the delay elements are designed. This is usually the first step of a TDC design, as the element delay is the critical parameter.

- T is the period of the signal propagating in the delay line, and fixes the bin size t_{LSB} of the TDC. Therefore, it cannot be used to optimise the DLL behaviour.

- I_{CP} and C can be chosen to optimise ω_{n}. The charge pump can be designed to have I_{CP} adjustable. C is usually not adjustable.

7.2.6 Limitations and Sources of Errors, Jitter

In this paragraph, only the jitter mechanisms that are particular and intrinsic to the DLL operation with a bang-bang phase detector are treated. Other sources of jitter, noise and crosstalk are considered at system level in a dedicated chapter (ch. 7.6).

A bang-bang phase detector cannot keep the control voltage constant [Mor07]. If the output is exactly one cycle delayed with respect to the input, the phase detector will make the circuit leave this optimum into one direction, generating a phase error that will be corrected in the next cycle and so on. The control voltage is oscillating around the optimum value, and ideally, the phase detector outputs change every clock cycle. Due to imperfections, such as unbalance in the charge pump, jitter or imperfections in the phase detector, it is likely that the phase detector outputs are frequently constant for two consecutive clock cycles. Therefore, the maximum amplitude of the control voltage oscillations is $2\,\frac{I_{\mathrm{CP}}}{C}\,T_{\mathrm{clk}}$, giving a peak to peak value of $4\,\frac{I_{\mathrm{CP}}}{C}\,T_{\mathrm{clk}}$. For a TDC, the relevant information is not the magnitude of oscillation of an internal control voltage, but the tracking jitter $J_{\mathrm{Track,pp}}$, the peak to peak value of the variation of the bin size t_{LSB}:

$$J_{\mathrm{Track,pp}} = 4\,K_{\mathrm{VCDL}}\,\frac{I_{\mathrm{CP}}}{C}\,T_{\mathrm{clk}}$$

With an adjustable charge pump, the charge pump current I_{CP} can be made high for fast lock acquisition, and reduced for low tracking jitter once the DLL is in lock. In tracking mode, $J_{\mathrm{Track,pp}}$ should not exceed $\frac{1}{4}\,t_{\mathrm{LSB}}$.

7.2.7 Start-up Procedure

The DLL has two potential problems at start-up, which need special consideration. A DLL can lock to multiples of the clock period. At start-up, the delay of the VCDL can be initialised such that the phase detector indicates that the delay must be reduced, even though it has to be increased by more than half of the clock period. Let's analyse these two problems in detail:

Lock to Multiples of the Clock Period

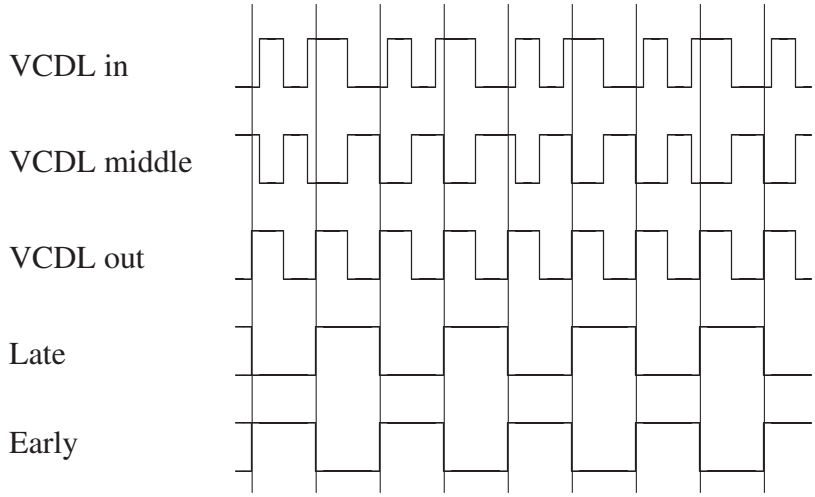
As only the beginning and the end of the VCDL are connected to the phase detector, the DLL might lock to a total delay that is a multiple of the clock period. In this case, multiple clock cycles propagate inside the delay line at the same time, and the resolution is not $\frac{1}{32}$ of the clock period, but a multiple of it (fig. 7.12).

This problem is avoided if the DLL is sure to start always too fast, i.e. the total delay of the VCDL is smaller than a clock cycle. In this case, the first possible state to lock to is the desired one. To implement this, it is sufficient to charge the loop filter capacitor at reset to the control voltage that corresponds to the smallest possible delay. In the case of the TDC130-0820, the delay decreases with increasing control voltage, and the filter capacitor is charged to the supply voltage at reset.

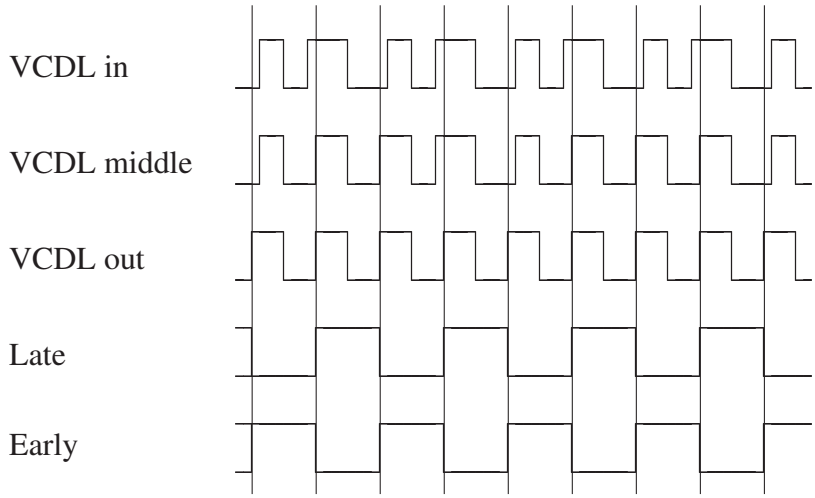
Wrong Phase Detector Indication

Another potential problem is that when starting the DLL, its delay might be so small that when the rising edge of a pulse arrives to the end of the line, the falling edge has not yet arrived to the beginning of the line. In this case, the phase detector, knowing the state of the beginning and the end of the line, but not the state of the intermediate elements, will give a 'late' indication, and make the charge pump try to further decrease the delay, which is already minimum. The 'late' indication will not disappear, as the clock pulse cannot reach the end of the line before it reaches the beginning.

As the DLL is known to start with minimum delay, it is also known that the phase detector should give an 'early' indication just after the reset. If this is not the case, then the DLL is a lot too fast, and the delay should be increased until the phase detector gives



(a) DLL locking to nominal delay (principle)
VCDL middle is in phase opposition to VCDL input and output.



(b) DLL locking to twice the nominal delay (principle)
VCDL middle is in phase with VCDL input and output. All other signals are exactly like in (a) when locking to nominal delay.

Figure 7.12: A DLL can lock to multiples of the nominal delay. The timescale is the same for both diagrams. VCDL middle is the signal at the VCDL tap half way between input and output. It is not visible to the phase detector, thus not used for controlling the loop

a consistent ‘early’ indication. A state-machine has been synthesised to implement this functionality (fig. 7.13).

7.3 PLL Implementation

The DLL requires an input clock frequency of $\frac{1}{N \cdot t_{\mathrm{LSB}}} = 1.28\,\mathrm{GHz}$, with $N = 32$ DLL elements and an element delay $t_{\mathrm{LSB}} = 24.4\,\mathrm{ps}$. However, the only external clock frequency that is available to all LHC detectors has a frequency of 40 MHz. A Phase Locked Loop can be used not only to provide a time base (ch. 5.1.4), but also to generate an internal clock based on an external reference with a different frequency. Recalling fig. 5.5 (p. 33), a VCO generates a clock and the loop, consisting of a frequency divider, a phase detector, a charge pump and a low pass filter assures that the divider output frequency is always equal to the reference clock frequency, and a fixed phase relation between those two clocks is maintained. This leads to a stable VCO output frequency, which is larger than the external reference clock frequency by a factor equal to the division ratio.

7.3.1 Voltage Controlled Oscillator

The VCO generates an output frequency that is a function of a control voltage, pretty much the same as the VCDL of a DLL generates a delay as a function of a control voltage. However, the VCO generates its own signal with its own frequency and jitter characteristics, while the VCDL delays an external signal, but does not change its characteristics other than the phase. As in the DLL, a phase detector followed by a filter will generate the control voltage as a function of the phase difference of the external reference signal and the output of, in the PLL case, the VCO. This voltage controls the frequency, not the phase of the VCO.

The VCO (fig. 7.14a) is implemented as a ring oscillator with differential buffers (fig. 7.14b). The delay of each buffer, and thus the frequency of the ring oscillator, depends on the tail current in the buffers, which is controlled by the control voltage. Bias circuitry generates the bias voltage for the NMOS load transistors. As the

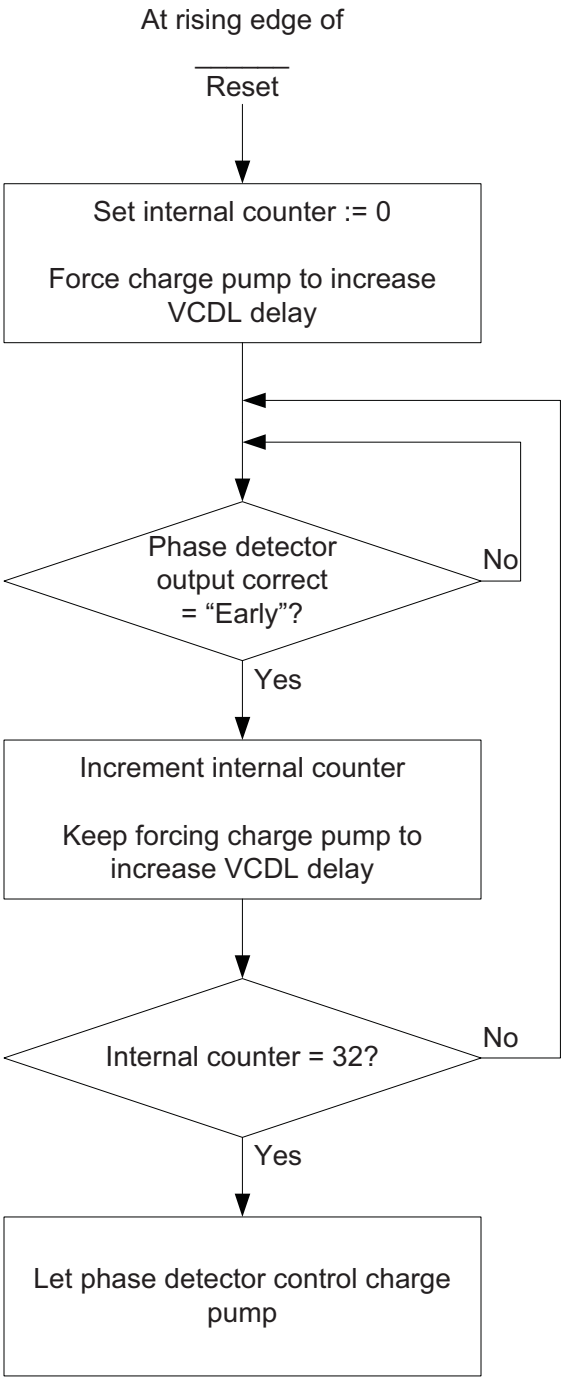


Figure 7.13: DLL Start-up State Machine

phase frequency detector inputs are single-ended, differential to single-ended conversion (fig. 7.14c) is necessary.

7.3.2 Phase Detector

The role of the phase detector in a PLL is basically the same as in a DLL. It provides an output signal as a function of the phase difference on its input. A fundamental difference between PLL and DLL is however that the phase detector output signal is used to generate a voltage that controls the frequency of the VCO, and not the phase to which the phase detector is sensitive. In other words, the loop adjusts the VCO frequency such that the phases of the VCO output and the external reference match. As the phase is the integral of the frequency, this is equivalent to considering the VCO as an integrator, introducing a new pole into the transfer function.

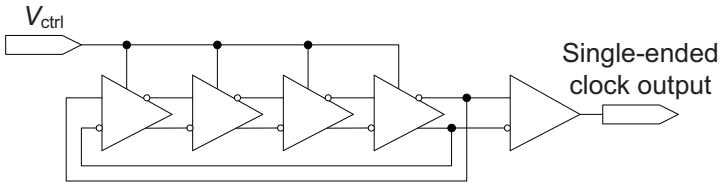
In this chapter, for simplicity of the phrasing, VCO output frequency actually refers to the frequency of the VCO output signal after frequency division, as it is this signal that is an input to the phase detector.

Two options are considered, the Analogue Multiplier (appendix B.3, p. 118) and the Phase-Frequency Detector [RCN03, Mor07].

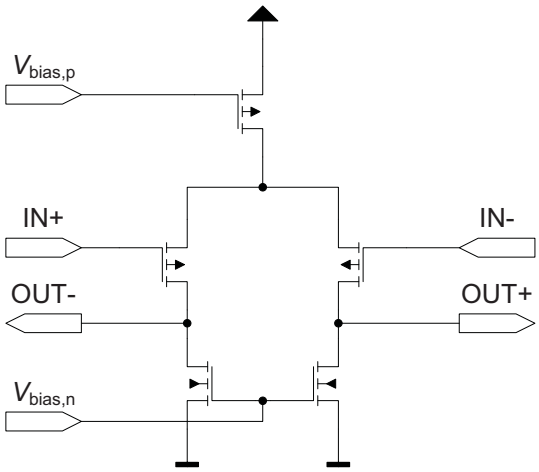
Phase-Frequency Detector

The Phase-Frequency Detector (PFD) consists of 2 D flip flops (fig. 7.15). One D flip flop gets set by the rising edge of the external reference clock, the other one by the rising edge of the VCO output clock. They are reset once both of them are set at the same time. On the contrary to the analogue multiplier phase detector, the phase-frequency detector is sensitive both to frequency and to phase, even when not part of a feedback loop.

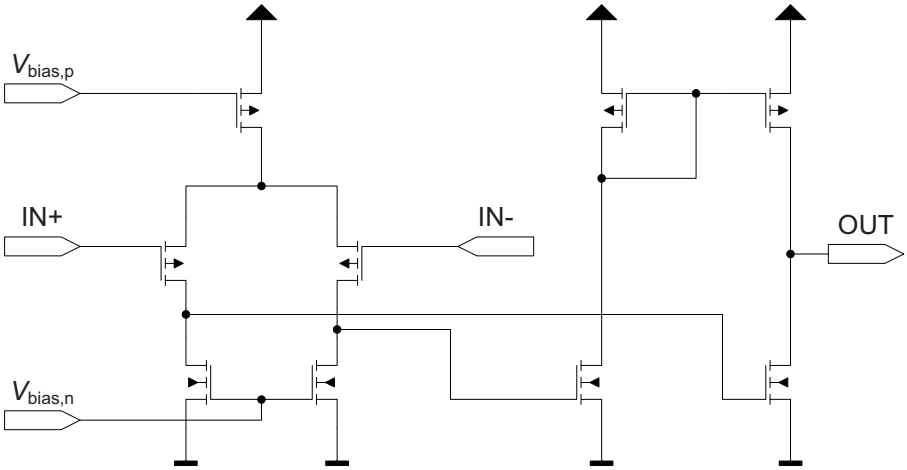
When the frequencies of the VCO and the external reference match, the phase frequency detector acts as a phase detector (fig. 7.16). The pulse diagrams show the error signal, which is the difference late $-$ early. The time it takes to reset the phase frequency



(a) Block schematic



(b) Delay element



(c) Differential to single-ended converter

Figure 7.14: VCO

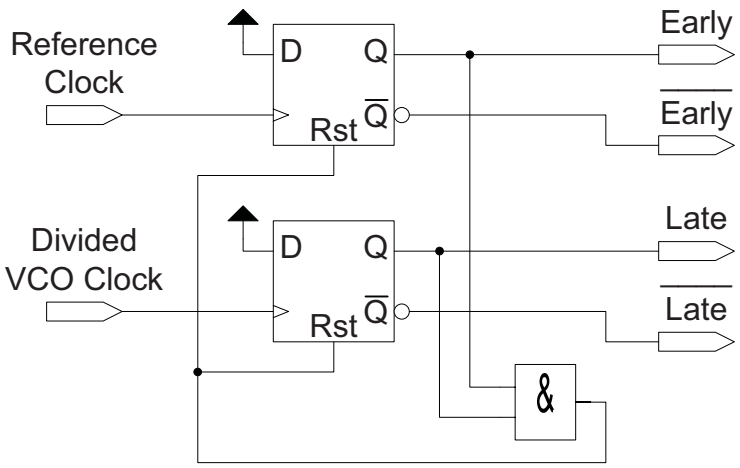


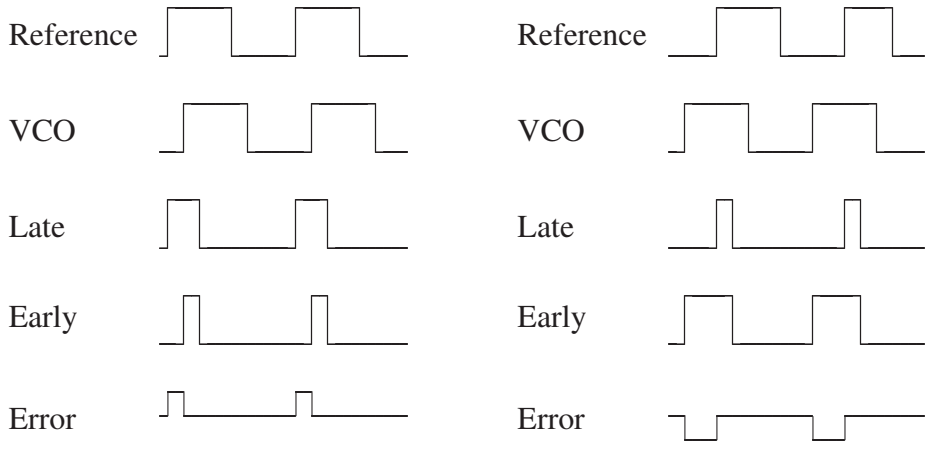
Figure 7.15: PLL Phase-Frequency Detector (simplified)

detector is exaggerated in the diagrams, as it usually too short to be visible. Neglecting this time, the average of the error signal is proportional to the phase difference:

$$\text{average}(\text{late} - \text{early}) = \frac{\varphi_{\text{Ref}} - \varphi_{\text{VCO}}}{2\pi}$$

When the frequency of the VCO does not match the external reference frequency, the phase frequency detector acts as a frequency detector (fig. 7.17). If the frequency of the VCO is different from the reference clock frequency, also the leading edges of the VCO signal are not as frequent as those of the reference signal. Consequently, one of the PFD flip flops is often clocked more than once before a reset is generated. This leads to a consistent ‘late’ signal if the VCO is slow or ‘early’ signal if the VCO is fast with respect to the reference clock. Only at the beginning of operation and during the resets, the opposite signal can be activated for a short time. Even if at the start of operation, the VCO signal is DC, the PFD gives the correct ‘late’ indication, regardless of whether the VCO output is low or high. As the PFD always gives the correct indications, disregarding possible errors in the very first cycle after start-up, a PLL using a PFD does not need any dedicated start-up logic to force the VCO to get close to the reference frequency.

It is worth noting that in frequency acquisition mode, the PFD output gives information on the frequency, and not the phase. Thus, the VCO does not perform an integration



(a) VCO lags (by $\frac{1}{4}\pi$) (b) VCO leads (by $\frac{1}{2}\pi$)

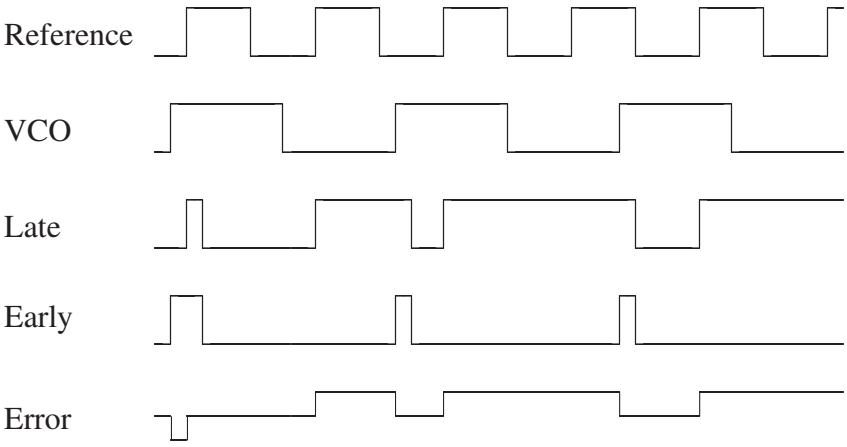
Figure 7.16: Phase Frequency Detector: Pulse Diagrams with phase error
Error = Late – Early

of the signals' frequencies, and does not introduce a pole to the transfer function. The PLL is a first order system during frequency acquisition.

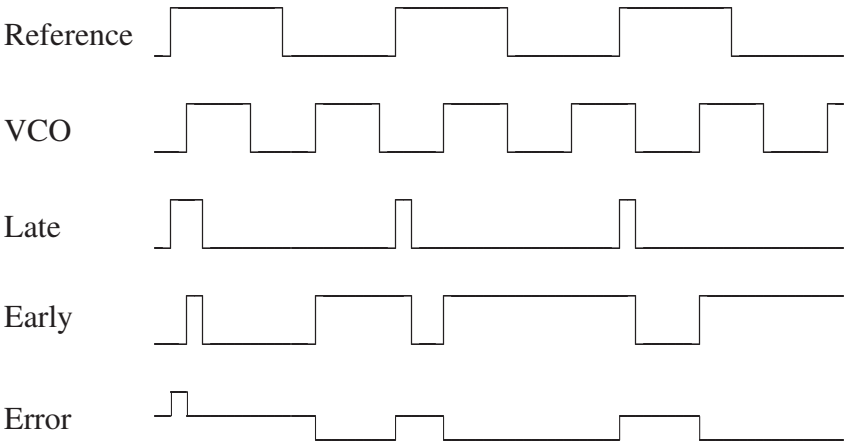
Again, both types of phase detector require different loop filters and the choice should be based on the combined properties.

7.3.3 Loop Filter

As a DLL, a PLL needs a loop filter to convert the digital phase detector output into an analogue control voltage. This filter needs to have low pass characteristics. In a PLL, the filter output voltage controls the VCO. Contrary to a DLL, the loop of a PLL contains two poles. One is introduced by the VCO and one by the low pass loop filter. The case of a PFD in frequency acquisition mode is disregarded, as this mode is only used for a short time after start-up. As second order systems can be unstable, it is desirable to introduce a zero intro the loop transfer function to compensate one pole. Therefore, loop filters for PLLs are different from DLL loop filters. Two possible



(a) VCO slow



(b) VCO fast

Figure 7.17: Phase Frequency Detector: Pulse Diagrams with frequency error Error = Late – Early

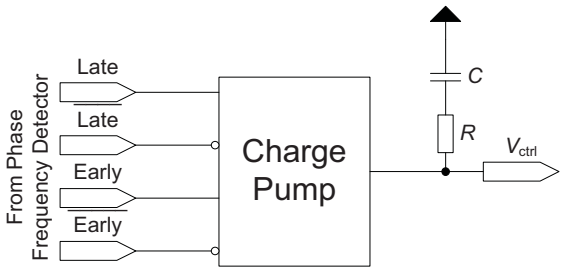


Figure 7.18: Charge pump with RC filter

implementations are a passive RRC filter (appendix B.4, p. 121) and a charge pump with an RC filter [RCN03, Mor07].

Charge Pump and RC Filter

Again, a solution with infinite gain is a charge pump followed by a filter (fig. 7.18). As the charge pump operation frequency is limited, this solution is suitable for use with a PFD, where the output signal frequency is equal to or, in frequency acquisition, lower than the input signal frequency, but not with an analogue multiplier where the output contains components at the sum of the input frequencies.

As for the DLL, the so-called charge pump is actually a current source. This time, it provides a current I_{CP} with constant magnitude and a sign depending on the phase detector outputs, or no current at all:

$$I_{\mathrm{CP}}(t) = I_{\mathrm{CP,mag}}\, a(t) = I_{\mathrm{CP,mag}} \times \begin{cases} +1 & \mathrm{Late} = \mathrm{High}, \mathrm{Early} = \mathrm{Low} \\ 0 & \mathrm{Late} = \mathrm{Early} \\ -1 & \mathrm{Late} = \mathrm{Low}, \mathrm{Early} = \mathrm{High} \end{cases}$$

This requires some minor modification if the DLL charge pump described above (ch. 7.2.3) is to be reused: while before, the $\overline{\mathrm{late}}$ signal was identical to the early signal, and $\overline{\mathrm{early}}$ identical to late, now those signals are different and have to be separated (fig. 7.19).

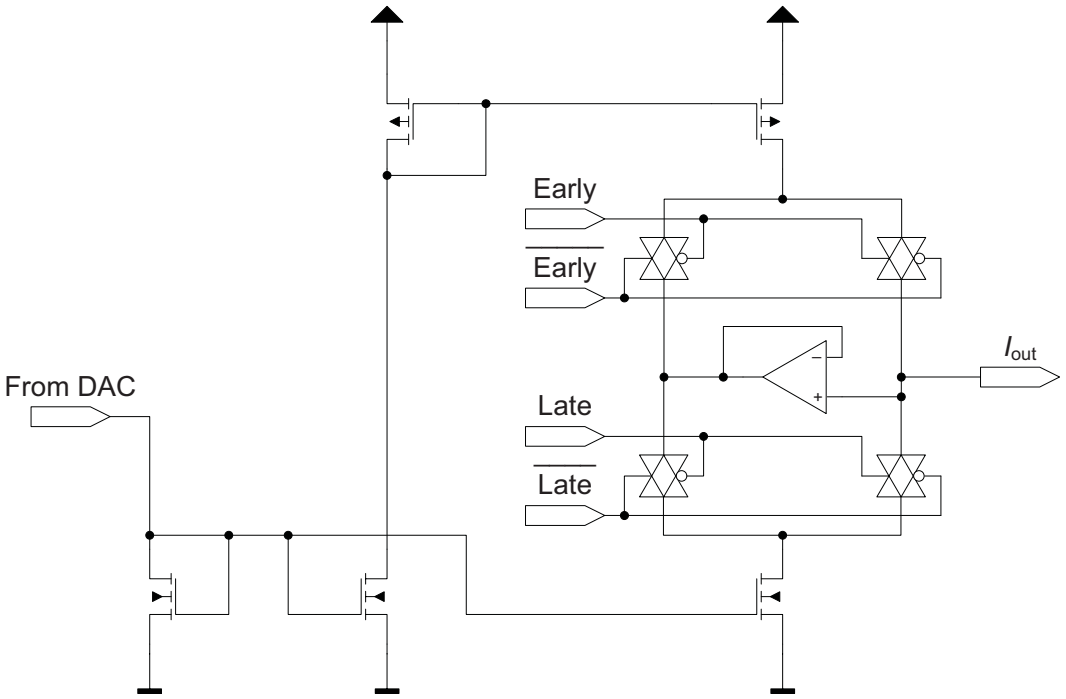


Figure 7.19: PLL charge pump implementation

The control voltage V_{ctrl} can be decomposed into a part across the capacitor, V_{C}, and a part across the resistor, V_{R}:

$$\begin{aligned} V_{\text{ctrl}}(t) &= V_{\text{R}}(t) + V_{\text{C}}(t) \\ &= R I_{\text{CP}}(t) + \frac{1}{C} \int_0^t I_{\text{CP}}(t)\mathrm{d}t + V_{\text{C}}(t=0) \\ &= I_{\text{CP,mag}} \left(R\,a(t) + \frac{1}{C} \int_0^t a(t)\mathrm{d}t \right) + V_{\text{C}}(t=0) \end{aligned}$$

V_{R} is called the proportional term, as it is proportional to the charge pump output current I_{CP}. V_{C} is the integral of that current, multiplied by a constant factor, and thus called integral term. The magnitude $I_{\text{CP,mag}}$ is a factor both of the proportional and the integral term, while R and $\frac{1}{C}$ can be used to adjust the ratio between integral and

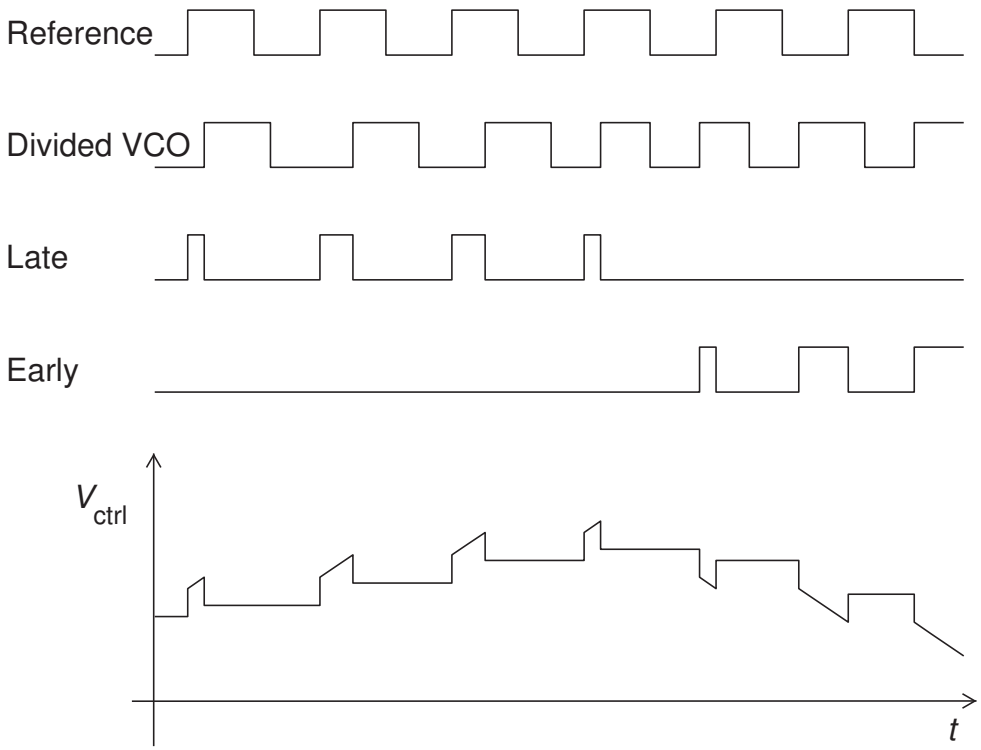


Figure 7.20: PLL charge pump operation. The ratio between integral and proportional term and the frequency changes of the VCO are exaggerated for better visibility. When early and late are reset to low, the respective other signal has been high for a very short time. These pulses are not shown for simplicity.

proportional term. Fig. 7.20 shows control voltage during operation of a charge pump in a PLL.

The transfer function

$$H(s) = I_{\mathrm{CP,mag}} \frac{1 + sRC}{sC}$$

has a pole at the origin

$$f_{\mathrm{P}} = 0$$

which means that it has infinite DC gain, and a zero at

$$f_{\mathrm{Z}} = \frac{1}{2\pi RC}$$

Note that f_P and f_Z are independent from each other, and the zero can be placed according to the requirements.

7.3.4 Choice of Phase Detector and Loop Filter

The characteristics of the analogue phase detector with passive RRC filter and the bang-bang phase frequency detector with charge pump and filter capacitor are summarised in tab. 7.2. The XOR phase detector is not explicitly shown, as it behaves like an analogue phase detector with square-wave input signals.

	Analogue multiplier Passive RRC filter		Phase frequency detector Charge pump RC filter
	Corresponds to XOR phase detector and passive RC filter for DLLs		Corresponds to bang-bang phase detector, Charge pump and filter capacitor for DLLs
+	simple	–	more complicated
–	sensitive to signal shape, including duty cycle	+	insensitive to signal shape, only leading edges of signal used
–	cannot be designed to work at 0 phase offset	+	operates at 0 phase offset by default
+	information about phase error magnitude	–	no information about phase error magnitude
–	no information about sign of phase error	+	information about sign of phase error
–	finite gain	+	infinite gain
–	gain is function of signal amplitude		
–	gain is function of phase offset		
–	small operating range $0 \ldots \pi$	+	infinite operating range: once operating range for phase detection exceeded, switches to frequency acquisition mode

	Analogue multiplier Passive RRC filter		Phase frequency detector Charge pump RC filter
+	keeps control voltage constant at 0 phase error	+	keeps control voltage constant at 0 phase error
–	stand-alone not sensitive to frequency differences	+	sensitive to both frequency and phase differences, even when without the feedback loop

Table 7.2: Comparison of PLL Phase Detector and Loop Filter combinations

7.3.5 Transfer function

The transfer function describes the behaviour of a PLL [Gar80, Mor07]. As mentioned before, in lock, the VCO and the loop filter introduce one pole each, and the loop filter one zero. The PLL in lock is therefore a second order system, which is not intrinsically stable. With knowledge of the transfer function in lock mode, the PLL can be designed to be stable at the operating point. In frequency acquisition mode, phase frequency detector is sensitive to frequency and the control voltage changes not depending on the phase, but on the frequency. The VCO does not perform an integration, and does not introduce a pole in this mode. This makes the PLL a first order system, which is intrinsically stable. A PLL is only temporarily in frequency acquisition mode if the frequency of the output signal is different from the target value. In this case, phase variations of the output signal are not relevant. Therefore, the transfer function is only used for a PLL in lock.

Let us recall the behaviour of the PLL's functional blocks:

- The VCO generates a signal with a frequency proportional[†] to the control voltage V_{ctrl}. The proportionality constant is called VCO gain K_{VCO}. The VCO signal is the output of the PLL.

[†]This simplification is only valid around the operating point, which is where the transfer function needs to be known.

- The VCO signal passes through a frequency divider, dividing the frequency by N.

- The phase frequency detector compares the phases φ_{Div} of this divided signal and φ_{Ref} of the reference input signal.
  The phase φ_{Div} is the integral of the divided VCO frequency $\frac{\omega_{\text{VCO}}}{Ns}$.
  The difference $a(t)$ of the phase frequency detector output signals is either $+1$ or -1 for a duration equal to $\frac{\varphi_{\text{Ref}}-\varphi_{\text{Div}}}{2\pi} T_{\text{clk,ref}}$ during a reference clock cycle and 0 otherwise.

- The charge pump converts the phase frequency detector output $a(t)$ into a current with instantaneous value $a(t)\, I_{\text{CP,mag}}$.

- This adds a charge of $\frac{\varphi_{\text{Ref}}-\varphi_{\text{Div}}}{2\pi} T_{\text{clk,ref}} I_{\text{CP,mag}}$ to the loop filter capacitor. The accumulated charge on the capacitor is $\frac{\varphi_{\text{Ref}}-\varphi_{\text{Div}}}{2\pi} I_{\text{CP,mag}} \frac{1}{s}$, leading to a voltage $V_{\text{C}} = \frac{\varphi_{\text{Ref}}-\varphi_{\text{Div}}}{2\pi} I_{\text{CP,mag}} \frac{1}{s} \frac{1}{C}$ across it.

- Assuming no voltage drop over the resistor, this voltage V_{C} is the VCO control voltage V_{ctrl}. The assumption is true if there is no current flowing to or from the capacitor and through the resistor, which is the case after the phase frequency detector reset. In addition, in lock, the average current through the resistor is zero. In the general case, $V_{\text{ctrl}} = \frac{\varphi_{\text{Ref}}-\varphi_{\text{Div}}}{2\pi} I_{\text{CP,mag}} Z_{\text{filter}}$ with $Z_{\text{filter}} = R + \frac{1}{sC}$ for an RC filter.

The VCO phase, output of the PLL, is thus

$$\varphi_{\text{VCO}} = \frac{1}{s} K_{\text{VCO}} V_{\text{ctrl}}(s) = \frac{1}{s} K_{\text{VCO}} \frac{\varphi_{\text{Ref}}(s) - \varphi_{\text{Div}}(s)}{2\pi} I_{\text{CP,mag}} Z_{\text{filter}}(s)$$

The transfer function $H(s)$ is defined as the ratio of output and input signal. Signal in this context is the phase:

$$H(s) = \frac{\varphi_{\text{VCO}}(s)}{\varphi_{\text{Ref}}(s)}$$

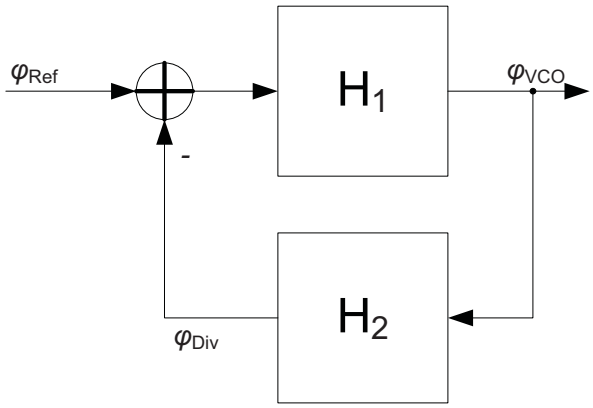


Figure 7.21: PLL block diagram for calculation of transfer function

Let's split the PLL transfer function into two parts $H_1(s)$ and $H_2(s)$ (fig. 7.21), as the signal fed back to the phase frequency detector is not the same as the output signal. Let

$$\begin{aligned} H_1(s) &= \frac{\varphi_{\mathrm{VCO}}(s)}{\varphi_{\mathrm{Ref}}(s) - \varphi_{\mathrm{Div}}(s)} \\ H_2(s) &= \frac{\varphi_{\mathrm{Div}}(s)}{\varphi_{\mathrm{VCO}}(s)} \end{aligned}$$

so that $H_1(s)$ describes the transfer from the phase detector to the VCO output and $H_2(s)$ the feedback from the VCO output to the phase detector.

$$\begin{aligned} \varphi_{\mathrm{VCO}}(s) &= \frac{1}{s} K_{\mathrm{VCO}} V_{\mathrm{ctrl}}(s) \\ \Longrightarrow H_1(s) &= \frac{\varphi_{\mathrm{VCO}}(s)}{\varphi_{\mathrm{Ref}}(s) - \varphi_{\mathrm{Div}}(s)} = \frac{1}{2\pi s} K_{\mathrm{VCO}} I_{\mathrm{CP,mag}} Z_{\mathrm{filter}}(s) \end{aligned}$$

$$\begin{aligned} \varphi_{\mathrm{Div}}(s) &= \frac{1}{N} \varphi_{\mathrm{VCO}}(s) \\ \Longrightarrow H_2(s) &= \frac{\varphi_{\mathrm{Div}}(s)}{\varphi_{\mathrm{VCO}}(s)} = \frac{1}{N} \end{aligned}$$

Thus, the transfer function is

$$\begin{aligned} H(s) &= \frac{\varphi_{\mathrm{VCO}}(s)}{\varphi_{\mathrm{Ref}}(s)} = \frac{H_1(s)}{1+H_1(s)\,H_2(s)} \\ &= \frac{N\,K_{\mathrm{VCO}}\,I_{\mathrm{CP,mag}}\,Z_{\mathrm{filter}}(s)}{2\pi N\,s + K_{\mathrm{VCO}}\,I_{\mathrm{CP,mag}}\,Z_{\mathrm{filter}}(s)} \\ \Longleftrightarrow H(s) &= \frac{1+\tau_{\mathrm{Z}}\,s}{\frac{1}{\omega_{\mathrm{n}}^2}\,s^2 + \frac{2\,\xi}{\omega_{\mathrm{n}}}\,s + 1} \end{aligned}$$

where $\tau_{\mathrm{Z}} = RC$ is the time constant of the RC filter zero, $\omega_{\mathrm{n}} = \sqrt{\frac{I_{\mathrm{CP,mag}}\,K_{\mathrm{VCO}}}{2\pi C}}$ the natural frequency and $\xi = \frac{RC}{2}\,\omega_{\mathrm{n}}$ the damping factor. The loop gain K is defined as $K = 2\,\xi\,\omega_{\mathrm{n}}$.

7.3.6 Start-up Procedure

A PLL with phase frequency detector can acquire lock without any special start-up procedure, as it is also sensitive to frequency. However, the filter capacitor is discharged when a reset is applied to the chip to have defined start conditions.

7.4 Hit Registers

Hit registers are the first memories that store the time measurement until the next stage can process the data. They consist of 32 D flip flops per channel, one for each element in the VCDL. The phase shifted outputs of the VCDL are connected to the data inputs of the D flip flops. The TDC hit input serves as clock signal to the hit registers. Two modes of operation are possible, one with high accuracy, but high power consumption, and one with lower power consumption at the price of lower accuracy. These modes correspond to the two hit controller modes described earlier (section 6.2, p. 47).

As the timing information is digitised in the output signal, the distribution of the output signals is not timing critical. Jitter on the output signal has no influence on the measurement.

7.4.1 Differential vs. Single-Ended Hit Registers

Having a differential VCDL as a time base and, on the full chip, synthesised single-ended processing logic, requires a differential to single-ended conversion somewhere between the VCDL and the processing logic. In the prototype, the readout logic is full custom design, so it could be designed differentially. However, the power consumption would be prohibitive. This leaves two options, either a conversion immediately after the VCDL and before the single-ended hit registers, or immediately after the hit registers, which then have to be differential. As a first step, single-ended and differential D flip flops have been simulated and compared.

Single-ended flip flops run at a maximum speed that is intrinsic to the technology and the dimensions of the transistors. This defines both average value and shape of the current consumption. The speed and power consumption of the differential circuit can be adjusted by the tail current source in a wide range. One of the main reasons for using differential circuitry is that the noise generated by the circuit is expected to be smaller that that generated by single-ended logic with the same functionality. To be able to compare the characteristics of both implementations, the differential circuit's tail current is adjusted such that both circuits recover from the metastable state at the same speed. Simulations have been carried out assuming three different configurations at the hit input of the TDC. The average supply currents of a single register with 1.2 V supply are shown in tab. 7.3. Note that the differential hit registers take about 20 times as much current. The currents depend on the signals at the hit input. Three cases are analysed:

1. High accuracy mode with 5 MHz clock at the hit input of the TDC, corresponding to an average hit rate of 5 MHz. This is clearly a very pessimistic case, realistically the hit rates are much lower.

2. Hit input is always low. This corresponds to the high accuracy mode of operation without hits arriving.

3. Hit input is always high. This corresponds to the low power mode without hits arriving.

	5 MHz hit rate Case 1	Hit input low Case 2	Hit input high Case 3
Differential	952 µA	1030 µA	1000 µA
Single-ended	54 µA	54 µA	166 nA

Table 7.3: Average supply current of one differential and one single-ended hit register ($V_{DD} = 1.2$ V)

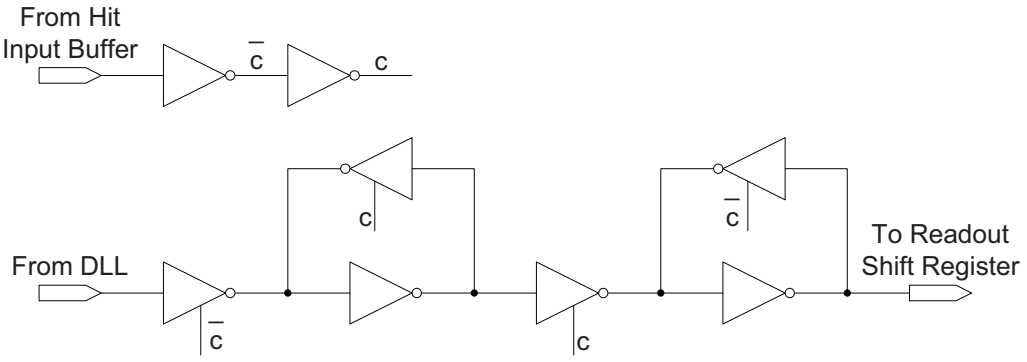
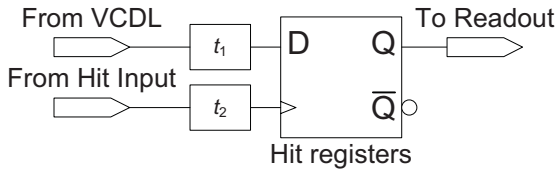


Figure 7.22: Implementation of a hit register

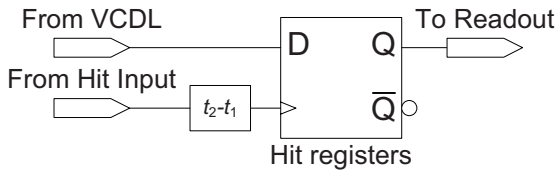
These are extreme cases. In high accuracy mode (case 2), the current consumption will be lower in presence of hits, in low power consumption mode (case 3), it will increase with the hit rate.

The standard deviation of the supply current in case 1, with 5 MHz hit rate, is 81 µA for the differential and 71 µA for the single-ended implementation.

These simulations include the differential to single-ended conversion, which needs to be faster for the case with single-ended hit registers, but they are not taking into consideration that the differential implementation requires the signals from the VCDL to be distributed differentially to the hit registers, which is power and space consuming. Thus, the values given for the differential implementation are underestimated. Moreover, besides the current consumption of the differential registers being already prohibitive, in addition, they generate more noise, in absolute terms, than their single-ended counterparts.



(a) Hit register with delayed input signals. t_1 and t_2 include the delay is the corresponding path both before and inside the hit register



(b) Simplified model of delays at the hit registers

Figure 7.23: Delay at hit register

Fig. 7.22 shows the chosen implementation of the hit register D flip flop. Using tristate inverters leads to a higher gain compared to an implementation with pass gates. Therefore, the register can resolve metastable states faster.

7.5 Hit Register Driving Circuitry

An ideal TDC digitises the time of a hit, i.e. its hit registers store the state of the VCDL at the time the hit arrives at the input of the chip. Delays make this situation a bit more complicated in the real world: The hit registers (fig. 7.23a) store the state of the VCDL output signal distribution lines, which is the state of the VCDL delayed by the differential-to-single-ended converters, the buffers and the distribution lines, at the time the hit registers are clocked, which is the time the hit arrives, plus the time of propagation through the input pads, the buffers and the distribution lines. In addition, the hit registers themselves sample the data with a relative delay, which can be positive or negative, depending on the detailed implementation, including routing, of the D flip flop. In a simplified model, the hit registers and the driving circuitry can be assumed ideal, and all delays are represented by one additional, non-ideal delay

stage (fig. 7.23b). This model shows that the delays of the different parts of the circuit cannot be distinguished, and neither can variations of them. The mean value of the delay causes a static offset between the real time of the hit and the measured time of the hit. This difference can be easily corrected for by an arithmetic subtraction, and its magnitude is, at first glance, irrelevant. The fluctuations of the delay however are seen as jitter in the measurement.

As shown later (ch. 7.6, p. 91), the most important effect causing timing jitter is the slew rate of signals when switching the hit registers from capture to store mode. The slower the transition between the two logic levels, the larger the effect of undesired signals on the timing measurement.

7.6 Sources of Errors

In this section, the most relevant sources of errors are discussed. In the first subsections, sensitive parts of the circuit are identified. Afterwards, the main noise mechanisms and the effects on the TDC performance are presented.

7.6.1 Slew Rate

Noise can change the signal voltage, adding a voltage $V_\mathrm{n}(t)$, but not its timing. For digital signals, noise is only relevant if it makes the signal voltage pass the threshold between the two logic states high and low. This is unlikely to happen for signal voltages that are close to 0 or the supply voltage V_DD. Once a signal switches from high to low or vice versa, it passes the threshold at some point in time. As the signal voltage gets infinitely close to the threshold voltage, any noise can make it pass the threshold earlier or later, adding t_n to the time of the transition (fig. 7.24). Noise on power and ground shifts the threshold voltage. In both cases, the result is jitter. The importance of this effect is inversely proportional to the slope of the time critical signal, called slew rate.

$$t_\mathrm{n} = \frac{1}{\frac{\mathrm{d}V}{\mathrm{d}t}} V_\mathrm{n}(t) \quad \text{(for } V_\mathrm{n} \text{ smaller than the threshold voltage)}$$

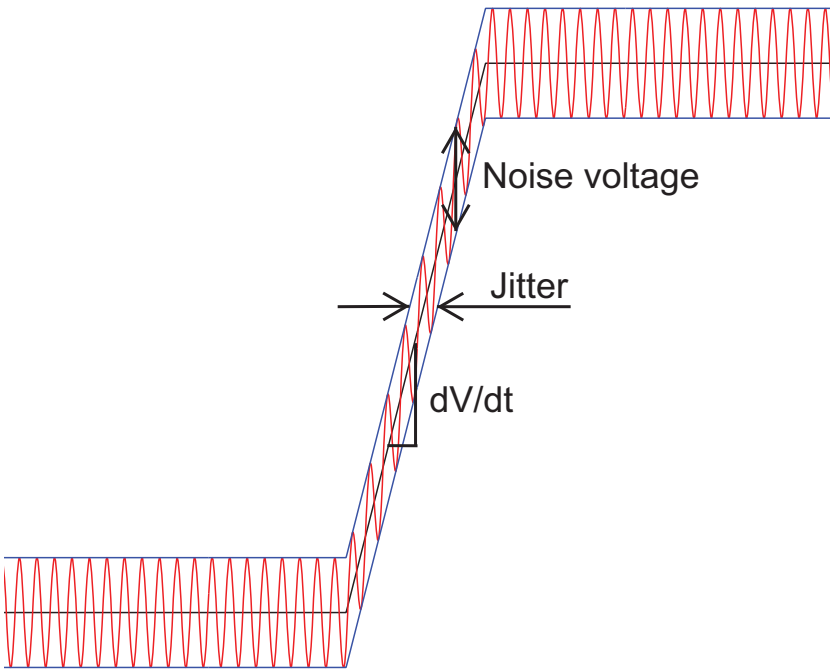


Figure 7.24: Relation between noise on a digital signal and timing jitter. For simplicity, the noise is represented by a single constant amplitude and frequency sine wave.

Therefore, it is important to keep those transition phases as short as possible, i.e. to minimise rise and fall times of the signals.

Tab. 7.4 shows the slew rates for selected signals of major importance to the TDC properties. This table is used to calculate the jitter caused by different effects such as power supply and ground variations, thermal, flicker and shot noise when digital signals are effected.

7.6.2 Noise on Control Voltages

The analogue PLL and DLL control voltages are, on the contrary to digital signals, immediately sensitive to changes of the voltage, which will lead to variations of the VCO frequency and the VCDL element delay, respectively. The VCO gain K_{VCO} and the VCDL gain K_{VCDL}, respectively, describe the effect of the control voltage on the VCO

Signal	Slew rate in $\frac{\mathrm{V}}{\mathrm{ns}}$	Inverse slew rate in $\frac{\mathrm{ps}}{\mathrm{V}}$
Hit register input from output of VCDL tap output buffer	40	25
Hit register input from output of hit input buffer	24	41
First inverter stage of buffer	21	48
Second inverter stage of buffer	41	24
Third inverter stage of buffer	17	56
Differential signal in VCDL	18	54

Table 7.4: Slew rate of signals of major importance to the TDC properties. As both buffers are the same and only the load to the last inverter is different, all slew rates except the last are equal. The relevant edges are shown.

Process	Supply voltage in V	Temperature in °C	VCO gain in $\frac{\mathrm{GHz}}{\mathrm{V}}$	VCDL gain in $\frac{\mathrm{ps}}{\mathrm{V}}$
Slow	1.35	125	2.06	−31.8
Typical	1.5	25	5.71	−87.9
Fast	1.65	−20	7.80	−142

Table 7.5: VCO and VCDL gain for different simulation corners

frequency and the DLL delay. For small noise voltages $V_{\mathrm{n}}(t)$, a linear approximation at the operating point is used to estimate the jitter to be expected:

$$J(t) = K V_{\mathrm{n}}(t)$$

Tab. 7.5 shows K_{VCO} and K_{VCDL} for the 24.4 ps bin size operating point for different simulation corners.

7.6.3 Power Supply and Ground Variations

The voltage on the power supply and ground wires can vary due to noise, which is fully random, coupling of signals into the ground and supply wires, or resistive voltage drops

on the lines, caused by a variation of the current drawn by other parts of the circuitry. The latter effects are not random, as they can be predicted if the whole system is known in detail. At the time of design of a certain block, the implementation of the rest of the circuit is often not yet fully known, the distances between blocks may change and filling capacitors may change the characteristics of the variations. In addition, even if the complete system, including input signal patterns, is known, the analysis is very complex. As long as the mentioned effects do not cause system instability, it is convenient to treat them as if they were random noise.

The VCDL is rather insensitive to power and ground variations, as a differential implementation is chosen. For the same reason, it also generates little noise. The standard deviation is in the order of 4 mA, the average 190 mA. The power supply rejection ratio is 28 dB, The common mode rejection ratio 33 dB. A switching delay elements creates an RMS voltage variation of 0.35 mV on a 0.5 Ω resistance in the supply wires. This makes the output of the delay elements vary by 28 μV and creates 0.7 fs of jitter, which is negligible.

The buffers both between the hit registers and the VCDL tap outputs, as well as the hit inputs, are implemented single-ended, and so are the hit registers themselves. The voltage drop V_R in a wire with resistance R is proportional to the current I. Assuming for simplicity that R is constant, the block with the largest current variation will cause the largest supply and ground voltage variations. Let's assume all blocks, buffers as well as hit registers, are individually connected to a global supply grid, which provides a stable supply voltage. Variations are compensated by large capacitor arrays and the resistance of the wires is reduced by design. In this case, all blocks can be considered independent, and the resistance of the supply lines is easy to estimate. According to simulations, the peak supply current of the VCDL tap output buffer is 17 mA, that of a hit input buffer 14 mA. With a resistance of 1 Ω in the supply lines, this generates peak voltage drops of 17 mV and 14 mV, respectively. Considering the slew rates inside and after these buffers, which consist of four stages of inverters, this leads to jitters of 2.6 ps and 2.4 ps, respectively, totalling 3.5 ps. The power and ground variations are not statistically independent. The four inverters that constitute the buffer always change state one after the other, with a very small delay. The delay between the change of the input signal and the output signal of the 4-stage buffer is simulated to be 150 ps, which

corresponds to 6.7 GHz. Power and ground variations with a smaller frequency affect all inverters of the buffer the same way.

7.6.4 Thermal Noise

Thermal noise generates a voltage due to random movement of electrons in a conductor or semiconductor [Raz01]. Let's first consider thermal noise in a resistor, and afterwards in a transistor.

A resistor with thermal noise can be decomposed for analysis into a series of an ideal, i.e. noiseless resistor, and a random voltage source. The spectrum of thermal noise is flat[†], with a power spectral density of $4k_BTR$, where $k_B = 1.38 \times 10^{-23}\ \frac{J}{K}$ is the Boltzmann constant and T the temperature. The average voltage generated by white noise is thus $V_{n,RMS} = \sqrt{4k_BTRf_{BW}}$, where f_{BW} is the bandwidth under consideration. In a circuit, resistors are always connected to a capacitor C, which can be parasitic and/or intentional. The combination of resistor and capacitor creates a low-pass filter with a cut-off frequency of $\frac{1}{RC}$. This formula is only valid for resistors whose noise bandwidth is limited by a capacitance. This leads to a total RMS noise voltage of $\sqrt{\frac{k_BT}{C}}$. Tab. 7.6 shows the thermal noise to be expected at selected critical points of the TDC. Even though the values for PLL and DLL are overestimated – the loops filter the noise inside the loop bandwidth – the values are too small to be significant. Inside the VCDL, the connection wires are short and their parasitic capacitance is small. The noise bandwidth is not limited by those capacitances, but by the bandwidth of the delay elements. As the delay of an delay element is a function of the current, it is more convenient to look at the RMS noise current: $I_{n,RMS} = \sqrt{\frac{4k_BT}{R}f_{BW}}$. The connections can be regarded as noise current sources, in series with the noise current sources that are intrinsic to the transistors, with $I_{n,RMS} = \sqrt{4k_BT\gamma g_m f_{BW}}$. γ, a coefficient in the order of 1, equals $\frac{2}{3}$ for long-channel transistors. The current passing through the delay elements is in the order of 4 mA for a delay of 24.4 ps. Frequencies higher than about 2.56 GHz are suppressed due to the limited bandwidth of a delay element. For a delay element, the total noise current, obtained by adding the power of all thermal noise

[†]The spectral density drops above 100 THz, which is many orders of magnitude higher than the signals we are interested in.

Signal	C in pF	$V_{\mathrm{n,RMS}}$ in µV	J_{RMS} in ps
Hit register input from VCDL	0.8	72	1.8
Hit register input from hit input	0.2	140	5.7
DLL control voltage	50	0.014	0.001
PLL control voltage	50	0.03	0.1

Table 7.6: Thermal noise at selected critical points. T is approximated to 300 K for simplicity. The values for DLL and PLL do not take into consideration that the loop filters noise that is inside the loop bandwidth.

sources, is in the order of 160 µA, which corresponds to 4% of the nominal current. This corresponds to 1.8 ps of jitter, according to simulations.

7.6.5 Flicker Noise

Flicker noise, also called $\frac{1}{f}$-noise, is present in MOSFETs [Raz01]. Its spectral density is proportional to $\frac{1}{f}$, making it negligible for higher frequencies. The effect is mainly due to charge carriers that a randomly trapped and randomly released at the interface between the Si crystal and the SiO_2 gate oxide. The RMS voltage of the flicker noise can be approximated as $V_{\mathrm{n,RMS}} = \sqrt{\int_{f_0}^{f_0+f_{\mathrm{BW}}} \frac{K}{C_{\mathrm{ox}}WL} \cdot \frac{1}{f}\,\mathrm{d}f}$, with K being a process-dependent constant, C_{ox} the capacitance per unit area, and W and L the dimensions of the MOSFET. For the delay elements, frequencies between 1.28 GHz and about 2.56 GHz are of concern, leading to a RMS voltage $V_{\mathrm{n,RMS}} = \sqrt{\frac{K}{C_{\mathrm{ox}}WL} \ln\left(\frac{2.56\,\mathrm{GHz}}{1.28\,\mathrm{GHz}}\right)}$. It is largest for the transistor with the smallest dimensions. It amounts to about 8.7 µV, and can therefore be completely neglected in the delay line.

For the hit registers, the lowest relevant frequency needs to be determined. Noise components with $f = 0$ cannot be present, as this equals perfect DC which is never changing. $f = 10$ µHz is the frequency of a signal that has one one period per day, 1 mHz corresponds to a period of about a quarter of an hour. Noise with such frequencies is not relevant, even temperature varies faster. Considering the frequency components between 10 kHz and 2.56 GHz, the flicker noise for the smallest transistor in the

hit registers is 30 μV, which is orders of magnitude lower than the expected supply and ground variations.

7.6.6 Shot noise

Shot noise is caused by electric current consisting of discrete moving charge carriers, rather than being a continuous flow [Raz01]. The RMS noise current is $I_{\mathrm{n,RMS}} = \sqrt{2eIf_{\mathrm{BW}}}$, where e is the charge of an electron and I the total current. In digital parts of the circuit outside the feedback loops, assuming exaggerated 20 mA and a bandwidth of 2.56 GHz leads to $I_{\mathrm{n,RMS}} = 4.0$ μA. For a bandwidth of 1.28 GHz and a current of 4 mA, which is the current in a VCDL delay element, $I_{\mathrm{n,RMS}} = 1.3$ μA is negligible. Simulations show that this corresponds to a jitter of 15 fs.

7.7 Readout and Configuration

The prototype has been designed to evaluate the timing resolution that can be achieved in the given 130 nm process. This is neither influenced by the way data is read out nor by the way the chip is configured. The design requirement for prototype readout and configuration is therefore simplicity and minimum number of required I/O pins. Simplicity is important to make sure that the prototype tests reveal information about the timing part and are unlikely to be impeded by problems in the readout and configuration part. The number of pins has to be minimised to keep the required chip area small while all important signals can be brought into or out of the chip.

Given these criteria, two shift registers, one for readout and one for configuration, have been identified as optimal.

7.7.1 Readout Shift Register

The readout shift register is used for transferring the data stored in the hit registers to off-chip test circuitry (fig. 7.25). It has a $44 \times 32\,\mathrm{b} = 1408\,\mathrm{b}$ wide parallel input connected to the 44 channels with 32 b each, and a serial output. A control signal is used to specify whether data is to be read in through the parallel input, or to be read out

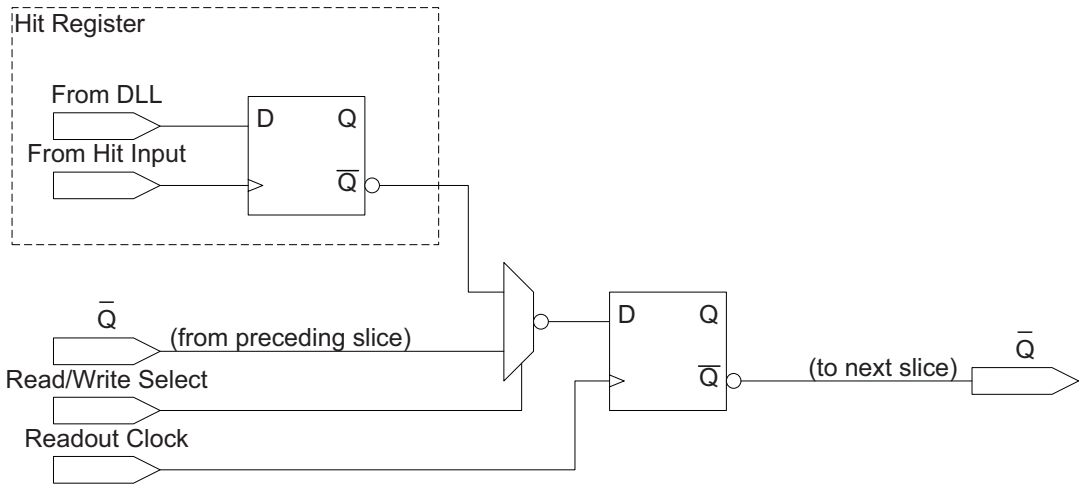


Figure 7.25: Readout shift register slice. The $\overline{Q}$ input of the very first slice in connected to the configuration data input, the $\overline{Q}$ output of the very last slice is connected to the readout data output pad.

serially. For verification, the shift register can also be serially loaded with external data from the configuration data input. In case of problems, this feature allows to distinguish between problems in the readout register and problems in the hit registers, the hit inputs or the DLL. Reading out the data of the shift register is done by shifting each bit in that register by one step per readout clock cycle, and shifting in data from the configuration data input. This means that all the data of the readout register is overwritten when read out. However, if no hits have arrived to the chip during readout, the data in the hit registers remains unchanged and can be transferred again to the readout registers and read out a second time. This can be useful to verify the proper operation of the hit input and the hit registers.

7.7.2 Configuration Shift Register

The configuration shift register (fig. 7.26) is meant for transferring data from the off-chip test circuitry into the chip. Data readout is only necessary for verification of the configuration shift register. If the data readout is destructive, the rest of the chip might not operate properly until the next reset. As this verification needs to be done only once per prototype, this is not a problem. At reset, a default configuration that is appropriate

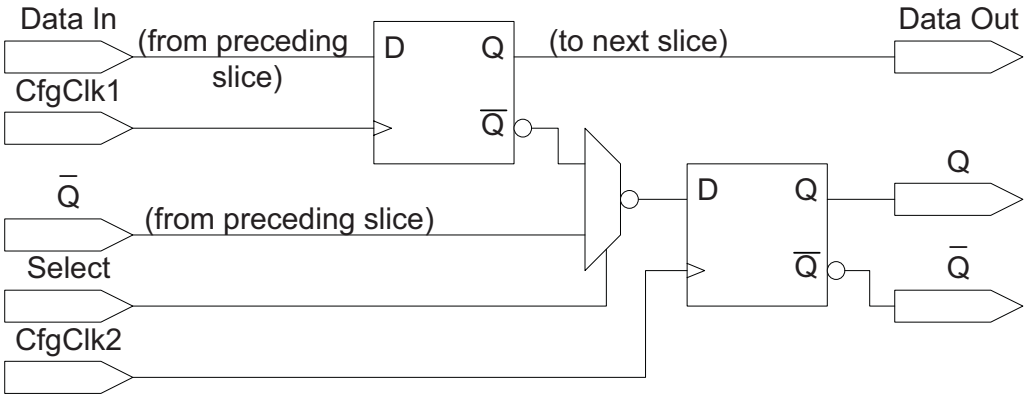


Figure 7.26: Configuration shift register slice. The Data In input of the very first slice is connected to the configuration data in pad. The output $\overline{Q}$ is connected to the $\overline{Q}$ input of the next slice. The outputs Q and $\overline{Q}$ are used to configure the circuitry, depending on whether the default configuration value is 0 or 1.

for a typical chip is loaded into the configuration register, allowing the chip to work properly without modification of the configuration, thus even if the configuration shift mechanism should be faulty.

A need for quick change of configuration might arise in case other parts of the prototype, such as the DLL start-up state machine, would not work as intended. In this case, a configuration forcing the charge pump to lower the voltage will be loaded and activated, slowing down the lock aquisition process which starts when the clock is enabled. Once the phase detector consistently gives the correct output signal, the control has to be passed quickly to the phase detector, thus the configuration needs to be changed quickly. The configuration shift register is composed of two simple stages, the first one with serial input and parallel output, and the second one with parallel input, parallel output, independent clock and, for verification, an additional serial output. In this way, the configuration data can be shifted in while being invisible to the rest of the chip and activated by transfer to the second stage. Then, another configuration can be loaded into the first stage and held ready for transfer.

8 Experimental Results

8.1 PLL Characterisation

8.1.1 Locking Range

A PLL has a locking range that is limited by the maximum speed at which a signal can be generated and propagated. The TDC130 PLL is designed to multiply a 40 MHz input signal up to a 1.28 GHz output. Measurements with an Agilent 81133A pattern generator have shown that the PLL can lock to frequencies of up to 72 MHz at the input when the PLL supply voltage is 1.2 V, and up to 103 MHz at the input using a 1.5 V supply. This leads to output frequencies of up to 2.3 GHz and 3.3 GHz, respectively. If the input frequency exceeds these limits, the PLL locks to twice the period of the input signal. Simulations with extracted parasitics have shown maximum frequencies of 4.7 GHz and 4.0 GHz. These simulations include layout parasitics without filling of empty areas in the metals. To comply with the metal density layout rules, dummy metal squares are automatically introduced to the layout before manufacturing, increasing the parasitic capacitances and explaining the differences between measurement and simulation.

8.1.2 Jitter

When signals inside the chip need to be measured, they have to pass through output buffers, and signals entering the chip have to pass through input buffers. Therefore, any jitter measurement of signals generated inside the chip using external instruments unavoidably includes the jitter of the buffers, in addition to the jitter of the measurement instrument. In order to be able to estimate which part of the measured jitter is due to

Source	$\frac{\sigma}{\text{ps}}$	$\frac{J_{\text{peak}-\text{peak}}}{\text{ps}}$
Pattern generator and oscilloscope	5.64	58
I/O buffers	7.62	79
Total	11.68	134
PLL	6.82	

Table 8.1: PLL jitter measurements: Contributions of pattern generator, oscilloscope, I/O buffers and PLL.

the buffers and the test setup and which part is the jitter of the PLL, the TDC130-0820 can be configured such that the signal at the PLL input is directly transferred to a test output, without passing through the PLL. The same test output can also be connected to the clock divider output. Note that PLL measurements show the jitter at the clock divider output, whose frequency is 40 MHz, while the VCO output is used to drive the DLL. Its frequency, 1.28 GHz, is too high to passed through the output buffers and thus this signal cannot be measured off-chip.

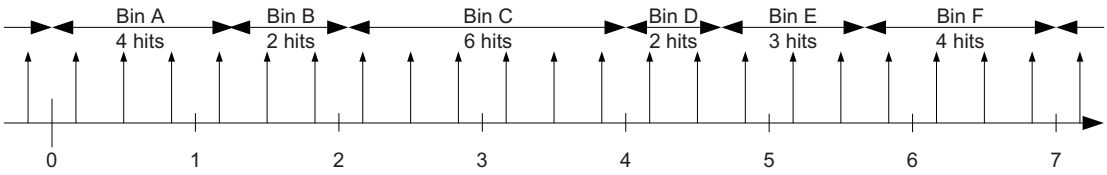
In a reference measurement, a Agilent 81133A pattern generator and a LeCroy Wavepro 7100 are connected together and the generator's and oscilloscope's jitters are measured.

Tab. 8.1 shows the contributions of the different parts of the measurement setup. Assuming these sources to be statistically independent, the value for the PLL is calculated to be 6.82 ps. This compares to a simulated value of 6.66 ps.

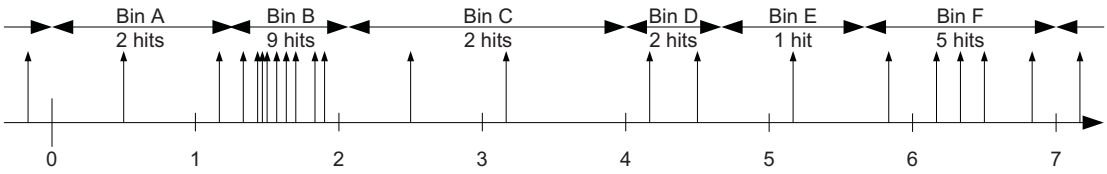
8.2 DLL Characterisation

8.2.1 DLL locking range

The DLL cannot be characterised independently from the hit registers. The DLL tap output signal frequencies are too high to be sent off-chip. The only way to characterise the DLL is to generate hits and to measure the properties of hit registers and DLL together, without being able to distinguish their contributions.



(a) Assuming a large number of random hits approximates equally spaced hits, the size a the bin is equal to the number of hits in that bin divided by the total number of hits times the dynamic range.



(b) If the hits are not equally spaced, for example due to beating, the number of hits in a bin does not represent its relative size.

Figure 8.1: Principle of code density test (simplified)

The DLL locking range has been measured sweeping the PLL clock input frequency and observing the DLL outputs using the hit registers. The pattern stored in the hit registers on the arrival of a hit clearly shows whether or not the DLL is in lock. The measurement has not been limited by the maximum frequency of the hit registers. The DLL has been found to lock to signals with frequencies between 950 MHz and 2.76 GHz, corresponding to bin sizes of 31.9 ps and 17.8 ps. When the DLL frequency is larger than 2.76 GHz, the DLL locks to two signals periods, as it was expected by simulations.

8.2.2 Linearity

Nonlinearity is the deviation of the TDC bin sizes from their average value (section 3.2). Sweeping the hit time in small steps through the dynamic range of the TDC gives all the required information to determine the transfer characteristics (fig. 3.3, p. 15). However, the time of a hit can only known if a TDC with a resolution much better than the TDC-under-test was available, or if the hit could be generated with such a resolution. In addition, jitter occurs at every stage of the chain from signal generation to time-to-digital conversion. Even if the exact time was known at one stage, jitter will change it before it is measured by the TDC. Disregarding these problems, after a com-

	uncalibrated		calibrated	
	in ps	in LSB	in ps	in LSB
σ_{DNL}	5.72	23.4%	4.02	16.5%
$\min_{1\leq i\leq 32}(\Delta t_i) - t_{\mathrm{LSB}}$	−9.03	−37.0%	−6.69	−27.4%
$\max_{1\leq i\leq 32}(\Delta t_i) - t_{\mathrm{LSB}}$	16.7	68.3%	12.0	49.1%
σ_{INL}	5.96	24.4%	4.99	20.5%

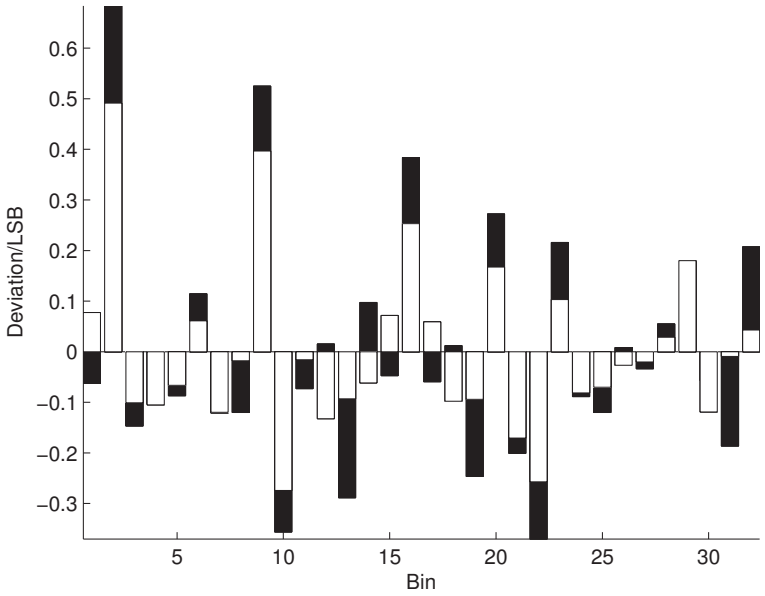
Table 8.2: Results of code density tests with 80 000 hits. The average bin size is 24.4 ps.

plete, jitter-free sweep with constant step size through the dynamic range, the number of hits measured to be in one time bin of the TDC is proportional to the size of that bin. This corresponds to the height of the bars of a histogram showing the numbers of hits per bin. These bars do not contain any information on when a particular hit arrived within a bin. Therefore, it is not necessary to know the exact time of any hit, the hit may even be randomly generated – it is sufficient to know that the hits are evenly distributed across the dynamic range of the converter (fig. 8.1). A test using random hits is called Code Density Test (CDT). It is an established method for measuring the linearity of a converter [JCG07, MC99].

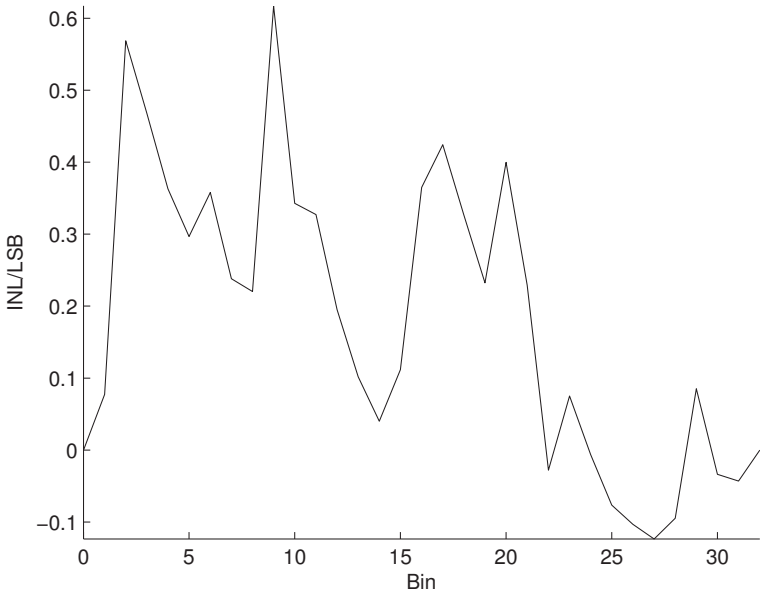
A hit source which is not synchronised with the TDC and its input clock generates random[†] hits distributed across the whole dynamic range, provided no beating occurs due to a fixed relation between the two generators' frequencies. It is convenient to use generators which are not very well frequency-stabilised – their frequencies drift rather quickly and beating will disappear eventually.

Fig. 8.2 and tab. 8.2 show the deviation of the individual bin sizes from the average value of 24.4 ps as measured by a code density test both with and without calibration of the VCDL. It can be seen clearly that the calibration reduces the deviation of every bin. While Monte-Carlo simulations of the VCDL have shown a variations of 30% of the nominal bin size, the measurements reveal much a higher non-linearity. The simulations do not include the buffers that are needed to drive the distribution lines

[†]For the purpose of the measurement, these hits are a sufficiently good approximation of random hits. Strictly speaking, they are not random distributed. With sufficient knowledge of the sources and the preceding hit, the next hit can be predicted.



(a) Bin sizes. The black bars show the size before, the white ones after calibration



(b) INL (after calibration)

Figure 8.2: Results of a CDT with a bin size of 24.4 ps

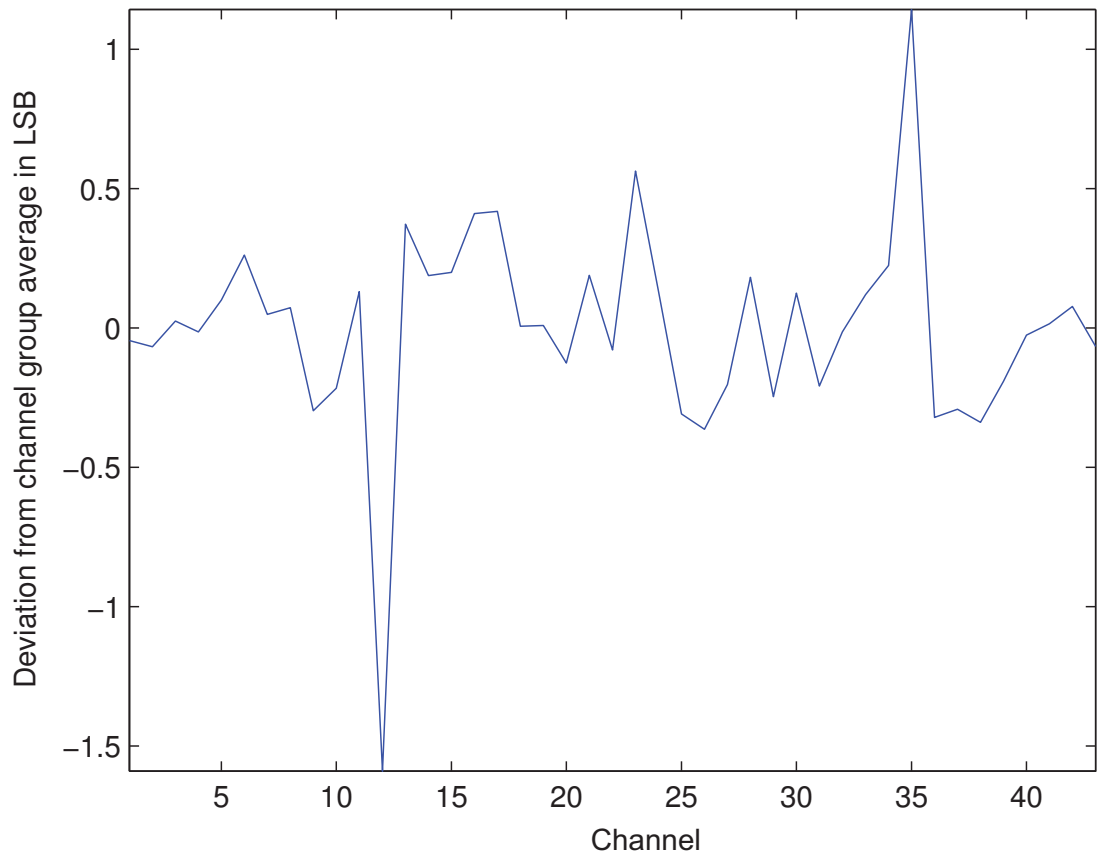


Figure 8.3: Offset due to the hit input buffers, averaged over 80 000 hits, with correction for PCB routing and LVDS input pads. The average bin size is 24.4 ps.

between the VCDL tap outputs and the hit registers. Those buffers have a simulated propagation delay of 150 ps in nominal conditions. The largest observed deviation from the average bin size is 16.7 ps without calibration, which corresponds to 11% of the buffer delay.

Fig. 8.3 shows spread of the delay of the hit input buffers, which are identical to the VCDL tap output buffers in order to distinguish VCDL imperfections from buffer imperfections. Recall that the channels of the hit registers are grouped into 8 channel groups that share the LVDS input pad, the routing on the PCB and the signal generator channel, but every individual channel has a dedicated hit buffer and hit registers. The average offset of the channel group specific and of the channel specific components

can thus be easily identified. The standard deviation of the channel specific offset is 37% of an LSB, or 9.11 ps.

It can be assumed that the buffers between the VCDL and the hit registers show the same or similar variations. They can be compensated partially, but not completely by the VCDL calibration, which has been designed to compensate VCDL mismatch variations as seen in the Monte Carlo simulations. The decision of implementing the calibration scheme in the VCDL elements has been justified by the experience of the previous chip, the HPTDC. There, the VCDL delay variations where compensated for by varying the delay after the tap output, not within the VCDL. It has been observed that once a VCDL element delay is too big or too small, following tap outputs also had to be set to a delay in the same order of magnitude. As example, let's assume one delay element at the beginning of the VCDL has a delay of 130% LSB, one at the end 70% LSB and all others exactly 100%. In this case, almost all elements have ideal delays. However, as the first bin is 30% too large, the calibration needs to reduce the delay after the tap output by 30% of an LSB to reach the ideal bin size. Reducing this delay, the following bin starts 30% of an LSB earlier. Therefore, the second the tap output calibration delay also needs to be reduced by 30% of an LSB, despite the fact that the second VCDL element's delay is exactly 100% of an LSB. This effect propagates until a VCDL element delay is compensating, which is, in this example, the very last one. Thus, many calibration delays may need to be configured to a rather large value, even if only a few delay elements are strongly mismatching. If in addition, two delays are too large without a delay between them being too small, the calibration delay might saturate to the maximum value. The scheme used in the TDC130 prototype overcomes this limitation, but the experiments show that calibration of the tap output buffers is also required.

8.2.3 Jitter

Jitter is measured using a hit that is generated with a fixed relation to the PLL reference input clock. Ideally, thus in absence of jitter, this hit always gives the same measurement result. Any deviation from that value is due to jitter, in the pulse generator, in the connection between pulse generator and the TDC hit input pad, input buffers, hit

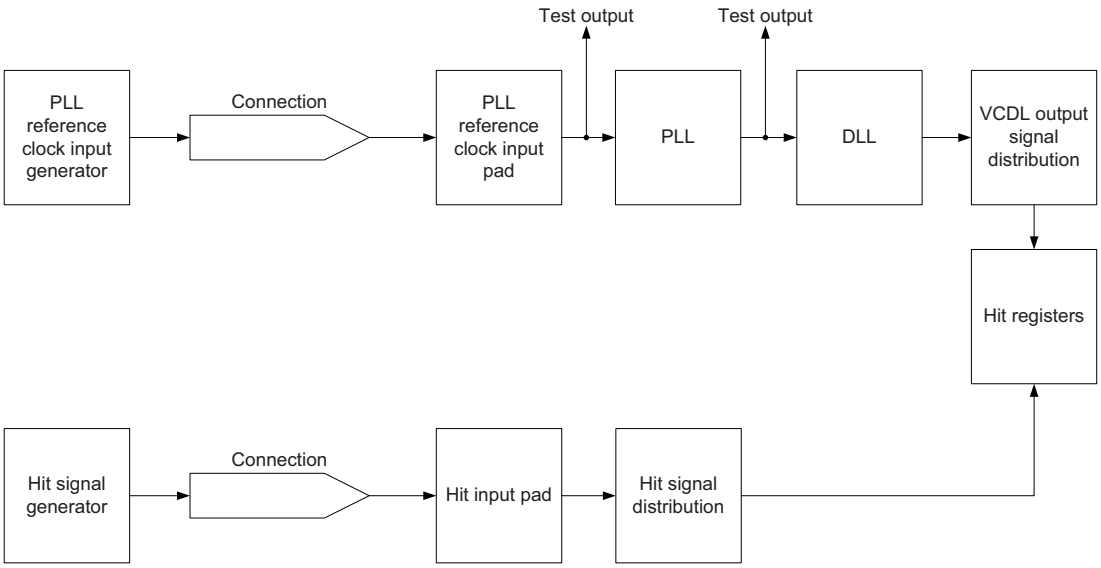


Figure 8.4: Signal paths from clock input and from hit input to hit register, showing as well the test signal outputs used to measure the jitter of the input buffers and of the PLL.

registers, the DLL, in the clock generator, the connection between clock generator and TDC input pad, in the clock input pad or in the PLL (fig. 8.4).

Measurements show a peak-to-peak jitter of $6t_{\mathrm{LSB}} = 146\,\mathrm{ps}$ with a standard deviation of $0.517t_{\mathrm{LSB}} = 12.6\,\mathrm{ps}$. Subtracting the jitter that can be attributed to the PLL (6.82 ps) leaves 10.6 ps for the DLL, the VCDL tap output buffers, the hit registers, the hit input buffers and both LVDS input pads. The DLL has been designed such that the RMS tracking jitter is in the order of 1 ps. Assuming that 50% of the jitter measured for one LVDS input pad and one single-ended output buffer is to be attributed to the LVDS pad leaves a total of 6.74 ps for the VCDL tap output buffers and the hit input buffers which have a total delay of 150 ps each.

Note that in the PLL measurement, the jitter of the LVDS input pad cannot be distinguished from the jitter of the single-ended standard cell output buffer. Here, the jitter of the output buffer is not relevant, as the data at the output is already digitised.

Supply domain	Logic (1.2 V)	DLL	PLL (1.2 V)	I/O (2.5 V)	Total
DLL at 1.2 V	48.9 mA 58.7 mW	123.8 mA 148.6 mW	10.3 mA 12.4 mW	36.9 mA 92.3 mW	312 mW
DLL at 1.5 V	48.9 mA 58.7 mW	137.1 mA 205.7 mW	10.3 mA 12.4 mW	36.9 mA 92.3 mW	370.8 mW
Simulation (extracted) DLL at 1.5 V	Hit registers 40 mA 48 mW	VCDL + buffers 186 mA 279 mW		LVDS pads 46.8 mA 117 mW	444 mW

Table 8.3: Measured power consumption, compared with simulation with extracted parasitics of dominant parts

8.3 Power dissipation

The power dissipation of the complete prototype is measured without hit activity and in high precision mode. During the measurement, the chip is clocked with a 40 MHz reference clock. PLL and DLL are in lock. The bin size is 24.4 ps. The DLL supply voltage can be either 1.2 V or 1.5 V. The 1.5 V option is only required for chips produced in the slow-slow process corner and operated with high temperature (125 °C) and the DLL supplied with 90% of the 1.5 V. This is the worst case, in all other corners, a DLL supply voltage of 1.2 V is sufficient to reach the bin size of 24.4 ps.

Tab. 8.3 shows the measured power of the chip as well as some power estimates based on simulation. The simulation includes extracted parasitics. For simplification, only the parts of the circuit that are considered dominant are simulated. A simulation of the full chip is not possible with the available equipment. The logic power domain contains the hit input buffers, the hit registers, the readout shift register, the configuration shift register and the DLL start-up state machine. The hit registers consume more than 80% of the power in that domain. However, it is possible that the simulated power is higher than in reality, and other parts in the same domain which are not simulated consume more than the remaining 20%.

The DLL domain consumption is estimated to be dominated by the VCDL and the buffers that drive the hit registers. Simulation of the buffers requires them to be loaded

by the distribution wires and the first inverter of the hit registers. To simplify the simulation, those wires and inverters are replaced by an RC arrangement. However, values for parasitic wire capacitances the technology design manual data gives vary by a factor of two, depending on whether the line is approximated as an isolated line with neighbouring metals or as a line with minimum spaced neighbours all around on the full length. Therefore, measuring a current that is 36% smaller than the simulated value is not surprising.

The I/O domain contains standard cell pads and 9 full custom LVDS pads, both including buffers. The consumption of the standard cell pads and buffers has been found negligible compared to the LVDS pads and buffers, which are simulated to take 5.2 mA each in the 2.5 V I/O supply domain. Again, measurements show that this simulated value is pessimistic.

9 Summary and Outlook

The goal of this work was the development of a multi-channel time-to-digital converter ASIC for high energy physics applications with a very high resolution in a standard digital CMOS 130 nm technology. After studying different TDC architectures and analysing existing TDCs, a new TDC has been designed. A novel delay element, making use of active inductors, transistors that show inductive behaviour around the operating point, has been conceived. Simulations have shown that the use of such an element can double the resolution from 48.8 ps to 24.4 ps. Aiming at LHC applications, the required clock frequencies are generated based on an external 40 MHz clock, as this is the LHC bunch crossing frequency. Special consideration has been paid to power consumption, a major issue in modern HEP detectors.

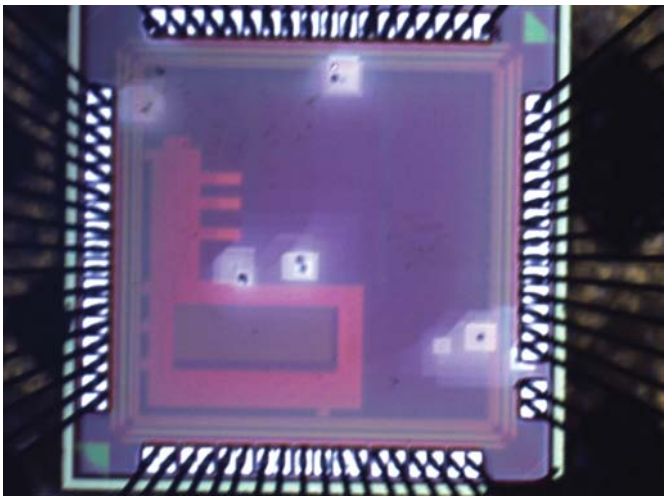


Figure 9.1: Photograph of the TDC130-0820 Prototype

A prototype (fig. 9.1) has been fabricated in order to confirm the functionality and performance of the design, especially the delay element. Based on previous experience

with the predecessor chip, the calibration scheme for the delay line which serves as a time base to all channels of the chip has been changed.

Experiments confirm the simulations. A resolution of 19 ps has been observed, the target resolution being 24.4 ps.

A future chip will contain a delay adjustment scheme to compensate for delay line tap output buffer variations in addition to the fine adjustment of the DLL delay elements included in the prototype. This will further increase the linearity. A new interpolation scheme based on DLL controlled delay lines has been devised to further improve the resolution.

A Calculations

A.1 RMS and Standard Deviation of LSB

Definition of variance σ^2:

$$\sigma^2 = E\left[(x-\mu)^2\right] = E\left[x^2\right] - E^2[x] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x)\,\mathrm{d}x$$

For a uniform distribution:

$$p(x) = \begin{cases} \dfrac{1}{x_{\max} - x_{\min}} & \text{within bin} \\ 0 & \text{else} \end{cases}$$

Assuming mean value $\mu = 0$:

$$p(x) = \begin{cases} \dfrac{1}{t_{\mathrm{LSB}}} & \text{within bin} \\ 0 & \text{else} \end{cases}$$

$$\Longrightarrow \sigma^2 = \frac{1}{t_{\mathrm{LSB}}} \int_{-\frac{1}{2}t_{\mathrm{LSB}}}^{+\frac{1}{2}t_{\mathrm{LSB}}} x^2\,\mathrm{d}x = \frac{1}{12}t_{\mathrm{LSB}}^2$$

$$\Longrightarrow \sigma = \sqrt{\frac{1}{12}}\,t_{\mathrm{LSB}} \approx \frac{t_{\mathrm{LSB}}}{3.5}$$

Definition of RMS value x_{RMS} (of $f(x) = x + \mu$):

$$x_{\mathrm{RMS}} = \sqrt{\frac{1}{t_{\mathrm{LSB}}} \int_{-\frac{1}{2}t_{\mathrm{LSB}}}^{+\frac{1}{2}t_{\mathrm{LSB}}} f(x)^2 \,\mathrm{d}x} = \sqrt{\frac{1}{t_{\mathrm{LSB}}} \int_{-\frac{1}{2}t_{\mathrm{LSB}}}^{+\frac{1}{2}t_{\mathrm{LSB}}} (x+\mu)^2 \,\mathrm{d}x} = \sqrt{\mu^2 + \frac{1}{12} t_{\mathrm{LSB}}^2}$$

With mean value $\mu = 0$:

$$\Longrightarrow x_{\mathrm{RMS}} = \sqrt{\frac{1}{12}}\, t_{\mathrm{LSB}} = \sigma$$

B Alternative Implementations

B.1 DLL Phase Detector: XOR

An XOR gate (fig. B.1) can be used as phase detector. Fig. B.2 shows its behaviour for different input phase relations. Note that in this application, the phase difference between input and output of the VCDL has to be 2π, while in most other DLL applications, the phase error, defined as the deviation from a phase difference of $\frac{\pi}{2}$, is considered. The instantaneous output voltage is either 0 or V_{DD}. The average voltage depends on the magnitude of the phase difference between VCDL input and VCDL output, but not on its sign. The XOR phase detector cannot distinguish a leading output phase from a lagging output phase [RCN03]. Therefore, within its operation range, the sign of the phase difference must not change. The phase detector cannot be designed to work with 0 phase difference. The frequency of its output signal is twice as high as the frequency of the input signal. This is convenient when a simple RC filter is used in the next stage, but can pose problems for charge pump filters.

The transfer function, the relation between the average voltage of the phase detector output and the phase difference between VCDL input and VCDL output, is shown in fig. B.3. The slope $K_{\mathrm{PD}} = \frac{\mathrm{d}V_{\mathrm{Error}}}{\mathrm{d}\varphi}$ is called phase detector gain. K_{PD} must be positive in order to achieve negative feedback in the DLL, thus the DLL has to operate with a static phase difference of $\frac{\pi}{2}$. The absolute value of the gain is $\frac{V_{\mathrm{DD}}}{\pi}$.

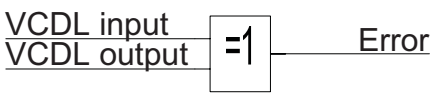
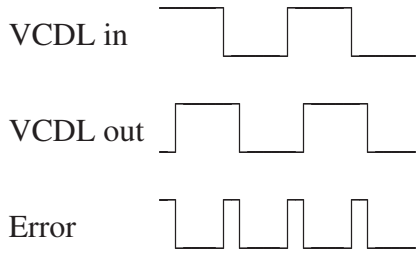
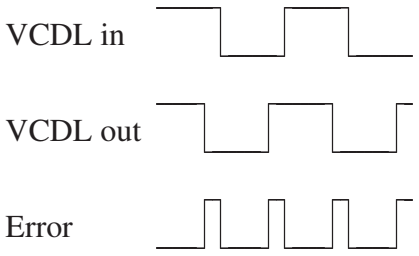


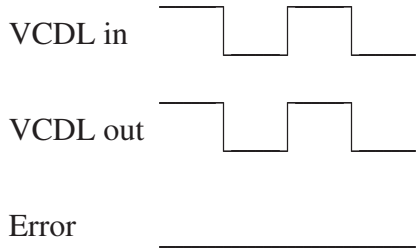
Figure B.1: XOR Phase Detector



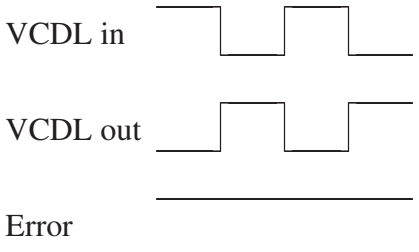
(a) Output lags by $\frac{1}{4}\pi$:
$\text{average}(V_{\text{Error}}) = \frac{1}{4}V_{\text{DD}}$



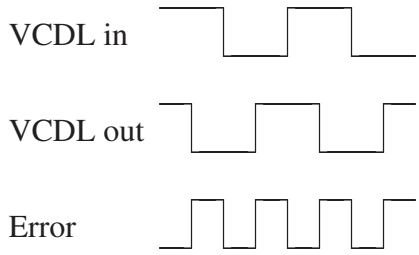
(b) Output leads by $\frac{1}{4}\pi$:
$\text{average}(V_{\text{Error}}) = \frac{1}{4}V_{\text{DD}}$



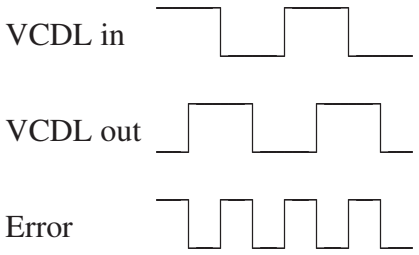
(c) Output in phase with input:
$\text{average}(V_{\text{Error}}) = 0$



(d) Output lags by π:
$\text{average}(V_{\text{Error}}) = V_{\text{DD}}$



(e) Output leads by $\frac{1}{2}\pi$:
$\text{average}(V_{\text{Error}}) = \frac{1}{2}V_{\text{DD}}$



(f) Output lags by $\frac{1}{2}\pi$:
$\text{average}(V_{\text{Error}}) = \frac{1}{2}V_{\text{DD}}$

Figure B.2: XOR Phase Detector: Pulse Diagrams

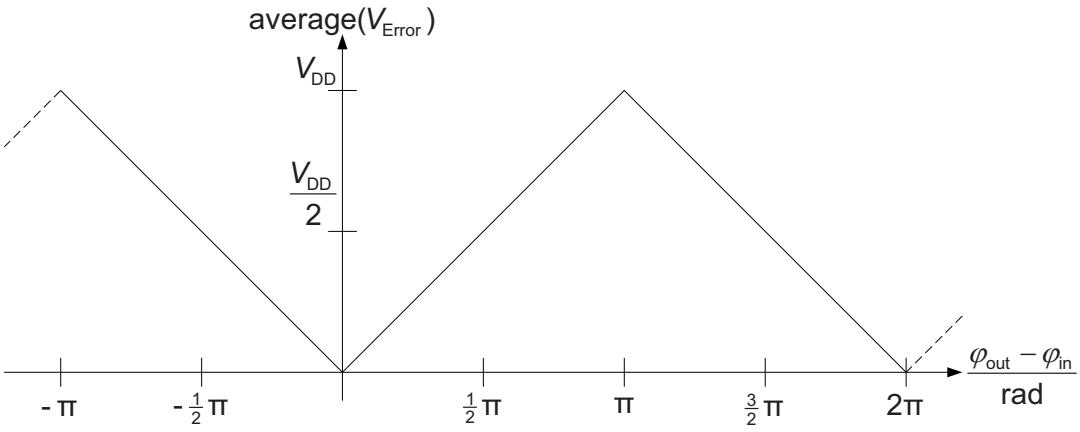
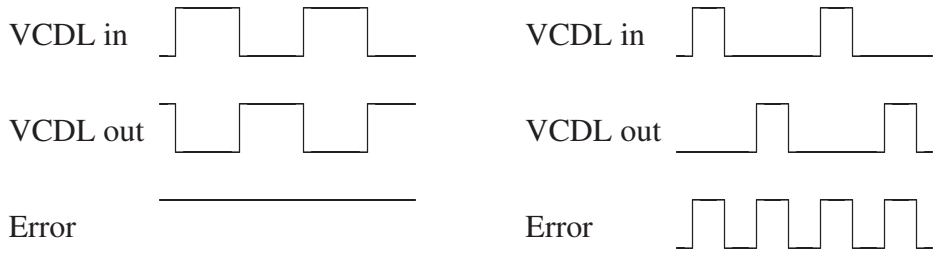


Figure B.3: XOR Phase Detector: Transfer function (for 50% duty cycle input signals)

The XOR phase detector is sensitive to duty cycle distortions both on the VCDL input and the VCDL output (fig. B.4 and B.5). If the duty cycle is not 50%, the average output can be the same for two different phase differences. This means the phase detector cannot distinguish two different phase relations, and cannot take corrective action. Therefore, the gain K_{PD} is 0 in the concerned range of phase difference. This effect is called gain saturation.

B.2 DLL Loop Filter: RC Filter and Amplifier

The simplest possible low pass filter is the RC filter (fig. B.6). It requires one input signal, and its output corresponds to the moving average of the input voltage. It is appropriate for use with an XOR phase detector. The time constant τ is fixed by the product $R \cdot C$. Assuming $\frac{1}{\tau}$ to be chosen much smaller than the frequency of the signal propagating in the VCDL, the output voltage of the RC filter is proportional to the phase error at the phase detector input: $V_{\mathrm{ctrl}} = (K_{\mathrm{LF}} K_{\mathrm{PD}}) \cdot \varphi_{\mathrm{error}}$, the proportionality constant being the loop filter gain K_{LF} times the phase detector gain K_{PD}. In order to compensate quickly for any detected phase error, it is desirable to have K_{PD} as large a possible. Recalling the definition of K_{PD} for XOR phase detectors, $K_{\mathrm{PD}} = \frac{V_{\mathrm{DD}}}{\pi}$, the only parameter that can be modified is V_{DD}, which has undesirable side-effects and limits imposed by the used chip technology. Alternatively, the loop filter gain K_{LF}, which is



(a) 50% duty cycle : average(V_{Error}) = V_{DD}

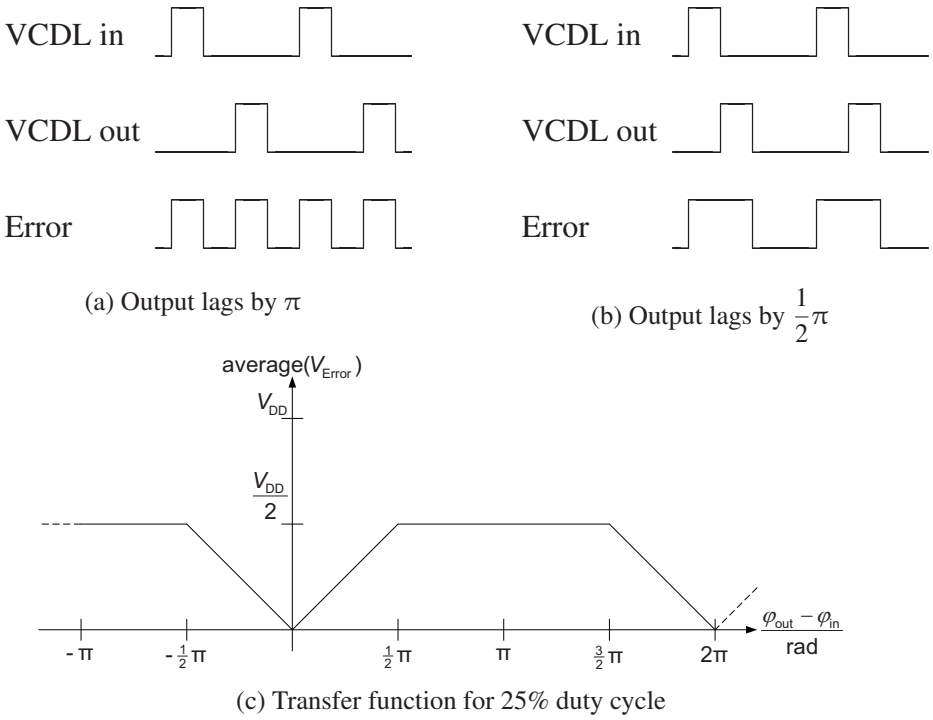
(b) 25% duty cycle : average(V_{Error}) = $\frac{1}{2}V_{\text{DD}}$

Figure B.4: XOR Phase Detector: For different duty cycles and identical phase relations, the XOR phase detector average voltage is different

1 for a passive RC filter, can be increased adding an amplifier with gain G after the RC filter or using an active low pass filter. This however introduces problems related to amplifiers into the circuit, namely secondary poles that can cause instability. Another possibility is increasing the gain of the VCDL, $K_{\text{VCDL}} = \frac{\mathrm{d}t_{\text{LSB}}}{\mathrm{d}V_{\text{ctrl}}}$. In both cases, small variations of the RC filter output voltage lead to large delay variations in the VCDL, thus jitter. This is fundamental to the XOR phase detector and RC filter principle, as the signal-to-noise ratio (SNR) of a chain is dominated by the stage with the smallest SNR. As the noise is similar in all stages, and the smallest signal is always the RC filter output voltage, adding amplifiers after the RC filter doesn't improve the overall jitter.

B.3 PLL Phase Detector: Analogue Multiplier

The Analogue Multiplier is a circuit frequently used in RF transmission applications, where it is also referred to as Mixer. It performs a multiplication of 2 input signals s_1



(a) Output lags by π

(b) Output lags by $\frac{1}{2}\pi$

(c) Transfer function for 25% duty cycle

Figure B.5: XOR Phase Detector: Gain Saturation: For different phase relations, the XOR phase detector average voltage can be the same if the duty cycle is not 50%

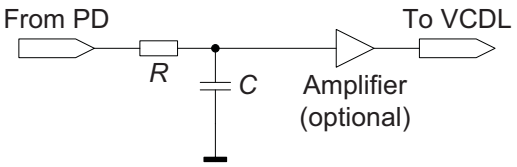


Figure B.6: RC Loop Filter with optional amplifier

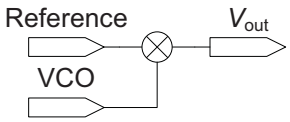


Figure B.7: PLL Phase Detector: Analogue Multiplier

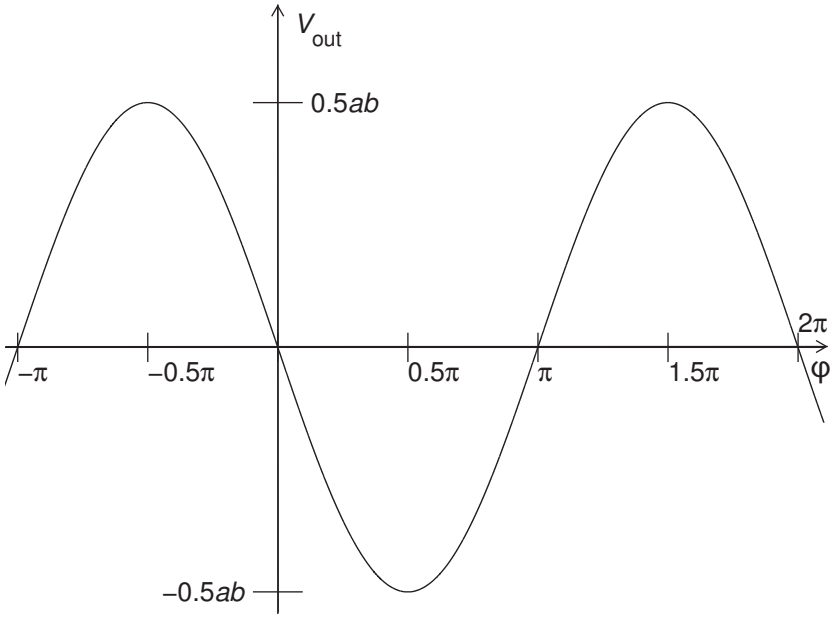


Figure B.8: Analogue multiplier transfer function

and s_2 in the time domain. Let $s_1(t) = a\cos(\omega t)$ and $s_2(t) = b\cos(\omega t + \varphi)$ be of equal frequencies $\omega = 2\pi f$, but with a phase difference φ. The product is

$$\begin{aligned} s_1(t) \cdot s_2(t) &= (a\cos(\omega t)) \cdot (b\cos(\omega t + \varphi)) \\ &= \frac{ab}{2}(\cos(\varphi) - \cos(2\omega t + \varphi)) \end{aligned}$$

and can be written as the sum of a phase difference dependent DC signal and a signal at twice the input frequency, which will be removed by a low pass filter. When using a passive low pass filter, it is convenient to have the signal to be filtered away at a very high frequency. For an active filter, very high frequency components require careful design. The output voltage of the low pass filter

$$V_{\text{out}} = \frac{ab}{2}\cos(\varphi)$$

is a function of the signals' amplitudes, making the gain change when the signal am-

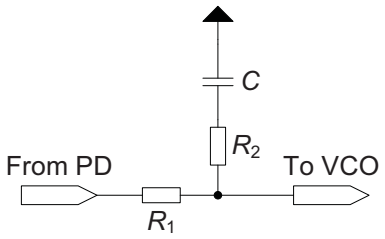


Figure B.9: Passive RRC Loop Filter

plitudes change, and a function of the cosine of the phase difference (fig. B.8). The phase detector gain

$$K_{\mathrm{PD}} = \frac{\mathrm{d}V_{\mathrm{out}}}{\mathrm{d}t} = -\frac{ab}{2}\sin(\varphi)$$

is zero for $\varphi = 0$ and $\varphi = \pi$, and its absolute value is maximum for $\varphi = \frac{\pi}{2}$. The analogue multiplier cannot be designed to operate at zero phase difference. The design phase difference should be $\frac{\pi}{2}$, and exceeding the range of $0 \leq \varphi \leq \pi$ leads to inversion of the sign of the gain and thus of the loop feedback. To acquire lock, difference of the input reference frequency and the divided VCO output frequency must be within the loop bandwidth. The phase detector itself is only sensitive to the phase of the signals, but not to their frequencies. It is the control loop as a whole that makes it sensitive to frequency differences, and this only within the loop bandwidth. The XOR phase detector can be considered as an analogue multiplier with square-wave input signals. Therefore, it is not analysed separately.

B.4 PLL Loop Filter: Passive RRC filter

The passive RRC filter corresponds to the passive RC filter for DLLs. The resistor R_2 is added to introduce a zero to the transfer function. It is suitable for use together with an analogue multiplier phase detector. Its transfer function is

$$H(s) = \frac{\frac{1}{sC} + R_2}{R_1 + \frac{1}{sC} + R_2} = \frac{1 + sR_2C}{1 + s(R_1 + R_2)C}$$

with a zero at

$$f_Z = \frac{1}{2\pi R_2 C}$$

and a pole, coupled to the zero, at

$$f_P = \frac{1}{2\pi (R_1 + R_2) C} = f_Z \frac{R_2}{R_1 + R_2}$$

The filter contains only passive devices, its gain is 1. As in DLLs, this leads to an undesirable phase offset. The gain can be increased adding an amplifier, with the same side-effects as for the DLL.

List of Symbols

D	Duty cycle
f	Frequency
f_{BW}	Bandwidth
i	Integer number
I	Current
j	Imaginary unit
J	Jitter
K	Gain in DLL and PLL
M, N	Number of delay elements in a DLL
N	PLL divider ratio
t	Time
$t_{\mathrm{DNL},i}$	Differential Non-Linearity of a converter
$t_{\mathrm{INL},i}$	Integral Non-Linearity of a converter
t_{LSB}	Lowest Significant Byte of a converter
t_{M}	Bin size of the secondary interpolation circuit
t_{max}	Dynamic range of a converter
t_{N}	Bin size of the main time base
T_{clk}	Clock period
Δt	Time difference, delay
Δt_i	Size of bin i
ξ	Damping factor
σ_{DNL}	Standard deviation of DNL
σ_{INL}	Standard deviation of INL
σ_{LSB}	Standard deviation of t_{LSB}
τ	Time constant

φ	Phase
ω	Angular frequency ($\omega = 2\pi f$)

List of Abbreviations

ADC	Analogue-to-Digital Converter
ALICE	A Large Ion Collider Experiment – an LHC experiment
ASIC	Application Specific Integrated Circuit
CCDL	Current Controlled Delay Line
CDT	Code Density Test
CERN	European Organization for Nuclear Research/Organisation européenne pour la recherche nucléaire
CLIC	Compact LInear Collider, a possible future e^+e^- collider
CMS	Compact Muon Solenoid – an LHC experiment
DFF	D flip flop
DLL	Delay Locked Loop
DNL	Differential Non-Linearity
FIFO	First In-First Out memory
FPGA	Field Programmable Gate Array
GBT	GigaBit Transceiver
HEP	High Energy Physics
HPTDC	High Precision TDC, predecessor of the TDC130
INL	Integral Non-Linearity
LHC	Large Hadron Collider, a 2×7 TeV particle accelerator at CERN
LIDAR	LIght Detection And Ranging
LSB	Least Significant Bit
MRPC	Multi Resistive Plate Chamber
NA62	A fixed target experiment, also known as P326 and NA48/3, at CERN's SPS accelerator

PFD	Phase-Frequency Detector
PID	Particle IDentification
PLL	Phase Locked Loop
RADAR	RAdio Detection And Ranging
RF	Radio Frequency
RMS	Root Mean Square
SNR	Signal-to-Noise Ratio
SPS	Super Proton Synchrotron, a 400 GeV proton accelerator at CERN
TDC	Time-to-Digital Converter
TDC130	New TDC in 130 nm-technology currently being designed
TDC130-0820	First TDC130 prototype
TOF	Time of Flight
TOT	Time Over Threshold
VCDL	Voltage Controlled Delay Line
VCO	Voltage Controlled Oscillator

Bibliography

[AAea09] A. Akindinov, A. Alici, and P. Antonioli et al. A topological trigger based on the time-of-flight detector for the alice experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 602(2):372 – 376, 2009. ISSN 0168-9002.

[AEF+04] P. B. Amaral, N. Ellis, P. Farthouat, P. Gallno, J. Haller, T. Pauly, H. P. Lima Jr., T. Maeno, I. R. Arcas, J. M. de Seixas, G. Schuler, R. Spiwoks, R. T. Teixeira, and T. Wengler. The ATLAS level-1 central trigger system. In *IEEE Nuclear Science Symposium Conference Record*, volume 3, pages 1673–1677, Oct. 2004. ISSN 1082-3654.

[AI96] Y. Arai and M. Ikeno. A time digitizer CMOS gate-array with a 250 ps time resolution. *IEEE Journal of Solid-State Circuits*, 31(2):212–220, Feb 1996. ISSN 0018-9200.

[ALI95] ALICE Collaboration. *ALICE: Technical proposal for a Large Ion collider Experiment at the CERN LHC*. CERN, Geneva, 1995. ISBN 92-9083-077-8.

[ALI02] ALICE Collaboration. *ALICE Time-Of Flight system (TOF): Addendum to the Technical Design Report*. CERN, Geneva, 2002. ISBN 92-9083-192-8.

[BCdR+06] H. H. Braun, R. Corsini, A. de Roeck, A. Grudiev, S. T. Heikkinen, E. Jensen, M. S. Korostelev, D. Schulte, I. V. Syratchey, F. A. Tecker,

W. Wuensch, and F. Zimmerman. Updated CLIC parameters 2005. Technical Report CERN-OPEN-2006-022. CLIC-Note-627, CERN, Geneva, May 2006.

[BCL+04] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock. *LHC Design Report*. CERN, Geneva, 2004. ISBN 92-9083-224-0.

[BGG+05] M. Bondila, V. A. Grigoriev, F. F. Guber, V. A. Kaplin, A. I. Karakash, O. V. Karavichev, T. L. Karavicheva, A. I. Klimov, N. Kondratieva, K. N. Kozlov, A. B. Kurepin, V. A. Loginov, V. V. Lyapin, T. Malkiewicz, V. V. Marin, A. I. Maevskaya, E. A. Meleshko, A. I. Reshetin, W. H. Trzaska, and S. Yamaletdinov. ALICE T0 detector. *IEEE Transactions on Nuclear Science*, 52(5):1705–1711, Oct. 2005. ISSN 0018-9499.

[BKP+98] B. Batyunya, S. Kiselev, G. Paic, K. Safarík, A. V. Smirnitsky, and B. V. Zagreev. Influence of the Time Resolution of the Time of Flight System in ALICE on the Measurement of Observables. Technical Report ALICE-INT-1998-08, CERN, Geneva, 1998.

[CAA57] Churchman, Ackoff, and Arnoff. *Introduction to Operations Research*. Wiley& Sons, 1957.

[Cec07] A. Ceccucci. NA62/P-326 Status Report. Technical Report SPSC-M-760. CERN-SPSC-2007-035, Nov. 2007.

[Chr04] J. Christiansen. *High Performance Time to Digital Converter, Manual Version 2.2 for HPTDC 1.3*. CERN, Geneva, Mar 2004.

[CMS00] CMS Collaboration. *Addendum to the CMS Tracker TDR*. CERN, Geneva, Feb. 2000.

[Cyp97] Are Your PLDs Metastable? Cypress Semiconductor Corporation, AN1211, Mar. 1997.

[DDTGG05] N. Da Dalt, E. Thaller, P. Gregorius, and L. Gazsi. A compact triple-band low-jitter digital LC PLL with programmable coil in 130-nm CMOS.

IEEE Journal of Solid-State Circuits, 40(7):1482–1490, July 2005. ISSN 0018-9200.

[DGLN95] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls. An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors. *IEEE Journal of Solid-State Circuits*, 30(4):412–422, Apr. 1995. ISSN 0018-9200.

[Gar80] Floyd M. Gardner. Charge-Pump Phase-Lock Loops. *IEEE Transactions on Communications*, 28(11):1849–1858, Nov. 1980.

[Hal09] G. Hall. The upgrade of the CMS Tracker for Super-LHC. Preprint: `www.imperial.ac.uk/research/hep/preprints/09-2.pdf`, 2009.

[Has97] E. Haseloff. *Metastable Response in 5-V Logic Circuits*. Texas Instruments, Feb. 1997. Application Report SDYA006.

[HEC89] J. U. Horstmann, H. W. Eichel, and R. L. Coates. Metastability behavior of CMOS ASIC flip-flops in theory and test. *IEEE Journal of Solid-State Circuits*, 24(1):146–157, Feb. 1989. ISSN 0018-9200.

[IEE01] IEEE standard for terminology and test methods for analog-to-digital converters. *IEEE Std 1241-2000*, 2001.

[Jai91] Jain. *The Art of Computer Systems Performance Analysis*. Wiley& Sons, 1991. ISBN 0-471-50336-3.

[JCG07] Le Jin, Degang Chen, and R. Geiger. Code-Density Test of Analog-to-Digital Converters Using Single Low-Linearity Stimulus Signal. In *25th IEEE VLSI Test Symposium*, pages 303–310, May 2007. ISSN 1093-0167.

[Kal] J. Kalisz. *Review of methods for time interval measurements with picosecond resolution*. Military University of Technology, Warsaw.

[KWG94] Beomsup Kim, T. C. Weigandt, and P. R. Gray. PLL/DLL system noise analysis for low jitter clock synthesizer design. In *IEEE International Symposium on Circuits and Systems, ISCAS 1994*, volume 4, pages 31–34, May/June 1994.

[Lef06] C. Lefevre. *The CERN accelerator complex. Complexe des accélérateurs du CERN*. CERN, Geneva, Jun 2006. CERN-DI-0606052.

[LHC01] LHCb Collaboration. *LHCb Outer Tracker Technical Design Report*. CERN, Geneva, 2001. ISBN 92-9083-200-2.

[Man96] J. G. Maneatis. Low-jitter process-independent DLL and PLL based on self-biased techniques. *IEEE Journal of Solid-State Circuits*, 31(11): 1723–1732, Nov. 1996. ISSN 0018-9200.

[MC99] M. Mota and J. Christiansen. A High-Resolution Time Interpolator Based on a Delay Locked Loop and an RC Delay Line. *IEEE Journal of Solid-State Circuits*, 34(10):1360–1366, Oct. 1999. ISSN 0018-9200.

[Mor07] P. Moreira. Introduction to Integrated Delay and Phase-Locked Loops and Applications. Padova Lecture notes, 2007.

[Mot00] M. Mota. *Design and Characterization of CMOS High-Resolution Time-to-Digital Converters*. PhD thesis, Universidade técnica de Lisboa, Lisbon, 2000.

[MRT+07] A. K. M. K. Mollah, R. Rosales, S. Tabatabaei, J. Ciclao, and A. Ivanov. Design of a Tunable Differential Ring Oscillator With Short Start-Up and Switching Transients. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(12):2669–2682, Dec. 2007. ISSN 1549-8328.

[Par08] A. Parenti. The CMS Muon System and Its Performance in the CMS Cosmic Challenge. *IEEE Transactions of Nuclear Science*, 55(1):113–121, Feb. 2008. ISSN 0018-9499.

[PMK02] P. Palojarvi, K. Maatta, and J. Kostamovaara. Pulsed Time-of-Flight Laser Radar Module With Millimeter-Level Accuracy Using Full Custom Receiver and TDC ASICs. *IEEE Transactions on Instrumentation and Measurement*, 51(5):1102–1108, Oct 2002. ISSN 0018-9456.

[Por73] D. I. Porat. Review of sub-nanosecond time-interval measurements. *IEEE Transactions on Nuclear Science*, 20(5):36–51, Oct. 1973. ISSN 0018-9499.

[Raz01] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, Singapore, 2001. ISBN 0-07-118839-8.

[RCN03] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits*. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, Jan. 2003. ISBN 0-13-090996-3.

[RRRK00] E. Raisanen-Ruotsalainen, T. Rahkonen, and J. Kostamovaara. An Integrated Time-to-Digital Converter with 30-ps Single-Shot Precision. *IEEE Journal of Solid-State Circuits*, 35(10):1507–1510, Oct 2000. ISSN 0018-9200.

[SRGR03] D. I. Sanderson, J. C. Rautio, R. A. Groves, and S. Raman. Accurate modeling of monolithic inductors using conformal meshing for reduced computation. *IEEE Microwave Magazine*, 4(4):87–96, Dec. 2003. ISSN 1527-3342.

[TS02] U. Tietze and Ch. Schenk. *Halbleiter-Schaltungstechnik*. Springer Verlag, 2002. ISBN 3-540-42849-6.

[WKG94] T. C. Weigandt, Beomsup Kim, and P. R. Gray. Analysis of timing jitter in CMOS ring oscillators. In *IEEE International Symposium on Circuits and Systems, ISCAS 1994*, volume 4, pages 27–30, May/June 1994.

[WWCL05] Jinn-Shyan Wang, Yi-Ming Wang, Chin-Hao Chen, and Yu-Chia Liu. An Ultra-Low-Power Fast-Lock-in Small-Jitter All-Digital DLL. In *IEEE*

International Solid-State Circuits Conference (ISSCC), Digest of Papers, volume 1, pages 422–607, Feb. 2005. ISSN 0193-6530.

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors Prof. Norbert Wermes and Paulo Moreira for the guidance and supervision of my PhD project.
I would like to thank my colleagues at CERN for the friendly working environment and the help I have received throughout the project.
I would like to thank my family and my friends for their support.