All-Digital Standard-Cell Based Audio Clock Synthesis

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
ERIC PASCAL ROTH
Dipl. El.-Ing. ETH
born April 14, 1975
citizen of Daenikon ZH and Luzern

accepted on the recommendation of
Prof. Dr. Wolfgang Fichtner, examiner
Dr. Steven Harris, co-examiner

2004
Acknowledgments

I would like to thank my supervisor, Prof. Dr. Wolfgang Fichtner, for his support and establishing the professional working environment at the Integrated Systems Laboratory (IIS). I also would like to thank Dr. Steven Harris for reading and co-examining this thesis.

A special thank goes to Dr. Norbert Felber for all the discussions, his scientific support, for the reading and commenting of this work.

This work has been carried out in cooperation with BridgeCo AG, Dübendorf. I would like to thank the ASIC design team of BridgeCo headed by Markus Thalmann for their input and support during the implementations. I am also thankful to my working colleague Thomas Boesch for the collaboration during the chip integrations and the proof-reading of the thesis.

I am very grateful to my parents for all they have done for me. I would like to express my gratitude to my family, my wife Nara and my son Tobias, for their loving support and patience during the past four years I was working on this project.
Contents

Acknowledgments v

Abstract xiii

Zusammenfassung xv

1 Introduction 1
  1.1 Motivation ................................. 1
  1.2 Goal of the Thesis .......................... 3
  1.3 Outline of the Thesis ........................ 3
  1.4 Contribution of the Thesis .................. 4

2 Timing Distribution in real-time Networks 7
  2.1 Real-time Computer Networks ................. 7
  2.2 Overview on IEEE 1394 ...................... 8
    2.2.1 Topology .............................. 9
    2.2.2 Asynchronous Transfers .................. 11
    2.2.3 Isochronous Transfers and Cycle Master ... 11
## CONTENTS

2.3 IEC 61883 Protocol .......................... 12
   2.3.1 Timestamps ............................ 13

2.4 Accuracy of Timing Distribution ............... 14
   2.4.1 Variable Transmission Latency .......... 15
   2.4.2 Transfer Delay Compensation .......... 16

2.5 Summary ..................................... 18

3 Phase-locked Loops for Sample Clock Recovery 21
   3.1 Jitter Limits of Audio Clocks .......... 22
      3.1.1 Distortions caused by Sampling Jitter .. 22
      3.1.2 Human Perception of Jitter in Audio Signals . 24
      3.1.3 Jitter Attenuation .................... 28
   3.2 Phase-locked Loops ......................... 29
      3.2.1 Analog PLL .......................... 30
      3.2.2 Semi-digital PLL ...................... 32
      3.2.3 All-digital PLL ....................... 35
   3.3 DCO Requirements ........................ 36
      3.3.1 Frequency Range ..................... 36
      3.3.2 Frequency Resolution ................. 37
      3.3.3 Intrinsic Jitter ..................... 39
   3.4 Review of Clock Generation Circuits ........ 39
      3.4.1 Voltage-controlled Oscillators ......... 39
      3.4.2 Voltage-controlled Ring Oscillators ..... 40
      3.4.3 All-Digital Ring Oscillators .......... 41
      3.4.4 Direct Digital Synthesis ............. 43
      3.4.5 Synchronous Clock Generation ........ 44
      3.4.6 Summary and Discussion ............... 46
4 Phase Shift DCO

4.1 Operation Principle ........................................ 49
4.2 Circuit Architecture ....................................... 51
  4.2.1 Delay Line ............................................. 52
  4.2.2 Accumulator ............................................ 60
4.3 Calibration ...................................................... 66
  4.3.1 Calibration with a Delay-locked Loop ............... 66
  4.3.2 Calibration with a Ring Oscillator ..................... 72
4.4 Design Methodology Issues ................................. 74
  4.4.1 Pipelining the Accumulation ............................ 74
  4.4.2 Timing Constraints ..................................... 76
  4.4.3 Choosing the Delay Line Length ......................... 80
  4.4.4 Placement and Routing ................................ 82
  4.4.5 Testability Issues ..................................... 82
4.5 Jitter Analysis ................................................ 85
  4.5.1 Period Jitter ........................................... 85
  4.5.2 Jitter due to Delay Line Resolution .................... 86
  4.5.3 Jitter caused by Non-linearity of the Delay Line .... 87
  4.5.4 Jitter caused by Power Supply Noise ................... 88
  4.5.5 Jitter due to Calibration Errors ...................... 89
  4.5.6 Total Jitter ............................................ 91
4.6 Implementations and Results ................................ 92
  4.6.1 Chip I - 0.6 μm CMOS ................................. 92
  4.6.2 Chip II - 0.25 μm CMOS ............................... 99
  4.6.3 Chip III - 0.18 μm CMOS .............................. 105
  4.6.4 Chip IV - 0.18 μm CMOS .............................. 108
4.7 Generation of System Clocks ............................... 110
4.8 Summary ...................................................... 110
5 Enhanced Ring DCO

5.1 Operation Principle .................. 113
5.2 Circuit Architecture .................. 116
   5.2.1 Pulse Generation .................. 116
   5.2.2 Decoder and Delay Line ............ 118
   5.2.3 Frequency Range .................. 119
   5.2.4 Frequency Resolution ............... 121
5.3 Frequency Calibration .................. 122
5.4 Design Methodology Issues ............... 124
   5.4.1 Clocking ......................... 124
   5.4.2 Placement and Routing .............. 125
   5.4.3 Testability ....................... 125
5.5 Jitter Analysis ....................... 127
   5.5.1 Jitter in Ring Oscillators ......... 127
   5.5.2 Jitter due to Limited Delay Resolution .... 127
   5.5.3 Jitter caused by Delay Line Non-Linearity ... 128
   5.5.4 Jitter caused by Power Supply Noise .... 128
5.6 Implementation and Results ............... 128
   5.6.1 Chip II - 0.25 μm CMOS ............. 128
5.7 Summary ......................... 132

6 Conclusion .......................... 135

6.1 Main Achievements .................... 135
6.2 Outlook ............................ 137
6.3 Future Work .......................... 137
Abstract

Consumer electronic (CE) devices connected by computer networks become more and more popular. Data transportation between CE devices must exhibit real-time behaviour to prevent interruptions of the data stream. This requires a synchronization of the data rates at the sending and the receiving streaming device, which is achieved with the transmission of timing information over the network. A phase-locked loop (PLL) in the receiver filters the jitter on the timing information and generates an accurate audio sample clock.

In order to save cost and to allow a high portability to different CMOS technologies, the PLL is proposed to compose of standard-cells only and integrated into the System-On-Chip (SoC) of the CE device. The key component of the all-digital PLL is the digitally-controlled oscillator (DCO), which generates the audio clock.

In this work, two alternative DCO circuits are proposed. The first is based on the phase shift principle, where the position of the output clock edges are pre-calculated, and a delay line shifts the edges of a fixed input clock to the predetermined position. An all-digital delay-locked loop (DLL) is used for the calibration of the delay line.

The second DCO architecture is based on a ring oscillator. The limited frequency resolution of conventional digital ring oscillators is extended by a method that switches rapidly between two adjacent frequency steps. A frequency calibration mechanism tunes the DCO to the target operation range.

Both DCO architectures are implemented on silicon for proof of
The measurement results demonstrate that high-quality audio clock recovery is achieved with the phase shift DCO. The ring oscillator based DCO offers a significantly lower circuit complexity, but due to the high clock jitter, it is only applicable to lowest-cost, and less quality demanding devices.

As an important goal of the thesis, the porting of the DCO architecture to different CMOS technologies could be realized within two person-days.
Zusammenfassung


Um Kosten zu sparen und um die Portabilität von einer CMOS Technologie zur nächsten zu gewährleisten, wird der PLL ausschließlich aus Standard-Zellen zusammengesetzt und auf dem System-On-Chip (SoC) des Gerätes integriert. Das zentrale Element des voll digitalen PLL ist der Digitally-Controlled Oscillator (DCO), der den Audio-Takt erzeugt.


Eine zweite, einfachere DCO Architektur basiert auf einem Ring-Oszillator. Herkömmliche digitale Ring-Oszillatoren haben eine begrenzte Frequenzauflösung. Es wird eine Methode vorgestellt, die
durch schnelles Umschalten zwischen zwei Frequenzeinstellungen die Auflösung verbessert. Um den DCO in den gewünschten Frequenzbereich zu ziehen, wird eine Methode zur Frequenzkalibrierung vorgeschlagen.


Die Portierung des DCOs auf verschiedene CMOS Technologien kann mit einem Aufwand von zwei Arbeitstagen bewerkstelligt werden, womit ein weiteres wichtiges Ziel der Arbeit erreicht ist.
Chapter 1

Introduction

1.1 Motivation

An increasing variety of digital consumer electronic (CE) devices is finding its way into almost every home. This includes DVD players, CD players, digital TV sets, home theater systems, and digital video cameras to mention just a subset.

Data is stored and processed digitally in today's CE equipment. The high data rates of video and multi-channel audio demand for high processing performance and a wide variety of supported interfaces.

The CE market is characterized by strong competition, high selling volumes, and a heavy cost pressure. The requirement for low-cost solutions prohibits the composition of these complex devices with individual integrated circuits (IC) mounted on a printed circuit board (PCB). Therefore, CE processing and interface devices are inevitably integrated on a System-on-Chip (SoC).

In order to exchange data among these devices, an interconnection network is necessary. When data are recorded or played back, the transportation must exhibit real-time behaviour to prevent interruptions in the data stream. Real-time transmission requires adjustment
of data rates between sender and receiver to avoid buffer over- and underflows.

Data rate synchronization is achieved with the exchange of timing information via the network. The receiver uses this information to adjust its local data rate. The timing packets are affected by variable transmission latency on the network, which adds timing jitter to the synchronization information. Therefore, a phase-locked loop (PLL) is required to filter the jitter and to recover the exact timing in the receiver. Due to the particularly high requirements on the regularity of audio conversion, the application addressed by this thesis is audio sample clock generation.

PLLs have a long tradition in communication systems and clock synthesis applications. Initially, PLLs were purely analog circuits with an analog voltage-controlled oscillator (VCO) as a key component. With the advances of digital ASIC technologies, parts of the control loop were implemented digitally. Today, the vast majority of PLLs uses both digital and analog building blocks (semi-digital). Truly all-digital PLLs (ADPLL) are rather rare, because no established technique exists for digitally-controlled oscillators (DCO) to replace the analog VCO.

Most ASIC vendors offer semi-digital PLLs as macro-cells, mainly intended for system clock generation in the range of several hundred MHz. The dynamic properties though of these macro-cells are not suitable for the low-frequency operating range of the audio clock recovery problem addressed in this thesis.

Integration of customized semi-digital or analog PLLs requires significant manual design effort. The layout of the analog building blocks must be re-designed for every technology change. This lack of flexibility increases the design time and risk. The short life cycle of CE products demands for fast time-to-market and makes these disadvantages a serious handicap. This limitation is especially severe for fab-less design houses, which avoid the dependency on a particular ASIC vendor.

As an alternative to the integrated semi-digital and analog PLLs, the analog building blocks can be placed off-chip. The analog elements
are then located on the PCB beside the SoC. This increases flexibility and lowers design time and risk, since the analog circuits are implemented with off-the-shelf components. However, the expenses for the additional ICs (in the order of dollars) results in higher system cost. The increased silicon cost of the SoC caused by the integration of the PLL is much lower (in the order of cents). This gap is a serious argument in CE products and will even grow with the advancement of ASIC technologies, allowing a higher density of the integrated circuits.

1.2 Goal of the Thesis

The goal of this thesis is the investigation of an all-digital clock recovery PLL with the following properties:

- To reduce cost, the PLL shall be fully integrated in the SoC.
- It shall be built of digital standard-cells only in order to scale with process advances, to offer high portability, and to be compatible with a digital design flow.
- The clock jitter performance shall be sufficient to enable acceptable audio quality.

The main issue of this thesis is the design of a suitable DCO. All additional ADPLL blocks are developed in a straight-forward manner.

1.3 Outline of the Thesis

First, the general requirements for audio real-time transmission are discussed in chapter 2. As an example network, the IEEE 1394 bus technology is summarized. A protocol for data transmission and timing distribution is presented and its limited accuracy is discussed.

Chapter 3 explains the need for audio sample clock recovery. The effects of sample clock jitter on the digital-to-analog converted audio
signal is discussed. The jitter limits for audible distortions are quantified with psychoacoustic principles. A comparison of the transmission jitter with the psychoacoustic jitter limit results in a 'minimal jitter attenuation' curve. Properties of analog, semi-digital, and all-digital PLLs are discussed, and a specification for the DCO in terms of frequency range, frequency resolution, and jitter performance is derived. A review summarizes the characteristics of existing clock generation circuits.

A novel DCO architecture is proposed in chapter 4. An arithmetic block calculates the time shift of an output clock edge relative to a stable input clock. A delay line shifts the edge to the predefined position. A calibration circuit matches the delay line to the actual delay conditions prior to operation.

In this thesis, the term digital is related to the binary representation of the signal amplitudes, which is maintained in the circuits proposed in this work. However, due to the implemented delay lines, the operation of several building blocks is asynchronous.

In chapter 5, an alternative, more simple DCO architecture is proposed. It is based on a ring oscillator. The limited frequency resolution inherent to conventional digital ring oscillators is enhanced. By rapidly switching among two adjacent delay settings, an average frequency can be produced with a high resolution. A frequency calibration block is proposed to tune the ring oscillator to the desired operating range.

Finally, the results of the thesis are summarized and conclusions are drawn in chapter 6.

1.4 Contribution of the Thesis

The main contribution of the thesis is the development and analysis of two DCOs suitable for audio sample clock recovery. A detailed description of the circuits and experimental results are given. Test integrations proved that standard cell-based clock synthesis is feasible with a jitter performance enabling sufficient audio quality for CE applications.
The main advantages of standard-cell designs mentioned above could be exploited. Practical experience has shown that porting of the DCOs from one ASIC technology to the next is possible within two person-days. Additionally, the digital design methodology offers a number of desirable side-effects, such as the change of PLL parameters during operation. This alleviates the trade-off between fast lock time and high jitter attenuation. Furthermore, test concepts are also proposed for the presented circuits.
Chapter 2

Timing Distribution in real-time Networks

2.1 Real-time Computer Networks

Computer networks referred to as Audio/Video (A/V) networks are characterized by their capability to transport real-time data. These data typically represent audio, video, or any other information that has temporary validity. The sender continuously produces a stream of samples at a fixed rate, such as a CD player. The receiver processes the data stream in the same order and with identical data rate, for instance a digital-to-analog converter (DAC) and speaker. This contrasts to file transfers, where data are copied from one device to another without tight time restrictions.

A fundamental characteristic of real-time systems is that data are coupled to a time window. This window defines the earliest and latest points in time where data can be accepted at a specific device. Data that arrive before the allowed time window cannot be stored due to limited buffer capacities. On the other hand, data arriving after a certain deadline are useless, because processing or representation could not be performed in a timely manner. The common use of broad- and
multicast transmissions prohibit re-transmission of corrupt, missing, or late data. Additionally, no guarantee exists that re-transmissions complete within the given time window.

This demands for a bandwidth management mechanism that ensures sufficient network capability for timely data delivery. This can be done in different ways, for example by choosing a network technology that provides much more bandwidth than maximum load occupies. Other approaches deploy more efficient, dynamic bandwidth reservation techniques.

The most dramatic consequence resulting from the properties of real-time systems is that devices exchanging real-time data are required to have a common sense of time. Time window, data rate, representation time, and bandwidth have only signification when related to a common reference time.

2.2 Overview on IEEE 1394

This section briefly describes the architecture of IEEE 1394, a widely used network technology with inherent real-time capabilities. In this thesis, most of the circuits explained in the chapters 4 and 5 were experimentally tested in an IEEE 1394 network.

Originally initiated by Apple Computer under the name FireWire, the serial bus architecture is specified in the IEEE 1394 standard. The first document was released in 1995 named IEEE 1394-1995 [1]. The extension IEEE 1394a-2000 [2] clarifies some points in the original standard and improves the serial bus architecture with additional features [3]. Today, most commercially available IEEE 1394 products and chip-sets support the IEEE 1394a standard. In the original specification cable speeds of 100, 200, and 400 Mbps are defined. The IEEE 1394b document [4] extends the data rate to 800, 1600, and 3200 Mbps and introduces optical fibers to the physical layer. Additionally, IEEE 1394b allows cable lengths of 100 m and more.
2.2.1 Topology

IEEE 1394 is a serial bus with up to 63 nodes structured as a tree, as illustrated in figure 2.1. On the physical layer ports are connected via point-to-point connections. Nodes with more than one port resynchronize data from an incoming port to the local clock before retransmission to all other ports.

![Bus topology diagram](image)

Figure 2.1: Bus topology with nodes (1, 2, 3, 4) having one or more ports (A, B, C)

Real-time capability of the IEEE 1394 bus is achieved by regulation of arbitration based on the bus topology. Every node holds information about the actual bus structure. At start-up and after topology changes the serial bus configuration is performed. It consists of three phases:

1. **Bus initialization:**
   During this phase, a reset signal is broadcast on the physical layer that causes all nodes to discard their previous topology information and to return to idle state.

2. **Tree identification:**

   The goal of the tree identification is to define the new topology. Every node identifies for all of its ports whether they are parent or child ports. A child port always connects to a parent port on the neighboring node (see figure 2.2). Parent ports points towards the root node and child ports
point towards leaf nodes. The singular node comprising only child ports is called root node. Figure 2.2 shows the serial bus introduced in figure 2.1 after tree identification has completed.

3. Self identification:
In this phase, a unique physical ID is assigned to every node. The procedure starts from the leaf node connected to the lowest port of the root. If this node is not a leaf, then the child on its lowest port is selected. This selection is repeated until the leaf is found (node 1 in the example shown on figure 2.2). The self ID count is increased for every node. Every port picks its ID when all its child nodes finished. The sequence in the bus shown in figure 2.2 is thus: node 1 → node 4 → node 2 → node 3. The root node obtains the highest physical ID.

Figure 2.2: Node 3 has become root node during tree identification
2.2.2 Asynchronous Transfers

Asynchronous packets are delivered without guaranteed latency limit and previous bandwidth reservation. A fairness interval ensures fair asynchronous arbitration. A round-robin priority scheme allows one bus access per fairness interval for nodes desiring asynchronous transfers. A minimum bus capacity of 20% is reserved for asynchronous packets.

The receiver node is identified by a 64-bit node address containing the 6-bit physical node ID. Delivery is confirmed by an acknowledge packet. A retry protocol regulates re-transmissions of erroneous transfers.

2.2.3 Isochronous Transfers and Cycle Master

Instead of an explicit node address, isochronous packets carry a 6-bit channel number. Since all traffic is broadcast, packets of a particular channel will be accepted by one or multiple listeners and ignored by all other nodes.

Bandwidth required for isochronous packets is reserved in advance at a singular node – the isochronous resource manager – that maintains a list of available bandwidth units. The root node acts as ‘cycle master’, which broadcasts cycle start packets (CSP) in regular 125 μs (8 kHz) intervals. The CSP serves as common clock source and indicates the beginning of a new bus cycle (see figure 2.3). Bandwidth reservation is related to the fraction of the 125 μs that a channel is allowed to occupy. Up to 80% (100 μs) of the cycle can be reserved for isochronous transfers. After the CSP has been sent, the bus is reserved for isochronous transfers. Short timing gaps signal the beginning of a new isochronous packet. When all isochronous traffic has finished, the remaining time of the cycle can be used for asynchronous packets. At the end of the isochronous phase and between asynchronous packets long subaction gaps indicate disposal for asynchronous arbitration.

All nodes participating in isochronous transfers maintain a 32-bit cycle time register (CTR) incrementing with 24.576 MHz (±100 ppm).
As illustrated in figure 2.4, the lowest 12 bits wrap after 3072 increments, resulting in a nominal wrap-frequency of 8 kHz. The next 13 bits increment at every wrap-around of the lower bits and count up to 8000 (1 second). The highest 7 bits count the second wraps.

The cycle master includes the value of its local CTR into the CSP. All nodes on the bus periodically overwrite their free-running CTRs with the values received with the CSP. With this method, divergence of CTRs is avoided and frequency synchronization within a bus is obtained.

2.3 IEC 61883 Protocol

The IEC 61883 standard [5, 6] defines a protocol for transmission of real-time data using IEEE 1394 (see figure 2.5). It specifies an additional packet header format for data within isochronous packets. It provides data flow management and connection management proto-
2.3. IEC 61883 PROTOCOL

cols for A/V data. Moreover it contains a method for synchronization of data rates relative to the CTR [7].

Figure 2.5: Audio transmission with IEC 61883 between IEEE 1394 devices. The cycle master distributes the common reference time.

2.3.1 Timestamps

This section focuses on the data rate synchronization specified in IEC 61883. The common isochronous packet (CIP) header added to isochronous packets contains a 16-bit timestamp called SYT. The SYT field specifies the presentation time in CTR notation at the receiver.

Figure 2.6 illustrates an example transmission. The transmitting node collects samples from the ADC. At the arrival of every eighth sample the actual cycle time is saved and added to a constant offset. The result is included as timestamp in the SYT field of the packet. The amount of samples that share a common timestamp is specified by the variable SYT_INTERVAL, which is defined by the protocol. The offset must be large enough to give time for transmission and processing on the receiver. On the other hand, an excessive offset will require large buffers in the receiver which is not desirable due to overall latency restrictions and limited memory resources.
The receiver compares the SYT field of incoming data to the common time reference - the cycle time register. As soon as the timestamp matches the CTR, data is re-presented. By relating the generation and re-presentation to the common reference time, sending and receiving devices ensure exact matching of their data rates.

Figure 2.6: Data rate synchronization between transmitter and receiver using timestamps according to IEC 61883 with a sampling rate of 44.1 kHz (SYT_INTERVAL = 8).

2.4 Accuracy of Timing Distribution

In the previous section the of timing distribution and data rate synchronization have been described. The precision of the timing distribution on the IEEE 1394 bus is discussed here.
2.4. ACCURACY OF TIMING DISTRIBUTION

2.4.1 Variable Transmission Latency

Since the CTR is updated upon receipt of the CSP, non-uniform intervals between CSP arrivals directly affect the regularity of the local cycle time. The irregularity originates in the variance of the transmission latency of the packets on the bus.

As shown in figure 2.7, serial data is transmitted on the IEEE 1394 bus in conjunction with a strobe signal. An EXOR of data and strobe produces a clock to sample data at the receiver. Data is resynchronized to the local clock before re-transmission. Since the system clocks of the sending and the receiving devices are independent, the delay from reception to re-transmission varies. According to the IEEE 1394-1995 standard, the frequency of the local clock of every IEEE 1394 node may deviate up to ±100 ppm from the nominal value. Therefore the local clock period $\text{clk\_per\_node}$ is node dependent.

$$\text{clk\_per\_node} = \text{clk\_per\_nominal} \times (1 + \text{freq\_deviation\_node})$$

The constant $\text{fifo\_depth}$ specifies the amount of bits that are
stored between receive and transmit circuits, it is the constant part of the delay.

The repeater delay \( rep_{\text{del}}(t) \) specifies the time between reception and transmission of a CSP at time \( t \). The IEEE 1394-1995 standard permits implementations that have a delay variance of more than one \( clk_{\text{per node}} \). However, since no practical reason exists for such an implementation, the maximum repeater delay variance is assumed to not exceed \( clk_{\text{per node}} \).

\[
rep_{\text{del node}}(t) = clk_{\text{per node}} \times (\text{fifo depth} + 1) - \text{mod}(t, clk_{\text{per node}})
\]

The effective delays of CSPs are samples of the continuous function \( rep_{\text{del node}}(t) \) at the moments when CSPs arrive \( t_{\text{csp arrival}} \). The arrival time of a CSP on node \( k+1 \) depends on the previous node \( k \) in the following way:

\[
t_{\text{csp arrival}k+1} = t_{\text{csp arrival}k} + rep_{\text{del}}(t_{\text{csp arrival}k})
\]

The bottom curve in figure 2.8 plots the transmission delays of CSPs over one hop. Due to the daisy-chain structure of the serial bus the repeater delays of all nodes in a serial line are cumulative. Figure 2.8 includes also the variable parts of the transfer delay when passing through several hops. The local clock frequency deviation of the nodes was assumed to be gaussian randomly distributed with a standard deviation of 30 ppm.

As one would intuitively expect, the amplitude of the variance increases with the amount of intermediate hops. For a chain of 16 hops, the delay varies by more than 100 ns. A similar result has been reported by Dunn in [8].

### 2.4.2 Transfer Delay Compensation

Figure 2.8 illustrates typical transmission delays composed of a variable and a constant part. The constant part causes the local CTR
2.4. ACCURACY OF TIMING DISTRIBUTION

Figure 2.8: Transmission delays in a IEEE 1394 bus for 1, 3, 9, and 16 hops.

to lag behind the cycle master’s CTR. Depending on the bus topology, this time offset differs between nodes. The representation time of audio data is subject to this displacement.

There are cases where it is desirable that presentation times have only a limited offset between nodes. For example in professional audio environments, where individual channels of a multichannel signal are played back at different nodes. Samples must then be re-presented with a delay offset of less than 1 µs (defined in [9]). This cannot be guaranteed without compensation of the CSP transfer delay.

The IEEE 1394a standard supplement defines a method to measure the round-trip delay between two nodes with ping packets. A device that requires offset compensation sends a ping packet to the cycle master node. Ping request and response packets are both affected by the variable repeater delays of intermediate nodes.

Measurements are assumed to be done at arbitrary moments that
have no correlation in time. Several pings can thus be viewed as independent random samples. The probability distribution of the measurement error is a superposition of the evenly distributed repeater delays of all intermediate nodes. Figure 2.9 shows the ping error probability density function for a chain of 16 nodes. The accuracy can be improved with multiple measurements and averaging of the results. \( N \) pings will reduce the standard deviation of the probability density function by \( \sqrt{N} \) compared to a single measurement.

![Figure 2.9: Probability density function of ping errors for a chain of 16 hops.](image)

### 2.5 Summary

The characteristics of the IEEE 1394 real-time network is outlined in this chapter. Distribution of a reference time common to all network nodes has been identified as an essential feature. The architecture
of IEEE 1394 has been summarized briefly, with focus on the timing distribution and the IEC 61883 timestamp protocol.

The accuracy of timing distribution has been investigated. The next chapter explains why additional measures are required to generate the desired quality of audio data representation.
Chapter 3

Phase-locked Loops for Sample Clock Recovery

In the previous chapter it has been shown that the accuracy of timing distribution on a IEEE 1394 network is affected by variable transmission latency. This variance causes the cycle time to be inaccurate.

Figure 3.1: The PLL removes jitter form the timing information and generates accurate sample clocks.

This chapter explains why irregularity of presentation instants of audio samples must be below a certain bound in order to maintain the desired audio quality. Phase-locked loops (PLL) are introduced as a method to derive high-quality clocks based on jittered timing information. The most important properties of analog, semi-digital, and all-digital PLLs are summarized.
3.1 Jitter Limits of Audio Clocks

The quality of an audio signal can be characterized in different ways. On one hand standardized measurement methods can be used. These measures can be represented as numbers, for example total harmonic distortion (THD) and dynamic range. The advantage is the fact that the performance of a given audio system is characterized by a few numbers, and comparison between equipment is eased.

On the other hand, the quality of an audio signal can be related to the human perception. Due to the fact that human hearing is complex, noticeable distortions are not always well represented by the measured figures and vice versa.

3.1.1 Distortions caused by Sampling Jitter

Timing variance of the sampling clock becomes relevant when audio signals pass a digital-to-analog converter (DAC) or analog-to-digital converter (ADC). The effects are similar in both cases. Jitter in the sampling clock causes a phase modulation of the signal. To simplify analysis, a sine signal on the ADC input is assumed. If jitter on the sampling clock is considered sinusoidal, the converted signal becomes:

\[ s(t) = A_s \sin(\omega_s(t + A_j \sin(\omega_j t))) \]  

(3.1)

With

- \( A_s \) = Signal amplitude
- \( \omega_s \) = Signal angular frequency
- \( A_j \) = Jitter amplitude
- \( \omega_j \) = Jitter angular frequency

This equation can be rearranged to

\[ s(t) = A_s \left( \sin(\omega_s t) \cos(A_j \omega_s \sin(\omega_j t)) + \cos(\omega_s t) \sin(A_j \omega_s \sin(\omega_j t)) \right) \]
The Bessel functions of the first kind describe the following relationship [16]:
\[
\cos(x \sin y) = J_0(x) + 2 \sum_{k=2i} J_k(x) \cos(ky) \quad i = 1, 2, \ldots
\]
\[
\sin(x \sin y) = 2 \sum_{k=2i-1} J_k(x) \sin(ky) \quad i = 1, 2, \ldots
\]
With
\[
J_0(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!(n+k)!} \left( \frac{x}{2} \right)^{n+2k}
\]
Since \( x = J_\omega s \ll 1 \), the following assumption is reasonable:
\[
J_0(x) = 1
\]
\[
J_1(x) = \frac{x}{2}
\]
\[
J_n(x) = 0 \quad \text{for } n \geq 2
\]
Thus, equation (3.1) can be approximated with
\[
s(t) = A_s \left( \sin(\omega_s t) + \frac{A_j \omega_s}{2} \sin \left( (\omega_s + \omega_j) t \right) - \frac{A_j \omega_s}{2} \sin \left( (\omega_s - \omega_j) t \right) \right)
\]
(3.2)
Sinusoidal jitter adds sidelobes that are separated by the jitter frequency \( \omega_j \) from the mainlobe. An example is shown in figure 3.2. The amplitude of the sidelobes is proportional to the signal frequency \( \omega_s \) and the jitter amplitude \( A_j \), but independent of the jitter frequency \( \omega_j \). This result is in agreement with the simulations and measurements by Harris [10].

Figure 3.3 plots the total harmonic distortion plus noise (THD+N) on the analog output of a DAC in dependence of the sample clock jitter amplitude. The DAC is assumed to be ideal beside the 24-bit quantization. Most commercial audio converters provide THD+N performance between -95 dB and -110 dB with an ideal sample clock (straight line in figure 3.3). According to figure 3.3, sine jitter amplitudes as low as 100 ps decrease the THD+N performance of commercial high-end audio converters.
Figure 3.2: 24-bit, 10 kHz, -10 dBFS signal affected by sine jitter with $A_j = 1$ ns and $\omega_j = 1$ kHz

Figure 3.3: THD+N of a 10 kHz signal with sinusoidal sampling jitter

### 3.1.2 Human Perception of Jitter in Audio Signals

This chapter explores the maximum jitter levels to prevent audible distortions. Dickmann defines in [11] an inaudible jitter profile based
on psychoacoustic principles [12, 13].

The following effects (among others) influence the human perception of audio signals [15]:

- **Absolute threshold of hearing:** Signals with a level below this frequency-dependent threshold cannot be perceived by general listeners.

- **Masking:** A signal becomes inaudible because of another signal with higher intensity. A set of masking curves is depicted in figure 3.4. The vertical lines are the tones (250 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz, 8 kHz) with an acoustic level of 110 dB. Every tone includes a curve that delimits the area where masking takes effect.

![Figure 3.4: Masking curves for 110 dB tones (colored) and the absolute hearing threshold (black). Signal levels below the masking curve cannot be perceived in presence of the corresponding tone.](image)

In [14], Dunn identifies a maximum inaudible sine jitter profile. It is reproduced on the left side of figure 3.5. This figure includes also
Figure 3.5: Limit curves for sinewave jitter amplitudes of Dunn and Dickmann (top) and corresponding effect on sinewave signals according to Dickmann (bottom). The distortions are just below the masking curves.

Dickmann’s profile that produces distortions that just fit underneath the masking curves. The effect of this fitted profile on the 110 dB tones introduced in figure 3.4 is plotted on the right in figure 3.5.
3.1. JITTER LIMITS OF AUDIO CLOCKS

The profile given by Dunn [14] is significantly lower for very low frequencies. The limitation of the jitter amplitude to below 1 \( \mu s \) is not due to psychoacoustic effects, but to prevent degradation of multichannel sound caused by timing offsets, as specified in [9] (see chapter 2.4.2). The analyses made by Dunn and Dickmann do not consider all details related to the masking theory. They must therefore be regarded as estimations.

In general, jitter usually contains components of multiple frequencies. Dickmann therefore estimated a jitter power spectral density (PSD) limit curve in [11] based on the masking theory. It is reproduced in figure 3.6.

![Figure 3.6: Jitter amplitude density limit curve estimated by Dickmann.](image)

In [17], Benjamin and Gannon report results from listening test using headphones to verify the noticeable effects of sampling clock jitter. The results are summarized in table 3.1. These values are significantly higher than those from above derived by psychoacoustic principles. When music was used as test signal, under the test conditions used, RMS jitter levels above 50 ns were acceptable. This inconsistency demonstrates that no generally accepted jitter limit profile is established.
Table 3.1: Hearing thresholds of jitter effects found by Benjamin and Gannon

<table>
<thead>
<tr>
<th>Signal frequency</th>
<th>Jitter frequency</th>
<th>Jitter amplitude at hearing threshold of jitter effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 kHz</td>
<td>2 kHz</td>
<td>40-170 ns</td>
</tr>
<tr>
<td>8 kHz</td>
<td>5 kHz</td>
<td>5-17 ns</td>
</tr>
<tr>
<td>20 kHz</td>
<td>17 kHz</td>
<td>7-13 ns</td>
</tr>
</tbody>
</table>

Audio converter architectures vary in their susceptibility to clock jitter. Measurements which relate clock jitter to theoretical audible artifacts or listening test must specify which converter is used in order to achieve comparable results. In [18], Travis and Lesso propose general methods of characterizing jitter and comment on the jitter susceptibility of various audio converter architectures.

### 3.1.3 Jitter Attenuation

A comparison with the jitter caused by variable transfer delays shown in section 2.4.1 can be made to derive the required jitter attenuation in the receiver. Figure 3.7 shows on top left the time domain representation of the transfer delay caused by 8 hops on the IEEE 1394 serial bus (without constant part of the delay). On the top right, the spectra of this jitter and the jitter tolerance curve from section 3.1.2 are plotted.

The spectrum of the repeater jitter contains a remarkable portion of discrete components. The attenuation must be sufficient to reduce them to below the limit curve. On the bottom left of figure 3.7 the required attenuation is given. Up to 200 Hz, no attenuation is necessary. Above 200 Hz, the jitter reduction must exceed 50 dB. This is in good agreement with the suggested minimum jitter attenuation mask from [30], as shown on the bottom right of figure 3.7.
3.2 Phase-locked Loops

When the timing distribution and the required accuracy of the audio sample clocks are compared, it becomes clear that jitter caused by variable transfer delays must be attenuated before distributed timing information can be used as sampling clock for audio converters. A common method is the phase-locked loop (PLL) technique. Phase-locked loops are control loops that generate an output signal with constant phase relationship to the input signal.

Figure 3.7: Repeater jitter time domain signal for 8 hops (top left), corresponding PSD and inaudible jitter profile (top right), required jitter attenuation (bottom left), jitter attenuation mask from [30] (bottom right)
3.2.1 Analog PLL

Phase-locked loops are originally of analog nature [19, 20, 21]. The operation principle is summarized in this section and the steps towards digital PLLs are described in the subsequent sections. PLLs consist of the three main building blocks shown in figure 3.8.

![Block diagram of an analog phase-locked loop](image)

- **Phase detector (PD):** It compares the phase of the input signal (reference signal) with the phase of the feedback signal. The output of the PD is ideally proportional to the phase difference.

\[ U_d(s) = K_d(\theta_i(s) - \theta_o(s)) = K_d\theta_o(s) \]

Where \( K_d \) is the PD gain in [V/\text{rad}] and \( \theta_i \) and \( \theta_o \) are the phase of the input and output signals respectively.

- **Low-pass filter (LPF):** Filters the output voltage of the phase detector with the transfer function \( F(s) \).

\[ U_c(s) = U_d(s)F(s) \]

- **Voltage-controlled oscillator (VCO):** Translates the filter output into a frequency. Due to the transformation of phase information into a frequency, it has the characteristic of an integrator with gain \( K_0 \) [rad/sV].

\[ \theta_o(s) = \frac{U_c(s)K_0}{s} \]
The transfer function of the closed loop becomes then

\[ H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{KF(s)}{s + KF(s)} \]

with

\[ K = K_d K_0 \]

Figure 3.9: Simple implementation of a first-order loop filter

A first order loop filter as depicted in figure 3.9 has the following transfer function:

\[ F(s) = \frac{1 + s \tau_2}{1 + s(\tau_1 + \tau_2)} \quad \tau_1 = R_1 C \quad \tau_2 = R_2 C \]

and leads to

\[ H(s) = \frac{K\frac{1 + s \tau_2}{\tau_1 + \tau_2}}{s^2 + s \frac{1 + K \tau_2}{\tau_1 + \tau_2} + \frac{K}{\tau_1 + \tau_2}} \]

which is a second-order transfer function with low-pass characteristic. The natural frequency is

\[ \omega_n = \sqrt{\frac{K}{\tau_1 + \tau_2}} \]

and the damping factor becomes

\[ \zeta = \frac{\omega_n}{2} \left( \frac{\tau_2 + \frac{1}{K}}{\tau_1 + \tau_2} \right) \]

A PLL employing a first-order loop filter is therefore a second-order system. Figure 3.10 shows the transfer functions of a second-order loop for various damping factors \( \zeta \).
3.2.2 Semi-digital PLL

Semi-digital phase-locked loops differ from their analog counterparts in that input, output, and feedback signals are digital. The phase-frequency detector (PFD) is a popular replacement for analog phase detectors. It consists of a state machine with three states (‘low’, ‘zero’, ‘high’). As illustrated in figure 3.11, the transitions between the states are triggered by the rising edges of the reference input (‘Ref’) and feedback signal (‘Fb’). If the PLL is locked, the states ‘low’ and ‘high’ are activated only during extremely short time spans. During the phase-locking process the relative time that the state machine remains in ‘low’ or ‘high’ state represents the phase error \[19, 20\]. The PFD stores the order of arrival of the reference and feedback edges. Prior to phase-locking, the frequency of the VCO must be locked. During this phase, the PFD produces an output signal corresponding to the frequency error. Incorrect locking on an integer multiple of the desired VCO frequency is therefore not possible. The outputs
of the PFD are usually combined to control a tristate buffer that remains in high-impedance during state ‘zero’ and outputs logic ‘1’ and ‘0’ in the states ‘high’ and ‘low’ respectively. This signal is then directly passed to an analog loop filter that controls the VCO. The phase detector’s gain $K_d$ depends on the output voltage levels of the tristate buffers [19]:

$$K_d = \frac{U_{\text{high}} - U_{\text{low}}}{4\pi}$$

![Circuit diagram and operation principle of PFD](image)

Figure 3.11: Circuit diagram and operation principle of PFD. It operates asynchronously, but it is tolerant to glitches and metastability.

With the binary clock in digital signals, usage of frequency dividers is simple. In fact, most semi-digital PLLs contain a frequency divider in the feedback path. It allows to generate VCO output frequencies that are an integer multiple of the reference input frequency. The division ratio $N$ has an impact on the overall PLL gain.

$$\theta_{fb}(s) = \frac{\theta_{o}(s)}{N}$$

$$K = \frac{K_dK_0}{N}$$
When the PLL has locked, the output of the PFD is permanently on high impedance, thus no current is flowing to the loop filter. The charge on the loop filter capacitance is kept constant (beside leakage effects) until the PFD output is activated. The loop filter depicted in figure 3.9 works thus as an integrator. The loop filter’s transfer function can then be approximated with

\[
F(s) = \frac{1 + s\tau_2}{s(\tau_1 + \tau_2)}
\]

and second order transfer function

\[
H(s) = \frac{K(1 + s\tau_2)}{s^2 + s\frac{K\tau_2}{\tau_1 + \tau_2} + \frac{K}{\tau_1 + \tau_2}}
\]

with natural frequency

\[
\omega_n = \sqrt{\frac{K}{\tau_1 + \tau_2}}
\]

and damping factor

\[
\zeta = \frac{\omega_n\tau_2}{2}
\]

Figure 3.12: Block diagram of a semi-digital phase-locked loop with digital blocks drawn shaded.

The digital blocks of the semi-digital PLL (PFD and frequency divider) can be integrated in a SoC, close to the network interface and
control blocks. The analog parts (LPF and VCO) are often located off-chip. Then, the one-bit representation of the phase error by the PFD is very advantageous, since only one pin is used to transfer the digital phase information to the analog loop filter. In total two pins are required - one tristate output and one input for the VCO output signal [23].

3.2.3 All-digital PLL

In all-digital PLLs (ADPLL) all analog building blocks are replaced by a digital representation. All relevant PLL signals are binary numbers instead of continuous voltages. Thus, the phase-frequency detector introduced in section 3.2.2 is modified to output a quantized number representing the phase error (see figure 3.13). The rising edges of the reference input signal and the feedback signal are used to start and stop a counter. The clock frequency of the counter $f_c$ is higher than the frequencies of the input signals to provide sufficient resolution. If the feedback signal arrives first, the counter value will be interpreted positive, if the feedback signal is observed first at the PD input, it is interpreted negative. The gain $K_d$ in $\frac{\text{step}}{\text{rad}}$ of the all-digital PD depends on the counting clock $f_c$ and on the reference frequency $f_r$ of the input signal.

$$K_d = \frac{f_c}{2\pi f_r}$$

![Figure 3.13: Digital phase detector](image)

As soon as the phase measurement has completed, i.e. the second of the two corresponding edges has occurred, the obtained phase er-
or is supplied to the loop filter. Digital replacements of analog loop filters can be realized with infinite impulse response (IIR) filters. The coefficients from the continuous-time s-domain can be transformed to the sampled-time z-domain [24]. Alternative digital loop filter implementations can employ an up/down counter that accumulates the measured phase error values. Since the phase error is available as a binary number, more sophisticated algorithms can be used to replace classical loop filters [25]. These algorithms can be implemented either in hardware or software.

In order to obtain a fully digital PLL, the voltage-controlled oscillator is replaced by a digital circuit. It generates a discrete frequency with gain $K_0$ in \( \frac{\sin \theta}{\text{step}} \), where ‘step’ is the minimal step width of the binary filter output.

The implementation of a digitally-controlled oscillator (DCO) is not as straightforward as for the PD or the loop filter. The next section discusses and compares various clock synthesis methods.

## 3.3 DCO Requirements

### 3.3.1 Frequency Range

An important property of a clock generation circuit is the range of frequencies that it can produce. Independent of whether it is a continuous range or a discrete set of output frequencies, this parameter has an impact on the possible applications. In digital audio two dominant sampling frequency sets have been established in the past: 44.1 kHz and 48 kHz and multiples thereof. Others exist, but have no relevance in consumer electronics.

However, providing the sampling frequency is not sufficient. Most of the commercially available audio converters use oversampling techniques and require a master clock that is a multiple of the sampling frequency \( f_s \). Typically it is in the range of \( 128 \times f_s \) to \( 512 \times f_s \). Also, the most popular digital audio interfaces - I²S and AES/SPDIF - require the same multiples for the interface operation [26, 27].
### 3.3. DCO REQUIREMENTS

<table>
<thead>
<tr>
<th>Sampling rate $f_s$</th>
<th>P'8 bit clock rate $64 \times f_s$</th>
<th>SPDIF data toggle rate $128 \times f_s$</th>
<th>Oversampling clock $256 \times f_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.1 kHz</td>
<td>2.822 MHz</td>
<td>5.641 MHz</td>
<td>11.289 MHz</td>
</tr>
<tr>
<td>88.2 kHz</td>
<td>5.644 MHz</td>
<td>11.289 MHz</td>
<td>22.579 MHz</td>
</tr>
<tr>
<td>176.4 kHz</td>
<td>11.289 MHz</td>
<td>22.579 MHz</td>
<td>45.158 MHz</td>
</tr>
<tr>
<td>48 kHz</td>
<td>3.072 MHz</td>
<td>6.144 MHz</td>
<td>12.288 MHz</td>
</tr>
<tr>
<td>96 kHz</td>
<td>6.144 MHz</td>
<td>12.288 MHz</td>
<td>24.576 MHz</td>
</tr>
<tr>
<td>192 kHz</td>
<td>12.288 MHz</td>
<td>24.576 MHz</td>
<td>49.152 MHz</td>
</tr>
</tbody>
</table>

Table 3.2: The most common audio clock frequencies. Oversampling clock rates of $384 \times f_s$ and $512 \times f_s$ also exist.

Table 3.2 summarizes the required clock rates for various oversampling factors and interface standards. Consumer electronics devices are usually required to comply with various sample rates and interface standards. The output frequency range of the DCO must cover oversampling frequencies of both 44.1 kHz and 48 kHz. Common frequencies are 11.289 MHz and 12.288 MHz or 22.579 MHz and 24.576 MHz. A minimum pull range of approximately 8.5% is required to cover both sample rates of each set.

#### 3.3.2 Frequency Resolution

Any quantized system is limited to a certain resolution. The set of possible output frequencies of a DCO cannot exceed the number of possible input control words. As mentioned above, an all-digital PLL is a sampled system in which phase updates occur only in periodic intervals. Between phase updates the content of the loop filter and the frequency setting of the DCO remains constant. Figure 3.14 illustrates waveforms of an ADPLL in locked state, where the desired DCO frequency is $N$ times the reference frequency $(N \times f_{ref})$. The DCO's frequency quantization error accumulates between phase updates to a phase error. The frequency step $f_{step}$ is assumed to be constant over the full output frequency range from $f_{min}$ to $f_{max}$. The peak-to-peak amplitude of the accumulated phase error $\varphi_{err}$ is
\[ \varphi_{err} = \frac{f_{step}}{f_{center}} T_{ref} = \frac{f_{step}}{f_{center} f_{ref}} \] (3.3)

With the DCO center frequency

\[ f_{center} = \frac{f_{max} + f_{min}}{2} \]

Figure 3.14: Effect of limited frequency resolution. The frequency toggles between adjacent frequency steps, causing a periodic jitter.

This phase deviation on the PLL output is equivalent to jitter. Unlike the jitter contributions from the input, this distortion is introduced after the low pass filter. It cannot be removed from the clock signal and affects the sample clock and its multiples directly. The waveform of the phase error is a periodic triangle. In order to determine the required frequency resolution, this phase accumulation can be simplified as a sinusoidal jitter contribution. Recalling the timestamp synchronization method from section 2.3.1, with for instance SYST_INTERVAL = 8 and sampling rate of 44.1 kHz, the ADPLL update frequency is

\[ f_{ref} = \frac{44.1 \text{ kHz}}{\text{SYST\_INTERVAL}} = 5.51 \text{ kHz} \]
3.3.3 Intrinsic Jitter

Intrinsic jitter is generated after the loop filter within the DCO or VCO. Low-frequency fluctuations can be followed by the control loop and do not affect the sample clock. Higher frequency jitter is neither filtered by the low pass filter nor followed by the control loop, and therefore transferred to the output without attenuation. Figure 3.15 shows the error transfer function from the DCO to the output of a second-order PLL with damping factor $\zeta = 0.7$. For jitter components with higher frequency than $\omega_n$ no attenuation takes effect. Consequently, the intrinsic jitter above $\omega_n$ of a VCO or DCO must satisfy the jitter requirements of sampling clocks stated in section 3.1.2.

3.4 Review of Clock Generation Circuits

This section reviews the most important features of conventional clock generators. As highlighted in the introduction, the focus of this thesis is on digital clock generation methods. Analog clock synthesizers are thus only mentioned, and not explained in detail.

3.4.1 Voltage-controlled Oscillators

Voltage-controlled oscillators are analog circuits that exist in various architectures. The frequency is controlled with a continuous voltage. Pull-range and jitter performance are strongly related and can vary in a broad range.

Voltage-controlled crystal oscillators (VCXO) use a quartz crystal as a resonator. VCXOs provide excellent jitter performance. However,
the frequency pull range is very limited (a few 100 ppm) and cannot serve for both 44.1 kHz and 48 kHz multiples. Professional audio equipment uses often two VCXOs, one for the 44.1 kHz multiples and one for the 48 kHz multiples, with some logic for determining which VCXO should be used.

### 3.4.2 Voltage-controlled Ring Oscillators

Ring oscillators consist of an odd number of inverting stages connected to a loop. By varying the delay of the stages, the frequency of the oscillation can be controlled. In classical ring oscillators, the delay is varied by changing the supply voltage of the stages, as illustrated in figure 3.16. The absolute frequency stability and jitter performance of ring oscillators are both limited. Jitter of ring oscillators has been discussed in various publications [28].
When employed in a PLL, the absolute frequency stability is guaranteed by the control loop mechanism. Ring oscillators provide very high pull ranges of up to a factor of two.

![Figure 3.16: A simple ring oscillator](image)

### 3.4.3 All-Digital Ring Oscillators

In contrast to their voltage-controlled counterparts, digitally controlled ring oscillators can be built of digital standard cells. Frequency control is obtained by changing the delay in the inverter loop. The operation principle is depicted in figure 3.17.

![Figure 3.17: All-digital ring oscillator](image)

The output period of the ring oscillator consists of the propagation delays through the ring of the positive and negative edge resulting in high and low output phases. If the delay \( d_{\text{ring}} \) is assumed symmetric for both transitions, the frequency of the ring oscillator is
This relationship is illustrated in figure 3.18. The slope of the curve increases towards small ring delays. Even for very large rings with an oscillation frequency of 25 MHz, the slope at 25 MHz is approximately 1.25 \( \frac{kHz}{ps} \) or 50 \( \frac{ppm}{ps} \). In section 3.3.2 it has been shown that audio sample clock recovery requires a frequency resolution of 1 ppm. Applied to a ring oscillator, this would require a delay resolution in the ring of 0.02 ps at 25 MHz. Digital delay generation with such a high accuracy is not realistic. In chapter 5, a method for improving the frequency resolution of all-digital ring oscillators is proposed.

Figure 3.18: Ring oscillator frequency vs. ring delay
3.4.4 Direct Digital Synthesis

Direct digital synthesis (DDS) is a waveform generation method that calculates the amplitude of the actual output signal and converts it to the analog domain [16]. The key element is a phase accumulator that permanently integrates the frequency control word (FCW) (see figure 3.19). Some of the accumulator’s most-significant bits address a ROM, that translates the phase information to a sine amplitude value. The DAC, followed by a reconstruction filter, generates a sinewave.

The frequency control word defines the amount of phase that is incremented per cycle and determines the output frequency. In general the DDS output frequency is

\[ f_{out} = \frac{f_0 \times FCW}{2^n} \]

With

- \( f_0 \) = fixed system clock frequency
- \( FCW \) = n-bit frequency control word
- \( n \) = accumulator width

Figure 3.19: Basic DDS arrangement and example waveforms.
Chapter 3. Sample Clock Recovery

The maximum frequency of a DDS is limited by the Nyquist frequency $f_0/2$. The frequency control word FCW is represented with $n$ bits, thus $2^{n-1}$ different frequencies can be generated between 0 and $f_0/2$. The frequency resolution is

$$f_{step} = \frac{f_0}{2} \times \frac{1}{2^{n-1}} = \frac{f_0}{2^n}$$

One of the most important features of DDS is that the frequency resolution can be increased by choosing an arbitrarily large $n$. The accumulator content is truncated to $m$ bits before it is applied to the sine lookup ROM. This truncation adds a periodic phase error which shows up as spurious signals on the output. The DAC has only $k$ bits resolution, with $k < m$ typically. This quantization limits the SNR and generates also spurious tones. DDS designs with spurious levels below $-60$ dB and frequency resolution better than 0.0005 ppm have been reported [29].

3.4.5 Synchronous Clock Generation

The simplest way of implementing a DCO is to use a constant clock $f_0$ and a programmable $n$-bit divider. Such a device is able to generate $2^n - 1$ frequencies with the following relationship:

$$f_{out} = \frac{f_0}{W} \quad \text{with} \quad W = \text{division ratio}$$

As shown in figure 3.20, the frequency resolution becomes very poor towards lower division ratios. Frequency steps become finer for large $W$, but to produce sufficiently high output rates (e.g., 12 MHz) with acceptable resolution (1 ppm), $f_0$ is required to be in the 10 THz range.

Another approach is fractional clock division. The basic idea is to use a sequence of 1/$n$ and 1/($n + 1$) divisions to obtain an average output frequency in between. The exact frequency is determined by
3.4. REVIEW OF CLOCK GENERATION CIRCUITS

Figure 3.20: The frequency resolution of a programmable divider improves with the division ratio. The required input clock $f_0$ becomes extremely high for the aimed 1 ppm resolution.

The relative occurrence of the two division ratios, which can be implemented with arbitrarily high resolution. The output period is a multiple of $1/f_0$, but varies periodically by one $f_0$ period. This variation is timing jitter and $f_0$ must be chosen to fulfill the requirements specified in figure 3.5.

The periodicity of the cycle variation depends on the desired division ratio. According to figure 3.5, the jitter amplitude is limited to well below 100 ps when located at the frequency with the highest jitter sensitivity. This would require $f_0$ to be larger than 10 GHz, which is not realistic for the discussed target applications.
3.4.6 Summary and Discussion

The need for sample clock recovery has been explained and various phase-locked loops architectures were classified. By quantifying the non-perceivable jitter levels on audio clocks and the amount of jitter generated by the timing distribution in IEEE 1394, the required jitter attenuation could be determined.

The characteristics of analog, semi-digital, and all-digital phase-locked loop circuits have been summarized. The requirements in relation to frequency range and frequency resolution have been discussed with a special focus on ADPLLs. Existent clock generation methods have been explained and the performance has been compared to the requirements mentioned above. Table 3.3 lists the characteristics of conventional clock generation circuits. The properties unsuitable for the application covered by this thesis are in bold font.

In the next two chapters, two new clock generation methods are proposed. They are intended to combine the advantages of the conventional circuits to fulfill the requirements of all-digital audio sample clock recovery.
### 3.4. REVIEW OF CLOCK GENERATION CIRCUITS

<table>
<thead>
<tr>
<th>Clock generation method</th>
<th>Frequency pull-range</th>
<th>Frequency resolution</th>
<th>Intrinsic Jitter</th>
<th>All-digital Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCO</td>
<td>high</td>
<td>n/a, continuous</td>
<td>low, depends on pull-range</td>
<td>no, analog technique</td>
</tr>
<tr>
<td>VCXO</td>
<td><strong>very low</strong></td>
<td>n/a, continuous</td>
<td>very low</td>
<td>no, analog technique</td>
</tr>
<tr>
<td>Voltage-controlled Ring Osc.</td>
<td>high</td>
<td>n/a, continuous</td>
<td>low</td>
<td>no, analog ring delay control</td>
</tr>
<tr>
<td>All-digital Ring Osc.</td>
<td>very high</td>
<td><strong>low</strong></td>
<td>low</td>
<td>yes</td>
</tr>
<tr>
<td>Direct Digital Synthesis</td>
<td>very high</td>
<td>very high</td>
<td>low</td>
<td>no, DAC and analog filter required</td>
</tr>
<tr>
<td>Integer Clock Division</td>
<td>very high</td>
<td><strong>very low</strong></td>
<td>very low</td>
<td>yes</td>
</tr>
<tr>
<td>Fractional Clock Division</td>
<td>very high</td>
<td>very high</td>
<td><strong>very high</strong></td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 3.3: Characteristics of various clock generation circuits
Chapter 4

Phase Shift DCO

A digitally controlled oscillator (DCO) is proposed in this chapter. It is tailored to the requirements of the applications explained in the previous chapters. The basic idea behind this DCO is to employ the phase accumulation principle of direct digital synthesis (DDS), but to replace the DAC with a dynamically controlled delay line [31, 32].

First, the operation principle is explained with a simplified circuit diagram. The formulae of the output frequency and output period are then adapted to the detailed circuit architecture. The need for calibration is explained and two alternative calibration techniques are proposed. Finally, the precautions in the design flow and the jitter performance of this DCO are discussed.

4.1 Operation Principle

DDS determines the output frequency with the amount of phase accumulated per clock cycle. In the proposed circuit, the phase information is used to delay a fixed input clock with a controlled delay line. This basic idea is illustrated in figure 4.1. A certain phase shift determined by the frequency control word (FCW) is added to the fixed
Figure 4.1: Illustration of the phase shift principle: A permanently increased phase shift applied to a fixed clock \( f_0 \) results in a clock with modified output frequency \( f_{\text{out}} \). The output frequency can be controlled with the shift.

The output frequency \( f_{\text{out}} \) ranges between \( f_0 \) (FCW = 0) and \( f_0/2 \) (FCW = \( 2^n \)). It is obvious, that the phase shift method produces only smaller frequencies than the input clock frequency. The output

\[
T_{\text{out}} = T_0 \left( 1 + \frac{\text{FCW}}{2^n} \right) \quad \text{and} \quad f_{\text{out}} = \frac{f_0}{1 + \frac{\text{FCW}}{2^n}} \quad (4.1)
\]
4.2. Circuit Architecture

Figure 4.2 shows a simplified circuit diagram of the proposed DCO. The phase accumulator is 3 bits wide and the actual FCW setting is 0x3 (hex). A delay line (DL) transforms the calculated phase shift into a delay for the individual clock edges. The 3 bits of the example
accumulator are mapped to 8 possible equally spaced delay shifts $[0, T_0/8, \ldots, 7T_0/8]$. Neither the accumulator register can be endlessly increased, nor the DL can produce an unlimited delay. Therefore, a wrap around mechanism is required. The wrap around is triggered, when the accumulated phase reaches a full input period $T_0$. Then, the accumulator overflows and generates a carry out signal. As shown in figure 4.3, the delay is reduced by $T_0$ and one input period is suppressed by a clock gate. With this method, the periodicity of the output clock edges is perfectly maintained.

The phase calculation is clocked by the input clock $f_0$. Since the wrap around causes one input clock edge to be omitted, there is one cycle with no phase calculation. Thus, the accumulation is stalled for one clock cycle.

### 4.2.1 Delay Line

Figure 4.3 shows that the accumulator wraps when the phase shift exceeds one input period $T_0$. Thus, the required delay range is between 0 and $T_0$. The demanded resolution of the delay line is strongly related to the jitter requirements, since the output clock edges can only be generated with the accuracy of the DL. To stay below the maximum inaudible jitter amplitude (shown in figure 3.5 on page 26) a resolution in the order of the allowed sine jitter amplitude ($< 50$ ps) is required. A detailed jitter analysis will be given in section 4.5.

If the full delay range would be composed of a single type of delay elements with full resolution, several hundred elements would be required. In order to limit the number of elements, but keeping the resolution high, two different types of delay elements are used [33, 34]:

- Coarse delay elements that cover a large delay range.
- Fine delay elements that cover one coarse delay step.

Beside the resolution, the linearity of the delay steps has also an impact on the accuracy of the output edges. Every mismatch in the
mapping of the accumulator’s content to the phase shift will result in periodic cycle jitter. The relationship between non-linearity and jitter will be addressed in section 4.5.

A further characteristic of the delay line is the minimal propagation delay. Ideally, it is zero, whereas in reality the delay ranges from $T_{\text{min}}$ to $T_{\text{min}} + T_0$. While not relevant for the accuracy, $T_{\text{min}}$ limits
Figure 4.4: The most important parameters of a delay line: Delay range, resolution, minimum delay, and non-linearity.

the operation speed of the DCO: No clock pulse may be applied to the DL input before all previous transitions have left the DL. Otherwise, the switching of the delay control inputs concurrently to the propagation of pulses will cause glitches and erroneous delays.

Coarse delay

The most intuitive delay line circuit consists of a series of \(L_{\text{coarse}}\) buffers, as shown on top of figure 4.5. The output signals of the buffers are separated by one buffer delay. They are fed into a large \(L_{\text{coarse}}\)-to-1 multiplexer, that selects one buffer output [35, 36]. The delay step \(T_{\text{coarse}}\) is determined by the propagation delay of one buffer. The minimum delay \(T_{\text{min}}\) is equal the propagation delay through the \(L_{\text{coarse}}\)-to-1 multiplexer. When \(L_{\text{coarse}}\) is a power of two, \(L_{\text{coarse}}\)-1 2-to-1 multiplexers are required to compose a \(L_{\text{coarse}}\)-to-1 multiplexer tree. The path from the inputs to the output passes through \(\log_2(L_{\text{coarse}})\) 2-to-1 multiplexers with delay \(T_{\text{2to1}}\). In [35], this method is used for both coarse- and fine-grained delay generation.

All paths through the multiplexer must have identical delays to ensure regular delay steps. This is very hard to fulfill, especially when the amount of delay steps \(L_{\text{coarse}}\) is large. Consider the 2-to-1 multi-
plexer tree depicted at the bottom of figure 4.5. Every 2-to-1 multiplexer has two inputs $a$ and $b$ and one output $z$. Multiplexer standard cells are typically not optimized for symmetric delays from $a$ to $z$ and $b$ to $z$. When the blue path is selected, the multiplexer delay consists of three $a$-to-$z$ delays, whereas the red path is built of three $b$-to-$z$ delays. Thus, the difference between $a$-to-$z$ and $b$-to-$z$ delays sums up to unequal multiplexer paths. The demand for balanced multiplexers is a serious limitation of this delay generation method.

An alternative coarse delay line is shown in figure 4.6. Instead of using one large multiplexer, every buffer is combined with a 2-to-1
selector, arranged in a chain [37, 38]. One multiplexer selects a buffer output whereas all other multiplexers select the output of the previous multiplexer.

\[\text{chain of 2-to-1 multiplexers:} \quad L_{\text{coarse}=8}\]

\[\text{in} \quad \begin{array}{c}
\text{out} \\
\text{delay control inputs}
\end{array}\]

\[\text{in} \quad \begin{array}{c}
\text{out} \\
\text{delay control inputs}
\end{array}\]

\[\text{--- delay control input } n_{\text{coarse}=1} \]
\[\text{--- delay control input } n_{\text{coarse}=6}\]

Figure 4.6: Coarse delay generation with a chain of 2-to-1 multiplexers.

On the bottom of figure 4.6 two example delay paths are marked. For a delay control input specifying one coarse step (marked blue), the total delay is composed of one buffer delay \(T_{\text{buf}}\), one \(a\)-to-\(z\) multiplexer delay, and one \(b\)-to-\(z\) multiplexer delay. The red line illustrates the path for a delay control input of 6. It composes of 6 buffer delays \(T_{\text{buf}}\), one \(a\)-to-\(z\) multiplexer delay, and 6 \(b\)-to-\(z\) multiplexer delays. Generally, all paths contain exactly one \(a\)-to-\(z\) delay, \(n_{\text{coarse}}\) buffer delays \(T_{\text{buf}}\), and \(n_{\text{coarse}}\) \(b\)-to-\(z\) delays, where \(n_{\text{coarse}}\) is the number of activated coarse steps. As an important advantage of this method,
unbalanced multiplexer paths do not affect the regularity of the coarse delay step width, since the number of a-to-z delays is constantly one and only the number of b-to-z delays is changed. The delay step \( T_{\text{coarse}} \) consists of \( T_{\text{buf}} \) plus a multiplexer delay \( T_{b\rightarrow z} \). The minimum delay is one a-to-z multiplexer delay \( T_{a\rightarrow z} \), and therefore significantly smaller than in the coarse delay generation method discussed above. Both types of coarse delay cells are summarized in table 4.1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>( L_{\text{coarse}})-to-1 mux(^1)</th>
<th>2-to-1 mux chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{\text{min}} )</td>
<td>( T_{2\rightarrow 1} \times \log_2(L_{\text{coarse}}) )</td>
<td>( T_{a\rightarrow z} )</td>
</tr>
<tr>
<td>( T_{\text{coarse}} )</td>
<td>( T_{\text{buf}} )</td>
<td>( T_{\text{buf}} + T_{b\rightarrow z} )</td>
</tr>
<tr>
<td>Required gates</td>
<td>( L_{\text{coarse}})-1 buffers, ( L_{\text{coarse}})-1 2-to-1 mux (balanced)</td>
<td>( L_{\text{coarse}})-1 buffers, ( L_{\text{coarse}})-2-to-1 mux</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of two coarse delay elements: The 2-to-1 multiplexer chain offers a significantly smaller minimum delay \( T_{\text{min}} \) and does not require balanced multiplexers.

Fine delay

The fine delay steps determine the resolution of the delay line, which must be less than 50 ps. This is below the minimal gate delay of most ASIC technologies. Two fine delay generation methods are illustrated in figure 4.7. The first one consists of a multiplexer and two identical buffers. One of the buffers is loaded by a small capacitance, while the other is not [39]. This results in a delay difference between the two paths \( T_{\text{fine}} \) that depends on the load and on the drive strength of the buffers, but not on the absolute delays of the buffers. The minimum delay \( T_{\text{min}} \) of this fine-delay line is the sum of \( L_{\text{fine}} \) propagation delays of all unloaded buffers with their corresponding multiplexers.

In order to reduce the minimum delay, the buffers can also be replaced by inverters and the non-inverting multiplexers by inverting multiplexers. Both inverting cells are significantly faster in most

\(^{1}\)Balanced multiplexers assumed: \( T_{2\rightarrow 1} = T_{a\rightarrow z} = T_{b\rightarrow z} \)
Figure 4.7: Fine delay generation methods: The upper two solutions are based on two identical buffers with different loads. The lower one uses the drive strength of a bank of tristate buffers to control the delay.

standard-cell libraries than their non-inverting counterparts. Depending on the technology, replacing the multiplexers and the buffers by NAND gates (center of figure 4.7) can improve the minimum delay further.
4.2. CIRCUIT ARCHITECTURE

The second type consists of a bank of tristate buffers driving the same node, as proposed by Hsu et al. [41]. The number of enabled buffers influences the delay of the cell. Hsu gives a formula for the delay of a cell in dependency of the amount of parallel buffers and the amount of activated drivers. In order to improve the linearity of the delay sequence, some buffers may drive permanently, and the increment of activated buffers may be not constant. In contrast the delay cell explained above, this method provides multiple delay steps per cell.

Delay line operation

It has been mentioned that the available minimal delay $T_{\text{min}}$ has an impact on the maximum achievable operation frequency. In figure 4.8 the contributions to the timing difference between a rising edge on the DL input and the subsequent falling edge on the DL output are shown. As a simplification, the propagation delays of positive and negative transitions are assumed to be identical. The minimal delay $T_{\text{min}}$ (marked with A in figure 4.8) and the required phase shift (B) sum up to the rising edge position at the output. The phase shift requires a range of one full period $T_0$. In order to prevent glitches and non-deterministic behaviour, the delay control input may only change when all transitions have left the DL. Therefore, the pulse width of the input signal must be considered. If the input clock is assumed to have a symmetric duty cycle, the pulse width is $T_0/2$ (C). The timing slack between the falling output transition and the next rising input edge is required to switch the delay control input (D). Obviously, not in every $T_0$ period an input signal to the delay line can be generated: For any phase shift larger than $T_0/2$, the transitions of consecutive pulses would occupy the DL simultaneously and thus conflict. In practice, the minimal delays of the delay elements require a pulse distance at the DL input of $3T_0$. The timing constraints of the DL are addressed in section 4.4.2 in more detail.

$$T_{\text{slack}} = 3T_0 - T_0 - \frac{T_0}{2} - T_{\text{min}} = \frac{3T_0}{2} - T_{\text{min}} \quad (4.2)$$
4.2.2 Accumulator

The length of the DL must cover a full input clock period $T_0$. Thus, it must be ensured that the coarse delay section can generate enough delay under all operating conditions. Process, temperature and voltage (PTV) variations cause propagation delays to vary by more than a factor of two between fastest and slowest conditions. The amount of coarse delay cells required to compensate $T_0$ is therefore not constant and depends on the delay conditions. Consequently, some extra delay elements are implemented. The same holds for the fine delay elements covering one coarse step. The goal of the calibration procedure is to determine the actual values $T_{\text{coarse}}$ and $T_{\text{fine}}$ dependent on the PTV condition. The calibration process is explained in detail in section 4.3.

As shown in figure 4.9, three delay relations hold:

1. The amount of coarse steps that fully fit into $T_0$ is $N_{\text{coarse}}$.
   \[ N_{\text{coarse}} = \text{floor}\left(\frac{T_0}{T_{\text{coarse}}}\right) \quad N_{\text{coarse}} \leq L_{\text{coarse}} \]

2. The number of fine delays that sum up to one coarse step is $N_{\text{fine}}$.
   \[ N_{\text{fine}} = \text{floor}\left(\frac{T_{\text{coarse}}}{T_{\text{fine}}}\right) \quad N_{\text{fine}} \leq L_{\text{fine}} \]
The quantity of fine steps that fill up the gap between the maximum coarse delay and the full period $T_0$ is $N_{\text{rest}}$.

$$N_{\text{rest}} = \text{floor} \left( \frac{T_0 - N_{\text{coarse}} T_{\text{coarse}}}{T_{\text{fine}}} \right) \quad N_{\text{rest}} \leq N_{\text{fine}}$$

Figure 4.9: Coarse, fine and rest delay relations: With $N_{\text{fine}}$, $N_{\text{coarse}}$, and $N_{\text{rest}}$ a delay with the range $T_0$ can be generated with a resolution of $T_{\text{fine}}$.

**Adaptive Modulo Generation**

The number ranges in the arithmetic of the phase accumulation must be adapted to the delay relations. The format for the FCW (shown on top of figure 4.10) conforms to the modified arithmetic. The total phase increment is divided into coarse, fine, and fractional fine step increments. The values in the fields of the FCW shall not be larger than the corresponding amount of delay cells available ($N_{\text{fine}}$, $N_{\text{coarse}}$, and $N_{\text{rest}}$). Thus, the maximum value allowed in the highest K bits shall not exceed $N_{\text{coarse}}$. If the coarse step increment is equal to $N_{\text{coarse}}$, the L-bit field for the fine step increment must not be set to a value larger than $N_{\text{rest}}$. Otherwise a phase increment corresponding to a delay larger than one input period is specified.

The field named ‘subfine’ of the FCW in figure 4.10 is used for phase increment fractions smaller than a fine step. Since this part of
the accumulator is not directly used for delay control, it is not subject to the adaptive modulo generation. As a first step of the new phase calculation, the lowest M bits of the FCW are added to the lowest M bits of the accumulated value. The most significant bit of the M+1 bit result (carry-out) is connected the carry-in bit is set for the next higher arithmetic block. The lowest M bits are stored as the new
Second, the carry is added with the next higher L-bit field representing the amount of fine step increments and the corresponding accumulated value. Since results higher than \( N_{\text{fine}} \) would produce a fine delay larger than one coarse step, the sum is passed through a modulo-\( N_{\text{fine}} \) operation and a carry bit for the coarse step accumulation is generated.

Last, the coarse part is accumulated, taking the carry bit of the previous stage into account. The results of the coarse and fine accumulation are passed to a logic within the adaptive modulo block, that checks whether the accumulated delay exceeds \( T_0 \). The condition is detected by comparing the coarse sum \( S_{\text{coarse}} \) with the maximum coarse value \( N_{\text{coarse}} \) and the fine sum \( S_{\text{fine}} \) with the rest fine value \( N_{\text{rest}} \). If it is true, the delay steps equivalent to one \( T_0 \) period (\( N_{\text{coarse}} \) and \( N_{\text{rest}} \)) are subtracted from the actual sums \( S_{\text{coarse}} \) and \( S_{\text{fine}} \), and the input of the delay line is delayed by an additional \( T_0 \) clock cycle. The cycle-wrap logic can be described with the following pseudo-code:

```plaintext
if (S_{\text{coarse}} > N_{\text{coarse}})
    OR (S_{\text{coarse}} = N_{\text{coarse}} AND S_{\text{fine}} \geq N_{\text{rest}}) then
        S_{\text{fine}} := S_{\text{fine}} - N_{\text{rest}};
        S_{\text{coarse}} := S_{\text{coarse}} - N_{\text{coarse}};
        Cycle\_wrap := TRUE;
    else
        Cycle\_wrap := FALSE;
end if;
```

**Decoder Blocks**

The decoding process maps the accumulated phase shift values to the delay line control inputs (see figure 4 11). The K-bit coarse field of the ACCU is expanded to \( 2^K \) control signals. Since every delay step has one enable signal, \( 2^K \) must be identical to the coarse delay line length \( L_{\text{coarse}} \). As explained on page 54, only one of the coarse enable signal is activated at a time. The coarse value is therefore decoded into one-hot control input signals.
Figure 4.11: The coarse and fine decoders map the accumulated values to the delay line control inputs.

The L-bit field representing the fine accumulated value is mapped to $2^L$ control lines of the fine DL. The amount of control inputs is equal to the fine DL length ($2^L = L_{\text{fine}}$). The fine decoder does not produce one-hot signals, but enables as many signals as specified by the fine ACCU value. As an example for an fine ACCU value of 3, the first 3 DL control inputs are activated.

**Refining the Frequency Formula**

The equation 4.1 on page 50 needs to be modified to apply to this scheme. The period of the output clock $f_{\text{out}}$ varies from $3 \times T_0$ to $4 \times T_0$, and the accumulator follows the adaptive modulo arithmetic.
4.2. CIRCUIT ARCHITECTURE

Figure 4.12: Detailed timing diagram of the phase accumulation

In order to obtain a formula for the output frequency depending on the FCW, the effective phase increment $T_{inc}$ and the coarse-, fine-, and subfine-fields of the FCW must be related:

$$P_{inc} = \frac{(FCW_{coarse}N_{fine} + FCW_{fine})2^M + FCW_{subfine}}{(N_{coarse}N_{fine} + N_{rest})2^M}$$

with

$$0 \leq P_{inc} \leq 1$$

and

$$T_{inc} = T_0 P_{inc}$$

The output period and output frequency can be written as follows:

$$T_{out} = 3T_0 + T_{inc} = T_0(3 + P_{inc}) \quad \text{and} \quad f_{out} = \frac{f_0}{3 + P_{inc}} \quad (4.3)$$

**Generation of lower output frequencies**

The timing diagram in figure 4.12 and formula 4.3 are derived for an operation, where output frequencies in the range of $\frac{f_0}{3}$ to $\frac{f_0}{4}$ can be produced. However, by adding the corresponding amount of wait cycles in the accumulation sequence, the range can be extended to cover frequencies between $\frac{f_0}{C}$ and $\frac{f_0}{C+1}$ ($C \geq 3$).
4.3 Calibration

The phase accumulation and delay generation addressed in the previous section are based on the values $N_{\text{coarse}}$, $N_{\text{fine}}$, and $N_{\text{rest}}$. These values depend on the delay relations illustrated in figure 4.9 on page 61. The calibration block is responsible for measuring these relations and to provide $N_{\text{coarse}}$, $N_{\text{fine}}$, and $N_{\text{rest}}$ to the DCO. This measurement must be performed before the DCO operation starts. Additionally, the calibration must be repeated in order to track changes of process, temperature, and voltage (PTV) conditions. The interval between calibration updates depends on how fast PTV conditions change. Typically, this is rather slow and a calibration interval in the magnitude of seconds is appropriate.

Two calibration methods are proposed in this section. One uses a delay-locked loop (DLL) to directly compare delay steps. The second employs a ring oscillator and determines the delay relations with frequency measurements. Both methods use delay lines identical to the DL in the DCO.

4.3.1 Calibration with a Delay-locked Loop

DLLs are commonly used to compensate signal delays in high-speed communication circuits. A conventional DLL is depicted in figure 4.13. The basic idea is to match the phase of two signals (B, C in figure 4.13) that are of the same origin (A) but pass through different paths. A commonly used setup employs a non-delayed path and a path with exactly one period delay. A phase detector (PD) determines the difference of the arrival times of the original (B) and delayed (C) signal edges. This information is used to control the variable delay path, such that signal edges arrive concurrently at the PD inputs. In the given example, the delay control block will adjust the variable delay to a full input period. The output of a DLL is typically tapped at a specific position of the DL (for example in the center).

The DLL used for calibration contains a DL identical to the DCO DL in every path (top of figure 4.14). The delay control inputs of
both DLs can be changed in order to examine the relations of the delay steps.

The calibration to obtain $N_{\text{coarse}}$ is performed as a first step. $N_{\text{coarse}}$ is determined by finding the number of coarse steps that must be activated (DL 1) to produce a delay equivalent to one input period $T_0$ (DL 2). DL 2 remains in its minimal setting, but its input is delayed by one $T_0$ period (figure 4.14). The amount of activated coarse elements in DL 1 is consecutively increased with a counter. In the beginning, the PD will report prior arrival of signal A compared to signal B. After every added coarse step in the DL 1, a phase comparison is performed. As soon as the delay increment exceeds $T_0$, the PD will indicate that signal B has arrived first. The measurement is stopped, and $N_{\text{coarse}}$ is determined by decreasing the counter value in the delay control by one step.

The diagram on the bottom of figure 4.14 serves for clarification. The arrival times of signals A and B are marked for every measurement step. In the first step (when no coarse element is activated in DL 1), signal A arrives after the minimal delay $T_{\text{min}}$ and signal B after $T_{\text{min}}$ and the input delay $T_0$. The linear approximation ends when the amount of coarse steps activated in DL 1 cause a delay larger than $T_0$.

In order to determine $N_{\text{fine}}$, the number of fine steps producing a delay equivalent to one coarse delay must be found. Figure 4.15 shows
the DLL in a slightly different configuration. Instead of retarding the input of DL 2 by $T_0$, only one coarse delay is activated in DL 2. DL 1 is set again to minimum delay configuration. Initially, signal B will be observed to be late compared to signal A. The fine delay setting of DL 1 is now increased after every measurement. This procedure is repeated until the PD reports arrival of signal A after signal B. The

Figure 4.14: The DLL configured for coarse step calibration: The coarse steps are increased until the coarse delay exceeds one input clock period.
4.3. CALIBRATION

Figure 4.15: In the fine calibration step, one coarse step is compared to an increasing number of fine steps.

The purpose of the third calibration step is the determination of $N_{rest}$ (see figure 4.16). This is the number of fine steps filling the gap between the largest coarse step and $T_0$. DL 2 is configured identically to the coarse step calibration with a $T_0$ delay. In DL 1 $N_{coarse}$ coarse elements are permanently activated. The amount of fine elements is again found with stepwise fine delay increments.
Phase Detector

Phase detectors are used to measure the phase difference between two signals. In this particular application it is sufficient to identify the order of arrival, since the delay control block handles the linear approximation to the desired delay setting. The most important requirement for a PD in this application is a small uncertainty window when the signals arrive shortly one after another.
In principle, a simple D flip-flop (top left of figure 4.17) is adequate as phase comparator [42]. In the beginning of every calibration iteration, signal B lags behind signal A. The rising edge of signal A will always sample a zero until signal B is leading. Unfortunately, flip-flops in standard cell libraries are typically not optimized for symmetric setup and hold times and the effective sampling time is not exactly defined within this window. This can cause a systematic measurement mismatch.

![Diagrams of phase detector implementation variants](image)

Figure 4.17: Phase detector implementation variants: D-flipflop (top left), cross connected D-flipflops (top center), latch (top right), and NAND-based flipflop (bottom).

On the top center of figure 4.17 a phase detector composed of two flip-flops with cross connected clock and data input signals is shown [43]. The improvement of such a combination is that a result
of "00" or "11" indicates that the sampling has occurred within the setup-hold window of the flip-flops. The uncertainty of the flip-flop's sampling windows remains the same.

A simple latch as shown on the top right in figure 4.17 provides very low setup and hold times [40, 44]. However, when inputs A and B rise simultaneously, oscillations may result. A simple combinational logic on the input prevents this, however at the cost of increased setup and hold times. Another approach is the construction of a flip-flop based on NAND standard-cells [45, 46, 47, 48, 49], as illustrated on the bottom of figure 4.17. This circuit samples input B with the rising edge of signal A. The shaded elements are only used to balance capacitive loads and to equalize delay paths on both inputs.

No phase detector architecture can be recognized as superior under all circumstances. The effective propagation delays and setup and hold times of the available standard-cells must be considered in detail. Some standard-cell libraries contain flip-flops optimized for the sampling of asynchronous signals. These devices offer excellent setup and hold times, and thus may result as the simplest and most accurate phase detectors for this application.

### 4.3.2 Calibration with a Ring Oscillator

As an alternative to the DLL-based calibration method, the DL can be configured as a ring oscillator [35] with an inverter connecting the output to the input of the DL, as illustrated in figure 4.18. A frequency measurement block counts the oscillations of the ring. The fixed clock \( T_0 \) is used as a reference and applied to a second counter. By starting and stopping the counters at the same instant, the counter can be used to compare the frequencies [54].

In a first measurement, the input clock frequency \( f_0 \) and the ring oscillation frequency \( f_{ring} \) with all delay cells disabled are compared. The ratio \( R_{init} \) of the counter values \( P_0 \) and \( P_{ring} \) is equal to the ratio of the frequencies.

\[
\frac{P_0}{P_{ring}} = \frac{f_0}{f_{ring}} = \frac{T_{ring}}{T_0} = R_{init}
\]
4.3. CALIBRATION

To perform the coarse calibration, the corresponding frequency ratio $R_{\text{coarse}}$ for a delay increment of $T_0$ must be calculated.

$$\frac{T_{\text{ring}} + T_0}{T_0} = R_{\text{init}} + 1 = R_{\text{coarse}}$$

The frequency measurement is repeated with increasing coarse delay. The highest amount of active coarse steps with a ratio below $R_{\text{coarse}}$ is the coarse calibration value $N_{\text{coarse}}$. Based on the obtained coarse setting ($N_{\text{coarse}}$), the number of activated fine delays is increased until $R_{\text{coarse}}$ is observed. The resulting amount of fine steps is $N_{\text{rest}}$.

The fine calibration step is performed in a similar way. The ratio $R_{\text{fine}}$ is measured with one activated coarse element. Then, the fine steps are increased (with all coarse elements disabled) until this ratio is again observed. The resulting number of fine steps is the fine calibration value $N_{\text{fine}}$.

$$\frac{T_{\text{ring}} + T_{\text{coarse}}}{T_0} = \frac{T_{\text{ring}} + N_{\text{fine}}T_{\text{fine}}}{T_0} = R_{\text{fine}}$$

Figure 4.18: Calibration Circuit based on Ring Oscillator
**Frequency Measurement Resolution**

The counter lengths must be sufficiently large to obtain the required measurement resolution. For a resolution of \( T_{\text{fine}} / 2 \) at the highest delay setting, \( P_{\text{ring}} \) periods must be accumulated before comparison. \( T_{\text{cal}} \) is approximately the time required for a complete calibration.

\[
P_{\text{ring}} = \frac{T_{\text{ring}} + T_0}{T_{\text{fine}}}
\]

\[
T_{\text{cal}} = P_{\text{ring}}(N_{\text{coarse}} + N_{\text{rest}} + N_{\text{fine}})(T_{\text{ring}} + \frac{T_0}{2})
\]

For example, a ring is assumed to have an initial delay \( T_{\text{ring}} \) of 15 ns and coarse and fine steps of 150 ps and 25 ps, respectively. With an input clock period \( T_0 \) of 5 ns, the resulting duration of the calibration \( T_{\text{cal}} \) is 1.15 ms.

**4.4 Design Methodology Issues**

The operation of the delay line and its characteristics have been explained in the previous sections. The DL itself is an asynchronous circuit that needs special attention to guarantee reliable functioning. In this section, the rules for designing a DL are discussed. In the following subsection, the pipelining of the DCO’s accumulation logic is discussed. It has an effect on the timing of the DL’s control input signals.

**4.4.1 Pipelining the Accumulation**

In section 4.2.2 on page 60 the phase accumulation block has been described and it has been shown that only in every third clock period a new accumulation value is generated. Therefore, three cycles are available for calculation and pipelining can be introduced. Figure 4.19 shows the block diagram of the DCO with pipeline registers drawn shaded. Appropriate locations for the pipeline registers are between...
4.4. DESIGN METHODOLOGY ISSUES

the subfine and fine addition and between the coarse addition and the cycle-wrap logic. Decode registers are inserted between the decoder and the DL.

![Diagram of accumulation block](image)

Figure 4.19: The accumulation block is divided into three pipeline stages and the delay control inputs are registered.

As illustrated in the timing diagram on the bottom of figure 4.19, the decode register and the DL input $f_1$ change simultaneously. As mentioned above, the delay control inputs must have settled before
any transition on the DL input occurs. The next subsection discusses measures and rules to ensure sufficient timing slack for delay control input settling.

4.4.2 Timing Constraints

The main contributions to the total propagation delay of the DL have been illustrated in figure 4.8 on page 60. This section discusses the timing constraints in further detail. For the timing analysis of the DL depicted in figure 4.20, the operation schedule with a nominal input pulse distance of 3 $T_0$ is adopted. A clock gate removes the unused clock pulses. The input clock at node A is assumed to arrive at the clock gate and clock input pins of all flip-flops simultaneously (without skew). The initial level of signal A propagating through the DL is low. Nodes B, C, D, and E are all logically the same and thus also low. According to figure 4.19 on page 75, the input pulse of the DL ($f_i$) and the delay control setting are performed simultaneously (with the same input edge). Since the setting of the multiplexers must be finished before the delay pulse arrives at the multiplexer input, the following condition must be fulfilled:

- The propagation delay from the rising edge at node A through the flip-flops ($t_{pdFl,1}$) and the select inputs of the multiplexers to the multiplexer outputs ($t_{pdMk,1}$) must be shorter than the minimal delay from node A through the delay line input B to the inputs of the same multiplexer ($t_{A\rightarrow B\rightarrow Mk,1}$).

$$t_{pdFl,1} + t_{pdMk,1} < t_{A\rightarrow B\rightarrow Mk,1} \quad \text{for all delay cells } k \quad (4.4)$$

This requirement is hard to fulfill for the first elements, since the propagation delay of an AND gate is significantly smaller than the propagation delays of flip-flops and multiplexers. An additional delay $D_{in}$ must be inserted between clock gate and DL input to ensure this relationship (dotted in figure 4.20).

- The difference between minimum coarse delay ($t_{B\rightarrow C,min,1}$) and maximum coarse delay ($t_{B\rightarrow C,max,1}$) must be at least $T_0$.

$$t_{B\rightarrow C,max,1} - t_{B\rightarrow C,min,1} \geq T_0 \quad (4.5)$$
The difference of minimum fine delay \( (t_{D\rightarrow E,\text{min},1}) \) and maximum fine delay \( (t_{D\rightarrow E,\text{max},1}) \) must be at least one coarse delay step \( (T_{\text{coarse}}) \):

\[
t_{D\rightarrow E,\text{max},1} - t_{D\rightarrow E,\text{min},1} \geq T_{\text{coarse}} \tag{4.6}
\]

The minimum delay through the entire delay line has been simplified in equation 4.2 on page 59. The delay composition must be specified in more detail for the timing analysis. Especially, the propagation delays of rising and falling transitions can differ significantly and must thus be analyzed separately.

- Only the rising edge is relevant for the phase calculation. However, the falling edge leaves the DL last, and is thus also relevant for the timing. The minimum delay of a rising edge \( (t_{A\rightarrow E,\text{min},1}) \) plus the highest possible phase shift \( T_0 \) plus the width of the output pulse at maximum delay shift \( T_0 \) \( (tpw,T_0,\text{out}) \) must be less than the input pulse distance of 3 \( T_0 \):

\[
t_{A\rightarrow E,\text{min},1} + T_0 + tpw,T_0,\text{out} < 3T_0 \tag{4.7}
\]

The pulse width at the output at phase shift \( T_0 \) \( (tpw,T_0,\text{out}) \) is composed of the input pulse width \( (tpw,in) \) plus the propagation delay of the negative edge at phase shift \( T_0 \) \( (t_{A\rightarrow E,T_0,1}) \) minus the propagation delay of the positive edge at phase shift \( T_0 \) 

![Figure 4.20: Detailed Delay Line](image)
(\(t_{A\rightarrow E,T_0,\dagger}\)):

\[ t_{pw,T_0,\text{out}} = t_{pw,in} + t_{A\rightarrow E,T_0,\dagger} - t_{A\rightarrow E,T_0,\dagger} \quad (4.8) \]

Since

\[ t_{A\rightarrow E,T_0,\dagger} = t_{A\rightarrow E,min,\dagger} + T_0 \quad (4.9) \]

equation 4.7 can be rewritten as follows:

\[ t_{pw,in} + t_{A\rightarrow E,T_0,\dagger} < 3T_0 \quad (4.10) \]

Thus, the pulse width at the input and the propagation delay of the negative edge at phase shift \(T_0\) are relevant for the determination of the maximum operation speed. If the left term of equation 4.10 is smaller than 2 \(T_0\), the operation scheme can be changed, so that in every second input cycle a phase increment is performed. In order to allow increased operation speed, the input pulse width \(t_{pw,in}\) can be shortened by means of a clock chopper (figure 4.21).

![Figure 4.21: Clock Chopper](image)

Another aspect that must be considered is the pulse width at the DL output. Equation 4.8 uses the difference of the rising and falling propagation delays to determine the output pulse width. The upper bound for \(t_{A\rightarrow E,T_0,\dagger}\) is given by equation (4.10). A lower bound is also required, since an overly fast propagation of the falling edge would result in a very short output high time \(t_{pw,\text{out}}\). In the extreme case, the negative edge can overhaul the rising edge, causing the pulse to vanish.
• The difference between positive and negative propagation delays must leave enough high time on the DL output.

\[ t_{A \rightarrow E,1} > t_{pu, out, min} - t_{pu, in} + t_{A \rightarrow E,1} \] for all delay settings (4.11)

The difference between positive and negative propagation delays has an additional effect. Whereas the positive delays’ relations are well known from the calibration, the falling edges will typically not have the same relations. Thus, the output pulse width will typically not be equal for all delay line settings, and the negative edge will contain high amount of jitter. If required, a duty-cycle control circuit on the DL output can overcome this handicap and produce well-defined falling edges. It also relaxes the requirement on the length of \( t_{pu, out, min} \).

Figure 4.22 shows a classical duty-cycle correction circuit, that triggers only on the positive edges. A simple toggle flipflop ensures safe 50 % duty-cycle generation with the drawback that it halves the frequency, as illustrated in figure 4.22.

![Diagram of a toggle flipflop](image)

Figure 4.22: A toggle flipflop can be used as duty cycle correction circuit.

The output of the DL is used as clock signal on subsequent blocks, it is thus required to be free of all kinds of glitches. The timing
constraints specified above ensure that no conflicts between multiple
signal transitions occur within the DL.

Another requirement is that the switching of the delay setting itself
does not cause any transitions. Therefore, switching occurs only
when all nodes in the delay path are low. Then, only the select signals
of some multiplexers toggle during the re-configuration. Both multi-
plexer data inputs are constant zero. Clearly, this change is prohibited
to cause any transition on the selectors’ outputs by construction of
the multiplexers.

4.4.3 Choosing the Delay Line Length

The length of the coarse DL $L_{\text{coarse}}$ and fine DL $L_{\text{fine}}$ must be fixed
during the design of the DCO. The number of used coarse elements
$N_{\text{coarse}}$ and fine elements $N_{\text{fine}}$ depends on the operating conditions
and on $T_0$ and is determined during the calibration process. The
length of coarse and fine DLs must be chosen to satisfy the following
relations for all operating conditions:

$$L_{\text{coarse}} \geq N_{\text{coarse}}$$

$$L_{\text{fine}} \geq N_{\text{fine}}$$

The fastest operating conditions and the longest input period $T_0$
must be considered for the determination of $L_{\text{coarse}}$. $L_{\text{coarse}}$ is ob-
tained by dividing $T_0$ by $T_{\text{coarse}}$. $L_{\text{coarse}}$ is obtained by dividing $T_{0,\text{max}}$ by $T_{\text{coarse,\text{min}}}$.

The fine step size $T_{\text{fine,\text{max}}}$ is the maximum allowed fine step.
If the operating conditions in the coarse and fine sections are as-
sumed to be identical, $T_{\text{coarse,\text{max}}}$ and $T_{\text{fine,\text{max}}}$ can be assumed to
have the same ratio as $T_{\text{coarse,\text{min}}}$ and $T_{\text{fine,\text{min}}}$, since the step width
of the fine element (top in figure 4.7) changes proportionally to all
other propagation delays. Therefore, $N_{\text{fine}}$ is theoretically constant
for all operating conditions. $N_{\text{fine}}$ is gained by dividing $T_{\text{coarse,\text{max}}}$
by $T_{\text{fine,\text{max}}}$. $L_{\text{fine}}$ is determined by adding some fine cells to $N_{\text{fine}}$
to obtain sufficiently high safety margin that compensates for delay
4.4. DESIGN METHODOLOGY ISSUES

The total amount of delay elements is minimal when $L_{\text{coarse}}$ and $L_{\text{fine}}$ are equal.

Figure 4.23 shows the number of delay cells in dependency of the coarse step for an example configuration with $T_0 = 10$ ns and $T_{\text{fine,max}} = 20$ ps. It can be seen that the total number of delay cells is minimal when $L_{\text{coarse}}$ and $L_{\text{fine}}$ are equal. The mentioned safety margin of the fine cells is not considered in the figure.

The mapping of the bit fields in the FCW and the accumulator register to the cells is most effective when the number of both delay cells types are a power of two ($L_{\text{coarse}} = 2^K$ and $L_{\text{fine}} = 2^L$). This is not mandatory, but it eases the design of the decoder and the arithmetic.

Additionally, the minimum delay must again be considered. The description of the delay cells illustrated in figures 4.6 and 4.7 indicates that the fastest path through the coarse delay line does not depend on the coarse delay length, whereas the minimal fine path passes through

![Figure 4.23: Required Delay Cells for $T_0 = 10$ ns and $T_{\text{fine}} = 20$ ps. The total amount of delay elements is minimal when $L_{\text{coarse}}$ and $L_{\text{fine}}$ are equal.](image)
all fine elements. Thus, the fine delay line length has a significant impact on the timing described above, whereas the amount of coarse cells has not.

The optimum lengths of coarse and fine delay lines depend on various parameters. $L_{\text{coarse}}$ and $L_{\text{fine}}$ must be determined depending on the specification of every individual DCO implementation.

4.4.4 Placement and Routing

The accuracy of the delay generation depends heavily on the regularity of the delay steps. While the use of identical standard-cells ensures architectural consistency, a non-regular placement of the cells still leads to physical inhomogeneity. Differences of wire lengths and wire loads between the standard-cells cause the propagation delays to vary. Besides the regularity of the layout, the local spread of the delay cells must be considered. A compact placement is necessary in order to minimize the length of the wires and the local variation of PTV.

Beside the regularity within the DL, an identical construction of the operational DL and the calibration DLs is important. The outputs of the two calibration DLs must be locally very close, since they compose the inputs of the phase detector. Figure 4.24 shows an example delay line placement. The upper two illustrations depict an example layout for the coarse and fine delay line introduced in section 4.2.1 on pages 56 and 58, respectively. The capacitive load of the fine delay elements is implemented with an input pin of a standard cell. A possible arrangement of the delay lines in a floorplan is shown on the bottom of figure 4.24. Symmetry and proximity of all three DLs is of highest importance in order to have maximum matching of process and temperature conditions.

4.4.5 Testability Issues

The goal of the physical verification is to ensure the absence of physical faults within the cells or wires of the DCO. The accumulator
Figure 4.24: Example DL Layout and Floorplan: The placement and routing of the delay cells must be compact and strictly regular. Operational DL and calibrations DLs must be proximate.

is a synchronous design, where conventional scan testing can be applied [55, 56, 57] and is not discussed further here. From a pure synchronous design perspective, a delay line has the same functionality as a buffer. It has only one output and a large number of selector inputs, that have no effect on the logic state of the output. During
scan test, the clock rate is lowered significantly and the timing of the DL is irrelevant. A scan test can therefore only find faults that have an impact on the logical state of signals. For more reliable testing that checks also for the correct delay relations, a functional built-in self test (BIST) is proposed in the second part of this section.

**Scan Test of the Delay Line**

A first step in gaining sufficient test coverage is to provide controllability of all inputs. The selector inputs are originating from registers and thus controllable with a scan chain. The DL input however is the clock signal dependent of the clocking during the scan procedure. It is thus very desirable to have the input of the DL controlled by a register. This can be obtained by placing a multiplexer in front of the DL that selects a register output during scan testing, as illustrated in figure 4.25.

Figure 4.25: Enhancements to improve the DL testability: An inverter closes the DL to form a ring oscillator. The effect of the delay settings can be checked with frequency measurements. Observability and controllability are improved with circuits at the input and output of the DL.

In order to gain observability of the DL output, it is fed to a scan flipflop. However, by having multiple paths with the same functionality, a high level of redundancy is introduced. A defective multiplexer
that selects always the same input will not be detected during scan test. In figure 4.25 the nodes, where simple stuck-at-faults can be detected are drawn with solid lines. It is mainly the delay path trough the DL. Nodes with possibly undetectable stuck-at-faults are drawn dotted. Note, that this is a very simplified testability analysis. Since the data outputs of the selector register are also connected to the scan data inputs of other registers, some faults might be detected, and others not.

Test pattern for scan chains are usually generated with automated test pattern generation (ATPG). For this highly regular circuit with its specialized functionality, manual support of pattern generation is required.

BIST for the Delay Line

The test coverage of the delay line can be further improved with a BIST: The output and the input of the DL are connected via an inverter. The frequency of the resulting ring oscillator can be measured by a frequency counter. The counter is started and stopped at predefined instants. The frequency is measured for all delay settings ($L_{\text{coarse}}$ coarse settings and $L_{\text{fine}}$ fine settings). The results are analyzed to lay inside a design tolerance field. This method circumvents the limitations of redundancy. Enhancing the BIST in order to control and observe the logic states of the DL leads to a full coverage of the DL, making the scan test obsolete.

4.5 Jitter Analysis

4.5.1 Period Jitter

Period jitter is the variation of the period $T$ of a clock, as illustrated on top in figure 4.26. In a measurement setup, an oscilloscope triggers on the rising edge of the clock, and the time variation of the next rising edge is observed.
Long-term jitter is an extension of the period jitter. The time span between triggered edge and observed edge is extended to a large number $N$ of clock periods (bottom in figure 4.26).

![Figure 4.26: Two types of clock jitter characterization: Period jitter (top) and long-term jitter (bottom).](image)

The DCO generates a variable output clock based on a fixed input clock. By adding a delay to selected input clock edges, any irregularity of the input clock is passed to the output clock edge. The output jitter of the DCO is thus the sum of the input jitter and the jitter generated within the DCO itself. With the assumption that the input clock comes from a high quality oscillator, its jitter contribution can be neglected.

Every output edge is derived from one input edge and does not depend on any previous states of the DCO. The concept of period jitter is therefore useful to describe the DCO jitter, since it measures the deviation from the ideal period for every edge individually.

### 4.5.2 Jitter due to Delay Line Resolution

The resolution of the DL is a theoretical upper bound for the timing accuracy of the output edges. An ideal digital DL produces delays with the resolution of one fine step width $T_{\text{fine}}$. Thus, the peak period jitter is $T_{\text{fine}}$. The truncation of the accumulator content to the delay line setting can be seen as a quantization process. This leads to an evenly distributed jitter in the interval $[-T_{\text{fine}}/2, T_{\text{fine}}/2]$ with a RMS jitter of $\frac{1}{\sqrt{3}} T_{\text{fine}}$. 
4.5.3 Jitter caused by Non-linearity of the Delay Line

The linearity of the DL depends on the regularity of the delay steps. Coarse and fine elements can be investigated separately. The layout of the standard-cells and of the connecting wires is assumed to be regular. Irregularity is caused by physical differences of the cells due to chip fabrication tolerances. Only the differences between delay steps within the same DL are relevant.

![Figure 4.27: Non-linearity of the DL](image_url)

Only coarse elements are considered to facilitate the analysis. This simplification is admissible, since the coarse steps contribute the main part of the full delay. In this simplified analysis $N_{rest}$ is considered zero, and $N_{coarse}$ coarse steps are assumed to match $T_0$ exactly.

The misalignment of an output edge is equal to the deviation of the generated delay from the linear delay slope, exemplified in figure 4.27. It is the sum of the tolerances of the preceding delay cells. The highest deviations from linearity are expected to occur towards the center of the DL, whereas the delay uncertainty is very low at the maximum and minimum delay settings. If the probability distribution of one coarse step is assumed to be normal, it can be characterized by the expected value $\mu_{coarse}$ and standard deviation $\sigma_{coarse}$:

$$\mu_{coarse} = T_{coarse} = \frac{T_0}{N_{coarse}} \quad \text{and} \quad \sigma_{coarse}$$
A rough estimate of the linearity can be made by analysing the probabilistic delay distribution at the center of the DL. The standard deviation $\sigma_{\text{center}}$ and the expectation $\mu_{\text{center}}$ are:

$$
\mu_{\text{center}} = \mu_{\text{coarse}} \frac{N_{\text{coarse}}}{2} \quad \text{and} \quad \sigma_{\text{center}} = \sigma_{\text{coarse}} \sqrt{\frac{1}{2} \frac{N_{\text{coarse}}}{2}}
$$

With these values the probability that a delay line exceeds a certain deviation from the nominal value $\mu_{\text{center}}$ can be calculated. However, this analysis only holds when all delay steps are independent and normal distributed. These prerequisites are heavily dependent on the technology.

The effect of the DL non-linearity on the period jitter is frequency-dependent. For a small phase increment, where the coarse delay settings of subsequent output edges are adjacent, only the differences between two neighboring steps influence the period jitter. When the phase increment comprises several coarse steps, the sum of all intermediate delay differences contributes to the period jitter. The absolute jitter reflects the jitter components distributed over many cycles. In contrast to the period jitter, the amplitude of the absolute jitter caused by non-linearity is independent of the phase increment.

The consequences of non-linearity depend on the probabilistic distribution of the delay cells. It is difficult to decide in advance whether a DL exceeds the acceptable tolerances or not. To verify the quality of the clock generation, the measurement results of the calibration can be used. The calibrator employing a ring oscillator measures the frequency changes of the delay steps. With this information the DL regularity and its effects can be determined.

### 4.5.4 Jitter caused by Power Supply Noise

The propagation delay of a standard cell depends on the supply voltage. Long-term variances are compensated by the calibration, but noise at higher frequencies is not. The DL is possibly affected by two types of supply noise:
Random noise caused by circuit activity on the device or by external disturbances. The noise is uncorrelated to the DCO operation. The absolute jitter can be estimated when the noise voltage at every cell and the voltage dependency of the delay is known.

Correlated noise has a timing relation to the edges propagating through the DL. This noise is caused by the activity of the DCO itself or any circuit operating with the same clock. The disturbance will occur at deterministic positions within the DL. Delay modulation will always affect the same delay elements. The resulting jitter is similar to the non-linearity jitter explained above.

A number of measures for power supply noise suppression are established, such as usage of separate supply pins, decoupling capacitors, and protections rings.

4.5.5 Jitter due to Calibration Errors

Incorrect calibration values cause errors in the process of edge generation. Two types of misalignments can be distinguished:

- The calibration values $N_{\text{coarse}}$ and $N_{\text{rest}}$ do not correspond to one input period $T_0$. The cycle wrap occurs at the wrong instant and causes a phase jump (figure 4.28).

- The coarse step $T_{\text{coarse}}$ is not exactly compensated by the fine steps $N_{\text{fine}}T_{\text{fine}}$. The misplacement of the edges is equal to the delay difference.

A possible source for calibration errors are measurement errors in the calibrator block, such as an incorrect result of the phase detector. If this error is deterministic and known, it can be compensated by modifying the calibration result.
A second source are deviations between the DL in the operational DCO and the DL in the calibrator. Such a mismatch can be of random or systematic nature. The analysis discussing the non-linearities can be extended for a purely random error. Again, as a valid simplification, only coarse steps are considered. With a standard deviation of coarse steps $\sigma_{\text{coarse}}$ and mean delay $\mu_{\text{coarse}}$, the distribution of the entire DL becomes:

$$\mu_{DL} = \mu_{\text{coarse}} N_{\text{coarse}}, \quad \sigma_{DL} = \sigma_{\text{coarse}} \sqrt{N_{\text{coarse}}}$$

The probability can now be calculated, that the operational DL and the calibration DLs differ by more than a predefined value.
4.5. JITTER ANALYSIS

Systematic mismatches can be caused by asymmetries in the circuit layout, differences between the supply voltages, or differing operating temperatures.

### 4.5.6 Total Jitter

The total jitter is the sum of all jitter contributions, as long as the various effects are independent from each other.

Strict circuit symmetry and proximity reduce the jitter of non-linearity and calibration errors. A second measure is to minimize $T_0$. This has the advantage of a decreased range of the DL, which improves the accuracy of the delay generation. The non-linearity and the calibration mismatches between operational and calibration DLs are reduced.

As a consequence, the propagation time of the edges in the DL is shortened and the impact of supply voltage disturbance is reduced.

Quantization due to the limited fine step resolution and phase jumps during cycle wraps due to incorrect calibration values can be both investigated on a per-cycle basis. The frequencies of occurrence depend on the phase increment and vary over the pull range of the DCO. The non-linearity of the DL manifests itself as a modulation of the output phase. The waveform of this jitter is spread over multiple output periods, whereas the frequency is dependent on the actual phase increment. Amplitude and frequency of jitter contributions caused by noise on the supply voltage are determined by the characteristics of the noise.

The DCO jitter can now be related to the requirements of audio clocks stated in chapter 3. While the bandwidth for audio signals is limited to approximately 20 kHz, the jitter frequencies explained above can be as high as half the output clock rate of up to 50 MHz. The amount of aliasing of high frequency distortions into the audio band depends on the architecture of the DA/AD converter. To satisfy the requirements at all configurations, the RMS jitter at all DCO output frequencies must be below the RMS value of the sine jitter described in figure 3.5 at the worst-case frequency (approximately 35 ps).
4.6 Implementations and Results

To proof the concept, various DCOs were implemented in different ASIC technologies. Due to the continuous improvement during the DCO development process, some of the implementations differ from the previously described architecture in certain details.

The quality of the DCO output clocks is determined with a time-domain jitter measurement. An oscilloscope is triggered on the rising DCO clock edge, and following edges of the same clock are investigated with a time histogram. The measurement setup is described in more detail in appendix A.1.

With the second measurement, the quality of an audio sine signal, D/A-converted with the DCO clock, is measured. A sine generator and a high-quality D/A converter are used in this setup, which is explained in appendix A.2.

4.6.1 Chip I - 0.6 µm CMOS

The goal of this first generation DCO was an experimental proof of concept. Chip I contains two DCOs with different fine delay generation methods [51].

Type A

The phase increment (see figure 4.29) is specified with a 22-bit frequency control word. The 5 most significant bits are used to control 32 coarse delay cells. The 12 least significant bits represent the sub-fine phase increment, while the 5 bits in-between control 32 fine grained delay elements. The accumulation of the fine and subfine parts of the FCW are combined within a single adder.

The pipelining differs from the method described in section 4.4.1 on page 75. The 64 decoded selector signals for the delay elements are unregistered. An additional clock cycle is therefore needed to decode the accumulated phase and to settle the setting of the delay elements. The resulting timing diagram is depicted in figure 4.30. The clock gate passes only every fourth input edge to the DL.
As illustrated in figure 4.29, the calibration circuitry is integrated in the arithmetic block. The main DL is shared between the two Finite State Machines (FSM) controlling the operation and the calibration. Multiplexers on the inputs of the decoders select between the delay settings from the accumulator and the calibrator. Every second input edge is used to verify the calibration values during operation. These calibration edges are removed by an additional clock gate at the output of the main DL. The DCO exhibits two modes (bottom of figure 4.30). Mode 1 performs the calibration in parallel.
with the operation, as described above. In mode 2, the calibration is triggered manually and the DCO operation is stopped for calibration. The output clock rate is doubled, since every input edge is now used for operation.

The second DL serves exclusively for calibration with a coarse section reduced to two elements. A multiplexer between the coarse and fine parts enables a bypass of the operational coarse section (dotted
The delay setting of the calibration DL is constant and no control signals are required with this method. The bypass is activated during the fine calibration step. The input signal from the clock gate is directly applied to the fine section in the upper DL. The edge propagates through one coarse element before entering the fine DL in the lower path. The bypass is disabled during coarse and rest calibration measurements. The phase detector required for the DLL is built of a simple D-flipflop.

<table>
<thead>
<tr>
<th>Technology</th>
<th>0.6 μm CMOS 5V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input clock $f_0$</td>
<td>100 MHz</td>
</tr>
<tr>
<td>Output clock $f_{out}$ range (Mode 1)</td>
<td>11.1 - 12.5 MHz</td>
</tr>
<tr>
<td>Output clock $f_{out}$ range (Mode 2)</td>
<td>20 - 25 MHz</td>
</tr>
<tr>
<td>Frequency resolution</td>
<td>0.5 ppm</td>
</tr>
<tr>
<td>Power consumption type A</td>
<td>164 mW</td>
</tr>
<tr>
<td>Circuit area type A</td>
<td>2.4 kGates, 1.3 mm²</td>
</tr>
<tr>
<td>Period jitter type A</td>
<td>220 ps pk-pk, 30.8 ps RMS</td>
</tr>
<tr>
<td>Delay step fine type A</td>
<td>35 ps mean, 3.1 ps std dev</td>
</tr>
<tr>
<td>Power consumption type B</td>
<td>213 mW</td>
</tr>
<tr>
<td>Circuit area type B</td>
<td>3.2 kGates, 1.8 mm²</td>
</tr>
<tr>
<td>Period jitter type B</td>
<td>293 ps pk-pk, 55.3 ps RMS</td>
</tr>
<tr>
<td>Delay step fine type B</td>
<td>4.3 ps mean, 3.2 ps std dev</td>
</tr>
<tr>
<td>Delay step coarse</td>
<td>540 ps mean, 14 ps std dev</td>
</tr>
<tr>
<td>Fine calibration error</td>
<td>70 ps</td>
</tr>
<tr>
<td>Coarse and rest calib. error</td>
<td>290 ps</td>
</tr>
</tbody>
</table>

Table 4.2: Performance characteristics of Chip I

**Type B**

The second DCO integrated on this chip differs only in the architecture of the fine delay elements. Instead of a multiplexer selecting two buffer outputs, the type B elements employ multiple tristate buffers driving the same node (see figure 4.30). The number of elements is also 32, but every cell offers 4 delay steps. A total of 7 bits from the accumulator are required to connect to the fine DL. The 2 additional
control lines reduce the subfine word to 12 bits. The FCW is also 22 bits wide.

**Time-Domain Jitter Measurement**

The time-domain jitter measurement result of the phase shift DCO is illustrated in figure 4.31. The RMS jitter of the observed DCO clock edge is plotted against the $\Delta T$ delay between the trigger edge and the observed edge (see appendix A.1 for details).

![Figure 4.31: Time-domain jitter of the phase shift DCO on Chip I.](image)

**Audio Signal Measurement**

The spectrum of a calculated sinusoidal audio signal D/A-converted with the DCO clock is shown in figure 4.32. THD+N of the audio signal is $-100.5$ dB for type A and $-99.7$ dB for type B (analog measurement). Due to the limited resolution of the A/D conversion within the audio spectrum analyzer, the FFT plots are very similar. A reference measurement and a detailed description of the measurement conditions are given in appendix A.2.
Figure 4.32: Audio spectrum of the sine converted with clock from phase shift DCO type A (top) and type B (bottom) on Chip I.
Discussion

Table 4.2 summarizes the performance characteristics of the two DCOs. Remarkable differences exist between the type A and type B implementations in terms of power consumption and chip area. The area increase is caused by the type B fine cell architecture that occupies 2.5 times more chip area than the type A fine cell. Additionally, the type B fine elements require 4 control lines, which increases the decoder and control circuit. The larger circuit size and the strong tristate buffers within the type B fine cells are mainly responsible for the higher power consumption.

The type B fine DL has a smaller step width, but the standard deviation is relative to the step width larger than in the fine DL of type A. This irregularity of the fine delay steps leads to non-linearity of the DL, which is the main reason for the significantly increased jitter of type B compared to type A (figure 4.31).

Figure 4.33: Microphotograph of Chip I
4.6. IMPLEMENTATIONS AND RESULTS

The step width and the propagation delay of the type A fine cell can be estimated with the information from the library’s data book. The estimation on the steps of type B fine cells is more difficult. It depends on the slew rates, driving strength, and output capacities of the cells and is influenced by analog artifacts. An analog simulation of the delay element is therefore required. This is in opposition to the digital design flow and needs analog cell models that are not always available from the ASIC vendors. Therefore, the type B architecture was no longer considered in the following integrations.

The measurement of the D/A-converted audio signal demonstrates that the jitter performance of the DCO clocks enables high-quality audio with THD+N values in the range of $-100$ dB.

4.6.2 Chip II - 0.25 $\mu$m CMOS

The second implementation is integrated together with a system-on-chip (SoC) including a SPARC-V8 compatible processor for system control and setup, large RAM blocks for data and program storage, and reconfigurable processors for audio and video stream processing [50]. Two identical DCOs sharing one calibration block are implemented on this chip.

The goal of this implementation was to investigate the effect of a smaller CMOS technology on the performance of the DCO. The DCOs are integrated on this SoC in order to save integration cost. They are not intended to work simultaneously with the other components on the chip. Therefore, no measures are implemented to reduce the effect of power supply noise caused by the processors’ activity. During the measurements, all other components on the chip are switched off.

Several improvements have been made to the architecture of the DCO illustrated in figure 4.34. The FCW is 25 bits wide and the lowest 13 bits specify the subfine phase increment. The following 5 bits represent the fine step increment, while the next 6 bits are used for the coarse step increment. The MSB determines whether the input clock is divided by 3 or 4 to increase the frequency pull range (see figure 4.35). The outputs of the decoders are registered for faster
Figure 4.34: DCO architecture of Chip II
operation. In contrast to the previous implementation, the input of the DL is not directly derived from the input clock, but connected to a flipflop. The flipflop stores the cycle-wrap output of the DCO arithmetic. Before its output enters the DL, a clock chopper shortens the high pulse width to approximately 1 ns. In the fine cell, the buffers are replaced by inverters and the multiplexer has an inverting output. The propagation delay of the inverting cells is significantly lower than the delay of buffers and non-inverting multiplexers. A toggle flipflop produces a symmetric output clock.

The calibrator is completely separated from the DCO. It contains two delay lines identical to the DL in the DCO. The calibration is triggered by setting a control register and completion is signaled by status registers and a processor interrupt signal. The phase detector consists of two cross connected flipflops.
Technology 0.25 μm CMOS 2.5V
Input clock $f_0$ 180 MHz
Output clock $f_{out}$ range 18 - 30 MHz
Duty cycle $f_{out}$ 50%
Frequency resolution 0.06 ppm
Power consumption 21 mW
Circuit area per DCO 3.2 kGates, 0.08 mm$^2$
Circuit area calibrator 1.8 kGates, 0.045 mm$^2$
Period jitter (CPU idle) 189 ps pk-pk, 27.8 ps RMS
Period jitter (CPU running) 389 ps pk-pk, 71 ps RMS
Delay step fine 23.3 ps mean, 3.1 ps std dev
Delay step coarse 274 ps mean, 10.7 ps std dev
Fine calibration error not measured\(^1\)
Coarse and rest calib. error not measured\(^1\)

Table 4.3: Performance characteristics of Chip II

**Time-Domain Jitter Measurement**

The time-domain jitter measurement result of the phase shift DCO in chip II is illustrated in figure 4.36. The RMS jitter of the observed DCO clock edge is plotted against the $\Delta T$ delay between the trigger edge and the observed edge (see appendix A.1 for details).

**Audio Signal Measurement**

The spectrum of a calculated sinusoidal audio signal D/A-converted with the DCO clock is shown in figure 4.37. THD+N of the audio signal is $-103.5$ dB (analog measurement). A reference measurement and a detailed description of the measurement conditions are given in appendix A.2.

**Discussion**

Table 4.3 summarizes the performance of the phase shift DCO implemented on chip II. The period jitter and the quality of the converted

\(^1\)A functional bug prevents reading of the calibration values.
Figure 4.36: Time-domain jitter of the phase shift DCO on Chip II.

Figure 4.37: Audio spectrum of the sine converted with clock from the phase shift DCO on Chip II.
audio signal improved significantly compared to chip I. Three reasons can be identified for this improvement:

- The power supply noise on this chip is reduced compared to chip I.
- The input clock \( f_0 \) is increased to 180 MHz. The DL range (and the delay uncertainty) is thus reduced.
- The resolution of the DL (fine step width) has improved.

Figure 4.38: Microphotograph of Chip II. The DCOs share the chip area with a SoC for multimedia processing.
4.6.3 Chip III - 0.18 \mu m CMOS

In this chip, the phase shift DCO is integrated as a component of a SoC for multimedia CE devices. The DCO in this implementation operates in the originally purposed environment. Other components are active during the measurements.

The architecture of the DCO in this implementation is identical to chip II. The design did profit from the easy portability of the architecture, which required only a matching of the delay elements to the standard-cell library. The operation speed of 100 MHz has been adapted to the available system clock frequency.

In order to reduce the effect of power supply noise caused by other components on the SoC, the power routing of the DCO is separated. Additionally, the DCO is placed in a corner of the chip to allow maximum local distance with the components with high activity.

The relationship between changes of operation conditions and calibration values has been measured with this chip. The results are plotted in figure 4.39. The left plot shows the calibration values to cover \( T_0 \) over various operating conditions. The rest value \( N_{rest} \) is included as a fraction of a coarse step and therefore not shown separately. The amount of fine delays \( N_{fine} \) remains constant over the full range of operating conditions.

The flat curve on the left plot of figure 4.39 represents the case when the calibration is disabled and the values are unchanged. The two rising curves demonstrate the results of the calibration. The upper line represents the values generated by the calibrator, whereas the lower graph constitutes the numbers with manually optimized jitter. It can be clearly seen that the offset between the two curves is approximately constant. This offset is due to an asymmetry of the phase detector, caused by the setup-hold window of the used flipflop. The right-hand figure plots the RMS period jitter at the same operating conditions. As expected, when no calibration is performed, the jitter increases excessively as the operating condition changes from the initial value. The other two curves show that the jitter remains at the same level when calibration is performed. The compensation of the offset halves the jitter from approximately 80 ps to below 40 ps.
Figure 4.39: Calibration results for the following operating conditions (see text for interpretation): 1: 70°C, 1.68V 2: 56°C, 1.71V 3: 42°C, 1.78V 4: 28°C, 1.84V 5: 14°C, 1.89V 6: 0°C, 1.95V

<table>
<thead>
<tr>
<th>Technology</th>
<th>0.18 µm CMOS 1.8V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input clock $f_0$</td>
<td>100 MHz</td>
</tr>
<tr>
<td>Output clock $f_{out}$ range</td>
<td>10 - 16.7 MHz</td>
</tr>
<tr>
<td>Duty cycle $f_{out}$</td>
<td>50 %</td>
</tr>
<tr>
<td>Frequency resolution</td>
<td>0.06 ppm</td>
</tr>
<tr>
<td>Power consumption</td>
<td>18.5 mW</td>
</tr>
<tr>
<td>Circuit area DCO</td>
<td>3.6 kGates, 0.077 mm$^2$</td>
</tr>
<tr>
<td>Circuit area calibrator</td>
<td>2 kGates, 0.043 mm$^2$</td>
</tr>
<tr>
<td>Period jitter (CPU idle)</td>
<td>276 ps pk-pk, 35 ps RMS</td>
</tr>
<tr>
<td>Period jitter (CPU running)</td>
<td>315 ps pk-pk, 39 ps RMS</td>
</tr>
<tr>
<td>Delay step fine</td>
<td>25.6 ps mean, 2.6 ps std dev</td>
</tr>
<tr>
<td>Delay step coarse</td>
<td>253 ps mean, 10.8 ps std dev</td>
</tr>
<tr>
<td>Fine calibration error</td>
<td>86 ps</td>
</tr>
<tr>
<td>Coarse and rest calib. error</td>
<td>256 ps</td>
</tr>
</tbody>
</table>

Table 4.4: Performance characteristics of Chip III

**Time-Domain Jitter Measurement**

The time-domain jitter measurement result of the phase shift DCO in chip III is illustrated in figure 4.40. The RMS jitter of the observed DCO clock edge is plotted against the $\Delta T$ delay between the trigger edge and the observed edge (see appendix A.1 for details).
4.6. IMPLEMENTATIONS AND RESULTS

Figure 4.40: Time-domain jitter of the phase shift DCO on Chip III.

**Audio Signal Measurement**

The spectrum of a calculated sinusoidal audio signal D/A-converted with the DCO clock is shown in figure 4.41. THD+N of the audio signal is $-102.2$ dB (analog measurement). A reference measurement and a detailed description of the measurement conditions are given in appendix A.2.

**Discussion**

Table 4.4 summarizes the performance characteristics of chip III. The period jitter and the quality of the converted audio signal slightly worse compared to chip II. This is mainly caused by two reasons:

- The $f_0$ clock of the DCO serves as system clock for the other components on the SoC. Therefore, activity in the remaining blocks causes power supply noise.

- The $f_0$ clock speed is adapted to the available system clock speed (100 MHz). Compared to chip II (180 MHz), this requires a larger DL range with more uncertainty.
4.6.4 Chip IV - 0.18 μm CMOS

This implementation was required to operate at a higher clock rate to produce output clocks at frequencies up to 50 MHz. A 300 MHz clock drives four implemented DCOs. The architectural changes are mainly motivated to enable the high-speed operation.

The DL has been adapted to this requirement with a reduced minimal delay. The fine DL that produces the main portion of the minimal delay is shortened to a length of 16 elements. The coarse elements are trimmed to provide a step of 200 ps to compensate the reduced fine delay range. The coarse section contains 64 cells to provide sufficient delay range. The FCW is 24 bits wide, the 3 most significant bits determine the frequency range (3 to 10 cycles), the next 6 bits specify the coarse increment, and only 4 bits are used for the fine increment. The subfine increment is represented with 11 bits.
The fine delay cell consists of NAND gates, as illustrated in figure 4.43. This reduces the initial delay and provides balanced propagation delays for rising and falling transitions. Table 4.5 summarizes the expected performance characteristics. Note, that this implementation is still in the design phase and no chip is available for measurements yet.
<table>
<thead>
<tr>
<th>Technology</th>
<th>0.18 μm CMOS 1.8V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input clock $f_0$</td>
<td>300 MHz</td>
</tr>
<tr>
<td>Output clock $f_{out}$ range</td>
<td>15 - 50 MHz</td>
</tr>
<tr>
<td>Duty cycle $f_{out}$</td>
<td>50 %</td>
</tr>
<tr>
<td>Frequency resolution</td>
<td>0.2 ppm</td>
</tr>
<tr>
<td>Circuit area per DCO</td>
<td>4.5 kGates, $\sim$0.064 mm$^2$</td>
</tr>
<tr>
<td>Circuit area calibrator</td>
<td>2.4 kGates, $\sim$0.034 mm$^2$</td>
</tr>
<tr>
<td>Delay step fine</td>
<td>25 ps$^4$</td>
</tr>
<tr>
<td>Delay step coarse</td>
<td>200 ps$^4$</td>
</tr>
</tbody>
</table>

Table 4.5: Performance characteristics of Chip IV

### 4.7 Generation of System Clocks

Four DCOs are contained in chip IV. Two are purposed to generate audio clocks as discussed in the previous chapters, whereas the other two are used to generate system clocks. Network interfaces are often coupled to a clearly defined clock frequency, such as 24.576 MHz for IEEE 1394, 50 MHz for IEEE 802.3 (Ethernet), or 48 MHz in universal serial bus (USB). By using the DCOs to produce the system clocks, special clock generation circuits can be omitted, and the flexibility to change the clock speed during operation is maintained.

### 4.8 Summary

A DCO has been presented in this chapter. Similar to DDS, it accumulates a phase increment to produce a variable output clock. The DAC used within DDS is replaced by a dynamically controlled delay line. The DL consists of coarse and fine delay cells, requiring a calibration to match the delay range to the actual PTV conditions. Two calibration methods are proposed: DLL-based and ring-oscillator-based. The former calibration method has been used for all implementations. Its drawback is the calibration error mainly caused by the mismatch.

$^4$Expected values, no silicon available.
of the phase detector within the DLL. The latter calibration method promises more precision, since frequency measurements can be performed with very high accuracy, provided that the measurement time is sufficiently long. However, no implementation exists to proof the feasibility and accuracy of the ring-oscillator calibration method.

The measurement results of the implemented DCOs show that the jitter performance is sufficient for high-quality audio. Chip III proofs the jitter performance can also be maintained in an activated SoC.

One of the main motivations for a standard-cell based DCO is the portability. The implementations in chip II and III differ only in a few architectural details and the transfer between the technologies serves as a test for the ease of portability. As a result, the design could be transferred with an effort of a few person-days, including the adaption of the DL and the testbenches. This is far below the development effort time of full-custom circuits.
Chapter 5

Enhanced Ring DCO

An alternative DCO is proposed in this chapter. Its operation principle and architecture differ from the phase shift DCO presented in chapter 4. It is based on the all-digital ring oscillator introduced in section 3.4.3 on page 41. The size and complexity of the circuit is heavily reduced compared to the phase shift DCO. It is therefore suitable for highly cost sensitive applications with reduced jitter requirements.

The frequency resolution of conventional all-digital ring oscillators has been identified as insufficient. This proposal enhances the classical approach and improves the frequency resolution.

5.1 Operation Principle

The frequency of the conventional all-digital ring oscillator is trimmed by controlling the propagation delay of the signal in the ring. The delay settings are held constant for every output frequency and the number of frequency steps is identical to the number of available delay steps (denoted as stars in figure 5.1).

The proposed enhancement interpolates the frequency steps to obtain frequencies in-between (small dots in figure 5.1). This is performed by rapidly switching between adjacent frequency steps. The
resulting frequency is proportional to the relative occurrence of the two delay settings. In the example configuration of figure 5.1, 8 delay cells in a ring oscillator allow 9 basic delays between 1.9 ns and 2.4 ns. The small dots indicate the frequencies obtained by alternation of adjacent delay samples.

Figure 5.1: A conventional all-digital ring oscillator generates only output frequencies with constant delay settings (stars). The proposed enhancement switches between adjacent delay settings to additionally obtain interpolated frequencies (dots).

Figure 5.2 shows the block diagram that corresponds to the example. The frequency control word (FCW) is 6 bits wide, whereas the 3 most significant bits control the constant delay setting of 7 cells. A pulse generator translates the 3 least significant bits into a series of pulses to control the remaining cell. The length of this pulse train is determined by the input word width of the pulse generator, in this example the train is 8 pulses long. The amount of high pulses is equal to the value of the input word. The example pulse trains on bottom of figure 5.2 serve for clarification. As an example, if the 3 bit wide input
word of the pulse generation block specifies a 010b (binary). 2 of the 8 slots in the pulse train are set to high. The delay cell attached to the pulse generation block is therefore activated in 2 out of 8 cycles. The pulse width matches the ring oscillator clock period and the setting takes effect during one ring oscillation. The pulse train is repeated permanently.

Figure 5.2: Enhanced Ring Oscillator: The most significant part of the FCW is decoded and applied to delay cells as constant frequency settings. The least significant part is converted into a pulse train that controls a single delay cell.
5.2 Circuit Architecture

5.2.1 Pulse Generation

The pulse generation block requires a counter (cnt in figure 5.3) that delivers the actual position within the pulse train. Every slot in this sequence is assigned to a bit of the FCW. To represent the weighting of the bits within the FCW, the bits are repeated dependent of their position. The MSB of the pulse generation input word FCW(2:0) is included in every second slot. The next bit is repeated in every fourth slot, and the LSB appears only once. Since the largest value that can be represented with the 3 bit field is 111b (7 dec), the eighth slot is not used and remains always low.

Figure 5.3: Pulse Generation: Every slot in the pulse train is reserved for a specific bit of the FCW. The bits are repeated at a rate corresponding to the bit position in the FCW.
A simple logic block maps the counter value to the corresponding bit of the FCW. The multiplexer chain illustrated in figure 5.3 performs the mapping. Every second value of the 3 bit counter \(\text{cnt}\) ends with a zero \((xx0b)\). This occurrence selects therefore FCW(2). Every fourth counter setting will end with \(x01b\) and is thus mapped to FCW(1). The combinations \(011b\) and \(111b\) occur exactly once per counter pass and select FCW(0) and the low bit, respectively. This method can be easily extended for larger word widths.

**Arrangement of the FCW-Bits in the Pulse Train**  
The arrangement within the pulse train of the repetitions of the FCW-bits is important. To illustrate the effect of a sub-optimal arrangement, a second – theoretical – method is debated (method B in figure 5.4).

As an example, one could apply FCW(2) during the first four slots, then FCW(1) for the next two slots, and FCW(0) in slot 6 (method B). The frequency of occurrence of the bits and the average frequency are the same as in the previously described method (method A in figure 5.4). However, the formation of blocks of slots representing the same FCW-bits has an important disadvantage. For a FCW of \(b100\), the first part of the slots will be enabled and the second half will be disabled. Figure 5.4 shows the effects of the two pulse generation methods for this setting. The average frequency lies between the alternated frequencies. The difference between the effective frequency and the required average frequency is a frequency error that causes a phase error accumulation. The phase error amplitude of the proposed method is significantly smaller compared to the alternative method. The faster switching of the delay element inhibits the accumulation of a large phase error. This effect is even more significant for larger word widths. The pulse generation circuit illustrated in figure 5.3 implements enhanced the arrangement of the FCW-bits with method A. The time spread between repetitions of this method ensures minimal phase error accumulation.
### Proposed Method A:

<table>
<thead>
<tr>
<th>cnt</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
<th>000</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulseTrain</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>low</td>
<td>FCW</td>
</tr>
<tr>
<td>slot</td>
<td>(2)</td>
<td>(1)</td>
<td>(1)</td>
<td>(2)</td>
<td>(1)</td>
<td>(2)</td>
<td>(1)</td>
<td>(2)</td>
<td>(2)</td>
</tr>
</tbody>
</table>

Example PulseTrain for FCW(2:0)=100b

<table>
<thead>
<tr>
<th>diff. from avg. period</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
</table>

Phase error

### Suboptimal Method B:

<table>
<thead>
<tr>
<th>cnt</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
<th>000</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulseTrain</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>FCW</td>
<td>low</td>
<td>FCW</td>
</tr>
<tr>
<td>slot</td>
<td>(2)</td>
<td>(1)</td>
<td>(1)</td>
<td>(2)</td>
<td>(1)</td>
<td>(2)</td>
<td>(1)</td>
<td>(2)</td>
<td>(2)</td>
</tr>
</tbody>
</table>

Example PulseTrain for FCW(2:0)=100b

<table>
<thead>
<tr>
<th>diff. from avg. period</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
</table>

Phase error

Figure 5.4: Effect of slot sequence on phase error for an example FCW of b100. The suboptimal method produces a higher phase error amplitude.

#### 5.2.2 Decoder and Delay Line

The decoder in Figure 5.2 is responsible for the generation of the constant delay settings. It enables as many delay elements as specified.
by the most significant part of the FCW.

The delay line in the ring oscillator contains the fine delay cells described in chapter 4. The circuit diagrams are reproduced in figure 5.5.

![Delay Control Diagrams](image)

**Figure 5.5: Fine Delay Cells**

### 5.2.3 Frequency Range

The frequency range is determined by the oscillation frequencies at maximum and minimum delay settings. One period of the ring oscillator includes the propagation of a rising and a falling edge through the ring.

\[
T_{\text{min}} = N_{\text{cells}}(T_{\text{fast},1} + T_{\text{fast},1})
\]

\[
T_{\text{max}} = N_{\text{cells}}(T_{\text{slow},1} + T_{\text{slow},1})
\]

with

- \( N_{\text{cells}} \) = Amount of delay elements in the DL
- \( T_{\text{fast},1} \) = Rising edge propagation delay of disabled cell
- \( T_{\text{fast},1} \) = Falling edge propagation delay of disabled cell
\[ T_{\text{slow},\dagger} = \text{Rising edge propagation delay of enabled cell} \]
\[ T_{\text{slow},\dagger} = \text{Falling edge propagation delay of enabled cell} \]

Figure 5.6: The frequency pull range is related to the minimum and maximum ring delay.

The absolute frequency range is:
\[ f_{\text{range abs}} = \frac{1}{T_{\text{min}}} - \frac{1}{T_{\text{max}}} = f_{\text{max}} - f_{\text{min}} \]

and the relative frequency range:
\[ f_{\text{range rel}} = 1 - \frac{T_{\text{min}}}{T_{\text{max}}} = 1 - \frac{f_{\text{min}}}{f_{\text{max}}} \]

The center frequency is:
\[ f_{\text{center}} = \frac{f_{\text{max}} + f_{\text{min}}}{2} \]
The relative frequency pull range of the ring oscillator is determined by the ratio between $T_{\text{max}}$ and $T_{\text{min}}$, as plotted in figure 5.6. For a large pull range, this ratio must be sufficiently high.

### 5.2.4 Frequency Resolution

The frequency resolution depends on the ratio $\frac{T_{\text{max}}}{T_{\text{min}}}$ and the word width $n$ of the FCW. The ratio between adjacent frequency steps is not constant. The minimum and maximum resolution steps are:

$$
\Delta f_{\text{step, max}} = 2^{-n} \left( \frac{T_{\text{max}}}{T_{\text{min}}} - 1 \right) \quad \text{when} \quad FCW = 0
$$

$$
\Delta f_{\text{step, min}} = 2^{-n} \left( 1 - \frac{T_{\text{min}}}{T_{\text{max}}} \right) \quad \text{when} \quad FCW = 2^n - 1
$$

Figure 5.7: The relative frequency resolution depends on the $\frac{T_{\text{max}}}{T_{\text{min}}}$ ratio and the word width $n$ of the FCW.
Figure 5.7 shows example curves of the relative frequency resolution for various $\frac{T_{\text{max}}}{T_{\text{min}}}$ ratios. The FCW normalized to $2^n$ is assigned to the x-axis and the frequency resolution on the y-axis is normalized to $2^{-n}$.

5.3 Frequency Calibration

The absolute values of $T_{\text{min}}$ and $T_{\text{max}}$ are dependent on the actual PTV conditions. The available frequency range is therefore not exactly predictable. PTV variations can shift the frequency range to a range, where the target output frequency $f_{\text{target}}$ is no more covered.

![Diagram](image)

Figure 5.8: A calibration block shifts the ring oscillator to the correct frequency range at startup.

Frequency calibration performed prior to operation circumvents this problem. Coarse delay elements are therefore added to the ring
5.3. FREQUENCY CALIBRATION

The ring oscillator frequency is compared to the target frequency to find the appropriate coarse delay calibration. The target frequency $f_{\text{target}}$ is a multiple of the reference frequency available at the PLL input. The ring clock is therefore divided before comparison with the reference frequency is performed.

The frequency calibration is optimal when the center frequency $f_{\text{center}}$ of the ring is equal to the target frequency. Therefore, the fine delay setting is kept in center position (half of the elements activated) during the frequency calibration. This ensures maximum tuning range around the target frequency during operation.

Figure 5.9: Before calibration: The desired frequency $f_{\text{target}}$ is not within the available frequency range. After calibration: The range is shifted towards $f_{\text{target}}$ by choosing the appropriate coarse setting (squares).
Several frequency search algorithms can be employed [41, 52, 53]. The linear search method starts from the lowest coarse delay setting and increases the number of activated elements until the frequency comparator detects that the ring frequency is lower than the target frequency (figure 5.9). Linear search is simple and efficient for the limited number of possibilities considered in this circuit, and more sophisticated search algorithms (e.g., binary search) are not required.

5.4 Design Methodology Issues

5.4.1 Clocking

Special attention must be given to the clocking strategy of the flipflops contained within the ring oscillator. As an elemental property of the ring oscillator, signal transitions permanently propagate through the ring. No instant exist, where all transients end to allow a safe change of the delay settings. An individual clocking instant for every delay cell solves this problem, as illustrated in figure 5.10. In the fine delay part, the flipflops providing the selector signal are clocked by the ring signal on the slow delay path. The propagation delay of the flipflop ensures that the select signal changes when the transition has left the multiplexer. The input pin of the flipflop contributes to the capacitive load of the slower path. In the coarse section, the flipflops are triggered by the input from the previous coarse element, since this signal arrives after the signal from the buffer.

The blocks to control the delay cells (operation and calibration) are also clocked by the ring signal. Otherwise, synchronization with an external clock would be required, which is extremely difficult due to the many clocking instants of the selector flipflops. The location where the ring clock is tapped must be chosen carefully to ensure that no timing violations occur when data are transferred from the output flipflops of the delay control blocks to the select flipflops.
5.4. DESIGN METHODOLOGY ISSUES

5.4.2 Placement and Routing

As discussed in chapter 4, inhomogeneous layout of the delay cells leads to non-regular delay steps. The example layout shown on page 83 can also be applied to this ring oscillator DCO.

5.4.3 Testability

Several properties of this DCO affect testability [56, 57].

- Redundancy within the delay line prevents detection of all possible faults.
- The combinational feedback path of the ring must be broken to prevent oscillation during testing.

Figure 5.10: Every delay element derives its own clock for the select flipflop. The clocks for the operation and calibration blocks are also derived from the ring.
• The individual clocking of the flipflops complicates the usage of established synchronous test methods.

Therefore, significant functional modifications during testing are necessary. The test circuitry is drawn red in figure 5.11. In order to break the feedback path, a multiplexer is inserted in the ring. The TestMode signal selects the test input signal TestIn that controls the DL. The inverted ring signal TestOut is used as an output to observe the logic state of the ring. The usage of a test clock (TestClk) common to all flipflop allows conventional scan testing. Therefore, multiplexers that select the test clock and disable the ring clocks during test mode are added.

![Figure 5.11](image_url)

Figure 5.11: The TestMode signal breaks the feedback path and provides all flipflops with a common test clock.

These measures do not improve the testability of the redundant logic. Similar to the test methodology discussed in chapter 4, a built-in functional test checks the individual delay paths. During this test, the TestMode signal is disabled. Measurements of the oscillation frequency at the different delay settings provide information about the functionality of each delay cell.
5.5 Jitter Analysis

The total jitter of the enhanced ring oscillator DCO is composed of jitter found in conventional ring oscillators and jitter effects related to the particularity of this DCO. The following subsections describe the individual contributions.

5.5.1 Jitter in Ring Oscillators

The jitter effects of conventional ring oscillators have been extensively investigated in several publications [28, 58, 59, 60, 61]. McNeill finds in [28] that the jitter performance of a ring oscillator can be characterized by a figure-of-merit. The jitter of the ring oscillator is independent of the number of stages and the oscillation frequency. The figure-of-merit is determined by the properties of the individual delay gates. Various transistor-level circuit parameters are identified by the papers referenced above, but they cannot be influenced, since only standard cells are considered in this work. However, fast switching times and increased power consumption have both been found to improve the jitter performance [61, 59]. Therefore, gates with strong drive strength should be chosen to build low-jitter ring oscillators.

5.5.2 Jitter due to Limited Delay Resolution

High frequency resolution is achieved by rapidly switching one of the delay cells. As has been shown, the obtained output clock is composed of two period lengths. During the activation of the delay cell a longer period results, whereas the disabled delay step contributes a shorter clock period. The difference between the clock periods is jitter. The amplitude of this jitter can be easily identified as the delay step width

\[ T_{\text{step}} = (T_{\text{slow},1} + T_{\text{slow},\uparrow}) - (T_{\text{fast},1} + T_{\text{fast},\downarrow}) \]

The period jitter toggles between the amplitudes \( \left[ -\frac{T_{\text{step}}}{2}, \frac{T_{\text{step}}}{2} \right] \), resulting in a square wave with a RMS jitter of \( \frac{T_{\text{step}}}{2} \).
5.5.3 Jitter caused by Delay Line Non-Linearity

In the phase shift DCO presented in chapter 4, the delay line non-linearity has been identified as a jitter contributor. The output edges of the phase shift DCO are directly arranged by the delay line. In the ring oscillator DCO discussed here, only one delay step is directly involved in dynamic clock period adjustment. All other delay cells are held constant for a given frequency setting. The delay non-linearity has therefore no effect on the jitter. However, irregular delay steps result in a non-linear relationship between frequency setting and output frequency. This imperfection is compensated by the closed-loop control of the PLL. Delay line non-linearity is therefore not crucial for the enhanced ring DCO.

5.5.4 Jitter caused by Power Supply Noise

Long-term variances on the power supply voltage are compensated by the PLL control mechanism, whereas high-frequency noise is not. Power supply noise can be among the most significant jitter contributors. This must be considered during physical layout of the ring oscillator circuitry. The most common measures against power supply noise are separation of supply pins, decoupling capacitors, and protections rings.

5.6 Implementation and Results

5.6.1 Chip II - 0.25 μm CMOS

Two enhanced ring oscillators have been implemented on a chip to verify the operation principle. The block diagram is shown in figure 5.12. It uses a 18-bit frequency control word. The 4 most significant bits are decoded to control 15 delay cells, whereas the 14 least significant bits are used to produce the pulse train for the switched delay element.

The frequency calibration block is not included in this test chip. All fine delay elements are used for operation. The clocking concept
has been modified compared to the method described in section 5.4.1. The select flipflops of the individual delay cells are clocked by the rising edge of the ring oscillation signal, tapped at the slower path of the delay cell. The clock input of the flipflop so contributes to the capacitive load in the slow path.

In order to simplify this prototype implementation, the decode and pulse generation blocks are driven by an external clock, which is independent from the ring oscillation clock. The sampling of the output flipflops of the decode and pulse generation blocks by the delay selector flipflops is subject to timing violations. As a result, an uncertainty of one ring clock period arises. Since the two clock domain frequencies are uncorrelated, the sampling of zeroes and ones corresponds in average with the calculated setting. With the sampling effect, the pulse train is also subject to the uncertainty of one ring clock period. The FCW-bits are not guaranteed to arrive at the

Figure 5.12: Block Diagram of the Enhanced Ring Oscillator implemented on Chip II.
delay cell in the optimal arrangement described in section 5.2.1 on page 116. This causes a higher phase error amplitude compared to the optimal arrangement.

<table>
<thead>
<tr>
<th>Technology</th>
<th>0.25 μm CMOS 2.5V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output center frequency ( f_{\text{center}} )</td>
<td>139 MHz</td>
</tr>
<tr>
<td>Output frequency range ( f_{\text{range}} )</td>
<td>126-152 MHz, 20 %</td>
</tr>
<tr>
<td>Duty cycle ( f_{\text{out}} )</td>
<td>50 %</td>
</tr>
<tr>
<td>Frequency resolution</td>
<td>0.7 ppm</td>
</tr>
<tr>
<td>Clock of decoder and pulse gen.</td>
<td>180 MHz</td>
</tr>
<tr>
<td>Power consumption</td>
<td>6.4 mW</td>
</tr>
<tr>
<td>Circuit area per DCO</td>
<td>0.6 kGates, 0.015 mm²</td>
</tr>
<tr>
<td>Period jitter (CPU idle)</td>
<td>387 ps pk-pk, 64 ps RMS</td>
</tr>
<tr>
<td>Period jitter (CPU running)</td>
<td>433 ps pk-pk, 76 ps RMS</td>
</tr>
<tr>
<td>Delay step fine</td>
<td>34 ps mean, 3.3 ps std dev</td>
</tr>
</tbody>
</table>

Table 5.1: Performance Summary of the Enhanced Ring Oscillator.

**Time-Domain Jitter Measurement**
The time-domain jitter measurement result of the enhanced ring DCO in chip II is illustrated in figure 5.13. The RMS jitter of the observed DCO clock edge is plotted against the \( \Delta T \) delay between the trigger edge and the observed edge (see appendix A.1 for details).

**Audio Signal Measurement**
The spectrum of a calculated sinusoidal audio signal D/A converted with the DCO clock is shown in figure 5.14. THD+N of the audio signal is −81.5 dB (analog measurement). A reference measurement and a detailed description of the measurement conditions are given in appendix A.2.

**Discussion**
The performance summary in table 5.1 shows that the circuit area is approximately five times smaller than the area of the phase shift DCO. Power consumption is thereby reduced by a factor of three.
Figure 5.13: Time-domain jitter of the ring DCO on Chip II.

The time-domain jitter measurement of the enhanced ring oscillator shows a massive increase of RMS jitter compared to the phase shift DCOs, especially when $\Delta T$ exceeds 1 us. In contrast to the phase shift DCO, the enhanced ring oscillator has no stable reference clock providing long-term stability. Thus, the enhanced ring oscillator has a significant amount of low-frequency jitter.

The audio spectrum of the enhanced ring oscillator implemented on chip II shows a wide skirt close to the 10.47 kHz signal. This distortion is caused by the mentioned low-frequency jitter of the ring oscillator. The THD+N value of $-81.5$ dB is approximately 20 dB worse than the measured performance of the phase shift DCOs.

In this measurement setup, the enhanced ring oscillator is operated in an open-loop configuration, i.e. not embedded in a PLL. According to section 3.3.3 on page 39, the PLL jitter transfer function from the DCO output to the PLL output attenuates jitter components below the PLL 'natural frequency' $\omega_n$. Since $\omega_n$ has to be very low (approximately 10 Hz) to fulfill the jitter attenuation requirements illustrated on figure 3.7 on page 29, only components very close to the 10.47 kHz
signal are attenuated. Therefore, the closed-loop configuration does not significantly improve the jitter performance of the enhanced ring oscillator clock.

5.7 Summary

In this chapter, an alternative frequency generation method is presented. In contrast to the phase shift DCO described in chapter 4, the enhanced ring oscillator uses a delay line in the closed ring. The limited frequency resolution inherent to quantized delays in ring oscillators is improved by a frequency-switching method. One of the delay cells is rapidly switched between two adjacent delay settings. The ratio of the occurrences of the two delay settings determines the output frequency.

The presented ring oscillator has a significantly smaller circuit area and complexity compared to the phase shift DCO. Additionally, the
calibration is performed in the ring and no additional hardware is required. Due to the simple arithmetic, design time and design risk are minimized.

These advantages come at the cost of a significantly worse jitter performance compared to the phase shift DCO, which results in a relatively low audio quality.
Chapter 6

Conclusion

The exchange of real-time audio data among CE devices requires data rate synchronization of sender and receiver. The timing information is distributed over the data network with limited accuracy, therefore a clock recovery circuit is needed to attenuate the jitter resulting from the time-scattered arrival of timing packets. In this thesis, the design of a standard-cell based DCO for all-digital PLLs (ADPLL) required for jitter attenuation and audio clock generation is presented.

6.1 Main Achievements

Identification of DCO Requirements

With the determination of the jitter limit for audio clocks and the required jitter attenuation for the clock recovery circuits, the requirements for the ADPLL are identified. The specification for the DCO is derived from the ADPLL requirements in terms of frequency range, frequency resolution, and intrinsic jitter limit.

High Quality Phase Shift DCO

The proposed phase shift DCO architecture employs a controllable delay line (DL) to shift edges of a fixed clock by a pre-calculated delay.
A DLL-based calibration method is introduced to compensate effects on the DL due to process-, temperature-, and voltage-variations (PTV). The circuit and the calibration procedure are discussed. A second calibration method based on a ring oscillator is proposed to achieve a higher accuracy. Furthermore, test- and design-methodology issues are addressed that come up with practical implementations of the phase shift DCO and the DL.

Phase shift DCOs with DLL-based calibrators have successfully been implemented on several ASIC technologies, thereby proving the feasibility of the concept. The excellent portability of the proposed circuits limits the effort to change to a new CMOS technology to two person-days.

The measurement results demonstrate that the jitter performance satisfies the identified requirements. High-quality audio with THD+N values below $-100$ dB are feasible.

**Low-Cost Ring Oscillator DCO**

An alternative DCO architecture based on an all-digital ring oscillator is presented. It has a simpler operation principle and is less complex than the phase shift DCO. The limited frequency resolution inherent to all-digital ring oscillators is overcome by rapidly switching between adjacent frequency steps.

A frequency calibration method to tune the DCO to the desired frequency range is introduced. Test- and design-methodology are discussed.

The ring oscillator DCO has successfully been implemented on an five times smaller circuit area compared to the phase shift DCO.

The measurements show that the ring oscillator DCO as high jitter at low frequencies due to the absence of a stable reference clock. The compact design with the rather modest measured audio quality (THD+N $-81.5$ dB), makes this DCO attractive for lowest-cost applications.
6.2 Outlook

Multimedia equipment will experience even more real-time connectivity in the future. An increasing number of independent audio and video streams will ask for multiple PLLs in every network device. Fab-less design houses certainly will tend to flexible and portable standard-cell based solutions, whereas large semiconductor companies have the additional option to develop dedicated semi-digital PLLs for their own ASIC technologies. Therefore, the future of standard-cell based clock synthesis is closely related to the future structure of the consumer electronics chip industry.

6.3 Future Work

The following topics to extend the work can be proposed.

- Design flow support for the DCOs. The composition of the DL depending on the operation frequency range and the delay characteristics of the delay cells could be generated automatically by appropriate software. The automatic generation of DL timing constraints for static timing analysis tools would further ease the design process.

- The calibration of the phase shift DCO based on a ring oscillator has not been implemented on a ASIC yet. The predicted calibration accuracy increase over the DLL-based calibration has to be proofed with a prototype implementation.
Appendix A

Measurements

A.1 Time-Domain Jitter Measurement

Figure A.1 shows the setup for the time-domain jitter measurements. The DCO is configured to generate a clock frequency of 12.288 MHz, which is a commonly used audio master clock frequency (256x48 kHz). A crystal oscillator serves for reference in order to explore the measurement setup limits. The DCO clock signal is applied to the channel input and the trigger input of a HP 54750A digitizing oscilloscope. The oscilloscope triggers on rising edges of the clock. The time histogram shows the distribution of the clock edge after a variable delay from the trigger. In figure A.2, the standard deviation of the resulting histogram (RMS jitter) is plotted against the delay between triggered edge and observed edge $\Delta T$. 
Figure A.1: Setup of the time-domain jitter measurement.

Figure A.2: Time-domain jitter of all implemented DCOs.
A.2 Audio Signal Measurement

A calculated sinusoidal 24 bit audio signal is D/A-converted with the DCO clock. The 12.288 MHz clock is applied to a Analog Devices AD21065 DSP [62], which calculates the samples of the sine signal. The samples are then passed to a high-quality D/A converter via an I²S interface [26] with a sampling frequency of 48 kHz and a signal amplitude of 0 dBFS. The Analog Devices AD1955 Multi-bit Σ-Δ DAC [63] was chosen as converter, since it offers excellent signal quality. The signal frequency is 10.47 kHz, because the jitter effects more apparent at high signal frequencies (see section 3.1.1). All signals produced by the DSP are derived from the 12.288 MHz clock. A high-quality D/A converter generates the analog signal, which is applied to the Audio Precision System One analyzer.

The A/D conversion prior to the FFT (1024 points) analysis within the audio analyzer has a limited resolution of 16 bit. Therefore, the plotted spectra of the high-quality DCOs (phase shift DCOs) look all very similar to the reference spectrum. However, the THD+N measurement is analog and offers a high accuracy.

![Diagram of audio signal measurement setup](attachment:image.png)

Figure A.3: Setup of the audio measurement.
Figure A.4: Spectrum of the sine (1024 points) converted with the reference clock (THD+N=−106.3 dB).
Bibliography


BIBLIOGRAPHY


Curriculum Vitae

Eric Roth was born in Zürich, Switzerland, on April 14, 1975. After finishing high school at the Kantonsschule Zürich-Oerlikon in 1994, he studied Electrical Engineering at the Swiss Federal Institute of Technology ETH Zürich with emphasis on digital VLSI design. In 2000, he received the Dipl. El.-Ing. ETH (M.Sc.) degree in Electrical Engineering.

The same year, he joined the Integrated Systems Laboratory (IIS) of ETH where he worked as a research assistant in the field of digital VLSI design. In cooperation with the industrial partner BridgeCo AG, Dübendorf, Switzerland, he was involved with analysis and development of cost-effective all-digital audio clock recovery circuits. His research interests include phase-locked loops, clock synthesizers, and VLSI circuits related to the transportation and processing of multimedia data.