

# **Masterarbeit**

**Entwicklung eines FPGA-basierten Time to Digital  
Converters**

**cand. el. Benjamin Blase**

**Karlsruher Institut für Technologie (KIT)  
Institut für Technik der Informationsverarbeitung (ITIV)**  
**Institutsleitung:**

**Prof. Dr.-Ing. Dr. h. c. Jürgen Becker**

**Prof. Dr.-Ing. Eric Sax**

**Prof. Dr. rer. nat. Wilhelm Stork.**

**Betreuer:**

**Dr.-Ing. Michael Dreschmann (ITIV)**

**Dr.-Ing. Siegwart Bogatscher (Bosch)**

**30.10.2015**



## **Zusammenfassung**

Für die Anwendung „Hochautomatisiertes Fahren“ entwickelt die Firma BOSCH verschiedene Sensoren, welche eine zuverlässige Erfassung der Fahrzeugumgebung ermöglichen. Neben Mono- und Stereokameras sowie Ulatraschall- und RADAR-Sensoren wird ein LiDAR-Scanner eingesetzt, welcher ähnlich einer Videokamera den Bildbereich in einzelne Bildpunkte aufteilt, in denen voneinander unabhängige Entfernungsmessungen durchgeführt werden. Die inherenten Vorteile von LiDAR-Sensoren (gute Tiefenauflösung und hohe Reichweite) werden dadurch um eine gute laterale Auflösung ergänzt.

LiDAR-Sensoren bestimmen die Messdistanz durch die präzise Laufzeitmessung eines, von einem Objekt im Bildbereich reflektierten, Laserpulses durch die Laufzeitmessung mittels sogenannter Time-to-Digital-Converter (TDCs). TDCs sind in der Regel als dedizierte integrierte Schaltkreise (ICs) realisiert. Insbesondere für Forschungsanwendungen werden jedoch immer komplexere TDCs auf Basis konventioneller Field Programmable Gate Arrays (FPGAs) entwickelt. Dabei werden die internen Strukturen der FPGAs genutzt um gängige TDC-Messprinzipien nachzubilden.

Für den Einsatz auf einem FPGA-basierten Prototypen des LiDAR-Scanners soll im Rahmen dieser Arbeit ein TDC-Modul entwickelt werden, welches die Laufzeit von LiDAR-Pulsen ohne zusätzliche ICs bestimmen kann. Hierzu werden bestehende Konzepte FPGA-basierter TDCs bezüglich ihrer Eignung für das Projekt evaluiert, sowie ein TDC-Modul für den Einsatz auf einem Xilinx Kintex-7 FPGA konzipiert, funktional implementiert und optimiert. Das fertige TDC-Modul wird mittels Laufzeitmessungen von Testsignalen evaluiert und getestet.

Der entwickelte TDC erreicht einen mittleren Messfehler von 11,72 Picosekunden, was ihn vorne im Feld FPGA-basierter TDCs seiner Klasse einordnet. Dies wird durch die Evaluation und konsequente Berücksichtigung der zu Grunde liegenden Hardwareeigenschaften, sowie durch das Nutzen eines alternativen Kodierungskonzeptes erreicht, dessen bessere Eignung für die Anwendung in einem TDC im Zuge der Arbeit nachgewiesen wird.



## Erklärung

Ich erkläre hiermit, dass ich die vorliegende Diplomarbeit selbstständig und ohne unzulässige fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Karlsruhe, den 30. November 2015

---

Benjamin Blase

# Inhaltsverzeichnis

<b>1 Einleitung und Zielsetzung</b>	<b>7</b>
1.1 Time to Digital Converter (TDCs) . . . . .	7
1.2 Motivation und Zielsetzung der Arbeit . . . . .	8
<b>2 Grundlagen</b>	<b>10</b>
2.1 Field Programmable Gate Arrays . . . . .	10
2.2 Xilinx 7-Series FPGA-Architektur . . . . .	11
2.2.1 Taktarchitektur . . . . .	11
2.2.2 Configurable Logic Block (CLB) . . . . .	13
2.2.3 CARRY4-Element . . . . .	14
2.2.4 Mixed Mode Clock Manager (MMCM) . . . . .	14
2.2.5 Block-RAM . . . . .	16
<b>3 FPGA-basierte TDCs und Messprinzipien</b>	<b>18</b>
3.1 FPGA-basierte Time to Digital Converter . . . . .	18
3.2 Messprinzipien und vorausgehende Arbeiten . . . . .	18
3.2.1 Vernieroszillator . . . . .	19
3.2.2 Tapped Delay Line . . . . .	19
3.2.3 Vernier Delay Line . . . . .	20
3.3 Wahl der Architektur für diese Arbeit . . . . .	20
<b>4 Konzept</b>	<b>22</b>
4.1 Messprinzip: Tapped Delay Line (TDL) . . . . .	22
4.1.1 Messen eines Zeitintervalls . . . . .	23
4.2 Dynamische Kalibrierung der TDL . . . . .	25
4.2.1 Code Density Test . . . . .	26
4.2.2 Erstellen einer Abbildungsvorschrift . . . . .	28
<b>5 Realisierung</b>	<b>33</b>

5.1	Tapped Delay Line . . . . .	33
5.1.1	Zurücksetzen der TDL . . . . .	34
5.1.2	Aufteilung der TDL und Multiphasenabtastung . . . . .	35
5.2	Priority Encoder . . . . .	36
5.2.1	Bubble Errors . . . . .	38
5.2.2	Kodierung mittels Kantenfilter . . . . .	40
5.2.3	Kodierung durch Zählen der Einsen . . . . .	41
5.2.4	Phasensynchronisation der Daten . . . . .	43
5.3	Datenfusion und Kalibrierungsmanagement . . . . .	43
5.3.1	Datenfusion . . . . .	44
5.3.2	Kalibrierungsmanagement . . . . .	44
5.4	Histogramm und Look Up Table . . . . .	45
5.5	Kanalmodul . . . . .	45
5.6	Ringoszillator . . . . .	46
5.7	Toplevelmodul TDC . . . . .	47
5.8	TDC-Wrapper . . . . .	49
5.9	Zusammenfassung . . . . .	50
5.9.1	Interfaces . . . . .	52
<b>6</b>	<b>Test und Ergebnisse</b>	<b>54</b>
6.1	Charakterisierung der Takt- und Ringoszillatorsignale . . . . .	54
6.1.1	Messaufbau . . . . .	54
6.1.2	Messergebnisse . . . . .	55
6.1.3	Statistische Unabhängigkeit des Ringoszillatorsignals zum Systemtakt	57
6.2	Theoretisch ermittelte Qualitätsmerkmale . . . . .	58
6.2.1	Quantisierungsstufen des TDC . . . . .	58
6.2.2	Quantisierungsfehler . . . . .	59
6.2.3	Auflösung . . . . .	61
6.2.4	Dynamikbereich . . . . .	62
6.3	Zeitmessungen . . . . .	62
6.3.1	Messaufbau . . . . .	62
6.3.2	Relativer mittlerer Fehler . . . . .	65
6.3.3	Absolute Genauigkeit . . . . .	66
6.4	Anforderungen an den externen Taktgeber . . . . .	67
6.4.1	Zahlenbeispiel zum Frequenzoffset . . . . .	69
6.5	Einsatz des TDC auf anderen FPGA-Modellen . . . . .	70
6.5.1	Virtex-7 FPGA . . . . .	70

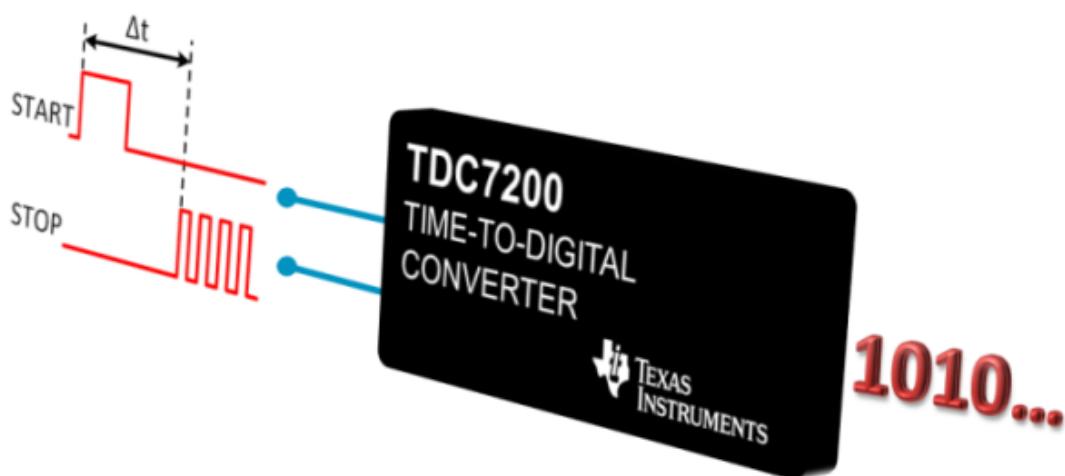
6.5.2 Artix-7 FPGA-Fabric eines Zynq-7000 SoC . . . . .	71
<b>7 Fazit und Ausblick</b>	<b>73</b>
7.1 Zusammenfassung der Arbeit . . . . .	73
7.2 Ausblick und weitere Arbeiten . . . . .	74
<b>Abkürzungen</b>	<b>I</b>
<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Literaturverzeichnis</b>	<b>V</b>

# Kapitel 1

## Einleitung und Zielsetzung

### 1.1 Time to Digital Converter (TDCs)

*Time to Digital Converters* (TDCs) sind elektronische Baugruppen, welche die zwischen einem Start- und einem Stopimpuls vergangene Zeit messen und in digitaler Form ausgeben. TDCs werden unter Anderem bei Experimenten der physikalischen Grundlagenforschung, als Laufzeitmesser für Messimpulse in der optischen oder radarbasierten Abstandsmessung [1], in der Forschung an integrierten Schaltkreisen und zur Signalauswertung in Hochgeschwindigkeitsdatentransferanwendungen eingesetzt [2]. Dafür sind Messpräzisionen von weit unter einer Nanosekunde notwendig. Dies kann von taktsynchronen Binärzählern nicht erreicht werden.



**Abbildung 1.1:** TDC7200 Time to Digital Converter von Texas Instruments mit 55ps Auflösung und einer Standardabweichung von 35ps. (Quelle: [ti.com](http://ti.com))

Time to Digital Converter gibt es in der Regel in der Form von Integrierte Schaltkreisen (ICs), aber auch als Messgeräte oder Einstekkarten für den PC zu kaufen. Im vergangenen Jahrzehnt wurden jedoch, aufgrund der dieser Technologie inherenten Vorteile, FPGA-basierte TDC-Implementierungen vor allem in der Forschung, wie zum Beispiel an der Positronen-Emissions-Tomographie [3] oder beim COMPASS Experiment am Teilchenbeschleuniger des CERN [4], eingesetzt. Diese Vorteile sind, unter Anderem, die kurzen Entwicklungszeiten, die einfache Anpassbarkeit an sich ändernde Anforderungen oder Umstände, kompakte Modulgrößen durch Datenerfassung und -verarbeitung auf dem selben Chip, sowie die Wiederverwendbarkeit der Hardware für andere Zwecke.

Als Qualitätscharakteristiken von Time to Digital Convertern gelten nach Gao [5]:

- Präzision der Messung
- Dynamikbereich: minimal und maximal messbares Zeitintervall
- Nichtlinearität der Quantisierungsstufen
- Messverzögerung: Zeit zwischen Messereignis und digitaler Ausgabe
- Totzeit: minimale Zeit zwischen zwei konsekutiven Messungen

## 1.2 Motivation und Zielsetzung der Arbeit

Für die Anwendung „Hochautomatisiertes Fahren“ entwickelt die Firma BOSCH verschiedene Sensoren, welche eine zuverlässige Erfassung der Fahrzeugumgebung ermöglichen. Neben Mono- und Stereokameras sowie Ultraschall- und RADAR-Sensoren wird ein LiDAR-Scanner eingesetzt, welcher ähnlich einer Videokamera den Bildbereich in einzelne Bildpunkte aufteilt, in denen voneinander unabhängige Entfernungsmessungen durchgeführt werden. Die inherenten Vorteile von LiDAR-Sensoren (gute Tiefenauflösung und hohe Reichweite) werden dadurch um eine gute laterale Auflösung ergänzt.

Für den Einsatz in einem FPGA-basierten funktionalen Prototypen des LiDAR-Scanners soll im Rahmen dieser Arbeit ein TDC-Modul entwickelt werden, welches die Laufzeit des LiDAR-Pulses ohne externen TDC-IC bestimmen kann. Die Zielarchitektur ist ein Kintex-7 Field Programmable Gate Array von Xilinx.

Der zu entwickelnde Time to Digital Converter soll die vergangene Zeit zwischen den steigenden Flanken zweier, 8ns langen, Messimpulsen möglichst präzise erfassen und in einem digitalen Format ausgeben. Er soll in der Lage sein, bis zu drei, rasch aufeinander folgende, Reflexionsimpulse auf der selben Signalleitung zu detektieren und zu messen,

etwa bei der Reflexion an der Heck- und Frontscheibe eines Fahrzeuges, sowie eines weiteren Hindernisses im Laserpulswege. Ein Dynamikbereich der Zeitmessung von 0 bis 2,4µs soll eine theoretische maximale Messdistanz von 360m erlauben. Die Impulsrate voneinander unabhängiger Messungen beträgt etwa 200kHz. Die Verfügbarkeit des Moduls zur Laufzeit soll möglichst ununterbrochen gewährleistet sein.

# Kapitel 2

## Grundlagen

### 2.1 Field Programmable Gate Arrays

Auf einem *Field Programmable Gate Array* (FPGA) lassen sich beliebige taktsynchrone und asynchrone digitale Logikschaltungen realisieren. Er besteht aus einer Matrix aus sogenannten Slices, welche man als *Fabric* des FPGA bezeichnet. Die Logikfunktionen werden durch *Look Up Tables* (LUTs) realisiert, welche in jedem Slice vorhanden sind. Diese asynchronen Elemente weisen einer Kombination von, je nach Hersteller und Modell, bis zu sechs binären Eingangssignalen gemäß einer, vom Benutzer programmierbaren, Wahrheitstabelle einen Ausgangswert zu. Für die taktsynchrone Datenverarbeitung gibt es in jedem Slice Flipflops, in denen Daten gespeichert werden können. Weiterhin enthält ein Slice noch asynchrone Elemente zur schnellen Berechnung von Übertragsoperationen.

In modernen FPGAs gibt es zudem eine Vielzahl an dedizierter Bausteine, wie etwa *Random Access Speicherelemente*, digitale Signalprozessoren, Signalmultiplexer oder A/D bzw. D/A-Wandler. Die Slices, sowie die dedizierten FPGA-Elemente sind über sogenannte *Switch Matrizen* miteinander verbunden. Über diese können beliebige Ein- bzw. Ausgänge der Elemente auf beliebigen Pfaden miteinander verbunden werden.

FPGA-Logik wird über Hardwarebeschreibungssprachen wie VHDL, Verilog oder SystemC auf funktionaler oder Registertransferebene beschrieben. Ein Synthesetool übersetzt den beschreibenden Code in ein Netz aus, auf dem FPGA vorhandenen, Funktionselemente (LUT, Flipflop, ...), die sogenannte *Netlist*. In einem nächsten Schritt, werden die Netlistelemente an konkrete Positionen des FPGA-Fabrics platziert und miteinander verbunden. Dieser Vorgang wird als *Place and Route* (PAR) bezeichnet.

Das Projekt wird auf einem Kintex-7 FPGA realisiert. Im Folgenden wird auf die, für die

Realisierung des Projektes relevanten, Elemente der Architektur von Xilinx 7-Series Field Programmable Gate Arrays eingegangen.

## 2.2 Xilinx 7-Series FPGA-Architektur

Die Realisierung eines TDC-Moduls erfordert die Kenntnis der zugrunde liegenden Hardwarelemente und Signalwege innerhalb des FPGA-Fabrics. Die folgenden Abschnitte bieten eine Übersicht über die hierfür relevanten, sowie in der Datenverarbeitung explizit instanzierten Architekturelemente der Xilinx 7-Series FPGAs.

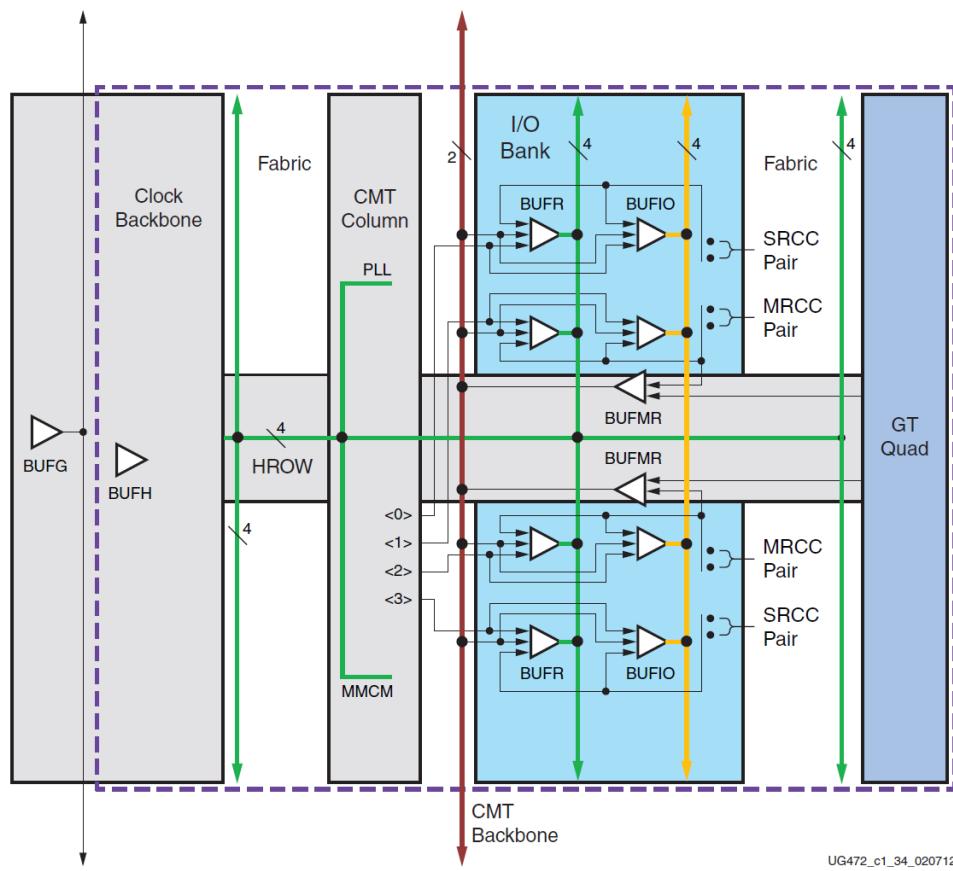
### 2.2.1 Taktarchitektur

Um taktsynchrone Datenverarbeitungslogik, das Haupteinsatzgebiet von FPGAs, möglichst hochfrequent betreiben zu können, ist eine gesonderte Verteilung des Taktsignals in das FPGA-Fabric über dedizierte Pfade notwendig. Diese sind so konzipiert, dass der Taktversatz an den Endpunkten möglichst minimiert wird. Die Taktverteilungspfade zeichnen sich durch eine geringe Signallaufzeit, sowie eine geringe Anzahl an nichtlinearen Schaltungen (Transistoren) im Signalweg aus. Aufgrund dessen werden die Taktspiele bei der Implementierung eines Time to Digital Converters auch für den Transport der Sensorimpulse genutzt. Deren Signalqualität hat direkte Auswirkungen auf das Messergebnis.

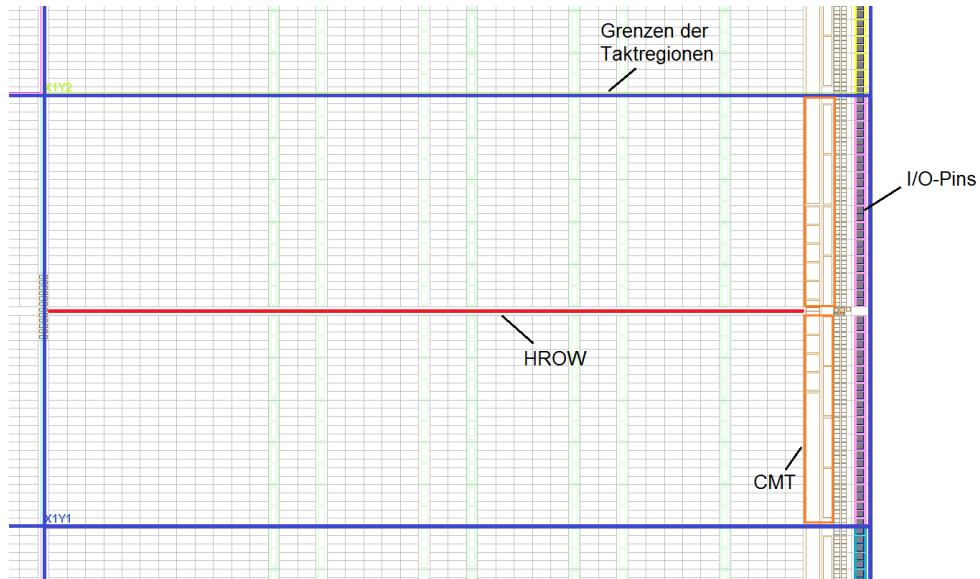
Xilinx 7-Series FPGAs sind in mehrere, in zwei Spalten angeordnete, Taktregionen unterteilt. Am Rand der Taktregionen befinden sich jeweils die Anschlüsse der I/O-Pins, sowie ein *Clock Management Tile* (CMT), welches die Treiber für die Taktverteilungspfade, sowie jeweils einen *Phase Locked Loop* (PLL) und einen *Mixed Mode Clock Manager* (MMCM) (s. Abschnitt 2.2.4) enthält. PLL und MMCM werden, unter Anderem, zur Synthese nichtnativer Taktfrequenzen eingesetzt.

Eingehende Signale oder synthetisierte Takte werden von regionalen High-Fanout Takttreibern (BUFR) in die horizontal verlaufende *Horizontal Clock Row* (HROW) in der vertikalen Mitte der Taktregion gespeist, von der aus das Signal vertikal dem FPGA-Fabric sowie in das Clock-Backbone zwischen den beiden Taktregionsspalten zugeführt wird [6] (s. Abbildung 2.1). Abbildung 2.2 zeigt den grundlegenden Aufbau einer Taktregion.

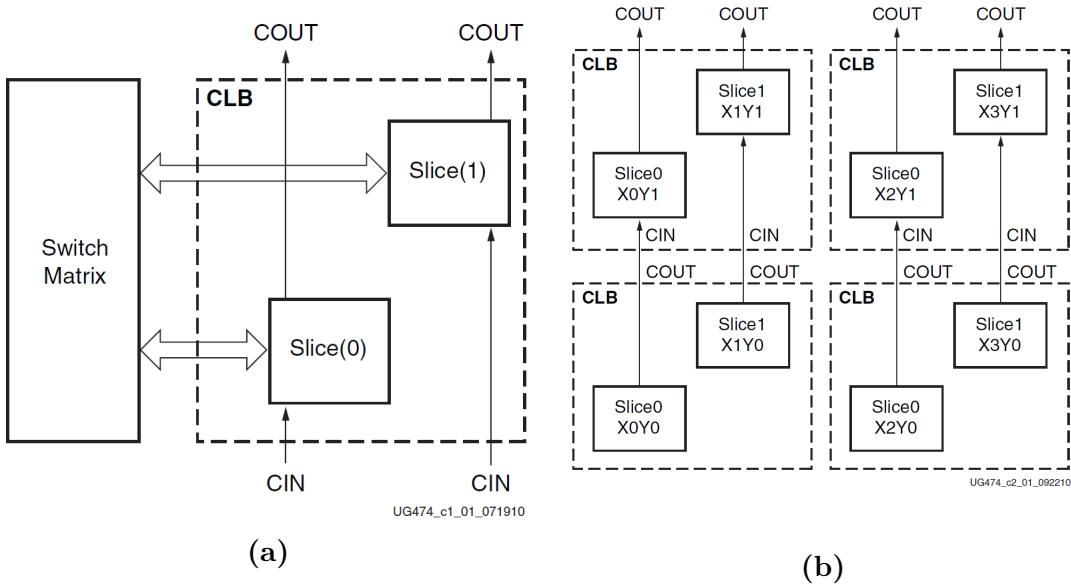
Vom Clock-Backbone werden die globalen Taktnetze gespeist, welche Elemente auf der gesamten Fläche des FPGA ansteuern können. Diese sind so konzipiert, dass eine Taktflanke möglichst zur gleichen Zeit an jeder Stelle des FPGA ankommt.



**Abbildung 2.1:** Struktur der Taktverteilungspfade innerhalb einer Taktregion von Xilinx 7-Series FPGAs. (Quelle: [6])



**Abbildung 2.2:** Aufbau einer Taktregion eines Xilinx Kintex-7 FPGAs. Gekennzeichnet sind die I/O-Pins, die Grenzen der Taktregionen, die Clock Management Tiles (CMTs), sowie die horizontal in der Mitte der Taktregion verlaufende HROW [6].



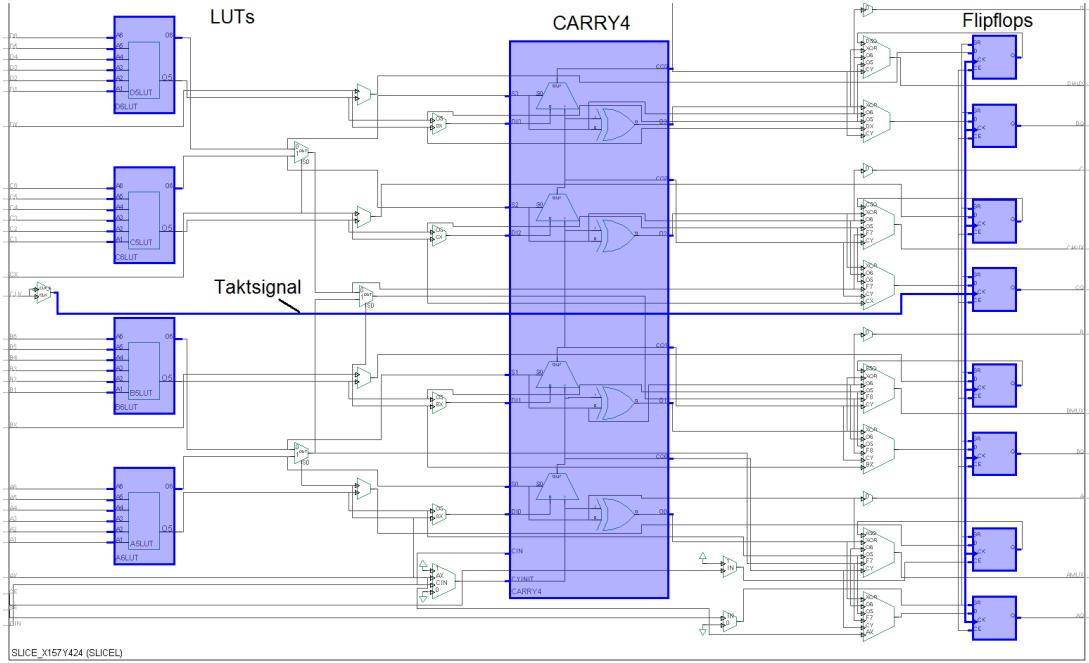
**Abbildung 2.3:** (a) Anbindung der Slices eines CLB an das FPGA-Fabric über eine gemeinsame Switchmatrix, (b) Matrix aus CLBs des FPGA-Fabrics mit dedizierten Verbindungen der CARRY4-Elemente zur Übertragsberechnung COUT und CIN. (Quelle: [7])

## 2.2.2 Configurable Logic Block (CLB)

Das Fabric der Xilinx 7-Series FPGAs besteht weitestgehend aus einer Matrix aus identischen *Configurable Logic Blocks* (CLBs). Jeder CLB enthält zwei Slices, welche die funktionalen Hardwarelemente beinhalten. Die Slices eines CLB sind über eine Switchmatrix mit anderen CLBs verbunden. Abbildung 2.3a zeigt die Anbindung der Slices an die Switch Matrix.

Jedes Slice enthält vier Look Up Tables (LUTs) mit jeweils sechs Eingängen und zwei Ausgängen. Diese können mit einer Wahrheitstabelle konfiguriert werden. So wird die funktionale Logik auf einem FPGA realisiert. Ein CARRY4-Element erlaubt die Implementierung schneller Übertragsoperationen (s. Abschnitt 2.2.3). Die CARRY4-Elemente sind ohne Umwege über die Switchmatrix mit den CARRY4-Elementen vertikal adjazenter Slices verbunden. Somit lassen sich lange Carry-Ketten bilden. Abbildung 2.3b zeigt die Anbindung der CARRY4-Elemente untereinander.

Acht Flipflops, deren Dateneingänge direkt an die Ausgänge der genannten Sliceelemente oder an die Eingänge des Slices angebunden sind, speichern die Daten taktsynchron. Alle Flipflops eines Slices werden mit dem selben Takt signal betrieben. Es ist nicht möglich, Flipflops eines Slices mit unterschiedlichen Signalen zu takten. Abbildung 2.4 zeigt den detaillierten Aufbau eines Slices. LUTs, CARRY4, Flipflops sowie das Takt signal sind hervorgehoben.



**Abbildung 2.4:** Detailansicht eines Slices. LUTs, CARRY4, Flipflops sowie das Taktsignal sind hervorgehoben. Das selbe Taktsignal schaltet alle Flipflops eines Slices.

### 2.2.3 CARRY4-Element

Um eine schnellen Übertragsberechnung, beispielsweise für Ripple-Carry-Addierer, realisieren zu können, gibt es in jedem Slice ein CARRY4-Element. Diese bestehen aus vier asynchron in Reihe geschalteten 2:1 Multiplexern (s. Abbildung 2.5). Diese Kette wird über den CYINIT-Port initialisiert. Die Ausgänge der Multiplexer werden als C0-Signal unverändert ausgegeben. Der höchstwertigste Ausgang eines CARRY4-Elements ist über eine dedizierte Signalleitung direkt an den Eingang des CARRY4-Elements im nächsten Slice angebunden und läuft nicht über die Switchmatrix des CLBs (s. Abschnitt 2.2.2). Diese Eigenschaft ermöglicht die einfache Implementierung einer, von Routingalgorithmen unabhängigen, asynchronen Kette aus CARRY4-Elementen.

### 2.2.4 Mixed Mode Clock Manager (MMCM)

Der Mixed Mode Clock Manager besteht aus einem Phase Locked Loop, sowie einer ergänzenden Beschaltung für zusätzliche Funktionalitäten. Der MMCM ist in der Lage eine große Anzahl an Frequenzen über die programmierbaren Taktteiler  $D$  (Eingangstaktteiler),  $M$  (Feedbacktaktteiler) und O0-O6 (Ausgangstaktteiler) zu synthetisieren.

Die Funktionalität des MMCM ist in Abbildung 2.6 dargestellt. Ein *Phase-Frequency De-*

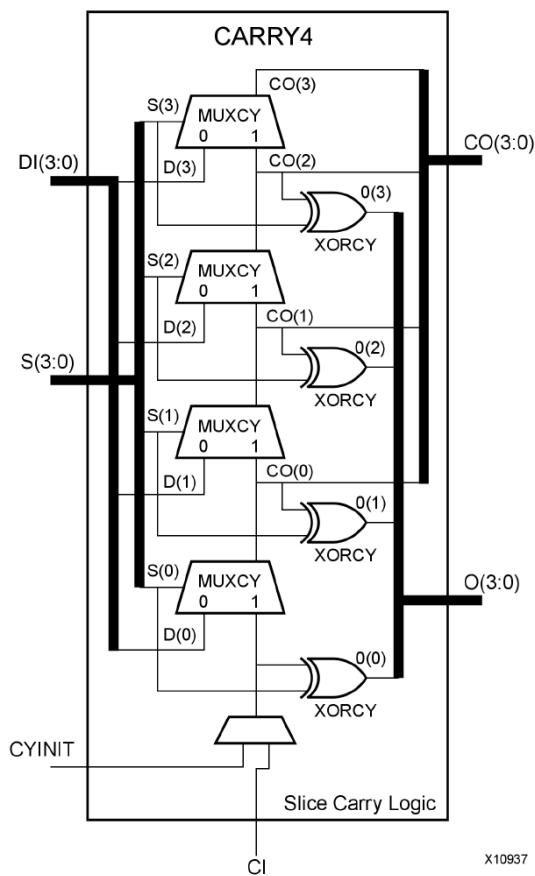
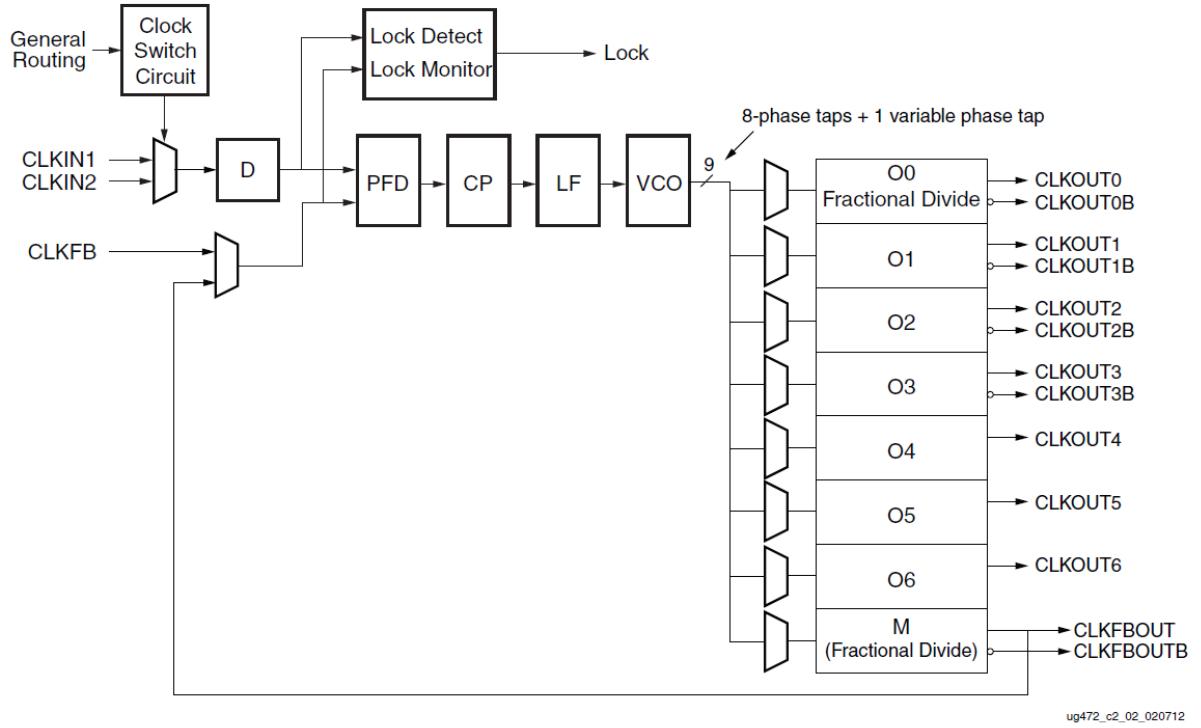


Abbildung 2.5: CARRY4-Element eines Xilinx 7-Series FPGAs. (Quelle: [8])



**Abbildung 2.6:** Schematischer Aufbau eines Mixed Mode Clock Managers. Ein PFD vergleicht Phase und Frequenz des geteilten Eingangstaktes. Eine Gleichspannung wird erzeugt welche einen VCO steuert. Die zu synthetisierenden Frequenzen können über die programmierbaren Taktteiler  $D$ ,  $M$  und  $O_0$ - $O_6$  gemäß Gleichung (2.1) eingestellt werden. (Quelle: [6])

*tector* (PFD) vergleicht Phase und Frequenz des Eingangstaktes mit der des Feedbacktaktes. Eine Ladungspumpe (*Charge Pump*, CP) und einen Tiefpassfilter (LF) erzeugen eine Gleichspannung, mit welcher ein *Voltage Controlled Oscillator* (VCO) angesteuert wird. Dessen Ausgangssignal ergibt, jeweils geteilt durch einen von sieben separat programmierbaren Taktteilern, die Ausgangstaktsignale mit einer Frequenz gemäß Gleichung (2.1).

$$f_{out} = f_{in} \cdot \frac{M}{D \cdot O} \quad (2.1)$$

Dieses System hat eine gewisse „Trägheit“ bei der Reaktion auf Änderungen des Eingangstaktes. Es lässt sich somit auch als Jitterfilter einsetzen.

## 2.2.5 Block-RAM

Das RAMB18E1-Element ist das grundlegende *Random Access* Speicherelement der Xilinx 7-Series FPGAs mit 18 KBit Speichergröße. Es unterstützt die gleichzeitige, unabhängige

Ansteuerung des Speichers auf zwei Ports mit unterschiedlichen Taktfrequenzen und eine Wortbreite von bis zu 18 Bit. Über jeden dieser Ports können Daten gelesen und geschrieben werden. Zwei RAMB18E1-Elemente können zu einem Random Access Speicher mit bis zu 36 Bit Wortbreite zusammengeschlossen werden (RAMB36E1).

Die RAM-Elemente des zu implementierenden TDC-Moduls werden über Macros der Xilinxbibliothek *Unimacro* instanziert. Dies erlaubt die einfache Implementierung der RAM-Elemente in einem spezifischen Modus und verbirgt, für diesen nicht notwendige, Interfaceports. Insbesondere werden die RAM-Blöcke des Designs im *Simple Dual Port* (SDP) Modus und als asynchroner First-In-Fist-Out Speicher (FIFO) eingesetzt. Im SDP-Modus dient jeweils ein Port als dedizierter Lese- bzw. Schreibport. Dies Vereinfacht die Ansteuerung. Im FIFO-Modus entfällt das Addressmanagement komplett. Der FIFO kann mit asynchronen Taktsignalen betrieben werden und ermöglicht so das Übertragen von Daten zwischen unterschiedlichen Takten.

# Kapitel 3

## FPGA-basierte TDCs und Messprinzipien

### 3.1 FPGA-basierte Time to Digital Converter

Zur Realisierung von Time to Digital Convertern in FPGAs wird versucht, gängige, von konventionellen TDCs bekannte, Messprinzipien in der Struktur des FPGAs nachzubilden. Dies ist für viele Methoden nur begrenzt möglich.

1997 gelang es Kalisz et al. den ersten Time to Digital Converter auf Basis eines kommerziell verfügbaren QuickLogic FPGAs mit einer Zeitauflösung von 200 ps und einem Dynamikbereich von 10 ns zu entwerfen [2].

Seither werden FPGA-basierte TDCs aufgrund von Weiterentwicklungen in der FPGA-Technologie, sowie innovativen und zunehmend komplexen Implementierungskonzepten immer kompetitiver.

### 3.2 Messprinzipien und vorausgehende Arbeiten

In diesem Abschnitt werden drei gängige Messprinzipien von Time to Digital Convertern beschrieben, sowie vorausgehende Arbeiten zu deren FPGA-Implementierungen vorgestellt. Die Machbarkeit einer FPGA-basierten Umsetzung kann so eingeschätzt werden.

### 3.2.1 Vernieroszillator

Der Vernieroszillator basiert auf dem Vergleich zweier periodischer Ringoszillatoren mit leicht unterschiedlicher Frequenz. Start- und Stopimpuls lösen jeweils einen Ringoszillator aus. Aus der Anzahl der vergangenen Oszillationsperioden bis zu einem Zeitpunkt an dem beide Taktflanken zeitgleich auftreten, kann auf das zu messende Zeitintervall geschlossen werden. Dies ist im Timingdiagramm in Abbildung 3.1 dargestellt. Während der Oszillationszeit können keine weiteren Messungen durchgeführt werden.

Ein großer Vorteil der Realisierung eines Vernieroszillators auf dem FPGA ist der geringe Ressourcenbedarf, sowie die hohe theoretisch erreichbare Präzision. Dafür ist es jedoch notwendig, Ringoszillatoren mit einer, zur Laufzeit bekannten, stabilen Frequenz zu implementieren. Bei Vernieroszillatoren in integrierten Schaltkreisen wird dies über eine Temperatur- und Spannungsregelung erreicht. Auf einem FPGA ist dies jedoch nicht ohne Weiteres möglich.

Literatur zu FPGA-basierten Vernier Oszillatoren gibt es nur wenig. Lin et al. [9] stellen Methoden und Strukturen zur Implementierung der notwendigen Bestandteile wie Ringoszillator und Phasendetektor, welcher die Koinzidenz der beiden Oszillatorflanken feststellt, vor. Im Ergebnisteil bleibt es jedoch bei Simulationsergebnissen und einer theoretisch errechneten Genauigkeit von etwa 30 ps.

### 3.2.2 Tapped Delay Line

In der *Tapped Delay Line* (TDL) wird die Zeit zwischen dem Eintreffen des Messimpulses und der steigenden Taktflanke eines bekannten Taktes interpoliert. Der Messimpuls propagiert durch eine Kette aus Verzögerungselementen, deren Ausgänge taktsynchron abgetastet werden. Aus der Propagationsweite des Messsignals in der Verzögerungskette zum Abtastzeitpunkt kann auf die Zeit zwischen Mess- und Taktflanke geschlossen werden. Die Struktur der TDL ist in Abbildung 3.2 dargestellt.

Das Tapped Delay Line Prinzip ist in der akademischen Literatur zu FPGA-basierten TDCs am besten dokumentiert. Neben Single-Sample-Implementierungen (eine Messung pro Messimpuls), wie dem Design von Pan et al. [10], welches einen relativen mittleren Fehler von 21ps aufweist, wird auch mit erweiterten Methoden experimentiert. Beim Wave Union TDC werden beispielsweise zwei Impulsflanken in der selben TDL generiert und gemessen. Durch das gemeinsame Betrachten beider Flanken entstehen zusätzliche, „virtuelle“, Quantisierungsstufen. Auch die Abtastfrequenzen steigen an die Grenzen der momentanen Technologie. Liu und Wang realisierten 2015 einen Wave Union TDC mit

einer Abtastrate von 710 Msps und einem relativen mittleren Fehler von 8,7ps auf einem Kintex-7 FPGA [11]. Dieser ist in der Lage, in 128 Kanälen parallel Messungen durchzuführen. Solche Kennzahlen sind auch bei kommerziell verfügbaren TDC-Chips unüblich.

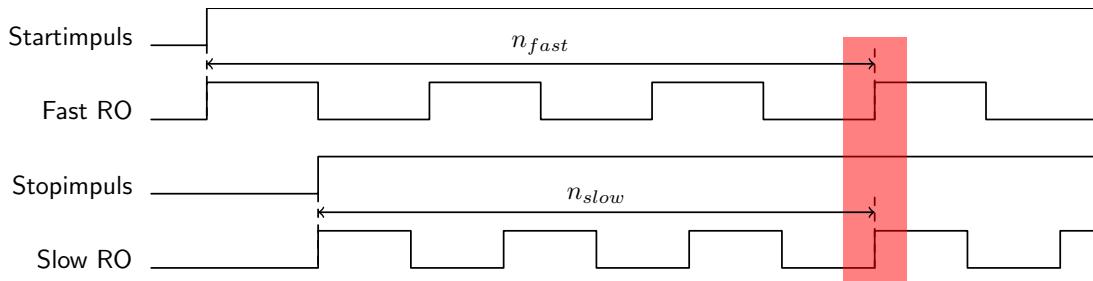
### 3.2.3 Vernier Delay Line

Die Vernier Delay Line ähnelt prinzipiell dem Vernieroszillator und strukturell der Tapped Delay Line. Start- und Stopimpuls propagieren durch Ketten aus Verzögerungselementen mit leicht unterschiedlichen Verzögerungszeiten. Die Ausgänge der langsamen Delay Line (Startimpuls) werden durch Flipflops abgetastet, welche mit den Ausgängen der schnellen Delay Line abgetastet werden. „Überholt“ der Stopimpuls, aufgrund der schnelleren Propagation durch die Kette, den Startimpuls, bleibt der Inhalt der folgenden Register unverändert (s. Abbildung 3.3). Prinzipiell lässt sich so eine höhere Präzision als mit einer Tapped Delay Line erreichen.

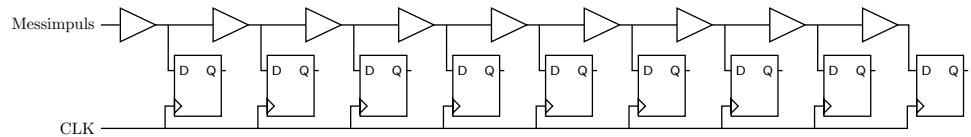
Auch die Vernier Delay Line auf Basis eines FPGA ist wenig dokumentiert. Die Schwierigkeit besteht darin, unterschiedlich schnelle Delay Lines in der FPGA-Struktur zu realisieren. Aloisio et al. lösten dies 2007 durch das parallele Platzieren zweier Delay Lines aus Buffern (schnell) und Latches (langsam) auf einem Xilinx Virtex 5 FPGA [12]. Die resultierende relative mittlere Messgenauigkeit von 80ps bleibt jedoch hinter modernen TDL-Realisierungen zurück.

## 3.3 Wahl der Architektur für diese Arbeit

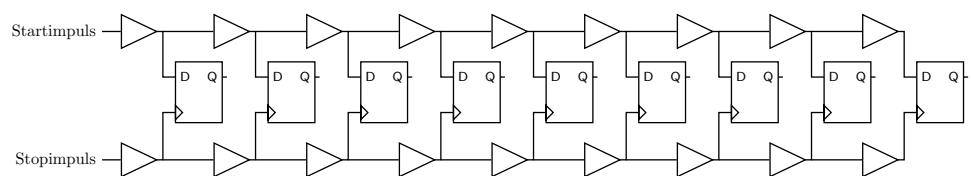
Aufgrund der an die Realisierung gestellten Anforderungen zur Multishotfähigkeit, der ungleich mehr vielfältigen akademischen Literatur, sowie der fraglichen Realisierbarkeit von Vernieroszillator-TDCs auf FPGA-basis, fällt die Wahl TDC-Architektur für diese Arbeit auf die Tapped Delay Line (TDL).



**Abbildung 3.1:** Timingdiagramm des Vernier Oszillatoren. Durch das Zählen von Oszillationsperioden des schnellen und langsamen Ringoszillators bis zum zeitgleichen Auftreten der positiven Flanke kann auf die Intervallzeit geschlossen werden.



**Abbildung 3.2:** Schematische Struktur der Tapped Delay Line. Der Inhalt der Kette aus Verzögerungselementen wird bei steigender Flanke des Systemtaktes in die Flipflops übernommen. Die Propagationsweite zum Abtastzeitpunkt ist ein Maß für die vergangene Zeit zwischen Messimpuls- und Taktflanke.

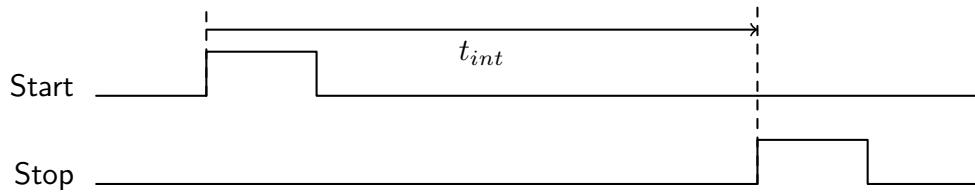


**Abbildung 3.3:** Schematische Struktur der Vernier Delay Line. Der Startimpuls wird stärker verzögert als der Stopimpuls. „Überholt“ der Stopimpuls aufgrund der schnelleren Propagation durch die Kette den Startimpuls, bleibt der Inhalt der folgenden Register unverändert. Die Propagationsweite zum Abtastzeitpunkt ist ein Maß für die vergangene Zeit zwischen Messimpuls- und Taktflanke.

# Kapitel 4

## Konzept

Ziel der Entwicklung des Time to Digital Converter ist die exakte Messung des Zeitintervalls zwischen den steigenden Flanken zweier unabhängiger Messimpulse (Abbildung 4.1).



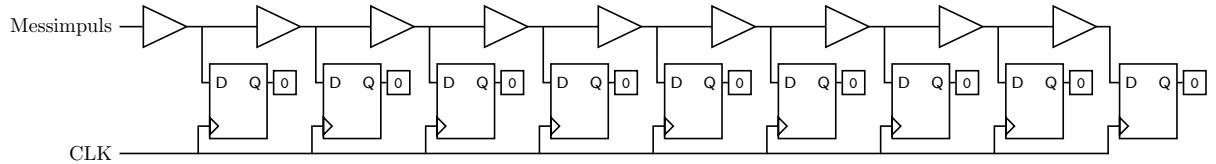
**Abbildung 4.1:** Zeitintervall zwischen den steigenden Flanken zweier Messimpulse.

In diesem Kapitel wird die in dieser Arbeit realisierte Messmethode, die Tapped Delay Line (TDL), näher erläutert und die platformbedingten Schwierigkeiten einer FPGA-basierten Umsetzung aufgezeigt.

### 4.1 Messprinzip: Tapped Delay Line (TDL)

Als Tapped Delay Line wird eine Struktur aus einer Kette in Reihe geschalteter Verzögerungselemente („Delay Line“) bezeichnet, deren Ausgänge von Flipflops, synchron zum Systemtakt, abgetastet werden (engl. „to tap“). Die Verzögerungskette ist auf dem FPGA als Kette aus CARRY4-Elementen (vgl. Abschnitt 2.2.3) aufgebaut. Der strukturelle Aufbau ist in Abbildung 4.2 schematisch dargestellt.

Ein Messimpuls trifft zu einem, relativ zur Systemtaktflanke zufälligen, Zeitpunkt am Eingang der TDL ein. Die Impuls propagiert durch die Delay Line. Die Signalpegel der Ausgänge der Verzögerungselemente werden zum Abtastzeitpunkt synchron als digitaler



**Abbildung 4.2:** Struktur der Tapped Delay Line. Die Ausgänge einer Kette aus, in Serie geschalteten, Verzögerungselementen werden mit taktsynchronen Flipflops abgetastet.

Wert in die Flipflops übernommen. In Abbildung 4.3 wird ein solcher Abtastvorgang im zeitlichen Verlauf dargestellt.

In jedem Taktzyklus wird von der TDL der Inhalt der Abtastflipflops parallel ausgegeben. Aufgrund ihres Erscheinungsbildes wird diese Bitfolge auch als *Thermometercode* bezeichnet. Die Information des Thermometercodes ist alleinig in der Signalfankenposition enthalten. Aus ihr lässt sich die vergangene Zeit zwischen Messimpuls- und Systemtaktflanke ermitteln. Diese Zeit ist streng monoton von der Propagationsweite des Messimpulses in der Delay Line abhängig.

Die Zeit zwischen Messimpulsflanke und Taktflanke des Abtasttaktes  $t$  (Abbildung 4.4) wird prinzipiell aus dem Anteil der Elemente des Thermometercodes  $n_1$ , in denen eine Eins detektiert wird, und der Gesamtlänge der Tapped Delay Line  $N$  gemäß Gleichung (4.1) berechnet.  $t_{clk}$  ist die Taktperiode des Abtasttaktes.

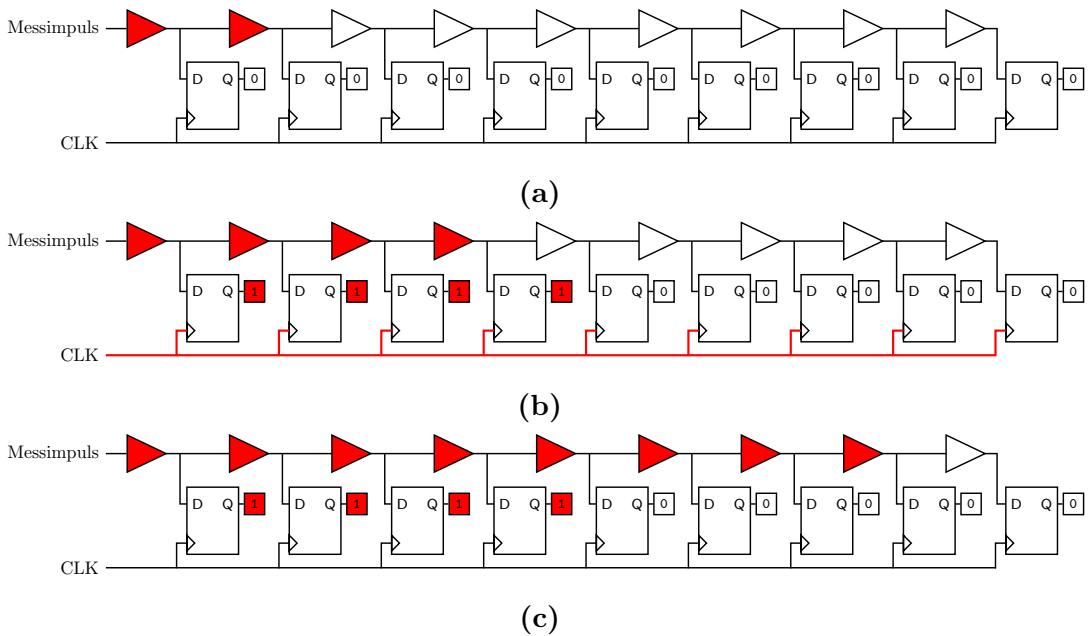
$$t = \frac{n_1}{N} \cdot t_{clk} \quad (4.1)$$

#### 4.1.1 Messen eines Zeitintervalls

Mit einer einzelnen Tapped Delay Line kann die Zeit zwischen der Flanke eines Start- bzw. Stopimpulses und der folgenden Taktflanke des Systemtaktes  $t_{Start}$  bzw.  $t_{Stop}$  gemessen werden. Um ein Intervall zwischen zwei Messimpulsflanken messen zu können, bedarf es zweier TDLs und eines binären Zählers  $n_{clk}$ . Dieser zählt die, zwischen den Messimpulsen vergangenen, ganzen Taktperioden.

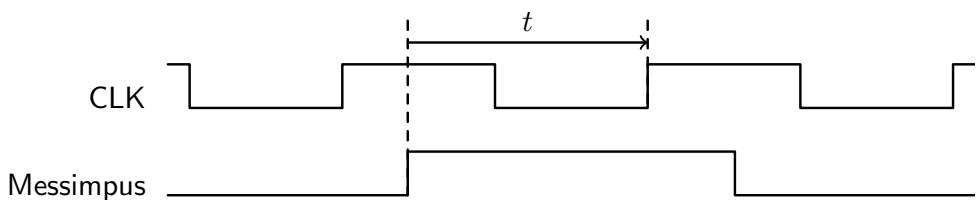
Gemäß Gleichung (4.2) ergibt sich aus diesen Komponenten das Gesamtzeitintervall  $t_{int}$ . Dies ist in Abbildung 4.5 veranschaulicht.

$$t_{int} = t_{Start} - t_{Stop} + n_{clk} \cdot t_{clk} \quad (4.2)$$

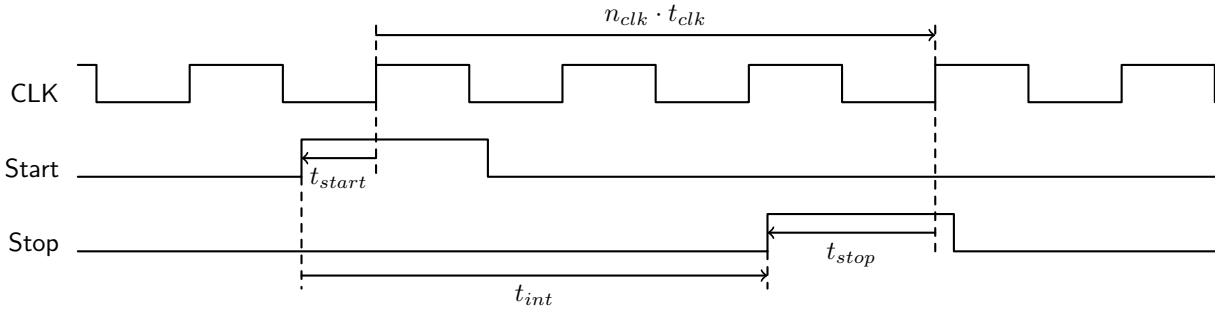


**Abbildung 4.3:** Zeitlicher Verlauf eines Messvorgangs der TDL.

(a) Der Messimpuls propagiert durch die TDL, (b) Zum Abtastzeitpunkt wird der momentane Zustand der TDL in die Register übernommen, (c) Der Messimpuls propagiert weiter durch die TDL. Der Registerinhalt bleibt dabei unverändert.



**Abbildung 4.4:** Veranschaulichung der von der Tapped Delay Line gemessenen Zeit  $t$  zwischen Messimpuls- und Taktflanke.



**Abbildung 4.5:** Timingdiagramm der Zeitintervallmessung des TDCs mit Messungen von Start- und Stopimpuls. Die Zeiten zwischen den Messimpuls- und Taktflanken des Systemtaktes  $t_{Start}$  und  $t_{Stop}$  ergeben zusammen mit der Anzahl der vergangenen ganzen Taktperioden  $n_{clk}$  gemäß Gleichung (4.2) die zu messende Zeit  $t_{int}$ .

## 4.2 Dynamische Kalibrierung der TDL

Die Verzögerungszeiten der einzelnen Verzögerungselemente der TDL haben folgende Eigenschaften, welche für die Zeitmessung von Nachteil sind:

- Die Verzögerungszeiten der einzelnen Elemente sind hochgradig ungleich,
- können zur Entwurfszeit nicht vorhergesagt werden,
- können von Chip zu Chip variieren,
- und variieren auch bei Änderungen von Temperatur und Versorgungsspannung des FPGAs.

Aufgrund dieser Variationen der Verzögerungszeiten ist der Zusammenhang zwischen der Propagationsweite des Messimpulses durch die TDL und der vergangenen Zeit nicht linear. Die genaue Ursache dieses Verhaltens kann nur vermutet werden, da es keine öffentlich zugängliche Informationen über die Chipstruktur und die zu Gunde liegenden Hardware gibt. Es wird jedoch davon ausgegangen, dass sie aus zwei Komponenten besteht, einer statischen, also für jeden Chip zeitlich konstanten, und dynamischen, zeitlich variierenden, Nichtlinearität [13].

Die statische Nichtlinearität resultiert aus der Logikstruktur des FPGAs. Jedes Slice des FPGAs beinhaltet vier Verzögerungselemente welche räumlich relativ dicht beieinander liegen. Die Signallaufzeit zwischen den Elementen ist dementsprechend klein. Zwischen den Slices müssen jedoch längere Signalwege zurückgelegt werden. Die Verzögerungszeit jedes vierten Verzögerungselementes wird somit um diese zusätzliche Signallaufzeit erhöht. Eine weitere Ursache für statische Nichtlinearitäten kann die Länge der Signalwege vom Ausgang des Verzögerungselementes bis zum Dateneingang der Flipflops, sowie vom

Takteingang des Slices zum Takteingang der Flipflops sein. Auch diese zählen zur Verzögerungszeit eines Elements hinzu, da ausschlaggebend für eine Detektion alleinig das (Nicht-)Vorhandensein einer Spannung am Dateneingang der Flipflops zum jeweiligen Abtastzeitpunkt ist. Auch sind, aufgrund des fehlerbehafteten Herstellungsprozesses des FPGAs, Variationen der Transistorgeschwindigkeiten zu erwarten.

Die dynamische Nichtlinearität resultiert aus der Variation der äußeren Umstände, wie der Chipspannung und Chiptemperatur, sowie der gegenseitigen elektromagnetischen Beeinflussung der schaltenden Logik. Durch das Nichteinhalten von Setup- und Holdzeiten können zufällige oder undefinierte metastabile Zustände am Ausgang des Flipflops entstehen [14].

Menninga et al. haben die statischen Fehler bei der Implementierung der Tapped Delay Line an unterschiedlichen Positionen des FPGA-Fabrics untersucht und festgestellt, dass die Nichtlinearität positionsabhängig stark (Faktor > 2) variiert [13]. Sie konnten auch zeigen, dass die räumliche Nähe von schaltender Logik auf der Chipfläche des FPGA den dynamischen Anteil der Nichtlinearität der Verzögerungszeiten einen messbaren Einfluss hat.

Um das TDC-Modul unter allen Umständen und ohne vorherige, aufwändige Vermessung des Chips einsetzen zu können, müssen die Verzögerungszeiten der einzelnen Verzögerungselemente zur Laufzeit erfasst werden können. Mit der Kenntniss dieser Verzögerungszeiten kann dann eine Abbildungsvorschrift erstellt werden, welche die Propagationsweite des Messimpulses in der Tapped Delay Line akkurat auf die vergangene Zeit abbildet. Somit kann die statische Nichtlinearität sowie der Temperaturanteil der dynamischen Nichtlinearität, welcher für kurze Zeiten als konstant betrachtet wird, kompensiert werden. Dieser Vorgang wird im folgenden als *Dynamische Kalibrierung* bezeichnet.

Eine Möglichkeit, die Verzögerungszeiten zur Laufzeit zu ermitteln, ist der *Code Density Test*.

#### 4.2.1 Code Density Test

Der sogenannte Code Density Test basiert auf dem zentralen Grenzwertsatz der Stochastik bei identischer Verteilung. Dieser sagt aus, dass, mit einem ausreichend großen Stichprobenumfang  $N$ , der Mittelwert der beobachteten Ereignisse eines Zufallsprozesses gleichbleibenden Charakters gegen die ihm zu Grunde liegende Wahrscheinlichkeitsverteilung konvergiert. Im Folgenden wird beschrieben, wie dies für das gegebene Problem genutzt werden kann.

Der Zeitpunkt des Eintreffens eines Testimpulses in der Tapped Delay Line wird, relativ zur Systemtakflanke, als gleichverteilt angenommen. Die Zeit zwischen Eintreffen des Testimpulses und der darauf folgenden Systemtakflanke  $t_{clk} - t_i$  ist dann gleichverteilt über ein Intervall  $[0, t_{clk}]$ . Die Wahrscheinlichkeit der Detektion der Messimpulsflanke an einer Position  $m$  im Thermometercode  $p_m$  entspricht dann dem Anteil der Verzögerungszeit des korrespondierenden Verzögerungselements  $t_m$  an der maximal erfassbaren Verzögerungszeit, welche  $t_{clk}$  entspricht:

$$p_m = \frac{t_m}{t_{clk}} \quad (4.3)$$

Das Auftreten oder Nichtauftreten der Messimpulsflanke wird, unabhängig für jede Signalflankenposition  $m$  in der TDL, als ein Bernoulli-Versuch mit  $p = p_m$  betrachtet. Werden  $N$  Messungen durchgeführt, so ist die Anzahl der „Erfolge“  $S_{N,m}$  des Bernoulli-Prozesses an der Position  $m$  in der TDL binomialverteilt.

Nach dem zentralen Grenzwertsatz konvergiert die beobachtete Auftretenswahrscheinlichkeit  $\hat{p}_m$  für  $N \rightarrow \infty$  gegen ihren Erwartungswert.

$$\hat{p}_m = \frac{S_{N,m}}{N} \quad (4.4)$$

$$\lim_{N \rightarrow \infty} |p_m - \hat{p}_m| = 0 \quad (4.5)$$

Die geschätzte Verzögerungszeit  $\hat{t}_m$  eines Verzögerungselementes ergibt sich nach Gleichung (4.3) aus der beobachteten Detektionswahrscheinlichkeit  $\hat{p}_m$ :

$$\hat{t}_m = \hat{p}_m \cdot t_{clk} \quad (4.6)$$

Bezogen auf die Tapped Delay Line bedeutet dies, dass aus  $N$  statistisch unabhängigen Testmessungen die Auftretenswahrscheinlichkeit  $p_m$  der Testimpulsflanke an einer Position  $m$  in der TDL und somit die Verzögerungszeit  $t_m$  abgeschätzt werden kann.

Das für diesen Kalibrierungsvorgang notwendige, vom Systemtakt unabhängige, Testsignal wird auf dem FPGA in einem frei laufenden Ringoszillator generiert. Der Ringoszillator ist in Abschnitt 5.6 beschrieben. Die Gleichverteilung des Auftretens der Ringoszillatorflanke über die Dauer einer Taktperiode des Systemtaktes, also die statistische Unabhängigkeit der Testimpulse, wird in Abschnitt 6.1.3 gezeigt.

Wird eine Impulsflanke an der Position  $m$  der TDL detektiert, wird der Inhalt eines Block-RAM-Elements an der Adresse  $m$  inkrementiert. Dieser Speicher wird als Kalibrierungshistogramm oder einfach Histogramm bezeichneten. Nach  $N$  Messungen ist das Kalibrierungshistogramm vollständig. Es beinhaltet nun die beobachteten Auftretenswahrscheinlichkeiten  $\hat{p}_m$  beziehungsweise Verzögerungszeiten  $\hat{t}_m$ .

Ein Maß für die Verlässlichkeit dieser Schätzung  $\hat{t}_m$  ist das sogenannte Konfidenzintervall, auch Vertrauensbereich genannt,  $[t_u, t_o]$ . Mit der ergänzten Methode von Clopper und Pearson lassen sich Konfidenzintervalle für die Schätzung der unbekannten Wahrscheinlichkeit einer Binomialverteilung aus dem Stichprobenumfang  $N$  und der beobachteten Wahrscheinlichkeit  $\hat{p}_m$  berechnen [15] [16]. Für ein Konfidenzniveau von  $\gamma = 1 - \alpha = 0,95$  sagt das Konfidenzintervall aus, dass bei unendlicher Wiederholung der Stichprobennahme für 95% aller Fälle  $t_m \in [t_{u,m}, t_{o,m}]$  gilt.

Um einschätzen zu können, wie viele Testmessungen  $N$  für eine akkurate Schätzung der Verzögerungszeiten notwendig sind, werden Histogramme mit den Stichprobenumfängen  $N = 2^k, k = 14, \dots, 18$  aufgenommen. Für jeden Versuch wird für alle Verzögerungselemente die beobachtete Verzögerungszeit  $\hat{t}_m = \hat{p}_m \cdot t_{clk}$  sowie das zugehörige Konfidenzintervall für  $\alpha = 0,05$  und  $\alpha = 0,01$  mit MATLAB berechnet.

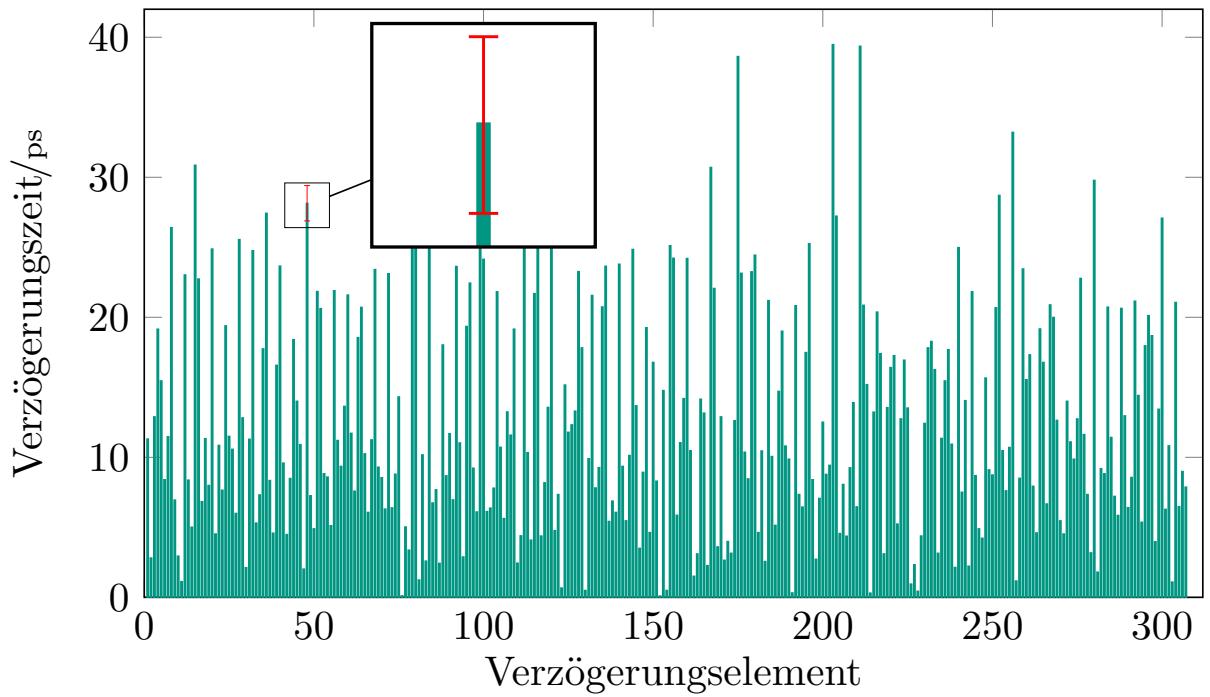
Für  $N = 2^{18}$  sind die beobachteten Verzögerungszeiten sowie beispielhaft ein Konfidenzintervall in Abbildung 4.6 dargestellt. Abbildung 4.7 zeigt die maximale und mittlere Länge der Konfidenzintervalle aller Stichprobenumfänge.

Es wird deutlich, dass eine hoher Stichprobenumfang für eine akkurate Schätzung der Verzögerungszeiten der Elemente von großer Wichtigkeit ist. Bei der Realisierung des TDC-Moduls wird deshalb ein Stichprobenumfang von  $N = 2^{18}$  implementiert.

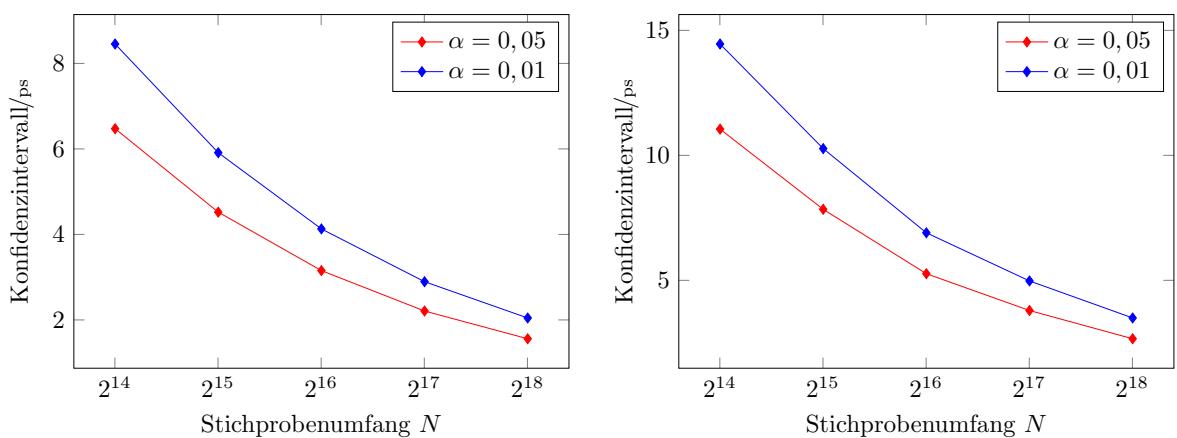
Zusammenfassend wurden mit dem Code Density Test aus  $2^{18}$  Testmessungen die Verzögerungszeiten der einzelnen Verzögerungselemente der Tapped Delay Line geschätzt und im Detektionshistogramm gespeichert.

#### 4.2.2 Erstellen einer Abbildungsvorschrift

Aus den ermittelten Propagationszeiten der einzelnen Verzögerungselemente wird nun eine Abbildungsvorschrift erstellt, welche jeder Signalfankenposition direkt einen Zeitwert zuordnet. Diese Abbildungsvorschrift wird als *Look-Up-Table* (LUT) bezeichnet. Sie wird gemäß Gleichung (4.7) als Summe der Propagationszeiten der passierten Verzögerungselemente gebildet.



**Abbildung 4.6:** Mit  $N = 2^{18}$  ermittelte Verzögerungszeiten  $\hat{t}_m$  der Tapped Delay Line. Beispielhaft ist bei  $m = 48$  das Konfidenzintervall für  $\alpha = 0,05$  eingezeichnet.



**(a)** Mittlere Länge der Konfidenzintervalle der Verzögerungszeiten  $\hat{t}_m$ .

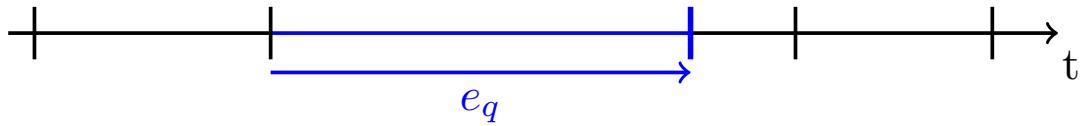
**(b)** Maximale Länge der Konfidenzintervalle der Verzögerungszeiten  $\hat{t}_m$ .

**Abbildung 4.7:** Mittlere und maximale Länge der Konfidenzintervalle für  $\hat{t}_m$  bei  $\alpha = 0,05$  und  $\alpha = 0,01$  in Abhängigkeit der Stichprobengröße.

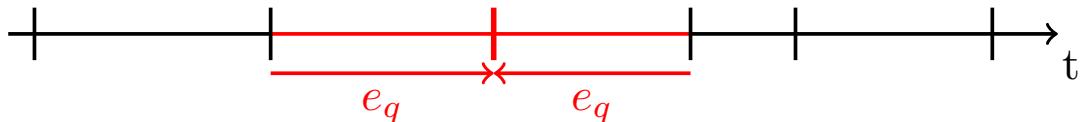
$$\begin{aligned}
 \text{LUT}(i) &= \frac{\hat{t}_i}{2} + \sum_{m=1}^{i-1} \hat{t}_m \\
 &= \frac{\text{HIST}(i)}{2} + \sum_{m=1}^{i-1} \text{HIST}(m)
 \end{aligned} \tag{4.7}$$

Durch die Addition der Hälfte von  $\text{HIST}(i)$  wird der, im LUT gespeicherte, Zeitwert in die Mitte des Detektionsintervalls verschoben. Der mittlere Quantisierungsfehler  $e_q$  der TDLs wird dadurch erheblich verringert. In Abbildung 4.8 ist diese Verschiebung veranschaulicht.

Auch der Look-Up-Table ist als Random Access Speicherelement realisiert. Legt man nun an der Adresse dieses Speichers die, in der TDL detektierten, Messimpulsflankenposition an, wird der entsprechende Zeitwert zwischen Messimpuls- und Taktflanke direkt ausgegeben. Der Inhalt von Histogramm und LUT ist in Abbildung 4.10 zu sehen. Abbildung 4.9 stellt die Generierung der Speicherinhalte anschaulich an einem Zeitstrahl dar.

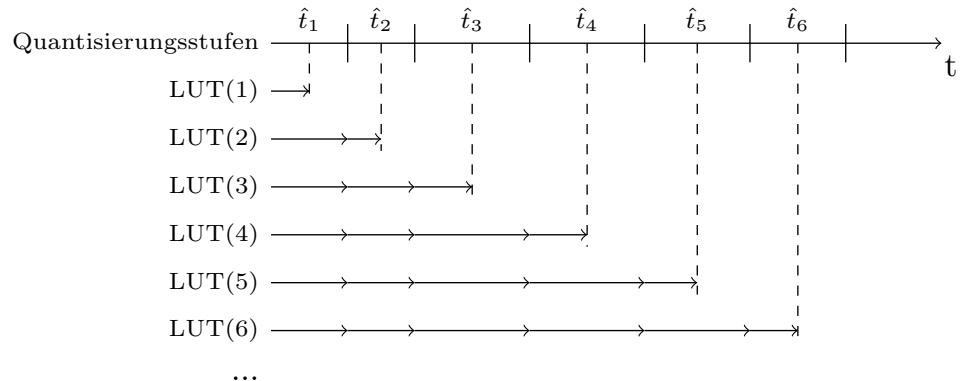


(a) Abbildung aller Werte des blau gekennzeichneten Bereichs des kontinuierlichen Zeitstrahls auf die markierte Quantisierungsstufe.

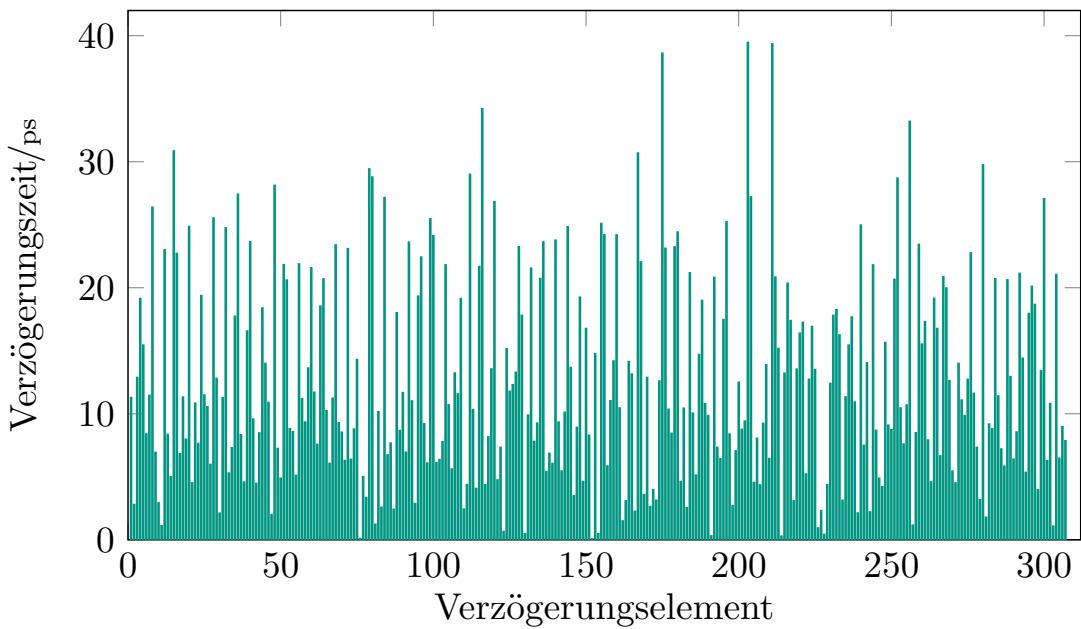


(b) Abbildung aller Werte des rot gekennzeichneten Bereichs des kontinuierlichen Zeitstrahls auf die markierte Quantisierungsstufe.

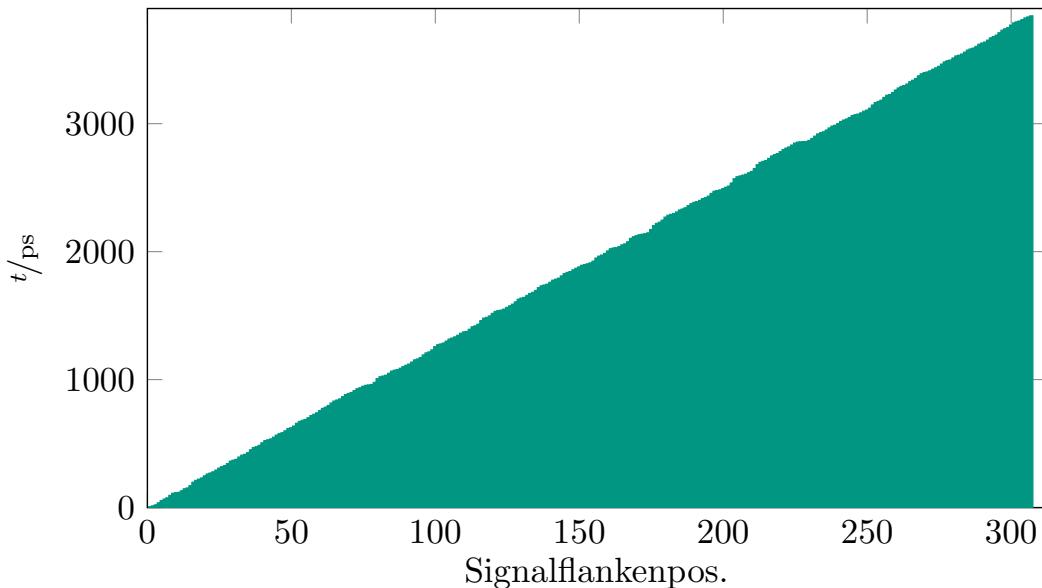
**Abbildung 4.8:** Durch das Verschieben der Quantisierungsstufe in die Mitte des Detektionsintervalls wird der maximale und mittlere Quantisierungsfehler  $e_q$  der TDL deutlich verringert.



**Abbildung 4.9:** Anschauliche Darstellung der Generierung des Look-Up-Tables als Summe der ermittelten Propagationszeiten der Verzögerungselemente an einem Zeitstrahl.



(a) Detektionshistogramm aus  $2^{18}$  aufgezeichneten Testimpulsen. Die hohe Nichtlinearität der Verzögerungszeiten ist deutlich zu erkennen.



(b) Der aus dem Histogramm generierte Look-Up-Table bildet die Impulsflankenposition in der TDL auf einen Zeitwert ab.

**Abbildung 4.10:** Kalibrierungshistogramm und Look-Up-Table. Der LUT ist die Aufsummierung nach Gleichung (4.7) der Verzögerungswerte des Histogramms.

# Kapitel 5

## Realisierung

Der TDC ist aus mehreren, nach Aufgaben logisch getrennten, Modulen aufgebaut, welche hierarchisch angeordnet sind. In Kapitel 5 wird die Implementierung der Module, angefangen bei der Datenerfassung in der TDL, bis hin zu einem taktübergreifenden Interface zu den dem TDC übergeordneten Logik, hierarchisch von unten nach oben beschrieben. Am Ende des Kapitels werden die Aufgaben und Anordnung der Module, sowie deren Datenschnittstellen zusammengefasst.

### 5.1 Tapped Delay Line

Grundlegender Baustein des TDC ist die Tapped Delay Line. Bei der Realisierung einer TDL muss unbedingt die zu Grunde liegende Hardwarearchitektur berücksichtigt werden. Wünschenswerte Eigenschaften der Verzögerungselemente sind möglichst geringe und homogene Propagationszeiten, um eine hohe Zeitauflösung und Linearität zu erreichen. Die Implementierung soll möglichst unabhängig vom *Place and Route* (PAR) Algorithmus sein, so dass das Verhalten der TDL nachvollziehbar und reproduzierbar ist. Ihre Abtastfrequenz wird so gewählt, dass die Propagationszeit einer Messimpulsflanke durch die Struktur  $t_{TDL}$  unter allen Umständen länger ist als die Taktperiode des Systemtaktes  $\frac{1}{f_s}$ . Es besteht sonst die Möglichkeit, dass eine Messimpulsflanke durch die TDL propagiert ohne detektiert zu werden:

$$t_{TDL} > \frac{1}{f_s} \quad (5.1)$$

Die Delay Line ist aus CARRY4-Elementen (s. Abschnitt 2.2.3) aufgebaut, welche in

jedem Slice der Xilinx 7-Series FPGAs vorhanden sind. Diese erfüllen die genannten Anforderungen am ehesten. Sie haben den Vorteil, über die Slicegrenzen hinweg mit schnellen, dedizierten Singalleitungen miteinander verbunden zu sein, ohne einen Umweg über die Switchmatrix des CLB zu machen (s. Abschnitt 2.2.2). Die Datenausgänge der vier Verzögerungselemente des CARRY4 werden sliceintern mit den Eingängen der Flipflops verbunden. Dies schließt ein unvorhersehbares, externes Routing durch den Algorithmus aus.

Aus CARRY4- sowie Flipflopelementen wird die TDL mittels Generateschleifen explizit instanziert. Ihre Platzierung wird, relativ zueinander, durch das vom PAR-Algorithmus erkannte Attribut **RLOC** festgelegt.

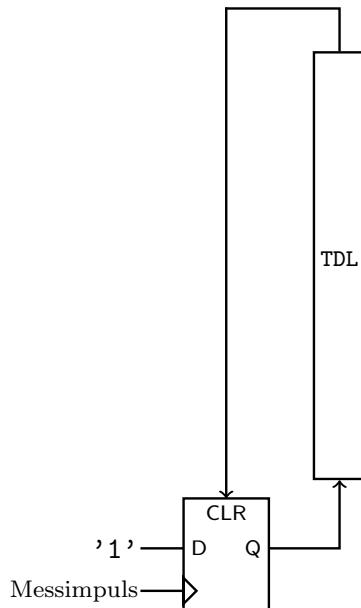
### 5.1.1 Zurücksetzen der TDL

Um mehrere, kurz aufeinander folgende, Messimpulse zuverlässig detektieren zu können (Multishotfähigkeit), muss sichergestellt werden dass, sich die TDL beim Eintreffen jedes separaten Messimpulses in ihrem Ausgangszustand befindet (der Zustand aller Verzögerungselemente ist Null).

Dies wird durch das Vorschalten eines Flipflops mit asynchronem Clear-Port bewerkstelligt, welcher den Ausgang des Flipflops auf Null setzt (s. Abbildung 5.1). Die Messimpulsflanke schaltet am Takteingang des Flipflops die Datenübernahme von dem, mit konstant Eins beschalteten, Dateneingang. Der Ausgang des Flipflops ist mit dem **CYINIT**-Port des ersten CARRY4-Elements der TDL verbunden.

Nachdem die Messimpulsflanke durch die gesamte Delay Line propagierte und ein Abtastvorgang erfolgt ist, wird das Signal auf den asynchronen Clear-Port des Flipflops rückgekoppelt. Der Ausgang des Flipflops wird zurückgesetzt. Die Delay Line wird so, unabhängig vom Zustand des Messimpulses am Eingang des Flipflops, in ihren Ausgangszustand zurückversetzt.

Die Slicearchitektur der Xilinx 7-Series FPGAs erlaubt es nicht, mehrere Elemente des Slices mit verschiedenen Taktsignalen zu betreiben (vgl. Abschnitt 2.2). Im ersten Slice der Tapped Delay Line werden die Ausgänge des CARRY4-Elements nicht abgetastet, da ein Flipflop mit dem Messimpuls als Taktsignal beschaltet ist.



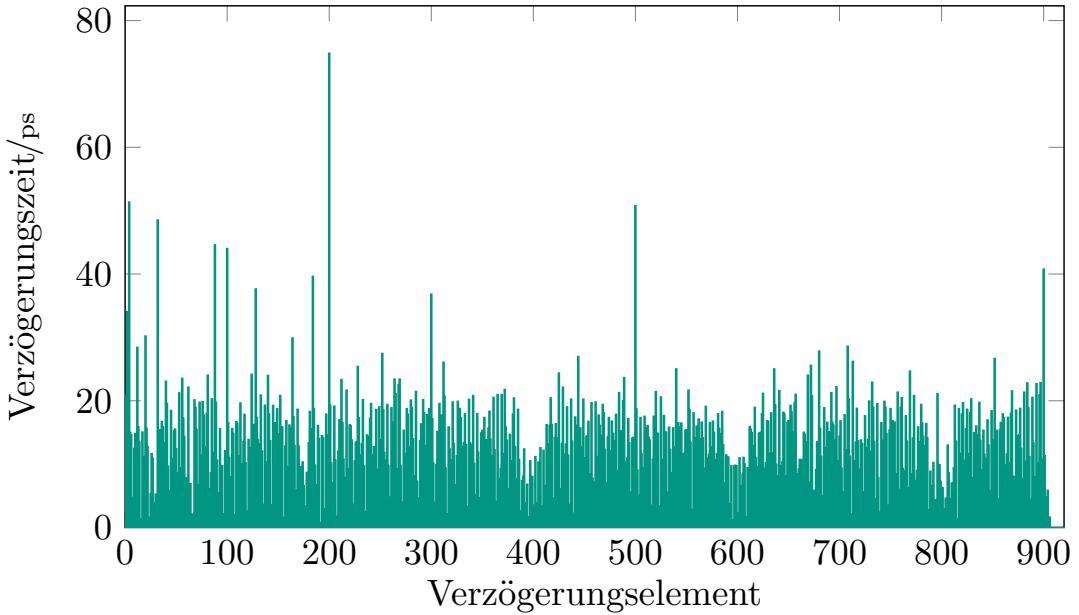
**Abbildung 5.1:** Schematische Darstellung der Triggerstruktur der TDL. Der Ausgang des letzten Verzögerungselementes der TDL ist auf den asynchronen CLEAR-Port des Triggerflipflops zurückgekoppelt. Dies bewirkt ein Zurücksetzen der TDL in ihren Ausgangszustand bevor der Messimpuls vorüber ist.

### 5.1.2 Aufteilung der TDL und Multiphasenabtastung

Eine erste Implementierung des TDC-Moduls wird mit einem 100 MHz Systemtakt betrieben. Die notwendige Länge der TDL beträgt etwa 900 Verzögerungselemente, welche in 225 Slices untergebracht werden können. Eine CARRY4-Kette dieser Länge erstreckt sich vertikal über fünf Taktregionen. Zum Abtasten der TDL wird deshalb ein globales Taktnetz verwendet.

Die Betrachtung der, gemäß Abschnitt 4.2 ermittelten, Verzögerungszeiten der einzelnen Elemente (Abbildung 5.2) zeigt hohe Delays und Einbrüche im Histogramm an den Positionen 200, 400, 600 und 800, welche den Grenzen der Taktregionen entsprechen, sowie an den Positionen 100, 300, 500 und 900, den Positionen der HROW in der Mitte der Taktregionen. Dies wird auf einen Taktversatz des Abtasttaktes aufgrund unterschiedlicher Länge der Taktverteilungspfade zu den Flipflops zurückgeführt [13].

Um dieses Phänomen zu vermeiden, soll die gesamte TDL in einer halben Taktregion untergebracht werden. Aus den gemachten Beobachtungen zur Propagationszeit eines Signals durch eine TDL, wird die hierfür nach Gleichung (5.1) notwendige Abtastfrequenz  $f_s$  auf etwa 800MHz geschätzt. Eine Datenverarbeitung mit dieser Frequenz ist auf dem FPGA nicht möglich.



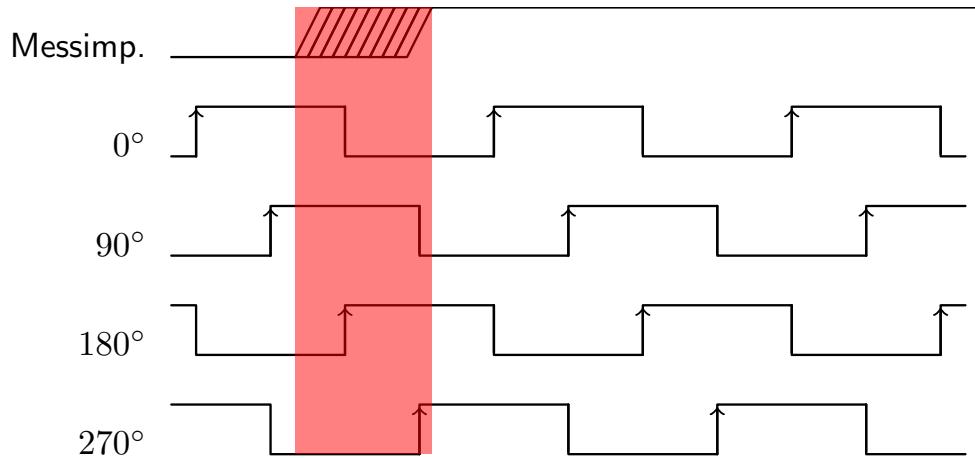
**Abbildung 5.2:** Ermittelte Propagationszeiten der Verzögerungselemente einer ersten TDC-Implementierung mit 100 MHz Abtasttakt. An den Positionen der Grenzen der Taktregionen, sowie an den Positionen der HROW sind lange Delays und Einbrüche im Histogramm zu erkennen. Die Ursache hierfür sind Taktverschiebungen des Abtasttaktes aufgrund unterschiedlich langer Taktverteilungspfade.

Im Konzept der finalen Implementierung werden deshalb vier kurze, parallel angeordnete, TDL-Abschnitte mit der Länge einer halben Taktregion eingesetzt. Dies entspricht 100 Verzögerungselementen, von denen 96 abgetastet werden. Die Abschnitte werden mit um jeweils  $90^\circ$  phasenverschobenen Taktsignalen abgetastet. Somit wird sichergestellt, dass über die gesamte Taktperiode eine Messimpulsflanke detektiert werden kann. Das Timingdiagramm in Abbildung 5.3 macht dies deutlich. Die Abtastfrequenz wird auf 260MHz erhöht. Sie ist so gewählt, dass die Propagationszeit des Messimpulses durch einen TDL-Abschnitt länger ist als eine viertel Taktperiode.

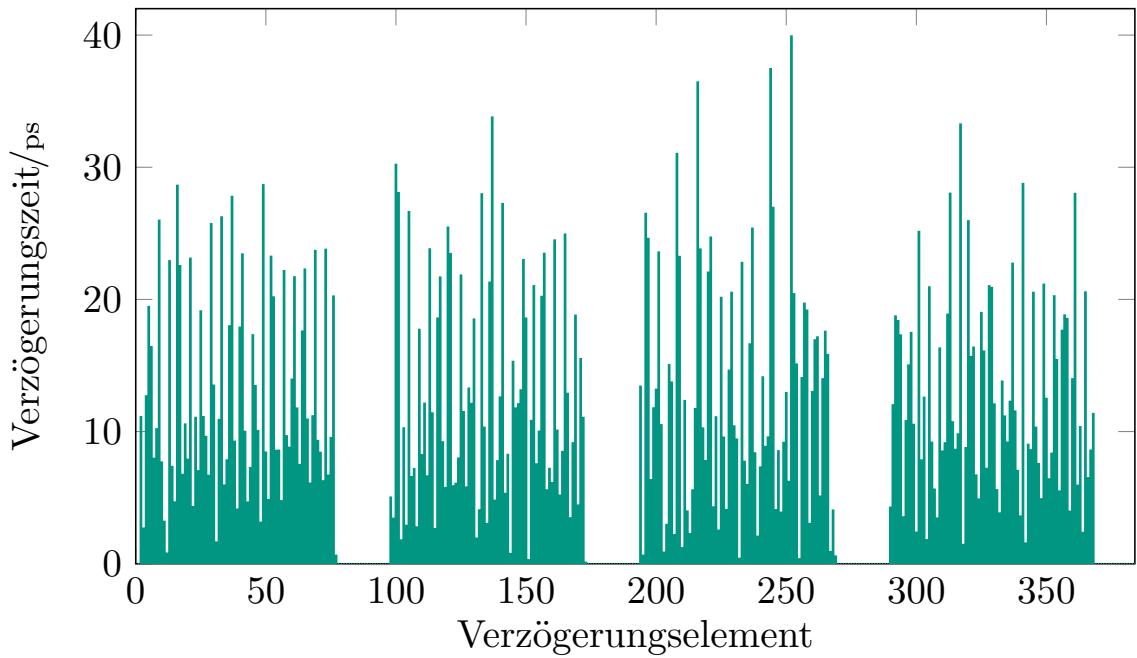
Im Histogramm der vier TDL-Abschnitte werden diese hintereinander angeordnet. Abbildung 5.4 zeigt ein solches gemeinsames Detektionshistogramm. Die Lücken zwischen den Abschnitten entstehen durch das Einhalten der Bedingung aus Gleichung (5.1).

## 5.2 Priority Encoder

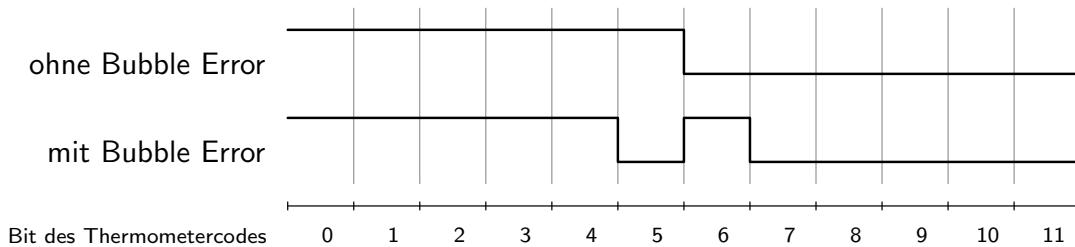
Jeder der vier Abschnitte der TDL generiert in jedem Taktzyklus ein thermometerkodiertes, 96 Bit breites Datenwort (s. Abschnitt 4.1). Die Messimpulsflankenposition im Thermometercode soll in eine binäre Repräsentation überführt werden. Ein ähnliches Ziel



**Abbildung 5.3:** Timingdiagramm der vier TDL-Abschnitte. Die Abtastfrequenz ist so gewählt, dass die Zeit, während der Messimpulsflanke durch die TDL propagiert (rot markierter Bereich), mindestens eine steigenden Taktflanke, also einen Abtastvorgang, enthält.



**Abbildung 5.4:** Ermittelte Propagationszeiten der Verzögerungselemente einer Vier-Phasen-TDL der finalen Implementierung mit einem 260 MHz Abtasttakt.



**Abbildung 5.5:** Thermometercode mit und ohne Bubble Error.

wird bei der Detektion des ersten Bits einer Bitfolge zur Priorisierung von Interruptrequests eines Microcontrollers verfolgt, weswegen sich die Bezeichnung „Priority Encoder“ eingebürgert hat.

### 5.2.1 Bubble Errors

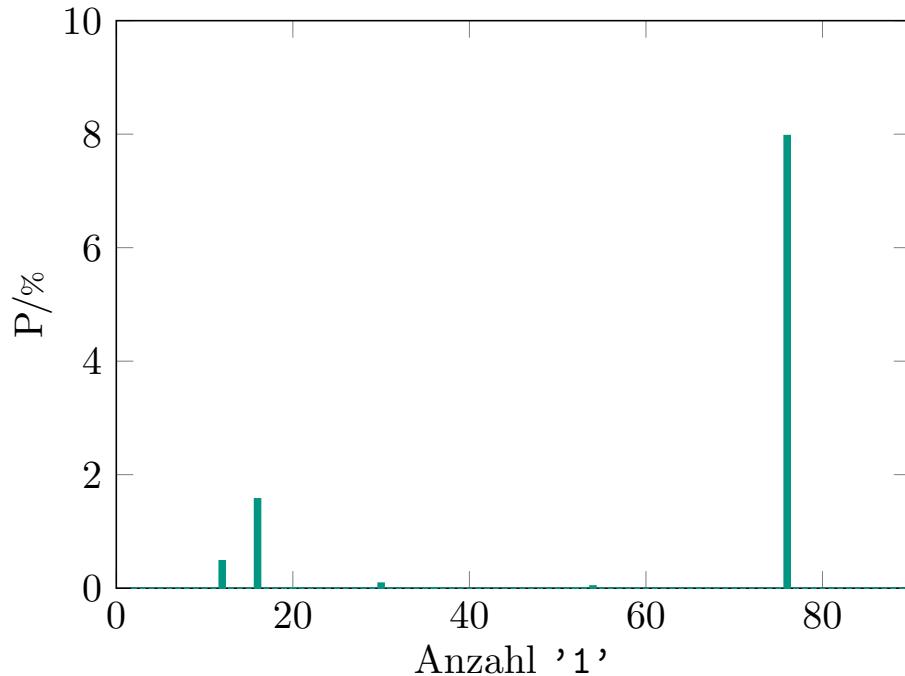
Beim Abtasten der Delay Line treten in der Umgebung der Impulsflanke „Blasen“ im Thermometercode auf. Die Ursache des Auftretens dieser „Bubble Errors“ kann nur vermutet werden, da die Chipstruktur des FPGAs nicht öffentlich bekannt ist. Abbildung 5.5 zeigt einen beispielhaften Thermometercode mit und ohne Bubble Error.

Eine mögliche Erklärung ist, dass die Laufzeiten zwischen den einzelnen CARRY4-Ausgängen und den Flipflops, sowie die Laufzeiten der Taktverteilung an die Flipflops innerhalb eines Slices ungleich sind [17]. Nach den Definitionen in Abschnitt 4.2, ist das Phänomen dann als statisch zu klassifizieren. Ein weiterer möglicher Grund ist das nicht vorhersehbare Verhalten eines Flipflops, wenn dessen Setup- und Holdzeiten nicht eingehalten werden [18]. Diese bedeutete ein nicht deterministisches, dynamisches, Auftreten dieser Fehler.

Die Art der Bubble Errors, statisch oder dynamisch, ist für die Wahl der Kodierungsme thode, welche im Priority Encoder eingesetzt wird, ausschlaggebend. Im Folgenden soll sie experimentell festgestellt werden.

### Experimentelle Charakterisierung

Der 96 Bit breite Thermoetercode des ersten Abschnittes der Tapped Delay Line wird einem Debugmodul übergeben. Dieses zählt die Gesamtanzahl der Einsen in der Bitfolge, ohne deren Position zu berücksichtigen. Entspricht diese Anzahl einem festgeleuten Wert, wird der Thermometercode über eine serielle Schnittstelle an einen angeschlossenen PC übertragen und zur späteren Analyse gespeichert. So lassen sich eine hohe Anzahl an Thermometercodes mit der selben Anzahl an Einsen aufzeichnen.



**Abbildung 5.6:** Prozentuales Auftreten von Abweichungen des abgetasteten Thermometercodes von den präferierten Mustern über der Gesamtzahl der im Muster vorhandenen '1'.

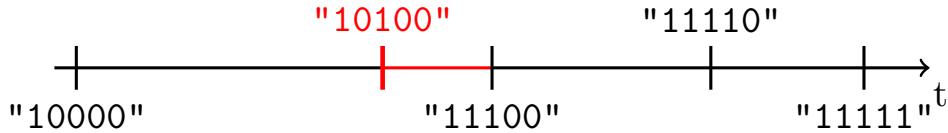
Automatisiert werden zu jeder möglichen Gesamtanzahl an Einsen,  $10^5$  Bitfolgen des ersten TDL-Abschnittes aufgezeichnet. Durch einen Vergleich dieser Daten kann gefolgert werden, ob Bubble Errors zufällig oder deterministisch auftreten.

Es wird festgestellt, dass es zu jeder Anzahl von Einsen im Thermometercode eine präferierte Anordnung der Bitfolge gibt, von der nur selten Abweichungen auftreten.

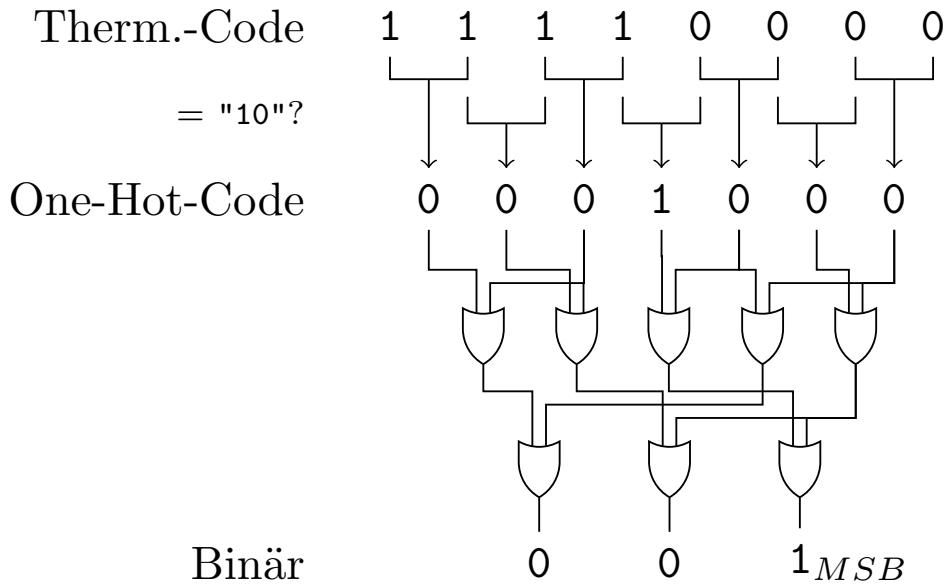
Thermometercodes mit 12 Einsen haben beispielsweise vorwiegend das Muster 0xFFFF. Nur in 0,48% aller Fälle wird ein anderes Muster, 0x2BFF, festgestellt. In Abbildung 5.6 ist die beobachtete Auftretenswahrscheinlichkeit solcher Abweichungen vom präferierten Muster für jede Gesamtzahl an Einsen im Thermometercode des ersten TDL-Abschnittes dargestellt.

Dies bedeutet, dass Bubble Errors nicht zufällig sind, sondern die einzelnen Bits des Thermometercodes im zeitlichen Verlauf in einer deterministischen Reihenfolge zu Eins werden (Beispiel Abbildung 5.7). Bubble Errors sind somit nach der Definition aus Abschnitt 4.2 als statische Fehler einzurufen.

Welche Auswirkung die festgestellte, statische Art der Bubble Errors auf die Kodierungsverfahren hat, wird in den folgenden Unterabschnitten gezeigt.



**Abbildung 5.7:** Zeitliche Änderung des Thermometercodes. Auch beim Auftreten eines Bubble Errors (rote Markierung), werden die Bits des Thermometercodes in einer festen Reihenfolge zu Eins.



**Abbildung 5.8:** Kodierung eines 8 Bit Thermometercodes mittels Kantenfilter. Die Impulsflankenposition im Thermometercode wird in einem One-Hot-Bitstring markiert und über einen Baum aus OR-Gattern binär kodiert.

### 5.2.2 Kodierung mittels Kantenfilter

Eine geläufige und einfache Methode zur binären Kodierung der Impulsflankenposition im Thermometercode, ist die Filterung mit einem eindimensionalen Kantenfilter.

Der Thermometercode wird nach dem Auftreten der Bitfolge "10" durchsucht. In einer sogenannten „One-Hot“-Bitfolge, in der nur ein Element Eins ist, wird das Auftreten des Musters markiert. Aus der One-Hot-Bitfolge wird mittels eines asynchronen Baums aus OR-Gattern die binär kodierte Impulsflankenposition generiert. In Abbildung 5.8 ist dies schematisch dargestellt.

Treten Bubble Errors im Thermometercode auf, werden mehrere Kanten im One-Hot-Code markiert, was zu einer fehlerhaften binären Kodierung führt. Dies ist in Tabelle 5.1, Spalte A veranschaulicht.

Um die Markierung mehrerer Kanten im One-Hot-Code zu vermeiden, muss ein längeres

Filter "100" oder gar "1000" gewählt werden. Es wird dann wieder nur eine Flanke detektiert (Tabelle 5.1, Spalte B).

In Abschnitt 5.2.1 wurde festgestellt, dass Bubble Errors vorwiegend statische Fehler sind und somit die zeitliche Reihenfolge, in der die Bits des Thermometercodes zu Eins werden, deterministisch ist. Deshalb geht durch das Verwenden eines langen Kanntenfilters relevante Information verloren (rot markierter Bereich des Zeitstrahls in Abbildung 5.7). Die Methode ist damit für die Anwendung in einem TDC weniger gut geeignet.

	ohne Bubble Error	mit Bubble Error	
		(A) Filter "10"	(B) Filter "100"
Therm.-Code	'111000000'	'110100000'	'110100000'
One-Hot	'00100000'	'01010000'	'00010000'
Binär	'0011' $_b = 3_d$	'0110' $_b = 6_d$	'0100' $_b = 4_d$

**Tabelle 5.1:** Auswirkungen von Bubble Errors auf das Umwandlungsergebnis beim Verwenden der eindimensionalen Kantenfilter (A) '10' und (B) '100'.

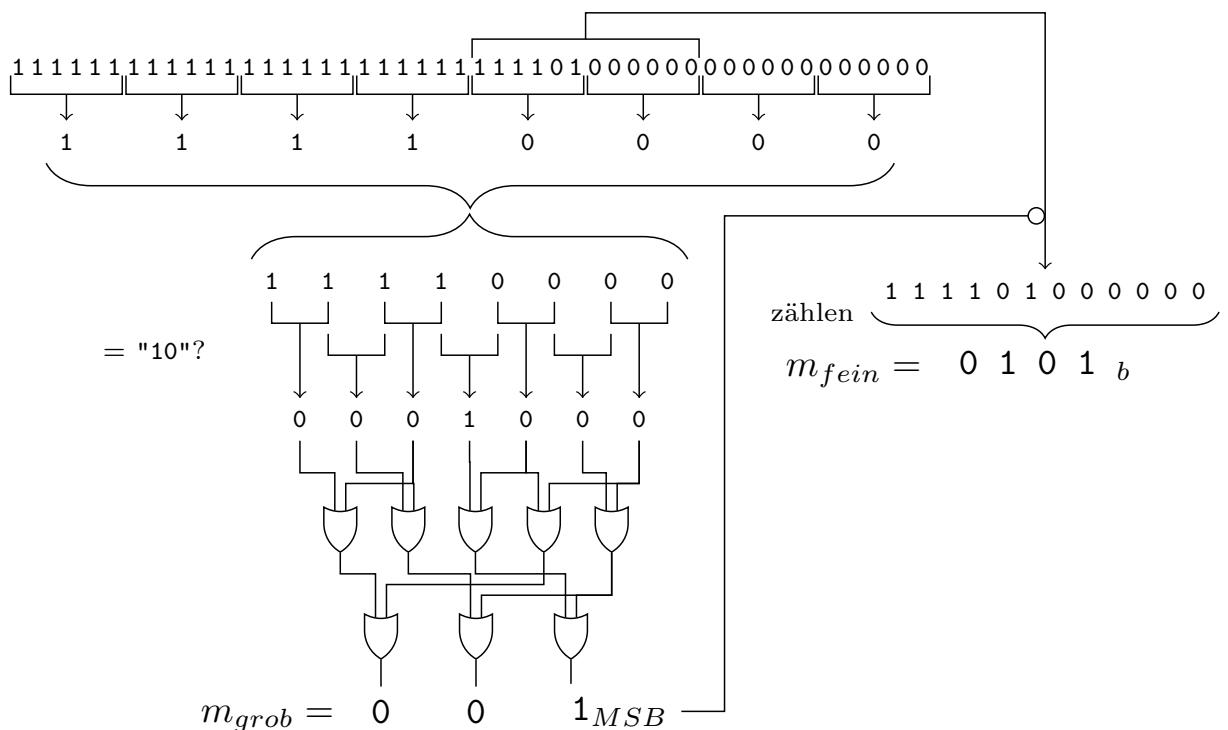
### 5.2.3 Kodierung durch Zählen der Einsen

Anstatt der Flankendetektion mit einem Kantenfilter, werden im implementierten Priority Encoder die Einsen der abgetasteten Bitfolge gezählt. Die Reihenfolge ihres Auftretens ist somit irrelevant. Um eine laufzeiteffiziente Implementierung zu gewährleisten, wird die Zählung nur im Umfeld der Messimpulsflanke durchgeführt, in dem Bubble Errors auftreten können.

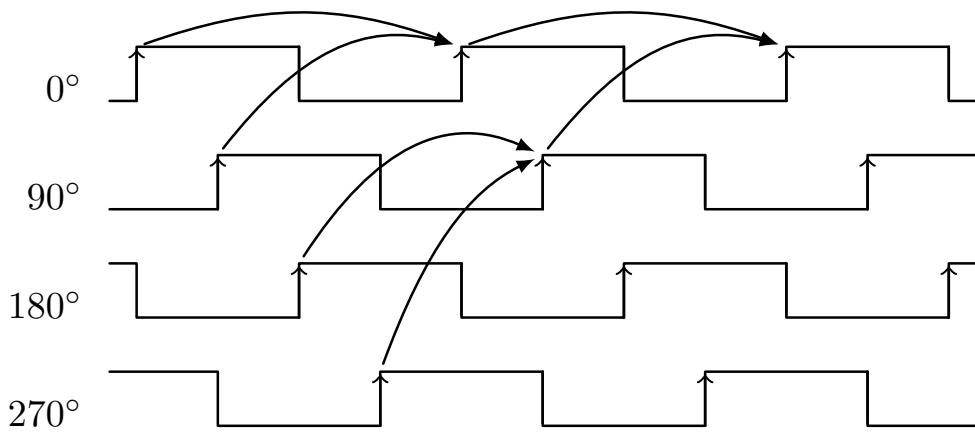
In einem ersten Schritt wird die Position der Messimpulsflanke grob bestimmt. Dies geschieht durch den Vergleich von, jeweils sechs Bit breiten, Abschnitten mit der Bitfolge "111111". Es entsteht ein kleinerer  $\frac{96}{6} = 16$  Bit breiter Thermometercode. Mit der Kantenfiltermethode aus Abschnitt 5.2.2 wird die grobe Position  $m_{\text{grob}}$  binär kodiert.

Auf Basis der groben Position wird im zweiten Schritt der entsprechende, zwölf Bit breite, Ausschnitt des Thermometercodes, in dem die Impulsflanke auftritt, durch einen Baum aus Multiplexern in ein „Zählregister“ übertragen. Die Einsen des Zählregisterinhalts  $m_{\text{fein}}$  werden letztendlich in einem dritten Schritt gezählt. Die binär kodierte Anzahl der Einsen im Thermometercode  $m$  setzt sich aus der groben Position des Übergangs und den gezählten Einsen zusammen (Gleichung (5.2)). Der Vorgang ist in Abbildung 5.9 skizziert.

$$m = m_{\text{grob}} \cdot 6 + m_{\text{fein}} \quad (5.2)$$



**Abbildung 5.9:** Binäres kodieren der Impulsflankenposition durch zählen der Einsen. Die grobe Taktflankenposition wird durch abschnittsweises Vergleichen mit "111111" und dem Kantenfilter aus Abschnitt 5.2.2 bestimmt. Auf Basis dessen, werden die Einsen eines entsprechenden, zwölf Bit breiten, Ausschnittes des Thermometercode gezählt. Am gewählten Beispiel ergibt sich die Impulsflankenposition zu  $m_{grob} \cdot 6 + m_{fein} = 29$ .



**Abbildung 5.10:** Phasensynchronisation der Daten der vier TDL-Abschnitte. Die Synchronisation erfolgt in zwei Schritten um Verletzungen der Setupzeiten der Flipflops zu vermeiden.

### 5.2.4 Phasensynchronisation der Daten

Um die Daten der vier Abschnitte der Tapped Delay Line weiter verarbeiten zu können, werden sie auf eine gemeinsame Taktphase ( $0^\circ$ ) überführt. Dies geschieht nach der Verarbeitung durch den Priority Encoder. Anstatt des 96 Bit breiten Thermometercodes der TDL, müssen so nur noch die Datenbits der Binärrepräsentation der Messimpulsflankenposition synchronisiert werden.

Der Versuch die Daten der  $270^\circ$ -Phase in einen mit  $0^\circ$  getakteten Flipflop zu übernehmen führt zur Verletzung der einzuhaltenen Setup- und Holdzeiten. Um die Daten aller Abschnitte einer Taktperiode zusammen zu halten, erfolgt die Datensynchronisation aller TDL-Abschnitte in zwei Schritten. Die Daten der Phasen  $180^\circ$  und  $270^\circ$  werden zunächst in  $90^\circ$ -Registern zwischengespeichert. Die Übernahme in die  $0^\circ$ -Register kann dann zum nächsten Takt erfolgen. Das Timingdiagramm in Abbildung 5.10 stellt dies anschaulich dar.

## 5.3 Datenfusion und Kalibrierungsmanagement

Im Modul `d1_sync` werden die, nun phasensynchron vorliegenden, binär kodierten, Messimpulsflankenpositionen der vier Abschnitte der TDL zusammengeführt, sowie auf das Umschalten zwischen Mess- und Kalibrierungssignal reagiert.

### 5.3.1 Datenfusion

Da die Propagationszeit durch einen TDL-Abschnitt länger ist als eine viertel Taktperiode (s. Abschnitt 5.1), kann es vorkommen, dass die selbe Impulsflanke in zwei phasenadjazenten Abschnitten zugleich detektiert wird. Die Daten beider Abschnitte liefern die selbe Information. Ein Datum kann verworfen werden.

Aus Abbildung 5.10 ist ersichtlich, dass die zum selben Takt anliegenden, phasensynchronisierten Daten in der zeitlichen Reihenfolge  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  und  $270^\circ$  abgetastet wurden. Wird die selbe Impulsflanke in zwei TDL-Abschnitten detektiert, ist sie in jenem Abschnitt weniger weit fortgeschritten, in welchem sie zeitlich zuerst abgetastet wurde. Dieses Datum wird zur weiteren Datenverarbeitung priorisiert, das Andere verworfen.

Die Zuordnung der priorisierten Daten zu ihrem entsprechenden Abschnitt im Kalibrierungshistogramm erfolgt durch Addition eines Vielfachen der Länge eines TDL-Abschnittes (96). Dadurch entstehen die charakteristischen Lücken im Detektionshistogramm (vgl. Abschnitt 5.1). Die resultierende Gesamtposition wird als 9-Bit-Binärwort ausgegeben.

### 5.3.2 Kalibrierungsmanagement

Das Umschalten zwischen Mess- und Kalibrierungssignal wird über ein externes Signal `calib_en` gesteuert, auf welches der TDC keinen Einfluss hat. Dies erlaubt es übergeordneten Funktionen, welche die Zeitdaten des TDC verarbeiten, die Kalibrierungsmessungen in einem Zeitraum durchzuführen, in dem keine Messimpulse zu erwarten sind.

Jedoch kennen weder die übergeordneten Funktionen, noch der TDC den momentanen Zustand des Ringoszillatorsignals, welches zur Kalibrierung verwendet wird. Ist dieses beim Setzen von `calib_en` bereits Eins, schaltet der Triggerflipflop der TDL (vgl. Abschnitt 5.1.1) sofort. Die nun durch die TDL propagierende Testimpulsflanke ist nicht unabhängig vom Systemtakt und verfälscht somit das Detektionshistogramm.

Wird ein Umschaltvorgangs zwischen Kalibrierungs- und Messsignal festgestellt, wird eine Totzeit von acht Taktperioden initialisiert, während der keine Daten ausgegeben werden. Diese Zeit ist so gewählt, dass etwaige ungültige Messwerte die bisherigen Verarbeitungsschritte durchlaufen haben und somit verworfen werden. Die Information, ob ein Kalibrierungs- oder Messimpuls gemessen wurde, wird gemeinsam mit den Daten ausgegeben.

## 5.4 Histogramm und Look Up Table

Das Modul LUT ist ein Zustandsautomat, welcher das Erstellen des Detektionshistogramms aus den Messdaten der Kalibrierungsimpulse und die Generierung des Look-Up-Tables verwaltet. In ihm ist die Abbildungsvorschrift, welche jeder gemessenen Signalfankenposition einen Zeitwert zuordnet, gespeichert. Eingang des Moduls ist die neun Bit breite binäre Signalfankenposition, Ausgang ein 18 Bit breiter Zweitwert.

Histogramm und LUT sind als `SIMPLE_DUAL_PORT` Block-RAM-Elemente implementiert. Sie zeichnen sich über zwei unabhängige, deidizierte Ports für Lese- und Schreiboperationen aus. Sie speichern 18 Bit breite Datenworte mit einer Tiefe von 1024 Elementen (10 Bit Adresse).

Wird dem Modul eine Kalibrierungsmessung übergeben, liest es den Speicherinhalt des Histogramms an der entsprechenden Speicheradresse aus, inkrementiert diesen und schreibt ihn an die selbe Adresse zurück.

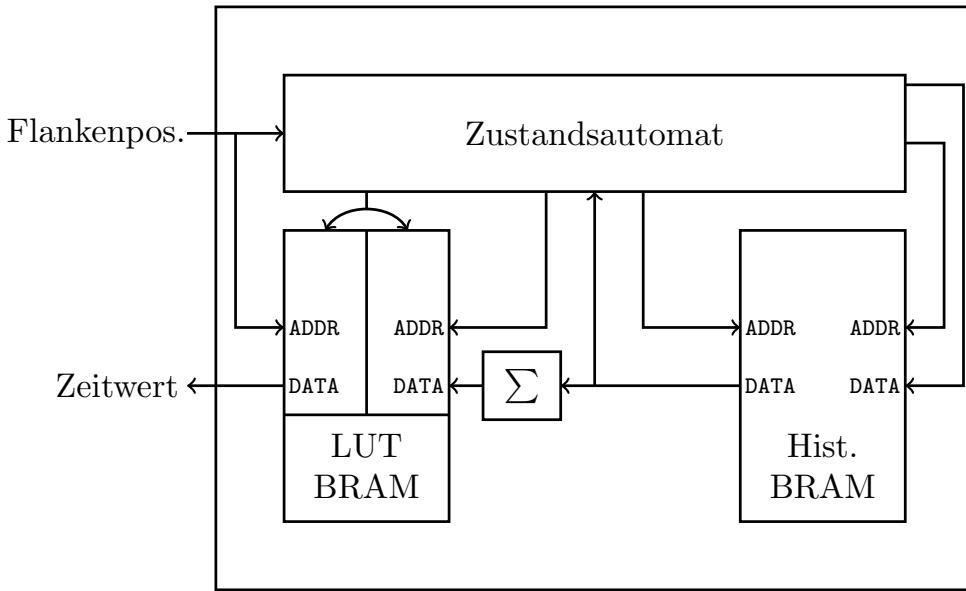
Sobald dies  $2^{18}$  mal geschehen ist, wird das Histogramm sequenziell ausgelesen und zu den Einträgen des LUT aufsummiert (vgl. Abschnitt 4.2).

Das Block-RAM-Element des Look-Up-Tables ist über das höchstwertigste Bit der Speicheradresse in zwei „Speicherseiten“ aufgeteilt. Während die eine Seite, im Zuge der periodischen Rekalibrierung, mit neuen Daten beschrieben wird, steht der alte LUT noch auf der anderen Speicherseite zur Verfügung. Ist der Generierungsvorgang abgeschlossen, wird das höchstwertigste Bit der Speicheradresse invertiert. Zu keinem Zeitpunkt stehen keine LUT-Daten zur Verfügung. Die in Abschnitt 1.2 geforderte ununterbrochene Verfügbarkeit des TDC kann somit garantiert werden. Der Aufbau des LUT-Moduls ist in Abbildung 5.11 schematisch dargestellt.

Nach dem Systemstart muss zunächst das erste Kalibrierungshistogramm mit  $2^{18}$  Testmessungen aufgebaut werden. Die dafür notwendige Zeit ist von der Oszillationsfrequenz des Ringoszillators abhängig, welcher die Kalibrierungsimpulse erzeugt.

## 5.5 Kanalmodul

Im Kanalmodul werden alle, für das Messen der Zeit von einer Messimpulsflanke bis zur nächsten Taktflanke des Systemtaktes notwendigen, Module instanziert. Die vom LUT-Modul ausgegebenen Zeitdaten  $t_{LUT}$  werden mit der Zeit einer Taktperiode in Picosekunden multipliziert. Mit der Division durch die Anzahl der Kalibrierungsmessungen ergibt



**Abbildung 5.11:** Struktureller Aufbau des LUT-Moduls. Ein Zustandsautomat verwaltet das Detektionshistogramm und den LUT. Durch Alternieren des ersten Addressbits des Speicherblocks bei der Neuberechnung des LUT stehen zu jeder Zeit gültige Werte zur Verfügung.

sich die Zeit  $t_{\text{Start}/\text{Stop}}$  zwischen der Messimpuls- und Taktflanke des Systemtaktes:

$$t_{\text{Start}/\text{Stop}} = t_{\text{LUT}} \cdot \frac{t_{\text{clk}}}{2^{18}} \quad (5.3)$$

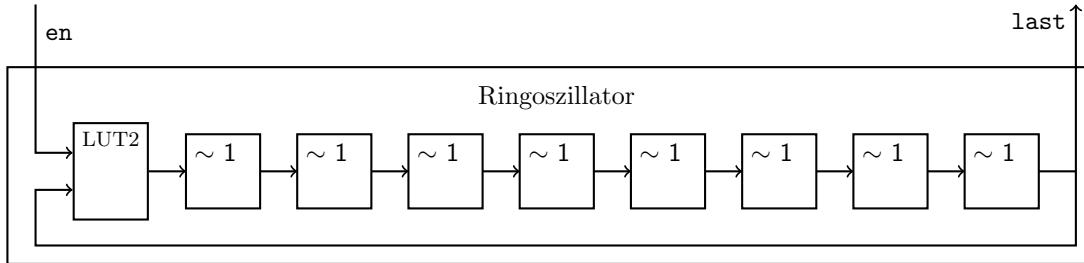
Auf Basis des Initialisierungszustandes des LUT und der Information, ob es sich um Daten eines Mess- oder Kalibrierungsimpulses handelt, wird entschieden, ob der berechnete Zeitwert gültig ist und ausgegeben werden kann.

## 5.6 Ringoszillator

Der Ringoszillator generiert die, für die Kalibrierung der Tapped Delay Line notwendigen, vom Systemtakt statistisch unabhängigen, Kalibrierungsimpulse (vgl. Abschnitt 4.2). Er besteht aus einer, über Parameter anpassbaren, ungeraden Anzahl an Invertern. Diese sind über das sogenannte *Gate* als Ring geschaltet. Das Gate ist als LUT2-Element instanziert und hat die Wahrheitstabelle aus Tabelle 5.2. Bei gesetztem Enablesignal `en` wird der Ausgang des letzten Invertorelements `last` invertiert wieder in die Inverterkette gegeben. Eine periodische, asynchrone Welle propagiert dann durch das System. Diese Struktur nennt man auch *Gated Ring Oscillator* (GRO). Sie ist in Abbildung 5.12 schematisch dargestellt. Das Signal `last` ist auch das Ausgangssignal des Ringoszillators.

en	last	
0	0	0
0	1	0
1	0	1
1	1	0

**Tabelle 5.2:** Wahrheitstabelle des LUT2-Gates des Ringoszillators. Bei gesetztem Enablesignal invertiert es den Ausgang des letzten Inverterelements und erwirkt so, dass eine periodische, asynchrone Welle durch den Ringoszillator propagiert.



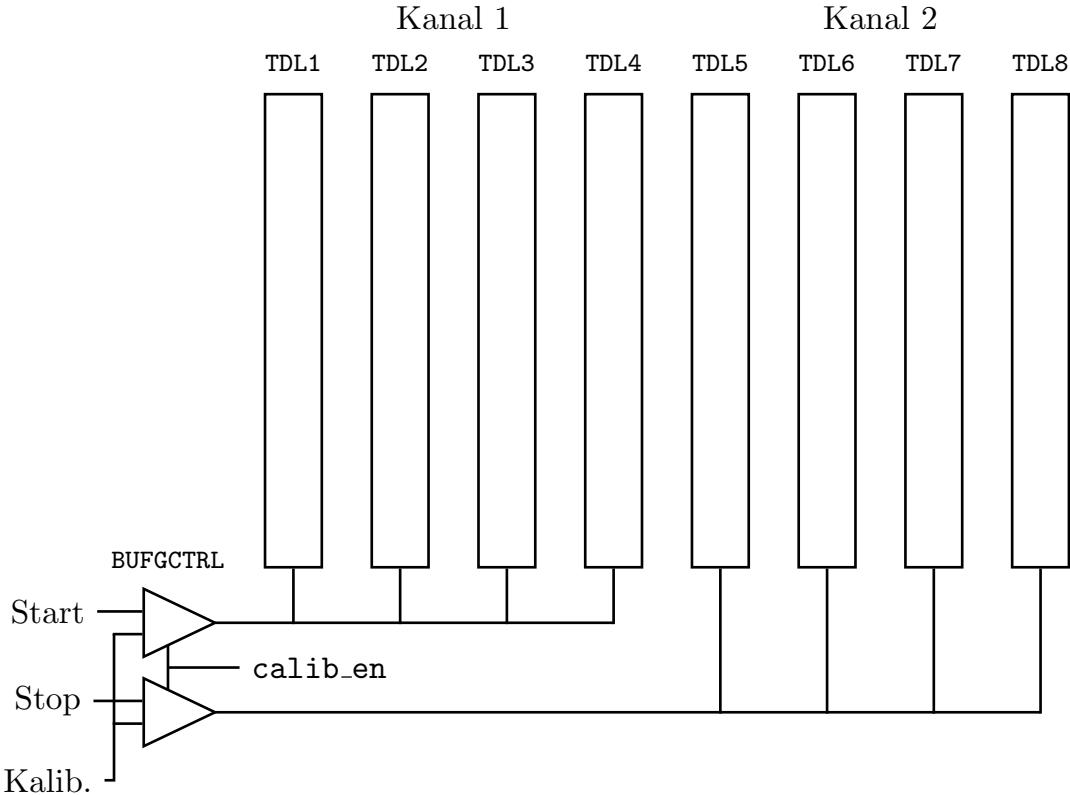
**Abbildung 5.12:** Struktureller Aufbau des Ringoszillators. Ist  $\text{en} = 1$ , so invertiert das LUT2-Element den Wert von  $\text{last}$ . Eine asynchrone Welle propagiert durch den Ringoszillator.

Initial ist es wichtig, den Ringoszillator für mindestens die Zeit einer halben Oszillationsperiode zurückzusetzen ( $\text{en} = 0$ ), um einen bekannten Ausgangszustand herzustellen. Wird die Resetdauer nicht eingehalten, kann es passieren, dass mehrere Signalfanken durch die asynchrone Schleife propagieren. Dies kann unvorhersehbare Auswirkungen auf den Kalibrierungsvorgang und somit auf die Messergebnisse haben.

Die Länge des Ringoszillators sollte so gewählt werden, dass er ein möglichst hochfrequentes Signal für eine schnelle Kalibrierung ausgibt. Frequenzen über einem Drittel der 260 MHz des Systemtaktes sind jedoch nicht sinnvoll, da die Inkrementierung des Detektionshistogramms (vgl. Abschnitt 5.4) drei Taktzyklen benötigt. Während dieser Zeit werden weitere Kalibrierungsmessungen ignoriert. Für die getestete Implementierung wird ein Ringoszillator mit einer Länge von 55 Elementen gewählt.

## 5.7 Toplevelmodul TDC

Das Modul TDC ist das funktionale Toplevelmodul des Designs. Es instanziert zwei Kanäle, jeweils einen für das Start- und Stoppsignal, den Ringoszillator dessen Signal zur Kalibrierung der TDLs verwendet wird und die Multiplexer, welche zwischen Mess- und Kalibrierungssignalen umschalten.



**Abbildung 5.13:** Das Umschalten zwischen Mess- und Kalibrierungssignal erfolgt mittels zweier BUFGCTRL-Taktmultiplexer. Die Steuerung erfolgt mit dem Signal `calib_en` durch die dem TDC übergeordnete Logik.

Zum Umschalten zwischen Mess- und Kalibrierungssignal werden BUFGCTRL-Taktmultiplexer zweckentfremdet. Zusatzfunktionen, wie die Phasenabgleichung der Signale zur Vermeidung von Glitches, sind zu Gunsten eines schnellen Wechsels zwischen den Signalen abgeschaltet. Diese Struktur ist in Abbildung 5.13 dargestellt.

Weiterhin werden im TDC-Modul die Zeitdaten der beiden Kanäle, welche das Start- bzw. Stopsignal messen, zusammengeführt. Liegt ein valides Datum  $t_{\text{Start}}$  am Ausgang des Startkanals an, wird dieses gespeichert und ein binärer Zähler  $n_{\text{clk}}$  (vgl. Abschnitt 4.1) initialisiert, welcher die nachfolgenden Taktzyklen zählt. Beim Eintreffen eines Datums des Stopkanals wird der momentane Zählerstand, zusammen mit der Stopzeit  $t_{\text{Stop}}$ , gespeichert. Die Messwerte für Start- und Stopimpuls, sowie der Zählerstand werden dann der Berechnung des Messergebnisses gemäß Gleichung (4.2) zugeführt. Diese ist als Pipeline implementiert um die Verarbeitung schnell aufeinanderfolgender Messdaten des Stopkanals („Multishots“) zu ermöglichen. Das Verarbeiten einer weiteren Messung des Stopkanals ist bereits zum nächsten Takt möglich.

Der Zähler  $n_{\text{clk}}$  hat eine Breite von 10 Bit. Dadurch ergibt sich für die Zeitintervallmessung

ein Dynamikbereich von  $\frac{2^{10}}{260\text{MHz}} = 3,938\mu\text{s}$ .

Zusammen mit den Messergebnissen wird ein ID-Bit ausgegeben, welches mit jeder neuen Messung eines Startimpulses alterniert. Dies erlaubt es übergeordneten Funktionen, zwischen den Messergebnissen verschiedener Startimpulse zu unterscheiden.

Über einen 13 Bit breiten Dateneingang kann der Wert der Taktperiode des Systemtaktes in Picosekunden, welcher für die Berechnung der Zeitwerte verwendet wird, von der dem TDC übergeordneten Logik angepasst werden.

## 5.8 TDC-Wrapper

Die Zielplattform des TDC wird mit 100 MHz betrieben, wohingegen der TDC ein 260 MHz Taktsignal benötigt. Die für eine kohärente Datenübertragung über die Taktgrenze hinweg und die Synthese des 260 MHz Taktsignals, notwendigen FPGA-Elemente sind im Modul `tdc_wrapper` implementiert.

Die ein Bit breiten Signale `CALIB_EN`, welches den Kalibrierungsvorgang steuert, Reset-Signal `RST`, sowie `READY`, welches der übergeordneten Logik anzeigt, ob die initiale Kalibrierung des TDC abgeschlossen ist, sind nicht timingsensitiv. Sie werden über zwei, mit dem Zieltakt betriebene, Flipflops einsynchronisiert um Metastabile Zustände zu vermeiden [14].

Die Übertragung der Messergebnisse des TDC, sowie der Daten einer etwaigen Anpassung der Taktperiode erfordert mehr Aufwand um die Eineindeutigkeit der Daten zu gewährleisten. Sie ist mittels zweier, als `FIFO_DUALCLOCK_MACRO` instanziierten, Block-RAM-Elemente realisiert. Diese erlauben asynchrone Taktsignale an Lese- und Schreibport. Die FIFOs werden im *First Word Fall Through* Modus betrieben. Befindet sich ein valides Datum im FIFO liegt dies zeitgleich mit dem entsprechenden `EMPTY`-Signal, welches Anzeigt ob Daten im FIFO liegen oder nicht, am Leseausgang an. Beim Setzen des Read-Enable-Signals `RDEN` wird es verworfen und das nächste Datenwort liegt am Ausgang an oder `EMPTY` wird wieder gesetzt.

Im Wrapper ist zudem ein 3-Bit-Zähler implementiert, welcher bei einem eingehenden Resetsignal eine Resetzeit von acht Taktzyklen für den TDC garantiert. So wird vermieden, dass im Ringoszillator ungültige Signale entstehen (vgl. Abschnitt 5.6).

Die Taktsynthese erfolgt in einem Mixed Mode Clock Manager. Er gibt zwei, zueinander 90° phasenverschobene, 260 MHz Taktsignale aus, welche die Tapped Delay Lines und die TDC-Logik treiben.

## 5.9 Zusammenfassung

In Kapitel 5 wurden die einzelnen Module des Time to Digital Converter beschrieben. Ihre hierarchische Anordnung ist in Abbildung 5.14 dargestellt. Die Module übernehmen zusammenfassend folgende Aufgaben:

- **Tapped Delay Line (TDL)**: Instanziert einen 96 Bit langen Dleay-Line-Abschnitt. Die Ausgänge der Verzögerungselemente werden abgetastet und parallel ausgegeben.
- **Priority Encoder (PE)**: Kodiert die Impulsflankenposition im Thermometercode als binäre Zahl.
- **Datenfusion (dl\_sync)**: Zusammenführung und priorisierung der Daten der TDL-Abschnitte, sowie Differenzierung zwischen Kalibrierungs- und Messimpulsen.
- **Look Up Table (LUT)**: Verwalten des Detektionshistogramms und Generierung des Look-Up-Tables, welcher die Impulsflankenposition auf einen Zeitwert abbildet.
- **Kanal (Channel)**: Berechnet aus den Daten des LUT die zugehörige Zeit zwischen Messimpuls- und Taktflanke in Picosekunden.
- **Ringoszillator (RO)**: Generiert das, für die Kalibrierung notwendige, vom Systemtakt unabhängige Testsignal.
- **TDC Toplevel (TDC)**: Berechnet aus den Zeiten von Start- und Stopkanal, sowie dem Zählerstand eines taktsynchronen Zählers, das zwischen den beiden Messimpulsflanken vergangene Zeitintervall in Picosekunden.
- **TDC-Wrapper**: Bettet den TDC in die Zielplattform ein. Instanziert einen MMCM zur Synthese der vom TDC benötigten Taktsignale, sowie FIFOs, mit welchen sich Daten zwischen zwei asynchronen Takten austauschen lassen.

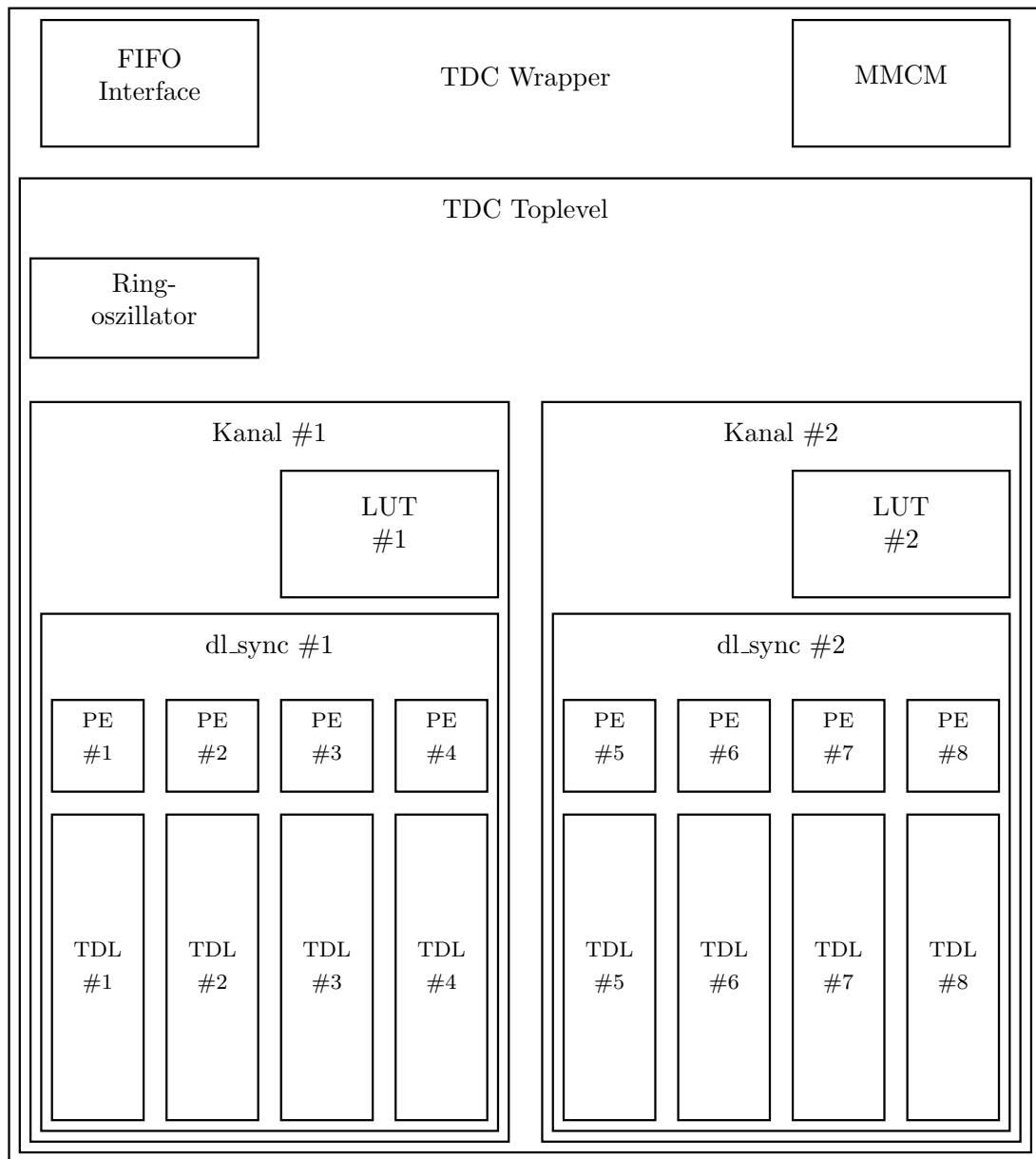


Abbildung 5.14: Hierarchische Anordnung der Module des TDC.

### 5.9.1 Interfaces

Die folgende Tabelle beschreibt, welche Daten zwischen den Modulen ausgetauscht werden.

Dateninterfaces der Module

TDL $\leftrightarrow$ PE

- 
- |   |   |
|---|---|
| → | <ul style="list-style-type: none"> <li>• <code>bins(95 downto 0)</code>: Abgetasteter Inhalt der Tapped Delay Line</li> </ul> |
| ← |   |

PE $\leftrightarrow$ dl\_sync

- 
- |   |   |
|---|---|
| → | <ul style="list-style-type: none"> <li>• <code>bin(6 downto 0)</code>: Binär kodierte Signalflankenposition im TDL-Abschnitt</li> </ul> |
| ← |   |

dl\_sync $\leftrightarrow$ Channel

- 
- |   |   |
|---|---|
| → | <ul style="list-style-type: none"> <li>• <code>bin(8 downto 0)</code>: Binäre Signalflankenposition über alle TDL-Abschnitte</li> <li>• <code>valid</code>: Zeigt gültige Daten auf dem Interface an</li> <li>• <code>calib_flag</code>: Zeigt an, ob ein Mess- oder Kalibrierungsimpuls auf dem Interface liegt</li> </ul> |
| ← | <ul style="list-style-type: none"> <li>• <code>calib_en</code>: Zustand des Signals, welches zwischen Mess- und Kalibrierungsimpuls umschaltet</li> </ul>   |

LUT $\leftrightarrow$ Channel

- 
- |   |   |
|---|---|
| → | <ul style="list-style-type: none"> <li>• <code>data(17 downto 0)</code>: Zeitwert des LUT-Speichers an der Adresse <code>bin</code></li> <li>• <code>init</code>: Zeigt an, dass die Generierung des initialen Look-Up-Table abgeschlossen ist</li> </ul>   |
| ← | <ul style="list-style-type: none"> <li>• <code>bin(9 downto 0)</code>: Binäre Signalflankenposition über alle TDL-Abschnitte</li> <li>• <code>valid</code>: Zeigt gültige Daten auf dem Interface an</li> <li>• <code>calib_flag</code>: Zeigt an, ob ein Mess- oder Kalibrierungsimpuls auf dem Interface liegt</li> </ul> |

Channel $\leftrightarrow$ TDC

---

→	<ul style="list-style-type: none"> <li>• <code>time(12 downto 0)</code>: Gemessene Zeit zwischen Messimpuls- und Taktflanke in Picosekunden</li> <li>• <code>valid</code>: Zeigt gültige Daten auf dem Interface an</li> <li>• <code>ready</code>: Zeigt an, ob der Look-Up-Table des Kanals initialisiert ist</li> </ul>
←	<ul style="list-style-type: none"> <li>• <code>calib_en</code>: Zustand des Signals, welches zwischen Mess- und Kalibrierungs-impuls umschaltet</li> <li>• <code>clk_period</code>: Zeit einer Taktperiode in Picosekunden</li> </ul>
	<b>TDC<math>\leftrightarrow</math>TDC_Wrapper</b>
→	<ul style="list-style-type: none"> <li>• <code>time(30 downto 0)</code>: Gemessene Zeit zwischen den Messimpulsflanken in Picosekunden</li> <li>• <code>valid</code>: Zeigt gültige Daten auf dem Interface an</li> <li>• <code>ready</code>: Zeigt an, ob der TDC bereit ist Zeitmessungen durchzuführen</li> <li>• <code>id</code>: Zwischen den Messungen alternierendes ID-Bit zur Trennung der Messungen unterschiedlicher Laserimpulse</li> </ul>
←	<ul style="list-style-type: none"> <li>• <code>calib_en</code>: Signal, welches zwischen Mess- und Kalibrierungsimpuls umschaltet</li> <li>• <code>period</code>: Zeit einer Taktperiode in Picosekunden</li> <li>• <code>period_valid</code>: Zeigt einen validen neuen Zeitwert für die Taktperiode an</li> </ul>
	<b>TDC_Wrapper<math>\leftrightarrow</math>Übergeordnete Logik</b>
→	<ul style="list-style-type: none"> <li>• FIFO Interface: Interface des FIFOs, mit dem die TDC-Daten über die Taktgrenze hinweg übertragen werden</li> <li>• <code>ready</code>: Zeigt an, ob der TDC bereit ist Zeitmessungen durchzuführen</li> </ul>
←	<ul style="list-style-type: none"> <li>• <code>calib_en</code>: Signal, welches zwischen Mess- und Kalibrierungsimpuls umschaltet</li> <li>• FIFO Interface: Interface des FIFOs, mit dem der Zeitwert für eine Taktperiode angepasst werden kann</li> <li>• <code>rst</code>: High Active Reset</li> </ul>

---

**Tabelle 5.3:** Interfaces zwischen den Modulen des Time to Digital Converter

# Kapitel 6

## Test und Ergebnisse

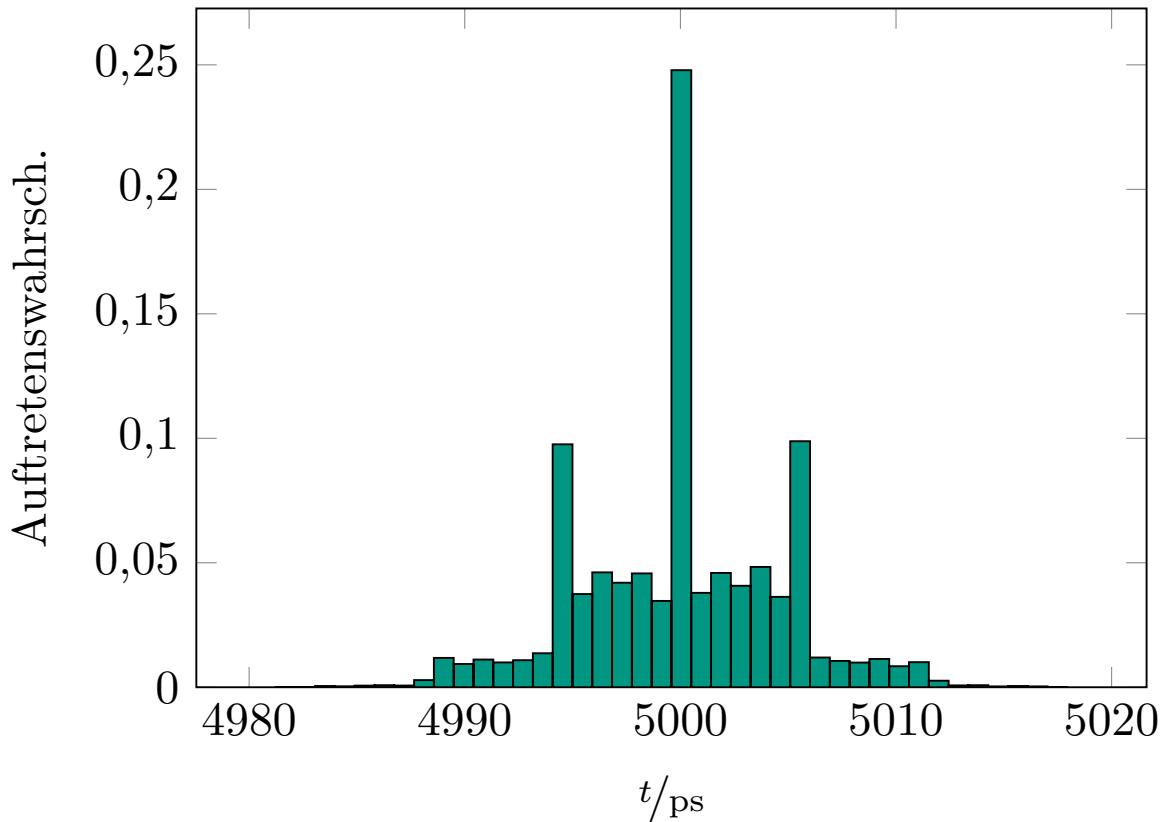
Dieses Kapitel beschreibt die Tests, welche mit dem entworfenen Time to Digital Converter auf der Testplattform, einem Xilinx KC705 Development Kit, durchgeführt werden. Die gewonnenen Erkenntnisse werden vorgestellt. In Abschnitt 6.1 werden die Signale der Taktgeber des Systems, sowie des Ringoszillators charakterisiert. Die Messgenauigkeit des TDC ist von ihrer Präzision linear abhängig. Die in Abschnitt 4.2 geforderte Unabhängigkeit des Ringoszillatorsignals vom Systemtakt wird experimentell bestätigt. Abschnitt 6.2 beschreibt den Quantisierungsfehler der einzelnen TDLs, sowie den gemeinsamen Fehler von Start- und Stopmessung. Letztlich werden in Abschnitt 6.3 die mit dem TDC gemachten Zeitmessungen beschrieben und bewertet.

### 6.1 Charakterisierung der Takt- und Ringoszillator-signale

Die Messpräzision des Time to Digital Converters hängt direkt von der Genauigkeit des Systemtaktes ab. In diesem Abschnitt werden die Charakteristika der Signale ermittelt, sowie die Unabhängigkeit von Systemtakt und Ringoszillatorsignal gezeigt.

#### 6.1.1 Messaufbau

Auf dem FPGA sind ein MMCM und ein Ringoszillator in einem Minimaldesign instanziert. Über drei SMA-Ports des KC705 Boards werden der native 200 MHz Systemtakt, der im MMCM synthetisierte 260 MHz Takt, sowie das Ausgangssignal des Ringoszillators ausgegeben. Die Signale werden mit einem Teledyne LeCory Wavesurfer 3024 Oszilloskop



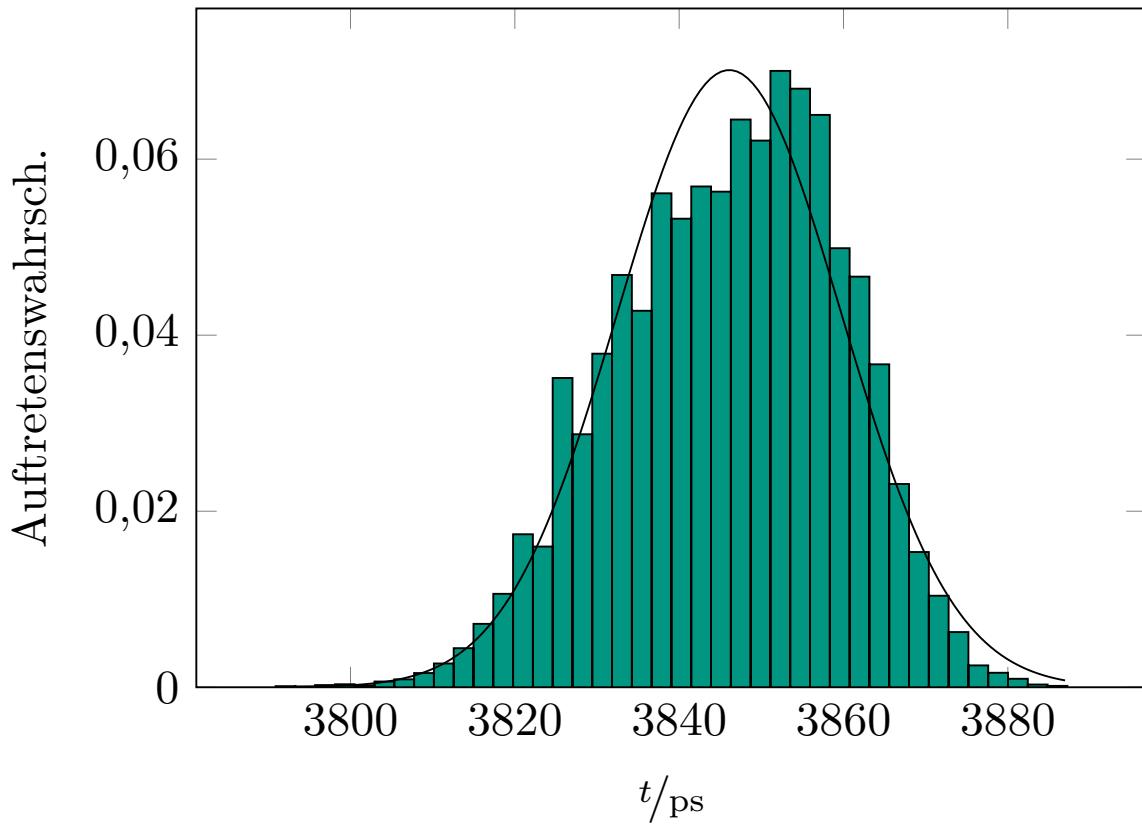
**Abbildung 6.1:** Histogramm des nativen 200 MHz Boardtaktes.  $\mu = 5,000\text{ns}$ ,  $\sigma = 4,677\text{ps}$ .

abschnittsweise mit 2 Gsps aufgezeichnet. Die einzelnen Abschnitte sind 500ns lang. Die Auswertung erfolgt in MATLAB.

### 6.1.2 Messergebnisse

#### 200 MHz nativer Boardtakt

Der 200 MHz Boardtakt wird von einem SiTime SiT9102AI-243N25E200 Oszillator (MEMS Resonator) synthetisiert. Dessen Jitterfehler (RMS) wird im Datenblatt als  $<2,5$  ps angegeben. Die Analyse von 38008 Taktperioden in Abschnitten von jeweils 500ns ergibt eine sehr akkurate mittlere Periode  $\bar{t}_{\text{clk}} = 5,000\text{ns}$  mit einer Standardabweichung (Jitter) von  $\sigma_t = 4,677\text{ps}$ . Das Histogramm der Taktperiode ist in Abbildung 6.1 dargestellt.



**Abbildung 6.2:** Histogramm des 260 MHz TDC-Taktes.  $\mu = 3,8461\text{ns}$ ,  $\sigma = 13,617\text{ps}$ .

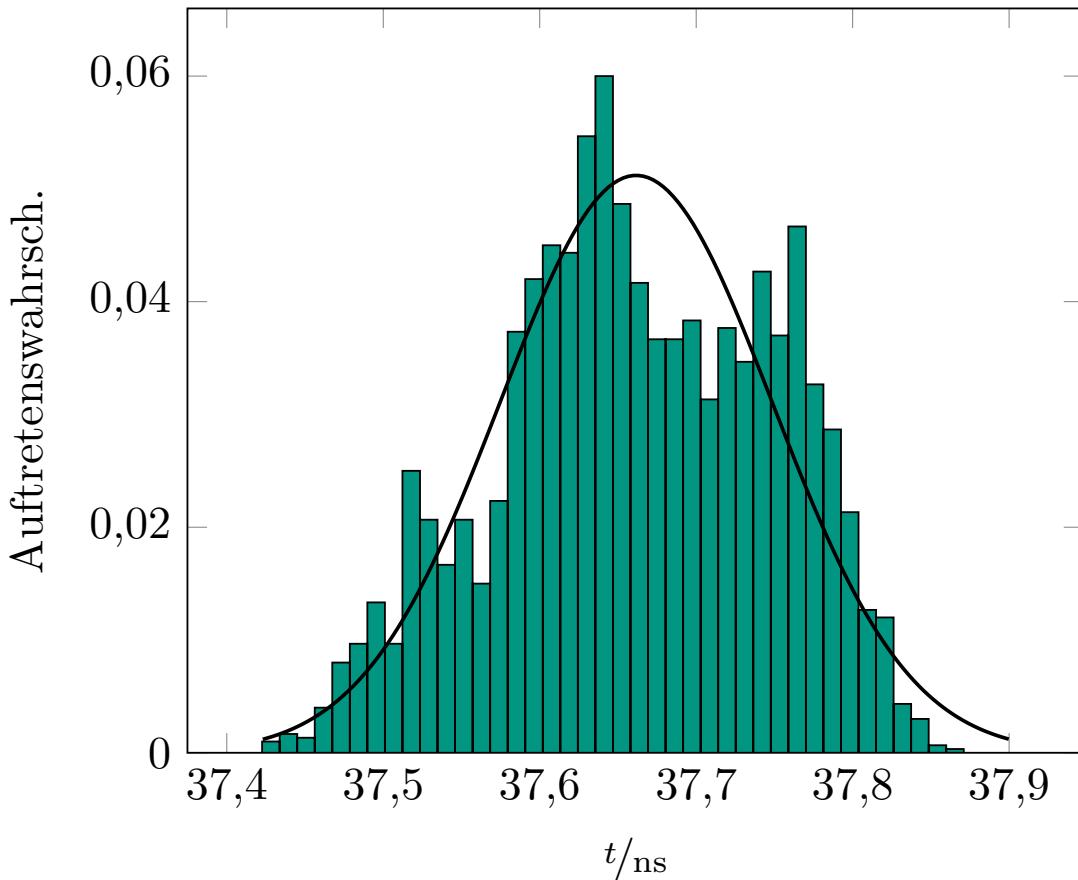
### 260 MHz MMCM-Takt

Der vom Mixed Mode Clock Manager aus dem nativen Boardtakt synthetisierte 260 MHz Systemtakt, mit dem der TDC betrieben wird, weist eine mittlere Taktperiode von  $\bar{t}_{clk} = 3,8461\text{ns}$  auf. Dies entspricht sehr genau dem erwarteten Wert ( $t_{clk,soll} = 3,84615\text{ns}$ ). Der mittlere Fehler (Standardabweichung) beträgt  $\sigma_t = 13,617\text{ps}$ . Im Vergleich zum nativen Takt ist das eine Verschlechterung um einen Faktor von etwa drei. Das Taktperiodenhistogramm ist in Abbildung 6.2 dargestellt.

### Ringoszillatorsignal

Der vermessene Ringoszillator mit einer Länge von 55 Inverterelementen wurde mittels `pblock` Constraint auf einer Fläche von 18 auf 22 Slices, abseits jedweder sonstiger Logik, im FPGA-Fabric platziert. Damit soll sicher gestellt werden, dass das Ergebnis reproduzierbar und weitestgehend unabhängig vom Place and Route Algorithmus ist.

Der Ringoszillatortakt weist eine mittlere Taktperiode von  $\bar{t}_{clk} = 37,66\text{ns}$  auf. Dies entspricht einer Frequenz von  $f = 26,55\text{MHz}$ . Seine Standardabweichung beträgt  $\sigma_t =$



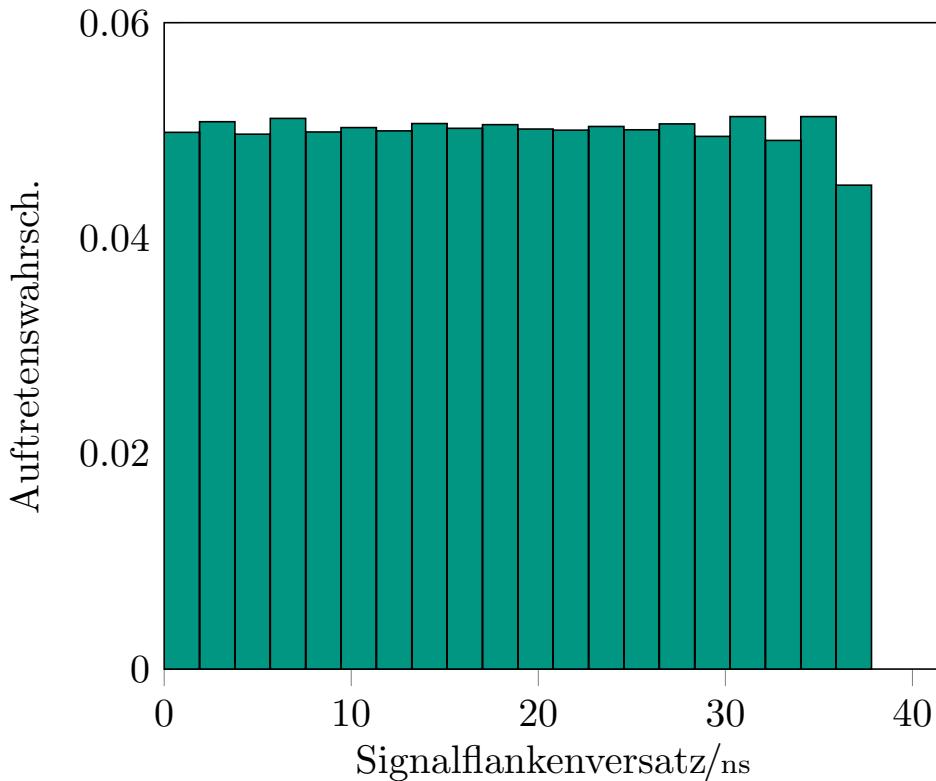
**Abbildung 6.3:** Histogramm des Ausgangssignals eines Ringoszillators mit einer Länge von 55 Inverterelementen. Er wurde mittels pblock Constraint auf eine Fläche von 18 auf 22 Slices eingeschränkt.  $\bar{t}_{\text{clk}} = 37,66 \text{ ns}$ ,  $\sigma = 87,03 \text{ ps}$ .

87,03ps. Das Histogramm ist in Abbildung 6.3 dargestellt.

### 6.1.3 Statistische Unabhängigkeit des Ringoszillatorsignals zum Systemtakt

Mit den gewonnenen Messdaten kann auch die Verteilung des Flankenversatzes zwischen Systemtakt und Ringoszillatorsignal ermittelt werden. Um eine Kalibrierung, wie in Abschnitt 4.2 beschrieben, zu erlauben, sollte der Flankenversatz möglichst gleichverteilt sein.

Zur Erstellung des Histogramms des Flankenversatzes in Abbildung 6.4 wurden 29366 Taktflanken des 260 MHz Systemtaktes relativ zu 3000 Flanken des Ringoszillatorsignals analysiert. Eine Gleichverteilung des Flankenversatzes ist klar ersichtlich. Der kleinere Wert des letzten Datenpunktes im Histogramm ist auf den Jitter von Ringoszillator und



**Abbildung 6.4:** Verteilung des Versatzes zwischen den Flanken des Ringoszillatorsignals zu den folgenden Taktflanken des Systemtaktes.

Systemtakt zurückzuführen.

## 6.2 Theoretisch ermittelte Qualitätsmerkmale

### 6.2.1 Quantisierungsstufen des TDC

Ein Qualitätsmerkmal von Analog-Digital-Wandlern, wie auch ein Time to Digital Converter einer ist, ist die Linearität der Quantisierung. Diese wird mittels der Differenziellen Nichtlinearität (DNL) und Integralen Nichtlinearität (INL) beschrieben.

Die Differenzielle Nichtlinearität für jede Quantisierungsstufe der Tapped Delay Line ist der um 0 zentrierte Quotient aus der realen und der idealen Höhe der Quantisierungsstufe (Gleichung (6.1)). Die ideale Höhe  $t_{q,ideal}$  einer Quantisierungsstufe ist der gesamte Messbereich  $t_{clk}$  geteilt durch die Anzahl an Quantisierungsstufen  $n_q$ .

$$\text{DNL}(i) = \frac{\text{LUT}(i+1) - \text{LUT}(i)}{t_{q,\text{ideal}}} - 1 \quad (6.1)$$

$$t_{q,\text{ideal}} = \frac{t_{\text{clk}}}{n_q} \quad (6.2)$$

Die Integrale Nichtlinearität (INL) ist die absolute Abweichung einer Quantisierungsstufe von der idealen, linearen Kennline. Sie kann als Summe aller vorherigen DNL-Werte berechnet werden:

$$\text{INL}(i) = \sum_{j=1}^i \text{DNL}(j) \quad (6.3)$$

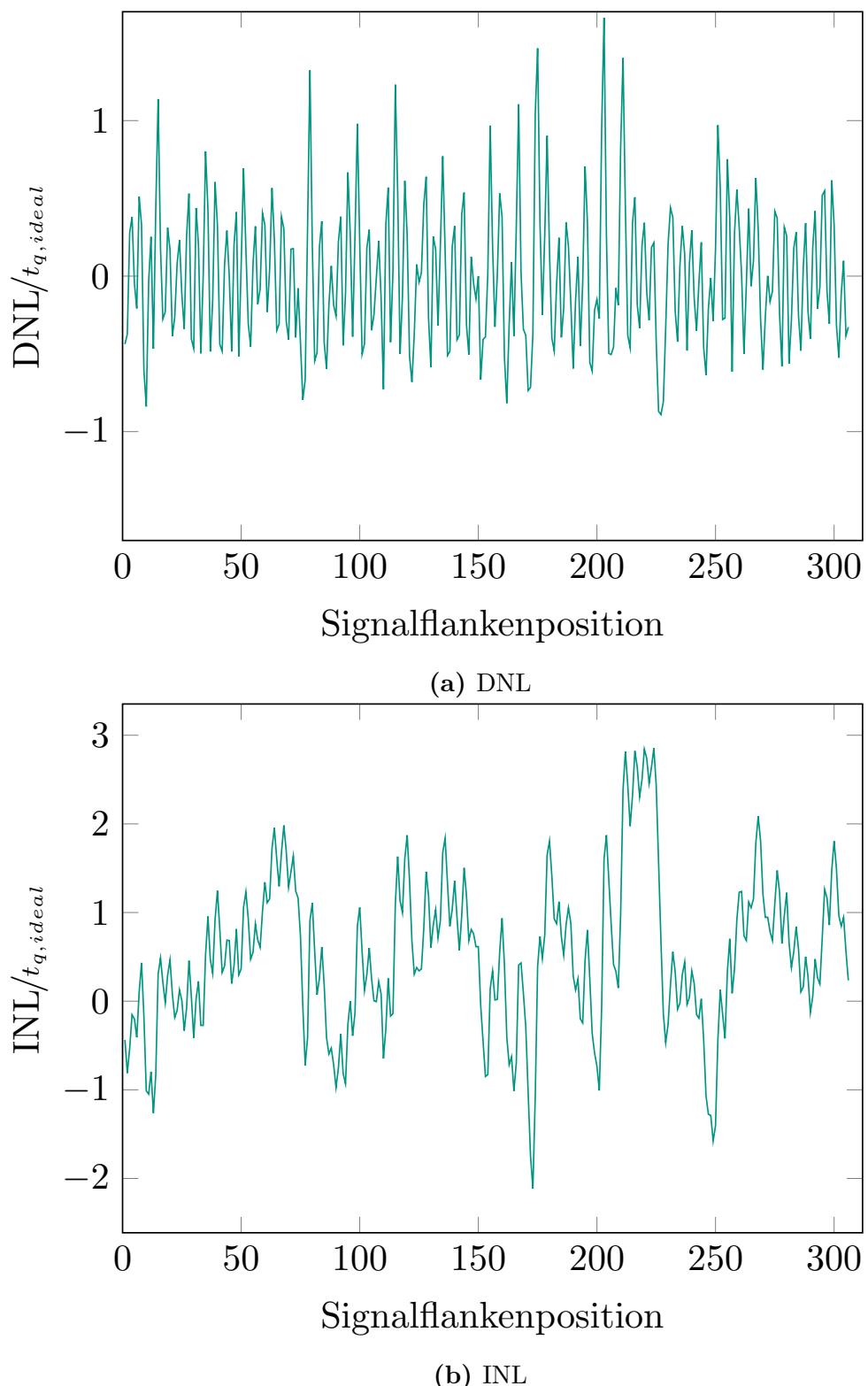
DNL und INL werden in der Regel als Vielfaches des idealen Quantisierungsschrittes  $t_{q,\text{ideal}}$  angegeben. Die Werte des TDC auf der Testplattform sind in Abbildung 6.5 dargestellt. Sie zeigen einen hohen Grad an Nichtlinearität. Welche Auswirkungen dieser Umstand auf das Messergebnis hat, wird in den folgenden Abschnitten verdeutlicht.

### 6.2.2 Quantisierungsfehler

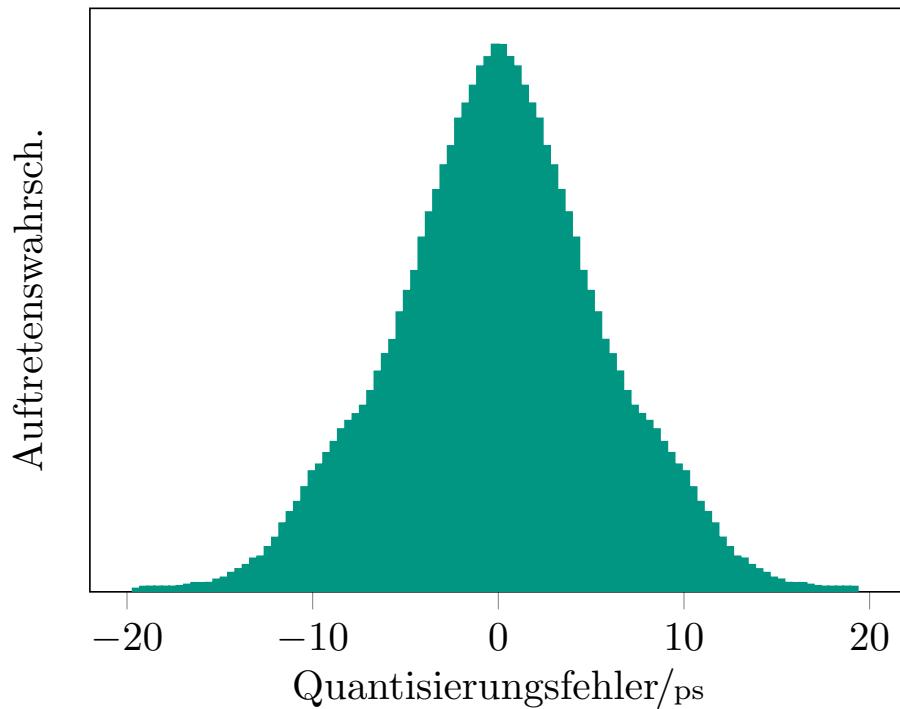
Aufgrund der hohen Nichtlinearität der Quantisierungsstufen der TDL, kann der Quantisierungsfehler nicht analytisch beschrieben werden. Stattdessen wird er numerisch aus dem realen Look-Up-Table der TDL ermittelt.

Unter Annahme von  $10^6$ , relativ zum Systemtakt gleichverteilt eintreffenden, Messimpulsflanken ergibt sich eine in Abbildung 6.6 dargestellte Wahrscheinlichkeitsverteilung des Quantisierungsfehlers. Die Verteilung wurde mit den realen Quantisierungsstufen der Tapped Delay Line berechnet. Der mittlere und maximale Quantisierungsfehler sind:

$$\begin{aligned} |\bar{e}_q| &= 4,467 \text{ps} \\ \max|e_q| &= 19,370 \text{ps} \end{aligned}$$



**Abbildung 6.5:** Verlauf der DNL und INL der Tapped Delay Line als Vielfache des idealen Quantisierungsschrittes  $t_{q,ideal}$ .



**Abbildung 6.6:** Numerisch berechnete Wahrscheinlichkeitsverteilung des Quantisierungsfehlers einer TDL.  $|\bar{e}_q| = 4,467\text{ps}$ ,  $\max|e_q| = 19,730\text{ps}$ .

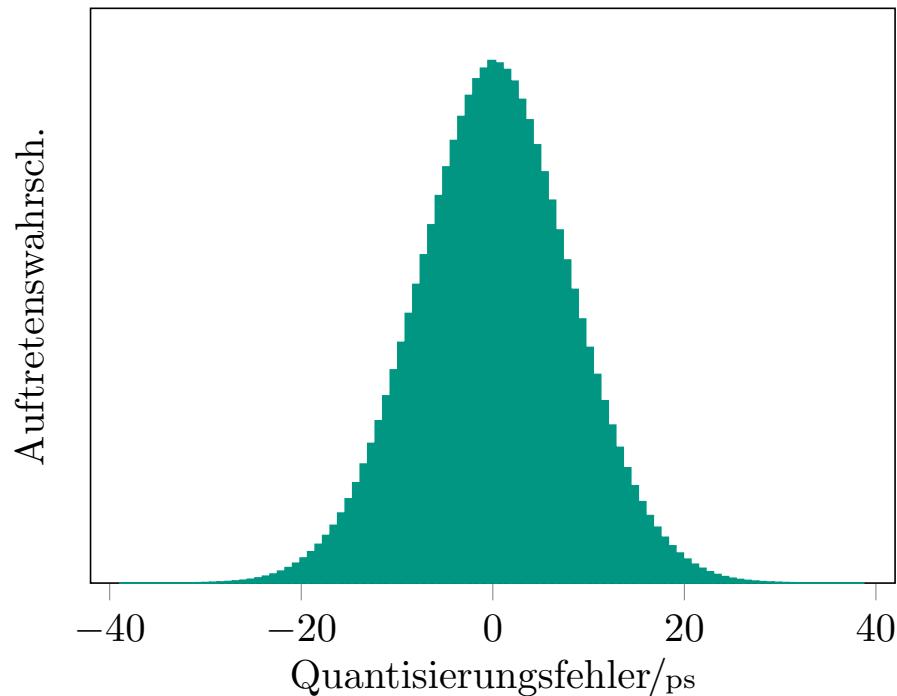
### 6.2.3 Auflösung

Als Auflösung wird bei Messgeräten oder -aufbauten der kleinste wahrnehmbare Unterschied der Messgröße bezeichnet. Bei annähernd äquidistanten Quantisierungsstufen ist dies ein aussagekräftiges Qualitätsmerkmal.

Bei nichtlinear quantisierten Systemen, wie dem FPGA-basierten TDC, ist der gemeinsame Quantisierungsfehler von Start- und Stopimpuls aussagekräftiger. Er resultiert aus einer Überlagerung der Einzelfehler der beiden Kanäle und ist annähernd normalverteilt (s. Abbildung 6.7). Der mittlere und maximale Fehler, sowie die Standardabweichung der angenäherten Normalverteilung ergeben sich zu:

$$\begin{aligned} |\bar{e}_q| &= 6,394\text{ps} \\ \max|e_q| &= 39,284\text{ps} \\ \sigma &= 8,052\text{ps} \end{aligned}$$

Im schlechtesten Fall muss bei einer einzelnen Zeitintervallmessung also von einem Fehler von  $e_q = 39,284\text{ps}$  ausgegangen werden.



**Abbildung 6.7:** Numerisch berechnete, annährend normalverteilte, Auftretenswahrscheinlichkeit des gemeinsamen Quantisierungsfehlers der Messungen von Start- und Stopimpuls.

#### 6.2.4 Dynamikbereich

Die Anzahl der ganzen Taktperioden  $n_{clk}$  wird von einem 10 Bit Zähler erfasst. Der Dynamikbereichs des TDC ergibt sich somit zu:

$$t_{min} = 0\mu s \quad (6.4)$$

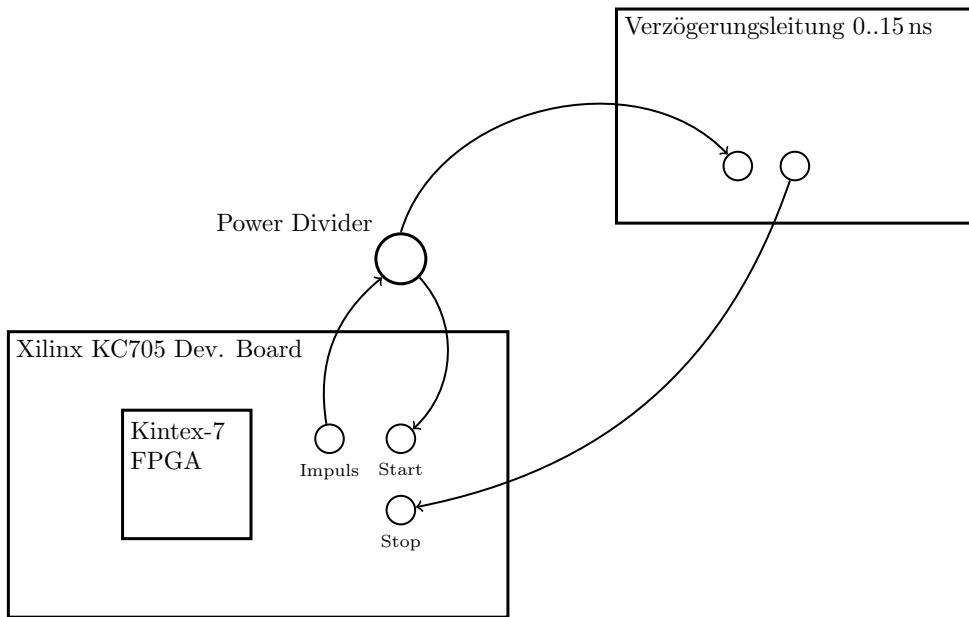
$$t_{max} = \frac{2^{10}}{260\text{MHz}} = 3,938\mu s \quad (6.5)$$

Der in Abschnitt 1.2 geforderte Dynamikbereich wird damit erreicht und überschritten.

### 6.3 Zeitmessungen

#### 6.3.1 Messaufbau

In diesem Abschnitt wird näher auf den Messaufbau eingegangen, mit dem die Zeitmessungen des TDC durchgeführt werden. Neben dem physikalischen Messaufbau wird auch die entworfen Softwareinfrastruktur beschrieben.



**Abbildung 6.8:** Skizze des grundlegenden Messaufbaus der durchgeföhrten Zeitmessungen.

Das entwickelte TDC-Designs wurden auf einem Kintex-7 XC7K325T-2 FPGA des KC705 Evaluation Kits von Xilinx implementiert. Ein Board-Wrapper instanziert, neben dem TDC, einen Microblaze Softcoreprozessor (Xilinx IP), welcher als Schnittstelle zwischen Benutzer und TDC fungiert. Der Microblaze kommuniziert über eine serielle UART-Schnittstelle mit einem PC. Mit einem Pythonscript werden übertragene Daten erfasst, gespeichert und ausgewertet.

Auf dem FPGA werden Messsignale in einem Ringoszillatator generiert. Diese werden über einen SMA-Port des KC705 Boards ausgegeben. Über einen Signalteiler wird dass Signal zum Einen direkt auf den FPGA zurückgeführt und als Startsignal der Messung verwendet. Zum Anderen wird es einer variabel einstellbaren Verzögerungsleitung zugeführt. Das verzögerte Signal geht als Stopsignal zurück auf den FPGA. In Abbildung 6.8 ist der Aufbau skizziert.

### Einstellbare Verzögerungsleitung

Zur variablen Verzögerung des Signals wird eine, vom Institut für Photonik und Quantenelektronik (IPQ) zur Verfügung gestellte, Verzögerungsleitung eingesetzt (Abbildung 6.9). Diese erlaubt es Verzögerungen von 0 bis 15 ns in Schritten von einer Nanosekunde einzustellen. Die für die Signalverzögerung eingesetzten Schaltungen sind nicht bekannt.



**Abbildung 6.9:** Einstellbare Verzögerungsleitung des Instituts für Photonik und Quantenelektronik (IPQ) für die Verzögerung der Messsignale.

## Software

Der Microblaze ist über drei AXI4-Stream Interfaces mit der Logik des Toplevel-Wrapper verbunden. Der Softcoreprozessor kann Daten mit den folgenden Befehlen anfordern, welche von der Logik unterstützt werden.

**0x0004:** Schreibe 512 konsekutive Messwerte des TDC in einen vom Microblaze auslesbaren FIFO.

**0x0008:** Schreibe den Inhalt des Detektionshistogramm bei der nächsten Kalibrierung in einen vom Microblaze auslesbaren FIFO.

**0x0010:** Schreibe die Bitfolge der Signalflanke im Thermometercode in einen vom Microblaze auslesbaren FIFO.

Die Software des Microblaze kann so Mess- und Histogrammdaten des TDC, sowie die in Abschnitt 5.2.1 verwendeten Ausschnitte des Thermometercodes anfordern. Diese Daten können auf dem Prozessor verarbeitet und über die serielle UART-Schnittstelle an einen externen PC gesendet werden. Pythonskripte lesen die serielle Schnittstelle aus, speichern die Daten und bereiten diese visuell auf.

## Platzierung der TDC-Elemente auf dem FPGA

Die TDLs sind am äußeren Rand der oberen Hälfte der Taktregion X1Y1 platziert, in welcher sich auch die Taktpins sowie der, zur Synthese der 260 MHz Takte verwendete, Mixed Mode Clock Manager befinden. Dies garantiert den kürzesten möglichen Signalweg des Taktsignals zu den Tapped Delay Lines. Die Eingänge der Messimpulse befinden sich durch die Architektur des KC705-Boards bedingt in den Taktregionen X0Y1 bzw. X0Y3.

Der Ringoszillator, welcher das Kalibrierungssignal generiert, wird mittels eines pblock-Constraints auf den Raum von 18x22 Slices eingeschränkt. Dies erlaubt eine vom Placementalgorithmus weitestgehend unabhängige, konstante Oszillationsfrequenz über mehrere Implementierungs durchläufe hinweg.

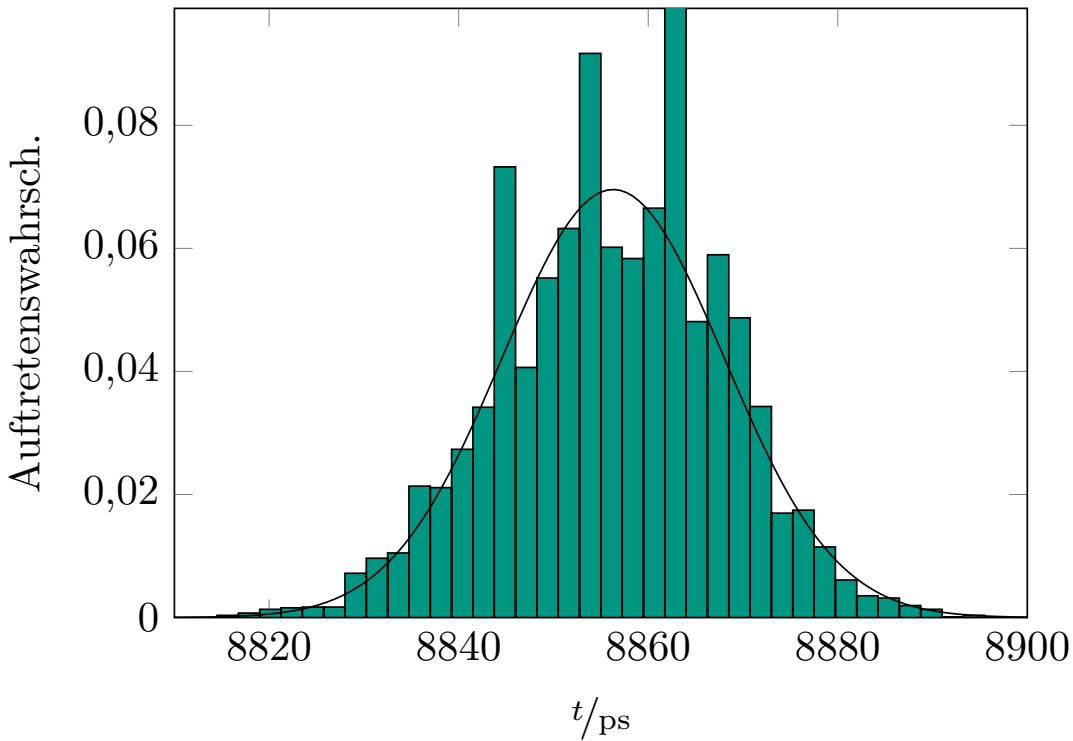
### 6.3.2 Relativer mittlerer Fehler

Der relative mittlere Fehler ist als die mittlere Abweichung des Messergebnisses von seinem Mittelwert definiert. Im Bezug auf FPGA-basierte Time to Digital Converter wird er oft anstatt der Auflösung als bezeichnende Kenngröße für die Messgenauigkeit herangezogen (s. Abschnitt 6.2.3). Er entsteht aus den jeweils im Start- und Stopkanal auftretenden Jitterfehlern des Abtasttaktes sowie den Quantisierungsfehlern der Messungen.

Zur Bestimmung des Fehlers werden 8192 Einzelmessungen bei einer konstant auf 0ns eingestellten Verzögerung durchgeführt. Die gemessene Zeit ist die Signallaufzeit durch die Kabel des Messaufbaus. Das Histogramm der Messungen ist in Abbildung 6.10 dargestellt. Die Werte sind annähernd normalverteilt. Der Mittelwert, der relative mittlere Fehler, sowie die maximale Abweichung vom Mittelwert sind:

$$\begin{aligned}\bar{t}_{int} &= 8856,35\text{ps} \\ \sigma_t &= 11,72\text{ps} \\ \max|t_{int} - \bar{t}_{int}| &= 41,35\text{ps}\end{aligned}$$

Der normalverteilte Charakter und die maximale Abweichung vom Mittelwert bestätigen die in Abschnitt 6.2.2 angestellten theoretischen Überlegungen zum Quantisierungsfehler. Der relative mittlere Fehler entspricht in etwa dem Jitterfehler des Abtasttaktes.



**Abbildung 6.10:** Histogramm aus 8192 Zeitmessungen bei konstant eingestelltem Verzögerungsintervall von 0ns.

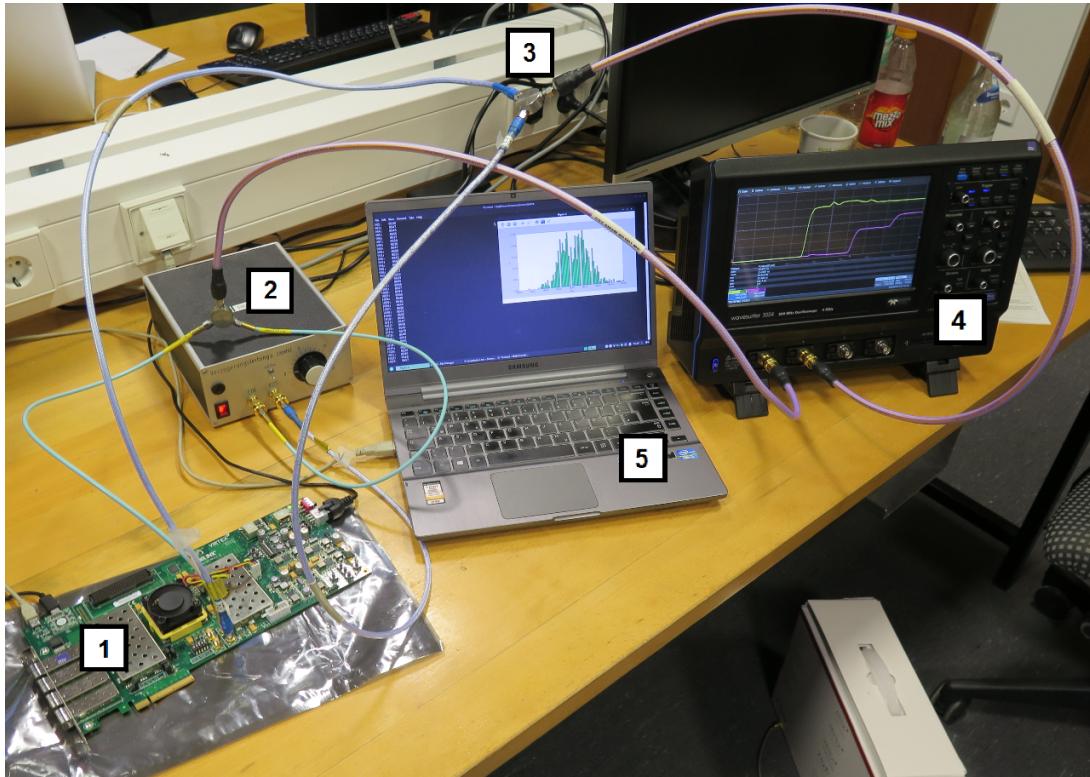
### 6.3.3 Absolute Genauigkeit

Die absolute Messgenauigkeit des TDC hängt linear von der Genauigkeit der Systemtaktperiode ab:

$$t_{int} = \left( \frac{\text{LUT}(m_{Start}) - \text{LUT}(m_{Stop})}{2^{18}} + n_{clk} \right) \cdot t_{clk} \quad (6.6)$$

In einer Messreihe werden die Zeitintervalle zwischen den Flanken von Start- und Stopimpuls für die 16 Stufen der Verzögerungsleitung gemessen. Die Messung wird zeitgleich mit dem Time to Digital Converter und dem Teledyne LeCroy Oszilloskop durchgeführt. Das Startsignal für den TDC ist intern im FPGA geroutet. Durch die sonst, zum Anschluss des Oszilloskops notwendigen, zusätzlichen Signaleiter ist der Eingangsspeicher des Stopsignals am TDC nicht mehr detektierbar. Ein Bild des Messaufbaus ist in Abbildung 6.11 zu sehen.

Mit dem TDC werden 1024 Messergebnisse aufgenommen und deren Mittelwert bestimmt. Die Ergebnisse des Oszilloskops sind etwa über die selbe Anzahl gemittelt. Die Messwerte von TDC und Oszilloskop sind gemeinsam in Abbildung 6.12 offsetkompensiert dargestellt. In den Vergrößerungen ist zu erkennen, dass die Ergebnisse mit größer werdendem



**Abbildung 6.11:** Messaufbau für die Vergleichsmessungen mit TDC und Oszilloskop. [1] FPGA Dev. Kit, [2] IPQ-Verzögerungsleitung, [3] Signalteiler, [4] Teledyne LeCroy Wavesurfer 3024, [5] Notebook mit Datenerfassungssoftware.

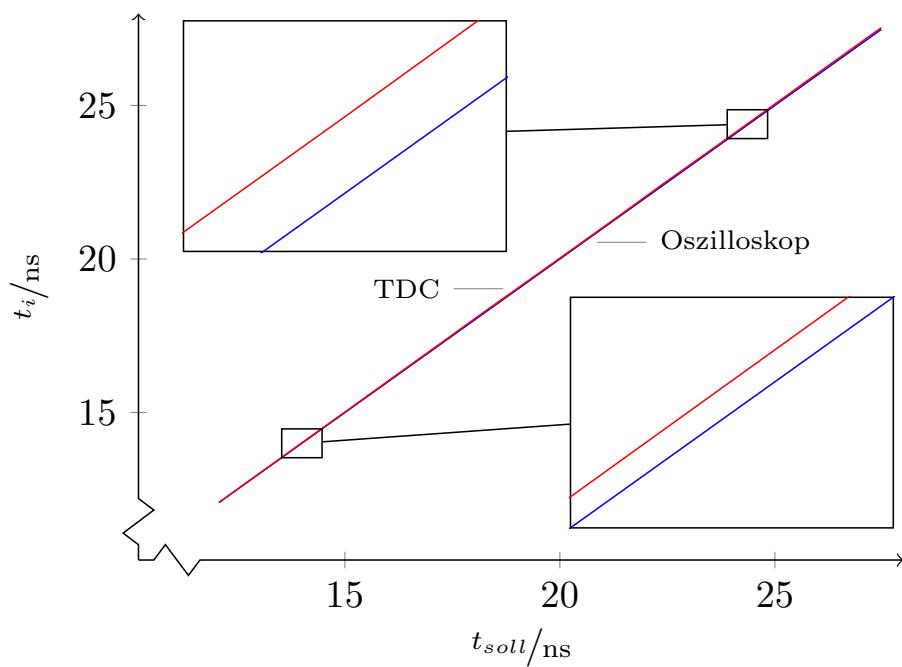
Zeitintervall immer mehr auseinander driften.

In Abbildung 6.13 ist die Differenz der Zeitmessungen dargestellt. Ein linearer Charakter des absoluten Fehlers über der Intervallzeit ist zu erkennen. Aufgrund der in Abschnitt 6.1 festgestellten Präzision des Systemtaktes wird davon ausgegangen, dass die Ursache der steigenden Abweichungen beim Oszilloskop liegt.

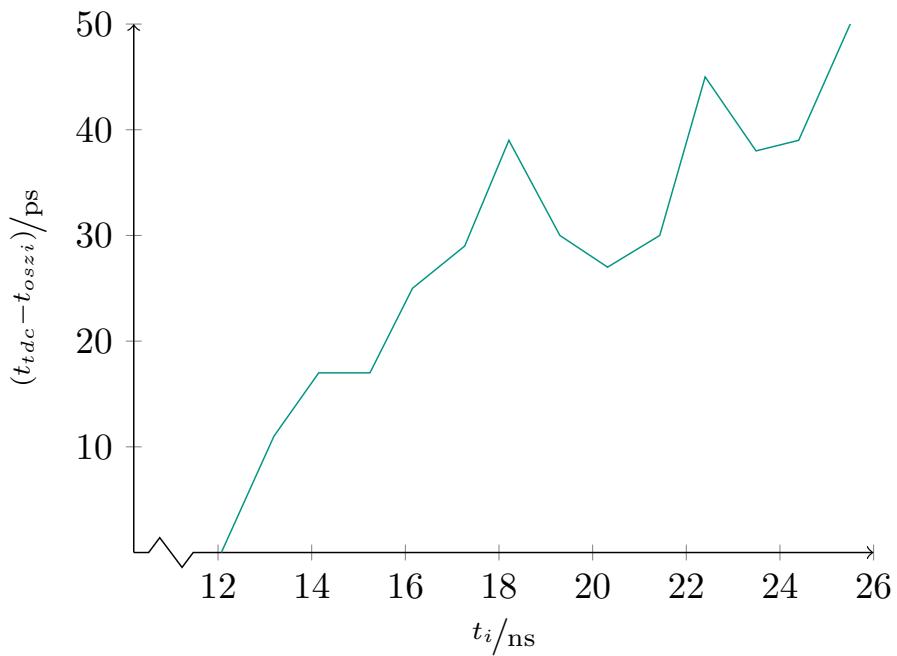
## 6.4 Anforderungen an den externen Taktgeber

Der Zeitbezug des Time to Digital Converter ist einzig über die genaue Kenntnis der Taktperiode des Abtasttaktes gegeben. Die Präzision der Zeitmessung ist von der des Systemtaktes direkt abhängig.

Eine Abweichung der Taktperiode wirkt sich linear auf das Messergebnis aus (vgl. Abschnitt 6.3.3). Ein statischer Offset, sowie eine bekannte Drift des Taktes zur Laufzeit, kann über das Anpassen des für die Berechnung verwendeten Zeitwertes der Taktperiode in Picosekunden ausgeglichen werden. Das Verwenden eines frequenzgenauen Oszillators



**Abbildung 6.12:** Offsetkompensierte, zeitgleiche Messungen des Zeitintervalls zwischen Start- und Stopimpuls mit dem TDC und dem Teledyne LeCroy Oszilloskop.



**Abbildung 6.13:** Offsetkompensierte Differenz der Zeitintervallmessungen des TDCs und des Oszilloskop. Ein linearer Charakter des absoluten Messfehlers ist erkennbar.

erspart diese nachträgliche Kompensation.

Die Abweichung des Abtastzeitpunktes von seinem Sollzeitpunkt, also die Ausprägung des normalverteilten Jitterfehlers zum Zeitpunkt des Abtastens, wirkt sich ebenfalls linear auf das Messergebnis einer TDL-Messung aus. Der relative Mittlere Fehler der Zeitmessung entspricht in etwa dem Jitterfehler des Abtasttaktes (s. Abschnitt 6.3.2).

Der Mixed Mode Clock Manager, mit dem das Abtastsignal erzeugt wird, wirkt als Jitterfilter [6]. Das Verhältnis zwischen Ein- und Ausgangsjitter des MMCM ist also nicht linear. Die folgende Tabelle zeigt Ein- und Ausgangsjitter des MMCM bei der Erzeugung eines 260MHz Signals. Die getesteten Eingangssignale sind der 200MHz Takt des KC705 Boards (vgl. Abschnitt 6.1) mit geringem Jitterfehler, sowie ein extern erzeugter 15MHz Takt mit vergleichsweise hohem Jitter.

Quelle	Eingangsjitter ( $\sigma$ )	Ausangsjitter ( $\sigma$ )
KC705 Boardtakt	4,68ps	13,62ps
Ext. Funktionsgenerator	76,52ps	19,17ps

**Tabelle 6.1:** Jitterfehler bei der Erzeugung von 260MHz Taktsignalen aus dem Boardtakt des KC705 Boards und einem externen Funktionsgenerator.

Die Wirkung des MMCM als Jitterfilter ist klar zu erkennen. Auch ist ersichtlich, dass eine hohe Stabilität des Taktgebenden Oszillators nur begrenzt positive Auswirkungen auf den resultierenden Jitter des synthetisierten Taktes hat.

#### 6.4.1 Zahlenbeispiel zum Frequenzoffset

Beispielhaft werden die Auswirkungen der Frequenzabweichungen eines Si511 Quartzoszillators von Silicon Labs auf die Messergebnisse des TDC aufgezeigt. Der Si511 kann mit einem integrierten Phase Locked Loop Frequenzen von 0,1 – 250MHz erzeugen. In diesem Beispiel wird eine Ausgabefrequenz von 100MHz gewählt. Der 260MHz Abtasttakt wird mittels eines Mixed Mode Clock Managers des FPGAs generiert. Die Frequenzeigenschaften des Si511 sind für *Stability Grade A* und *Stability Grade C* in Tabelle 6.2 aufgelistet.

Es soll ein Zeitintervall von 0,8μs gemessen werden. Ist  $S$  die Stabilität des Oszillators in ppm,  $t_{clk}$  die ideale und  $\hat{t}_{clk}$  die reale Taktperiode des Oszillators, ergeben sich die

	Stability Grade	
	A	C
Temp. Stabilität ( $-40^{\circ}\text{C} \dots 85^{\circ}\text{C}$ )	$\pm 100\text{ppm}$	$\pm 30\text{ppm}$
Periodenjitter (RMS)		1, 3ps
Periodenjitter (max.)		11ps

**Tabelle 6.2:** Frequenzeigenschaften des Si511 Quartzoszillators von Silicon Labs.

gemessenen Zeiten nach Gleichung (6.6) wie folgt:

$$\begin{aligned}\hat{t}_{clk}^{-1} &= 260\text{MHz} \cdot (1 + S \cdot \frac{260\text{MHz}}{100\text{MHz}}) \\ \Delta t_{clk} &= \hat{t}_{clk} - t_{clk} \\ t_{int} + \Delta t_{int} &= 0,8\mu\text{s} \cdot 260\text{MHz} \cdot (t_{clk} + \Delta t_{clk})\end{aligned}$$

Der Fehler des gemessenen Zeitintervalls  $\Delta t_{int}$  ist linear vom Fehler der Taktperiode des Abtasttaktes abhängig.  $\Delta t_{int}$  ist für  $S = 30\text{ppm}$  bzw.  $S = 100\text{ppm}$  Frequenzoffset:

$$\begin{aligned}\Delta t_{int,30} &= 62,4\text{ps} \\ \Delta t_{int,100} &= 207,9\text{ps}\end{aligned}$$

## 6.5 Einsatz des TDC auf anderen FPGA-Modellen

Der entwickelte Time to Digital Converter kann grundsätzlich auf allen Xilinx FPGAs der 7er Serie eingesetzt werden, da sich der Aufbau ihrer Configurable Logic Blocks nicht voneinander unterscheidet. Es gibt zwischen den Modellen jedoch Unterschiede, beispielsweise bei der maximalen Taktfrequenz und der Schaltgeschwindigkeit der asynchronen Logik.

### 6.5.1 Virtex-7 FPGA

Bei einer Testimplementierung auf einem Virtex-7 FPGA wird festgestellt, dass die Verzögerungszeiten der Verzögerungselemente in der TDL kürzer sind als auf dem Kintex-7 FPGA. Gemäß Gleichung (5.1) aus Abschnitt 5.1 muss deswegen die Abtastfrequenz erhöht werden. 300MHz haben sich hierfür als ausreichend herausgestellt. Am TDC mussten keine weiteren Änderungen gemacht werden. Der relative mittlere Fehler der Zeitintervallmessung verbessert sich, aufgrund der kleineren Quantisierungsstufen, geringfügig auf  $\sigma_t = 10,4\text{ps}$ .

### 6.5.2 Artix-7 FPGA-Fabric eines Zynq-7000 SoC

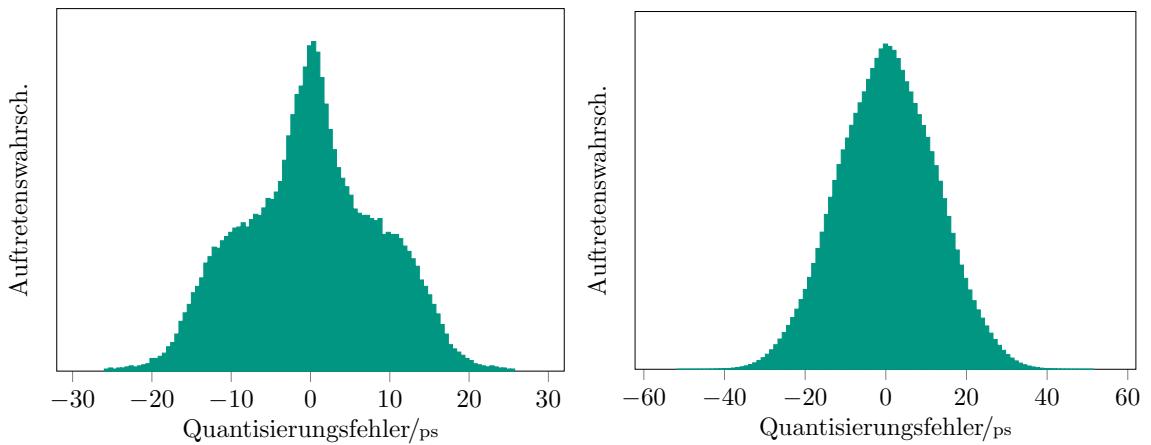
Der Zynq-7000 von Xilinx ist ein *System on Chip* (SoC) welches aus einem ARM Cortex-A9 Prozessor, sowie einem Artix-7-basiertes FPGA-Fabric besteht. Artix-7-FPGAs sind für den Einsatz in kostensensitiven Produkten mit hohen Fertigungsstückzahlen konzipiert. Sie zeichnen sich durch ihren geringeren Energieverbrauch und kleinen Formfaktor aus. Darunter leidet auch die Performanz des Systems.

In der Testimplementierung hat sich eine Abtastfrequenz von 220MHz als ausreichend für die Erfüllung von Gleichung (5.1) herausgestellt. Um Timingfehler zu verhindern, muss in der Datenverarbeitung bei der Berechnung der Signalfankenposition im Thermometercode, sowie beim Erstellen des Look-Up-Tables aus dem Detektionshistogramm eine weitere Pipelinestufe eingesetzt werden.

Die Verzögerungszeiten der einzelnen Elemente der Tapped Delay Line sind, verglichen mit dem Kintex-7 und Virtex-7, größer. Dies wirkt sich auf den Quantisierungsfehler des Systems aus. Mittlerer und maximaler Quantisierungsfehlers einer Zeitmessung ergeben sich, wie in Abschnitt 6.2.3 zu:

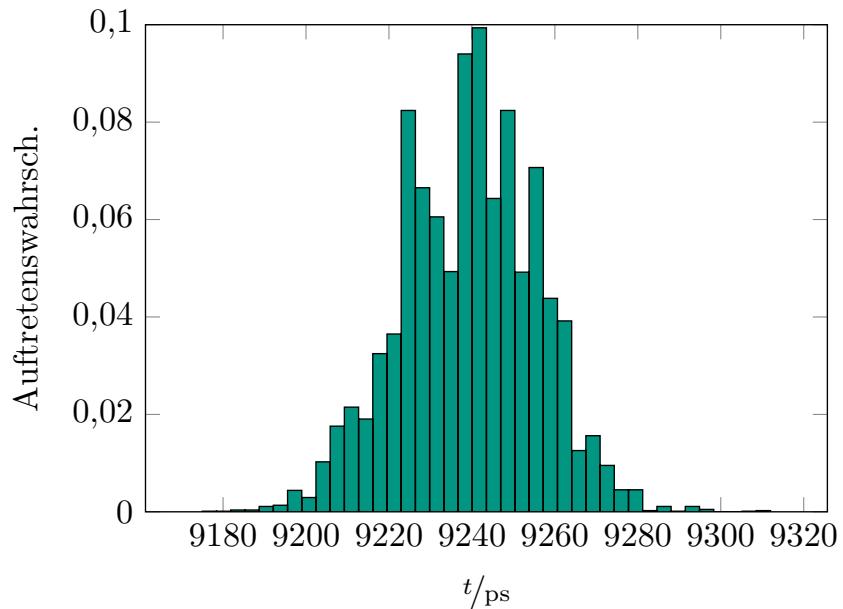
$$\begin{aligned} \bar{|e_q|} &= 9,695\text{ps} \\ \max|e_q| &= 52,351\text{ps} \end{aligned}$$

Abbildung 6.14 zeigt den Quantisierungsfehler eines Kanals, sowie den gemeinsamen Quantisierungsfehler der Zeitmessung. Eine Testmessung ergibt einen relativen mittleren Fehler von  $\sigma_t = 16,401\text{ps}$ . Gegenüber dem Ergebnis des Kintex-7 ist dies eine Verschlechterung von etwa 40%. Die Maximale Abweichung vom Mittelwert beträgt  $\max|t_{int} - \bar{t}_{int}| = 72,687\text{ps}$ . Das Histogramm des Messergebnisses ist in Abbildung 6.15 zu sehen.



(a) Verteilung des Quantisierungsfehlers eines Kanals.  
(b) Verteilung des gemeinsamen Quantisierungsfehlers der Zeitmessung.

**Abbildung 6.14:** Quantisierungsfehler der Testimplementierung auf einem Zynq-7000 SoC.



**Abbildung 6.15:** Histogramm aus 8192 Zeitmessungen mit auf einem Zynq-7000. Der relative mittlere Fehler beträgt  $\sigma_t = 16,401\text{ps}$ . Die Maximale Abweichung vom Mittelwert ist  $\max|t_{int} - \bar{t}_{int}| = 72,687\text{ps}$ .

# Kapitel 7

## Fazit und Ausblick

### 7.1 Zusammenfassung der Arbeit

Ausgehend von der Recherche FPGA-basierter TDC-Designs zum Messen des Zeitintervalls zwischen den Flanken zweier Messimpulse, wird ein Realisierungskonzept auf Basis der Tapped Delay Line Methode mit dynamischer Kalibrierung zur Laufzeit erstellt.

Die Tapped Delay Line wird, unter Berücksichtigung der, in den Kalibrierungshistogrammen der ersten Implementierungen gemessenen, Schwachstellen der zu Grunde liegenden Hardware, in vier phasenverschobenen Abschnitten realisiert. Ein „Multishotmechanismus“ wird konzipiert und implementiert, welcher es erlaubt kurz aufeinanderfolgende Messimpulse zu erfassen.

Die bessere Eignung der Kodierung der Impulsflankenposition im Thermometerkode mittels Zählen der Einsen gegenüber der konventionellen Filtermethode wird experimentell erwiesen. Die Kodierung wurde möglichst laufzeiteffizient umgesetzt.

Eine dynamische Kalibrierung zur Laufzeit wird reaktiv so realisiert, dass sie der übergeordneten Logik die Entscheidung überlässt, wann die Kalibrierungsphasen stattfinden sollen. Die Generierung des Look-Up-Table erfolgt im Hintergrund, so dass zu jeder Zeit Messungen durchgeführt werden können. Unter Annahme eines, relativ zum Systemtakt, gleichverteilten Kalibrierungssignals, werden die Konfidenzintervalle der geschätzten Verzögerungszeiten für verschiedene Stichprobengrößen  $N$  ermittelt. Das Ergebnis zeigt klar, dass eine Kalibrierung mit  $N = 2^{18}$  Testimpulsen zu bevorzugen ist. Die, für die Kalibrierung notwendige, Gleichverteilung des Flankenversatzens zwischen Takt- und Kalibrierungssignal wird nachgewiesen.

Mittels eines Wrappers wird die Komplexität der Einbindung des TDC als Teil eines Gesamtsystems so weit reduziert, dass als Interface nur die Lese- bzw. Schreibseite zweier asynchroner FIFOs, sowie drei Status- bzw. Steuersignale übrig bleiben. Ein ID-Bit erlaubt es Daten verschiedener Messungen voneinander zu unterscheiden.

Auf Basis der geschätzten Verzögerungszeiten der Delay-Line-Elemente wird der gemeinsame Quantisierungsfehler der Messung von Start- und Stopimpuls in MATLAB numerisch ermittelt. Im schlechtesten anzunehmenden Fall weicht die quantisierte Zeit 39,3ps von der real vergangenen Zeit ab. Der mittlere Quantisierungsfehler beträgt 6,394ps.

Die Auswertung der Testmessungen des TDCs weisen einen relativen Mittleren Fehler von 11,72 Picosekunden auf. Dies ordnet ihn vorne im Feld der „Single-Sample-TDCs“ ein. Form und Ausmaß des relativen Fehlers bestätigen die getroffenen theoretischen Überlegungen zum Quantisierungsfehler.

## 7.2 Ausblick und weitere Arbeiten

Ein Blick auf akademische Veröffentlichungen des vergangenen Jahres zeigt, dass diese Arbeit den TDCs auf Basis der Tapped Delay Line noch bei Weitem keine Grenzen setzt. Der „Wave Union“ TDC beispielsweise erweitert das Konzept der TDL durch das zeitgleiche Messen mehrerer Impulsflanken in der selben Struktur. Dadurch können insbesondere die wenigen Verzögerungselemente mit besonders großer Laufzeit in mehrere „virtuelle“ Abschnitte aufgeteilt werden. Einen anderen Ansatz verfolgt der sog. „Multichain Measurement“ TDC. Das Messsignal wird, kohärent verzögert, in mehreren TDLs gemessen und deren Ergebnis gemittelt. Hiermit wurden relative mittlere Fehler von 4,2ps erreicht [19].

Auch fehlt es an Evaluationen anderer TDC-Konzepte auf FPGA-basis, die über theoretische Überlegungen und Simulationen hinaus gehen. Neben den, in der Einführung genannten, Vernier-Methoden ist beispielsweise auch eine Nutzung von Transceiver- und Eingangsverzögerungselementen moderner FPGAs zur Zeiterfassung denkbar.

Um neue Einsatzgebiete für FPGA-basierte TDCs ist zu erschließen, ist abschließend noch die Implementierung auf Low-Cost/Low-Performance Chips mit besonderem Augenmerk auf die Energie- und Ressourceneffizienz, sowie die Robustheit gegenüber externen Einflüssen wie Spannungs- und Temperaturschwankungen oder elektromagnetischer Beeinflussung interessant.





# Abkürzungen

**CLB** Configurable Logic Block

**CMT** Clock Management Tile

**FPGA** Field Programmable Gate Array

**GRO** Gated Ring Oscillator

**Histogramm** Detektionshistogramm

**HROW** Horizontal Clock Row

**LUT** Look-Up-Table

**MMCM** Mixed Mode Clock Manager

**PAR** Place and Route

**PFD** Phase-Frequency Detector

**PLL** Phase Locked Loop

**SoC** System on Chip

**TDC** Time to Digital Converter

**TDL** Tapped Delay Line

**VCO** Voltage Controlled Oscillator

# Abbildungsverzeichnis

1.1	TDC7200 Time to Digital Converter von Texas Instruments . . . . .	7
2.1	Struktur des regionalen Taktverteilungsnetzes im FPGA . . . . .	12
2.2	Aufbau einer Taktregion . . . . .	12
2.3	Konnektivität der CLBs und Slices des FPGA-Fabric . . . . .	13
2.4	Detailansicht eines Xilinx 7-Series Slices . . . . .	14
2.5	Struktur des CARRY4-Elements in den FPGA-Slices . . . . .	15
2.6	Schematischer Aufbau eines MMCM . . . . .	16
3.1	Timingdiagramm des Vernier-Oszillators . . . . .	21
3.2	Schematische Struktur der Tapped Delay Line . . . . .	21
3.3	Schematische Struktur der Vernier Delay Line . . . . .	21
4.1	Zeitintervall zwischen den steigenden Flanken zweier Messimpulse . . . . .	22
4.2	Struktur der Tapped Delay Line . . . . .	23
4.3	Zeitlicher Verlauf eines Messvorgangs der TDL . . . . .	24
4.4	Timingdiagramm der in der TDL gemessenen Zeit zwischen Messimpuls- und Taktflanke . . . . .	24
4.5	Timingdiagramm der Zeitintervallmessung des TDCs . . . . .	25
4.6	Ermittelte Verzögerungszeiten einer Tapped Delay Line . . . . .	29
4.7	Mittlere und maximale Länge der Konfidenzintervalle in Abhängigkeit der Stichprobengröße . . . . .	29
4.8	Verschieben der Quantisierungsstufe in die Mitte des Zeitintervalls eines Verzögerungselementes . . . . .	31
4.9	Anschauliche Darstellung der Generierung des Look-Up-Tables . . . . .	31
4.10	Generierung des Look-Up-Tables aus den Daten des Detektionshistogramm	32
5.1	Triggerstruktur mit asynchronem Reset der Tapped Delay Line . . . . .	35
5.2	Propagationszeiten der Verzögerungselemente einer 100 MHz TDL . . . . .	36
5.3	Timingdiagramm der Vier-Phasen-TDL . . . . .	37

5.4	Propagationszeiten der Verzögerungselemente einer 260 MHz Vier-Phasen-TDL . . . . .	37
5.5	Bubble Errors im Thermometercode . . . . .	38
5.6	Prozentuale Abweichung des abgetasteten Thermometercodes von den präferierten Mustern in der TDL . . . . .	39
5.7	Zeitliche Änderungen des Thermometercodes mit Bubble Errors . . . . .	40
5.8	Kodierung mittels Kantenfilter. . . . .	40
5.9	Binäres kodieren der Impulsflankenposition durch zählen der Einsen . . . . .	42
5.10	Phasensynchronisation der Daten der vier TDL-Abschnitte . . . . .	43
5.11	Struktureller Aufbau dies LUT-Moduls . . . . .	46
5.12	Struktureller Aufbau des Ringoszillators . . . . .	47
5.13	Umschalten zwischen Mess- und Kalibrierungssignal . . . . .	48
5.14	Hierarchische Anordnung der Module des TDC . . . . .	51
6.1	Histogramm des nativen 200 MHz Boardtaktes . . . . .	55
6.2	Histogramm des 260 MHz TDC-Taktes . . . . .	56
6.3	Histogramm des Ausgangssignals eines Ringoszillators mit einer Länge von 55 LUT1-Elementen . . . . .	57
6.4	Verteilung des Versatzes zwischen der Signalflanken des Ringoszillatorsignals und des Systemtaktes . . . . .	58
6.5	Verlauf der DNL und INL der Tapped Delay Line . . . . .	60
6.6	Numerisch berechnete Wahrscheinlichkeitsverteilung des Quantisierungsfehlers einer TDL . . . . .	61
6.7	Auftretenswahrscheinlichkeit des Quantisierungsfehlers einer Zeitintervallmessung . . . . .	62
6.8	Skizze des grundlegenden Messaufbaus für Zeitmessungen . . . . .	63
6.9	Einstellbare Verzögerungsleitung zur Verzögerung der Messsignale . . . . .	64
6.10	Histogramm von 8192 Zeitmessungen . . . . .	66
6.11	Messaufbau für die Vergleichsmessungen mit TDC und Oszilloskop . . . . .	67
6.12	Vergleichsmessungen mit TDC und Oszilloskop . . . . .	68
6.13	Differenz der Zeitintervallmessungen des TDCs und des Oszilloskop . . . . .	68
6.14	Quantisierungsfehler der Testimplementierung auf einem Zynq-7000 SoC . . . . .	72
6.15	Histogramm aus 8192 Zeitmessungen mit auf einem Zynq-7000 . . . . .	72

# Tabellenverzeichnis

5.1	Auswirkung von Bubble Errors auf das Kodierungsergebnis beim Verwen- den von Kantenfiltern . . . . .	41
5.2	Wahrheitstabelle des LUT2-Gates des Ringoszillators . . . . .	47
5.3	Interfaces zwischen den Modulen des Time to Digital Converter . . . . .	53
6.1	Jitterfehler bei der Erzeugung von 260MHz Taktsignalen aus dem Board- takt des KC705 Boards und einem externen Funktionsgenerator. . . . .	69
6.2	Frequenzeigenschaften des Si511 Quartzoszillators von Silicon Labs. . . . .	70

# Literaturverzeichnis

- [1] D. Genschow, "A time to digital converter for use in ultra wide band radar sensor nodes," in *2015 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, 2015, p. 38.
- [2] J. Kalisz, R. Szplet, J. Pasierbinski, and A. Poniecki, "Field-programmable-gate-array-based time-to-digital converter with 200-ps resolution," *Ieee Transactions On Instrumentation And Measurement*, vol. 46, pp. 51–55, 1997.
- [3] K. J. Hong, E. Kim, J. Y. Yeom, P. D. Olcott, and C. S. Levin, "Fpga-based time-to-digital converter for time-of-flight pet detector," in *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, 2012, p. 2463.
- [4] M. Büchele, H. Fischer, F. Herrmann, K. Königsmann, C. Schill, and S. Schopferer, "The gandalf 128-channel time-to-digital converter," *Physics Procedia, Volume 37*, p. 1827-1834., vol. 37, pp. 1827–1834, n/a 2012.
- [5] G. Wu, G. Deyuan, W. Tingcun, C. Hu-Guo, and Y. Hu, "A high-resolution multi-channel time-to-digital converter (tdc) for high-energy physics and biomedical imaging applications," in *2009 4th IEEE Conference on Industrial Electronics and Applications*, 2009, p. 1133.
- [6] 7 Series FPGAs Clocking Resources User Guide, 1st ed., Xilinx Inc., June 2015.
- [7] 7 Series FPGAs Configurable Logic Block User Guide, 1st ed., Xilinx Inc., November 2014.
- [8] X. Inc., "Xilinx 7 series fpga and zynq-7000 all programmable soc libraries guide for hdl designs."
- [9] M.-C. Lin, G.-R. Tsai, C.-Y. Liu, and S.-S. Chu, "Fpga-based high area efficient time-to-digital ip design," in *TENCON 2006 - 2006 IEEE Region 10 Conference*, 2006, p. 1.
- [10] W. Pan, "A 20-ps temperature compensated time-to-digital converter (tdc) implemented in fpga," in *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*, 2013, p. 1.
- [11] C. Liu and Y. Wang, "A 128-channel, 710 m samples/second, and less than 10 ps

- rms resolution time-to-digital converter implemented in a kintex-7 fpga,” *IEEE Transactions On Nuclear Science*, vol. 62, pp. 773–783, 2015.
- [12] A. Aloisio, P. Branchini, R. Cicalese, R. Giordano, V. Izzo, and S. Loffredo, “Fpga implementation of a high-resolution time-to-digital converter,” in *2007 IEEE Nuclear Science Symposium Conference Record*, 2007, p. 504.
- [13] H. Menninga, C. Favi, M. W. Fishburn, and E. Charbon, “A multi-channel, 10ps resolution, fpga-based tdc with 300ms/s throughput for open-source pet applications,” in *2011 IEEE Nuclear Science Symposium Conference Record*, 2011, p. 1515.
- [14] M. Thakur, B. B. Soni, P. Gaur, and P. Yadav, “Analysis of metastability performance in digital circuits on flip-flop,” in *2014 International Conference on Communication and Network Technologies*, 2014, p. 265.
- [15] L. Sachs, *Angewandte Statistik: Mehtodensammlung*, 11th ed. Springer-Verlag.
- [16] C. Chen and R. W. Tipping, “Confidence interval of a proportion with over-dispersion,” *Biometrical Journal*, vol. 44, no. 7, p. 877, 2002.
- [17] J. Wu, “Several key issues on implementing delay line based tdc using fpgas,” *IEEE Transactions on Nuclear Science*, vol. 57, issue 3, pp. 1543-1548, vol. 57, pp. 1543–1548, jun 2010.
- [18] C. Favi and E. Charbon, “A 17ps time-to-digital converter implemented in 65nm fpga technology,” in *Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays - FPGA '09*, 2009, p. 113.
- [19] Q. Shen, S. Liu, B. Qi, Q. An, S. Liao, P. Shang, C. Peng, and W. Liu, “A 1.7 ps equivalent bin size and 4.2 ps rms fpga tdc based on multichain measurements averaging method,” *IEEE Transactions on Nuclear Science*, vol. 62, issue 3, pp. 947-954, vol. 62, pp. 947–954, jun 2015.