



Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Lehrstuhl für Entwurfsautomatisierung

**Development of Analytical Behavioral Models for
Digitally Controlled Edge Interpolator (DCEI) based
Digital-to-Time Converter (DTC) Circuits**

Sebastian Sievert, M.Sc. (TUM)

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender: Prof. Dr. Andreas Herkersdorf

Prüfer der Dissertation:

1. apl. Prof. Dr.-Ing. Helmut Gräß
2. Prof. Dr.-Ing., Dr.-Ing. habil. Robert Weigel

Die Dissertation wurde am 18.04.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 07.07.2017 angenommen.

Abstract

Shrinking of CMOS process technology aims at reducing area, power, and cost while increasing the operation frequency of the fabricated circuits. Due to the quadratic dependency of power consumption from voltage, the supply voltage is lowered for smaller technology nodes to save power. For conventional digital-to-analog converters (DAC) the reduced voltage headroom complicates the circuit design. However, time converter circuits as time-to-digital converters (TDC) or digital-to-time converters (DTC) benefit from technology scaling: Faster transistors and lower parasitic capacitances enable a reduced minimum inverter delay, and a finer tuning of RC time constants is enabled through decreasing minimum capacitor sizes.

The present thesis concentrates on the investigation of DTCs, which got increasing attention from academical and industrial research in the past decade. These circuit types belong to the class of DACs, where the analog domain is time. DTCs apply a time delay, controlled by a digital code word, on a reference input clock, allowing a dynamic modulation of the DTC output signal's period and thus frequency. This concept enables various applications in the areas of frequency synthesis and wireline/wireless transmitters and receivers. Similar to conventional DACs, important performance characteristics include the full scale delay, resolution, nonlinearity, and monotonicity, but also power consumption and jitter or phase noise.

DTCs that target high time resolution are usually segmented into a multi-stage architecture with subsequent coarse and fine delay tuning stages. Recent literature discussed several architectures for coarse and fine tuning, including phase interpolator (PI) circuits for fine delay tuning. While PIs have the advantage of a well-defined tuning range, designs presented so far lack high linearity and show only resolutions of up to 5 bit. The primary focus of the present thesis is the design, modeling, and verification of PI based DTCs. The investigated PIs are implemented as digitally controlled edge interpolators (DCEI), belonging to PI types that operate on digital rail-to-rail signals. Their architecture is based exclusively on digital circuit elements, enabling to leverage technology scaling even further.

Based on an existing 2 GHz three-stage DTC design with 11 bit resolution (7 bit provided by the PI fine tuning stage), PIs' nonlinearity sources are elaborately analyzed with an analytical circuit model to confirm and quantify different sources of nonlinearity discussed in the literature so far. As the major source of nonlinearity are shoot-through currents during the phase interpolation, it is imperative for a high linearity design to suppress them. Linear PI designs published to date implement this and allow up to 5 bit resolution, however, they have several architectural drawbacks, including the limitation that the digital code is only applied on the rising output edge while the falling edge needs to be used for resetting the PI. The present thesis presents a linearized 7 bit PI that prevents shoot-through currents with additional control logic. The linearized PI enables interpolation on rising and falling edges through implementation of retention cells, that are complementary to the interpolation cells and render the PI reset unnecessary. While this linearization concept can lead to ideally perfect linearity, its main drawback is the increased power consumption due to the additional control logic.

Therefore, a second novel PI architecture is developed in this thesis. A two-points PI exhibits two subsequent coarse and fine interpolations to double the full-scale interpolation range compared to the reference PI. This allows to reduce the reference three-stage DTC design to two stages, decreasing the power consumption and simplifying the overall DTC design. While it is usually no issue to double a given PI's interpolation range at the cost of a severely degraded linearity, the key innovation of the two-points PI is the prevention of the linearity degradation. The newly developed interpolation cells implement a k bit interpolation with a cell array of only 2^{k-1} instead of 2^k cells, with a minimum increase in the single interpolation cell's area compared to the reference PI. In order to increase the resolution to 10 bit, a novel implementation of a hybrid PI array, splitted into thermometrically and binary controlled parts, is presented. It differs in several aspects from conventional hybrid DAC implementations and is a key design aspect for low power designs with enhanced resolution.

DTC discussions in the literature primarily focus on performances such as static nonlinearity, resolution, operation frequency, or power consumption. However, dynamic effects that are triggered by DTC code activity lead to dynamic errors, which are visible as additional dynamic nonlinearity. Depending on the code activity and targeted linearity, they can have a non-negligible impact on the DTC application. The mechanisms leading to dynamic effects are analyzed in detail, identifying supply regulators with finite regulation bandwidth as major contributor. Therefore, a dynamic effects compensation circuit is developed that aims at mitigating dynamic errors at supply regulator level.

Circuit designs for the present thesis resulted in three test chips that were fabricated in 28 nm standard CMOS technology. The developed DTCs operate in a frequency range of 2–3 GHz and provide resolutions of up to 13 bit (up to 10 bit provided by the PI), equivalent to a time resolution of 48 fs for 2.5 GHz operation frequency. Test chip verification shows excellent matching between circuit simulations, analytical circuit models, and test chip measurements of the static DTC nonlinearity. Furthermore, the implemented dynamic effects compensation is validated to be functional, even if a detailed verification is limited by the instrument noise, which is in the same order of magnitude as the targeted DTC resolution.

Contents

List of Abbreviations	vii
List of Symbols	xi
1 Introduction	1
1.1 DTC Architectures and Circuit Design	2
1.1.1 Coarse Tuning Architectures	2
1.1.2 Fine Tuning Architectures	4
1.2 DTC Applications	5
1.2.1 Direct Digital Period Synthesis	6
1.2.2 Clock and Data Recovery Circuits (CDR)	7
1.2.3 DTC Assisted TDCs	9
1.2.4 Fractional-N Sub-Sampling PLLs (SSPLL) and Multiplying DLLs (MDLL)	10
1.2.5 Polar and Outphasing Transmitters	12
1.3 Motivation and Objectives	13
2 DTC Architecture and Characterization	15
2.1 Investigated Multistage DTC Architecture	15
2.1.1 Multi-Modulus Divider	16
2.1.2 Multiplexer and Delay Element	18
2.1.3 Phase Interpolator	19
2.1.4 Digital Data Path	20
2.2 DTC Performance Characteristics	20
2.3 Summary	23
3 Phase Interpolator Design and Modeling	25
3.1 Digitally Controlled Edge Interpolator	26
3.1.1 DCEI Model	28
3.2 Contention-Free Digitally Controlled Edge Interpolator	32
3.2.1 Design and Implementation	34
3.2.2 CF-DCEI Model	37
3.3 Digitally Controlled Two-Points Edge Interpolator	42
3.3.1 Design and Implementation	43
3.3.2 DCEI ² Model	48
3.4 Binary Bit Resolution Enhancement	50
3.4.1 Architecture of a Binary Extended Cell Array	51
3.4.2 Binary Unit Cell Implementation	53
3.5 Summary and Conclusion	55

4	Dynamic Effects in DTCs	59
4.1	Definition of Dynamic Errors and Dynamic INL	59
4.2	Root Causes of Dynamic Errors	60
4.2.1	Code-Dependent Current Consumption	62
4.2.2	Instantaneous Change of Average Current	63
4.2.3	Logic Current Consumption	65
4.2.4	Digital Control Signal Timing	65
4.2.5	Digital Control Signal Coupling	67
4.3	Dynamic Error Simulation	67
4.4	Compensation for Load Current Variations at Supply Regulator Level . . .	70
4.4.1	DCEI ² Compensation	71
4.4.2	MMD Compensation	73
4.4.3	Compensation Impact on Dynamic Effects	75
4.5	Summary and Conclusion	76
5	DTC Measurements	79
5.1	Measurement Setup	79
5.2	Quasi-Static CF-DCEI Nonlinearity	80
5.3	Quasi-Static DCEI ² Nonlinearity	82
5.3.1	INL Tuning	84
5.3.2	Binary Bit Implementation	85
5.4	Dynamic DTC Performance	86
5.4.1	DCEI ² Dynamic Effects Compensation	89
5.4.2	MMD Dynamic Effects Compensation	90
5.4.3	Dynamic Error Measurement Limitations	92
5.5	Summary and Conclusion	92
6	Conclusion and Outlook	95
	Appendices	99
A	DCEI Nonlinearity Model	101
B	CF-DCEI Nonlinearity Model	105
C	Switched Capacitor Fine Tuning Nonlinearity Model	113
	List of Figures	119
	List of Tables	123
	List of References	125
	List of Author Publications	135
	Acknowledgments	137

List of Abbreviations

ADC	Analog-to-digital converter
ADPLL	All-digital phase-locked loop
BBPD	Bang-bang phase detector
CDR	Clock and data recovery
CF-DCEI	Contention-free digitally controlled edge interpolator
CMOS	Complementary metal-oxide-semiconductor
CORDIC	COordinate Rotation DIgital Computer
CP	Charge pump
DAC	Digital-to-analog converter
DC	Direct current
DCDL	Digitally controlled delay line
DCEI	Digitally controlled edge interpolator
DCEI²	Digitally controlled two-points edge interpolator
DCO	Digitally controlled oscillator
DDPS	Direct digital period synthesis
DEL	Delay element
DFF	D flip-flop
DLL	Delay-locked loop
DNL	Differential nonlinearity
DPC	Digital-to-phase converter
DPS	Digital period synthesis
DTC	Digital-to-time converter
FCW	Frequency control word

FLL	Frequency-locked loop
FS	Full scale
FSR	Full scale range
IC	Integrated circuit
I-DAC	Current-steering digital-to-analog converter
INL	Integral nonlinearity
INTC	Interpolation cell
ISSCC	International Solid-State Circuits Conference
LDO	Low drop-out voltage regulator
LF	Loop filter
LO	Local oscillator
LSB	Least significant bit
LUT	Look-up table
MC	Monte Carlo
MDLL	Multiplying delay-locked loop
MMD	Multi-modulus divider
MOSFET	Metal-oxide-semiconductor field-effect transistor
MSB	Most significant bit
MUX	Multiplexer
NMOS	N-channel MOSFET
ODE	Ordinary differential equation
PA	Power amplifier
PCR	Periodic code ramp
PD	Phase detector
PI	Phase interpolator
PLL	Phase-locked loop
PMOS	P-channel MOSFET
PSD	Power spectral density

PVT	Process, voltage, and temperature variations
QVCO	Quadrature voltage controlled oscillator
RETC	Retention cell
RF	Radio frequency
RPM	Random phase modulation
RX	Receiver
SCCS	Short-circuit-current-suppression
SDR	Software-defined radio
SoC	System on a chip
SRAM	Static random-access memory
SSB	Single-sideband
SSPD	Sub-sampling phase detector
SSPLL	Sub-sampling phase-locked loop
TDC	Time-to-digital converter
TDL	Tapped delay line
TF	Transfer function
TX	Transmitter
VCDL	Voltage controlled delay line
VCO	Voltage controlled oscillator

List of Symbols

α	Normalized DTC code
$\Delta\phi_{\text{static}}$	DTC output phase difference due to code change
λ	Channel length modulation factor in Shichman-Hodges transistor model
$\mu_{n,p}$	Electron mobility in n/p doped silicon
ϕ	DTC output phase
ϕ_{dyn}	DTC output phase related to dynamic nonlinearity
ϕ_{in}	Input signal's phase in a phase filtering DLL
ϕ_{out}	Output signal's phase in a phase filtering DLL
ϕ_{pd}	Phase shift related to propagation delay t_{pd}
ϕ_{ref}	Reference signal's phase in a phase filtering DLL
ϕ_{static}	DTC output phase related to static nonlinearity
τ_{int}	RC time constant at interpolation node V_{int}
θ	Phase shift between $S_{1/2}$ in an outphasing transmitter
$\Delta\phi$	Phase enclosed between $I_{n1/2}$ of a phase interpolator input
b	Control word for binary interpolation cells
d_{acc}	Accumulated frequency control word
$\Delta i_{(k,k+j)}$	DTC current deviation from i_{nom} for code change $n = k \rightarrow k + j$
Δt	Time spacing between the rising edges of the two phase interpolator input signals
Δt_1	Interpolation range of the DCEI ² 's first interpolation
$\Delta t_{2,1/2}$	Interpolation range of the DCEI ² 's second interpolation
Δt_{c}	Coarse tuning time resolution
Δt_{uc}	Ultra coarse tuning time resolution
d_{sel}	MUX control signals in DDPS applications

$f_{\text{clk,MMD}}$	Clock signal generated by the MMD
$f_{\text{clk,MUX+DEL}}$	Clock signal generated by the MUX+DEL
$f_{\text{clk,PI}}$	Clock signal generated by the phase interpolator
f_{nom}	Nominal DTC output frequency
f_{offset}	Frequency offset at DTC output due to programming
f_{out}	DTC output frequency
f_{ref}	DTC reference frequency
i_{base}	Constant static DCEI ² current consumption
i_{bias}	LDO bias current
i_{clk}	Digital DTC current due to clocking
i_{comp}	LDO compensation current
i_{dig}	Total digital DTC current
$I_{\text{D,n/p}}$	Drain current of n- and p-type MOS transistor
$i_{\text{int},1}$	Static DCEI ² current consumption due to the first interpolation
$i_{\text{int},2}$	Static DCEI ² current consumption due to the second interpolation
i_{load}	Load current the DTC imposes on the LDO
i_{logic}	Digital DTC current due to logic circuitry
i_{nom}	Total DTC current
$i_{\text{nom},0}$	DTC current not influenced by code changes
$i_{\text{nom,coarse}}$	DTC current of all blocks processing the coarse tuned signal
$i_{\text{nom,fine}}$	DTC current of all blocks processing the fine tuned signal
$i_{\text{static,DCEI}^2}$	Total static DCEI ² current consumption
j_{coarse}	DTC code change in coarse tuning stage
k_{bin}	Phase interpolator resolution of binary controlled cells in bit
k_{coarse}	DTC coarse tuning number of bits
k_{DTC}	DTC number of bits
k_{fine}	DTC fine tuning number of bits
k_{MMD}	MMD number of bits

$k_{\text{MUX+DEL}}$	MUX+DEL number of bits
k_{PI}	Phase interpolator number of bits
k_{therm}	Phase interpolator resolution of thermometrically controlled array in bit
k_{total}	Total phase interpolator resolution in bit
n	DTC code word
n_{cycle}	DTC output cycle number after code change
$n_{\text{dyn,err}}$	Code sequence for dynamic error DTC tests
n_{mod}	Code sequency for modulation dynamic error DTC tests
n_{ramp}	Code sequency for frequency synthesis dynamic error DTC tests
t_{cross}	Signal's crossing time of threshold voltage $V_{\text{th,inv}}$
t_{d}	DTC output delay
$t_{\text{d,DEL}}$	Propagation delay of DEL path in MUX+DEL stage
$t_{\text{d,inv}}$	Propagation delay of inverter
$t_{\text{d,LSB}}$	DTC time resolution
$t_{\text{d,LSB,coarse}}$	DTC coarse tuning time resolution
$t_{\text{d,MUX}}$	Propagation delay of MUX path in MUX+DEL stage
$t_{\text{d,therm}}$	Delay for step between two neighboring thermometrically controlled DCEI cells
t_{int}	Rise time for $V_{\text{int}} : 0 \rightarrow V_{\text{th,inv}}$
$t_{\text{int},0}$	Minimum rise time at V_{int}
$t_{\text{int},0,\lambda \neq 0}$	Minimum rise time at V_{int} for $\lambda \neq 0$
$t_{\text{int},1,0}^{\text{sc}}$	Minimum of $t_{\text{int},1}^{\text{sc}}$ due to programming
$t_{\text{int},1}^{\text{sc}}$	Delay until $V_{\text{out}}^{\text{sc}}$ starts to switch in the switched capacitor based fine tuning
$t_{\text{int},2,0}^{\text{sc}}$	Minimum of $t_{\text{int},2}^{\text{sc}}$ due to programming
$t_{\text{int},2}^{\text{sc}}$	Relevant fall time at $t_{\text{int},1}^{\text{sc}}$ that influences the linearity of the switched capacitor based fine tuning
$t_{\text{inv,min}}$	Minimum inverter delay of a given technology
$t_{\text{out},0}^{\text{sc}}$	Minimum rise time at $V_{\text{out}}^{\text{sc}}$

$t_{\text{out}}^{\text{sc}}$	Rise time at $V_{\text{out}}^{\text{sc}}$
t_{pd}	Propagation delay
$t_{\text{r,f}}$	Rise/fall time of a rail-to-rail signal
$B_{1/2}$	Binary controlled 50% DCEI ² interpolation cell
$B_{1/4}$	Binary controlled 25% DCEI ² interpolation cell
B_x	Binary controlled DCEI ² interpolation cell
C	Capacitance
$C_{C,3}$	MMD division-by-3 compensation capacitance
$C_{C,5}$	MMD division-by-5 compensation capacitance
$C_{C,\text{LSB}}$	MMD LSB compensation capacitance
C_{int}	Capacitance of phase interpolator's interpolation node V_{int}
C_{load}	Decoupling capacitor of LDO supply regulator
C_{max}	Maximum value for C_{tune} in the switched capacitor based fine tuning
C_{min}	Minimum value for C_{tune} in the switched capacitor based fine tuning
C_{ox}	MOSFET gate oxide capacitance
C_{PG}	Gate capacitance of LDO pass-gate transistor
C_{tune}	Tuning capacitance of the switched capacitor based fine tuning
DEL_{out}	Delay element path output of MUX+DEL stage
$\Delta T_{1/2}$	Clock signal skew in source synchronous CDR circuits
$\Delta V_{\text{PG,div-3}}$	Voltage spike on V_{PG} due to MMD division-by-3 compensation
$\Delta V_{\text{PG,div-5}}$	Voltage spike on V_{PG} due to MMD division-by-5 compensation
$\Delta V_{\text{PG,LSB}}$	Voltage spike on V_{PG} due to MMD LSB compensation
I_1	Initial current of current integrating phase interpolators
$I_{\text{D,off}}$	Drain current in cut-off region
$I_{\text{D,sat},0}$	Drain current in saturation region for $\lambda = 0$
I_{max}	Maximum current of current integrating phase interpolators
$\text{In}_{1/2}$	Input signals of DTC fine tuning stages
INL_{dyn}	Dynamic INL

$\overline{\text{INL}}_{\text{dyn}}$	Average dynamic INL for each DTC code
$\sigma(\text{INL}_{\text{dyn}})$	Standard deviation of INL_{dyn} for each code
$\overline{\sigma(\text{INL}_{\text{dyn}})}$	Average of $\sigma(\text{INL}_{\text{dyn}})$ over all DTC codes
$\text{INL}_{\text{int},1/2}$	DCEI ² INL of first/second interpolation
INL_{max}	Positive or negative peak value of $\text{INL}[n]$
$\text{INL}_{\text{max}}^{\text{sc}}$	Positive or negative peak INL of switched capacitor based fine tuning
$K_{n,p}$	$\mu_{n,p}C_{\text{ox}}$
L_{eff}	Effective transistor length
$\mathcal{L}(f)$	Phase noise at frequency f
M	Relevant DTC code history depth for dynamic effects
$\text{MMD}_{\text{out},1/2}$	MMD output signals
MUX_{out}	Multiplexer path output of MUX+DEL stage
N	Maximum DAC code, e.g. for DTCs or PIs
$P_{\text{LO}}(f_{\text{LO}})$	Power of output carrier signal with frequency f_{LO}
$P_{\text{noise}}(f)$	Power of noise floor at frequency f
P_{nom}	Nominal output signal power
R	Resistance
R_{1-4}	Logic control signals of the CF-DCEI's retention cells
S_{1-4}	Logic control signals of the DCEI ² /CF-DCEI's interpolation cells
Sel_i	Select signal of i^{th} phase interpolator unit cell
$\text{Sel}_{i,1/2}$	Select signals of i^{th} DCEI ² interpolation cell
$\overline{S_{1-2}}$	Inverted logic control signals of the DCEI ² 's interpolation cells
$S_{\tau}(f)$	Jitter power spectral density at frequency f
$S_{\tau,s}(f)$	Spectral density of jitter at frequency f
T	Period of a signal
T_{acc}	Clock to output delay of accumulator in DDPS applications
T_{cell}	Thermometrically controlled DCEI ² interpolation cell
$T_{(k,k+j)}$	Period for DTC code change $n = k \rightarrow k + j$

T_{nom}	Nominal period
T_{out}	Period of DTC output signal
T_{ramp}	Period of DTC ramp generator
T_{trigger}	Period of DTC trigger signal for external measurements
T_{VCO}	Period of VCO output signal
V	Voltage
$V_{\text{C},3}$	MMD division-by-3 compensation voltage
$V_{\text{C},5}$	MMD division-by-5 compensation voltage
V_{casc}	LDO slow-loop output voltage
$V_{\text{C,LSB}}$	MMD LSB compensation control voltage
V_{DD}	Supply voltage
$V_{\text{DD,ext}}$	External supply voltage
V_{DS}	Drain-source voltage
V_{GS}	Gate-source voltage
V_{int}	Voltage at interpolation node of DTC fine tuning circuits
$V_{\text{int},1}$	First interpolation node of DCEI ²
$V_{\text{int},2}$	Second interpolation node of DCEI ²
$V_{\text{int}}^{\text{sc}}$	Tunable net of switched capacitor based fine tuning
V_{out}	LDO output voltage
$V_{\text{out}}^{\text{sc}}$	Output net of switched capacitor based fine tuning
V_{PG}	Voltage at LDO pass-gate transistor gate
V_{ref}	Bandgap reference voltage
V_{SS}	On-chip ground potential
V_{sup}	DTC supply voltage
$V_{\text{th,inv}}$	Threshold voltage of CMOS inverter
V_{tune}	Tuning voltage of DLL
$\text{VCO}_{\text{p/n}}$	Differential VCO output signals
W_{eff}	Effective transistor width

$Z_{1/2}$	Complex outphasing signals of an outphasing transmitter
Z_{out}	Complex output signal of an outphasing transmitter

1 Introduction

Several system architectures in modern system on a chip (SoC) integrated circuits (IC) require circuit blocks for frequency synthesis or modulation, such as for generation of digital clock signals or for wireline/wireless communication systems. As modern process technologies favor digital circuits, systems are preferably implemented digitally and signals move from the analog to the digital domain wherever possible. Therefore, it would be desirable to implement fully digital circuits for the generation of arbitrary digital signals, i.e., clock signals of constant frequency or modulated signals. A circuit that allows the generation of such signals, while providing the possibility of a fully digital implementation, is the digital-to-time converter (DTC). DTCs, also called digital-to-phase converters (DPC) or phase rotators, apply a time delay t_d (equivalent to a phase shift ϕ) on a reference signal, based on a digital input code. As their output is usually a periodic clock signal, phase shift corresponds to time delay according to the output signal's period T_{out} :

$$\frac{\phi}{2\pi} = \frac{t_d}{T_{out}}. \quad (1.1)$$

Due to the scaling of CMOS process technology, the supply voltage is lowered to reduce power dissipation. This results in loss of dynamic range for conventional analog-to-digital converters (ADC) and digital-to-analog converters (DAC). However, transistors also get faster, allowing time processing circuits as time-to-digital converters (TDC) or DTCs to increase their time resolution and therefore benefit from process scaling.

DTCs belong to the class of DACs, where the analog domain is time or phase. In most applications they operate on a rectangular input reference clock signal f_{ref} and produce a rectangular output clock signal f_{out} . Fig. 1.1(a) shows an overview of the DTC block. The digital code n controls the time delay (or the phase shift) of the output signal, while the relation between f_{ref} and f_{out} is determined by the DTC circuit architecture and its programming. Fig. 1.1(b) plots the relation between input and output waveforms exemplary for the case of $f_{out} = f_{ref}$, static DTC code n , and an output phase range of

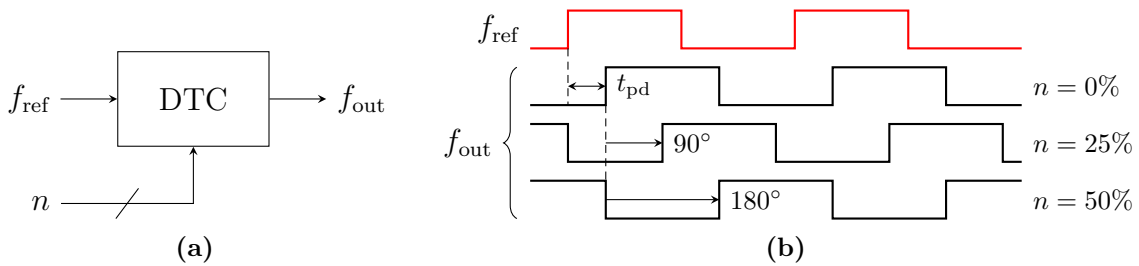


Figure 1.1 – Basic DTC operation: (a) top level overview on the DTC, and (b) example for relation between input reference signal and DTC output signal.

$0 - 2\pi$ for a code range of $0 - 100\%$. For $n = 0$ the delay between the input and output signal is only determined by the propagation delay t_{pd} that the signal takes to propagate through the DTC. If the code is increased to $n = 25\%$, the output is shifted by $\pi/2 = 90^\circ$ compared to the case of $n = 0$ (assuming a perfectly linear DTC).

The further introduction is structured as follows: Section 1.1 reviews and compares popular DTC architectures and discusses their advantages and drawbacks. Afterwards Section 1.2 gives an overview on typical DTC applications, highlighting the benefits of DTCs when used to enhance or replace conventional circuit architectures. Finally, Section 1.3 outlines the present thesis and defines the key research objectives.

1.1 DTC Architectures and Circuit Design

State-of-the-art DTCs show a resolution of down to $t_{d,LSB} = 19$ fs [1] and operation frequencies from the megahertz to the gigahertz domain. The best architecture can be chosen by trading off operation frequency, resolution, and jitter requirements, and depends highly on the targeted application. Before diving into the applications, typical DTC circuit architectures are reviewed.

If DTCs target high resolution, they are usually segmented into coarse and fine phase tuning stages. Common architectures for coarse tuning are the delay-locked-loop (DLL), divider, or multiphase voltage controlled oscillator (VCO) based approach, followed by a multiplexer (MUX). Their resolution $t_{d,LSB}$ is limited either by the minimum inverter delay $t_{inv,min}$ of the respective technology or by the frequency of their reference input f_{ref} . To overcome this limitation, subsequent fine tuning stages are used to provide a resolution of $t_{d,LSB} \ll t_{inv,min}$. For fine tuning switched capacitor circuits, phase interpolators (PI) or oversampling phase filters are used. In the following, the most popular concepts are described briefly and their advantages and limitations are discussed.

1.1.1 Coarse Tuning Architectures

Coarse tuning blocks aim at providing a wide dynamic range with a coarse phase resolution $t_{d,LSB,coarse}$ and high linearity. As this resolution is not sufficient for most applications, coarse tuning stages are constructed in a way that subsequent fine tuning blocks can be placed. The coarse tuning blocks discussed in the following have a resolution of k_{coarse} bit and generate $N = 2^{k_{coarse}}$ evenly phase shifted signals.

DLLs as shown in Fig. 1.2(a) consist of a voltage controlled delay line (VCDL) that is built from buffers or inverters, and regulate their delay in a negative feedback loop to ensure a delay range of 2π [3–10]. The phase of the output signal is compared to the reference input by a phase detector (PD). The PD output is then low pass filtered by the loop filter (LF), which controls a charge pump (CP) to adjust the VCDL's delay. Another possibility of delay control is current starving of the single delay elements [6]. The MUX taps N evenly phase shifted signals which are generated from a reference with $f_{ref} = f_{out}$. The advantage of this architecture is the regulated 2π range, that is accurate over process, voltage, and temperature variations (PVT). One drawback of this architecture is the long total delay of the buffer chain and hence accumulated jitter [11]. With an increasing number of delay stages the variance of each stages' delay increases along the line, with

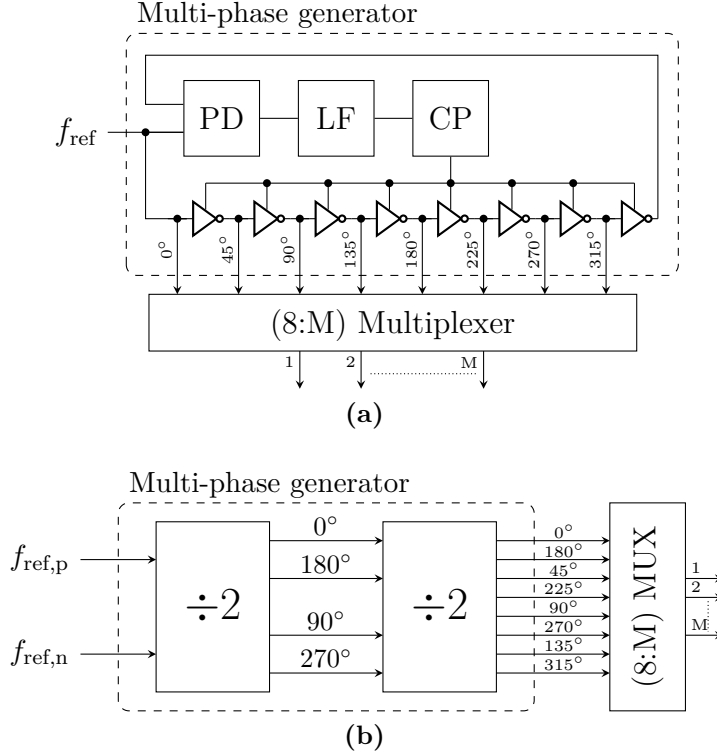


Figure 1.2 – DTC coarse tuning architecture examples with k_{coarse} bit resolution: (a) DLL based coarse tuning, and (b) divider based coarse tuning [2].

maximum in the middle of the delay line [12]. The non-ideal delay of each delay stage is one source of nonlinearity in the DLL architecture. Another implementation featuring the described delay line is the tapped delay line (TDL), which lacks the DLL’s control loop. This reduces the design complexity, saving the control loop design, but increases the nonlinearity and PVT sensitivity, which needs to be calibrated then [7, 13].

Divider based coarse tuning stages generate N signals with a phase shift of $\min. T_{\text{ref}}/2$, followed by a MUX [2, 7, 13, 14], as shown in Fig. 1.2(b). An I/Q divider topology that operates on pseudo-differential input reference signals allows to generate signals of 0° and 90° , plus their pseudo-differential counterparts with 180° and 270° . As in case of the DLL, a MUX selects the coarse tuning output from the divider output signals. Compared to the DLL, the divider achieves low jitter and does not require a control loop. The implementation is fully digital, without the need of analog control blocks such as a CP. However, the input reference signal needs to have multiple times the frequency of the generated output signal.

As last coarse tuning approach, a MUX could tap directly N phase shifted output signals of a VCO or a ring oscillator [15, 16], for example $N = 4$ for a quadrature VCO (QVCO). The advantage is the use of the same circuit for reference clock generation and coarse tuning. However, to date no coarse tuning stages with a high order of N have been reported, that use this type of architecture.

Depending on the fine tuning architecture, which in the following examples require $1 \leq M \leq 3$ adjacent phases as input, the MUX is implemented as $(N : M)$ MUX. By changing the MUX implementation accordingly, each coarse tuning concept can be

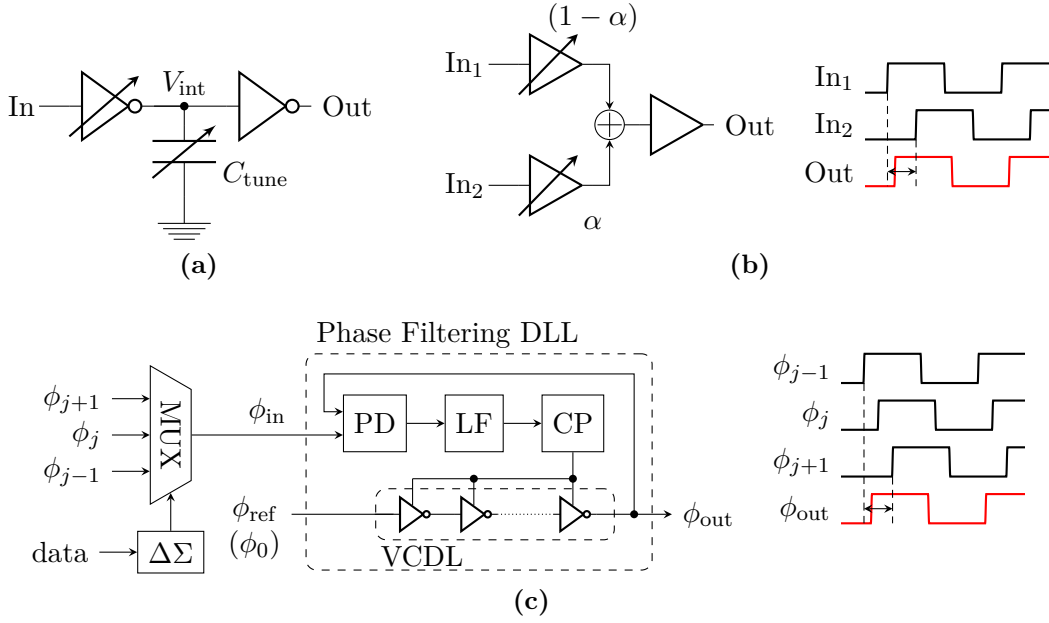


Figure 1.3 – DTC fine tuning architectures: (a) switched capacitor based delay cell or DCDL, (b) PI [2], and (c) DLL based phase filter [16].

combined with any of the following fine tuning architectures.

For high orders of N or a high operation frequency the MUX implementation can be complicated. In addition to the coarse tuning block itself, it can contribute to nonlinearity. In general, it can be seen as a phase selector, for which the standard CMOS MUX is not necessarily the best implementation. Therefore, other implementations include MUX like circuits based on D flip-flops (DFF), thus sensitive to rising edges of the coarse tuning output [17–19], or a combination of logic gates, implementing MUX functionality [8].

1.1.2 Fine Tuning Architectures

The following fine tuning architectures take the M output signals of the coarse tuning as input, and apply a phase tuning with high resolution. Therefore, their dynamic range should be limited to the resolution $t_{d,LSB,coarse}$ of the coarse tuning block, to preserve monotonicity over all DTC stages.

A fine tuning based on delay cells as shown in Fig. 1.3(a) tunes the RC constant of a node to modify the signal’s zero crossing time [3, 7, 9, 10, 13, 15, 20–32], with an optional tunable inverter to adjust for PVT [7]. They are also referred to as switched capacitor based fine tuning or digitally controlled delay line (DCDL). The advantage of delay cells is their high linearity, as linear tuning of C_{tune} results in linear shifting of the zero crossing. The dominating source of nonlinearity is the C_{tune} dependent slope at V_{int} , that modulates the turn-on time of the output buffer [1, 9]. Major drawbacks of this architecture are unwanted supply modulation through a code-dependent current consumption, high jitter through the degradation of the (dis)charge slopes, and a not well-defined delay range. Replica paths with inverted codes are necessary to equalize the current consumption over code [22–24, 26, 29] and calibration engines are used to cope with the undefined range [7].

DTCs integrated in phase-locked loops (PLL) use this fine tuning type commonly without coarse tuning [22–28, 30, 32, 33], relying only on a single stage DTC design. In general, multiple delay cells can be cascaded to ensure fast rise/fall times of the propagating signal, which improves sensitivity to supply and thermal noise [34] and prevents possible pulse swallowing at coarse tuning code changes [7].

PIs as shown in Fig. 1.3(b) have two input signals of identical frequency ($M = 2$), temporally shifted against each other by Δt , and produce an output signal weighted in time (or phase) domain from the inputs [1, 2, 6, 14, 16, 35–40]. The interpolation cells, to which the input signals are connected, are visualized as tunable buffers, but can be implemented differently. The PI output signal covers exactly Δt enclosed between $In_{1/2}$ over code α , but shows high systematic nonlinearity. A harmonic rejection technique has been implemented to linearize the PI [2], but at expense of a lower slew rate of the internal signals and hence, jitter. The present thesis focuses on PI based DTCs and gives a detailed analysis of different PI types and their systematic nonlinearity.

With a DLL as phase filter after a ($N : 3$) MUX, oversampling can be explored to increase the DTC resolution [7, 16]. The major difference of the phase filtering DLL from Fig. 1.3(c) compared to a regular DLL is the separation of PD and voltage controlled delay line (VCDL) input (compare Fig. 1.2(a) of the DLL based coarse tuning). The VCDL input ϕ_{ref} determines the frequency of the output signal, while ϕ_{in} determines its phase. To operate the DLL in a meaningful fashion, ϕ_{ref} and ϕ_{in} need to have an (on average) identical frequency. The reference ϕ_{ref} can be for instance equal to one of the coarse tuning signals (in front of the coarse tuning output MUX). The regulation loop consisting of PD, LF, and CP has a low pass characteristic due to the loop filter and locks the phase of ϕ_{out} to ϕ_{in} . If ϕ_{in} changes, ϕ_{out} follows with a delay determined by the control loop's bandwidth. The filtering effect of the DLL allows to switch between input signals with adjacent phases, and create an output signal with an average phase. This allows to apply oversampling and $\Delta\Sigma$ modulation well known from PLL implementations for fractional- N frequency synthesis [41]. The waveforms on the right hand side of Fig. 1.3(c) indicate the range of ϕ_{out} 's phase for a given set of input signals. As it is well defined by the spacing of the input signals, it does not need further calibration. In [16] an impressive resolution of 14 bit is reported, however, new DTC codes cannot be applied immediately due to the phase filter's settling time. This reduces its practical use to applications with sufficiently slow changing input codes. A similar fine tuning was implemented in [42], where the ($N : 3$) MUX in the coarse tuning stage was combined with the ($3 : 1$) MUX in front of the phase filter to a ($N : 1$) MUX. This removes one MUX from the signal path, thus removing sources of jitter and nonlinearity, as well as saving power.

1.2 DTC Applications

Many applications exist where DTCs are used to replace or enhance traditional architectures. Most of them came up only in the last decade and gained popularity through increasing DTC performance, resulting from architectural DTC enhancements and smaller technology nodes. Applications include usage in direct digital period synthesis (DDPS), clock-and-data-recovery circuits (CDR), in the feedback or reference path of a PLL, as fine delay in TDCs, or as direct phase modulators in polar or outphasing transmitters.

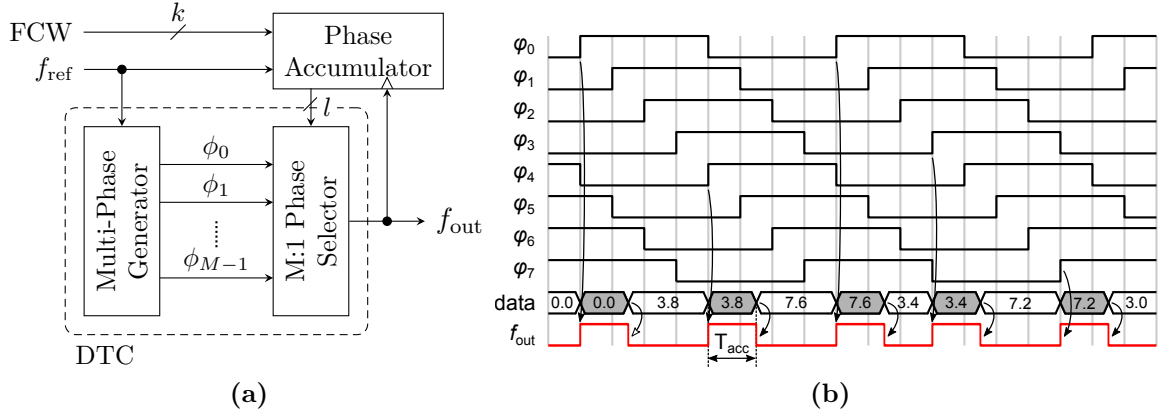


Figure 1.4 – DDPS frequency synthesis: (a) DDPS circuit architecture [18], and (b) example operation of a 3 bit DDPS block for generation of $f_{\text{out}} > f_{\text{ref}}$.

One of the first DTC implementations was presented at ISSCC in 1990 [43], where a 5 bit DLL with subsequent MUX was used in the context of CDR.

While DTCs in PLLs often operate close to the reference oscillator’s frequency, CDR and transmitter DTCs are required to operate at frequencies in the gigahertz range. This fact reflects in the architecture types chosen for the different applications. The following sections briefly introduce the mentioned DTC applications and highlight the advantages compared to prior DTC-less implementations.

1.2.1 Direct Digital Period Synthesis

DDPS, also called digital period synthesis (DPS), is a technique that allows to synthesize clock signals (including spread spectrum clocks) for use in digital clocking or in communication systems in a purely digital manner. It was firstly introduced by Mair et al. in 2000 [44]. In principle, the circuit re-combines M signals of identical frequency f_{ref} but different phases $\phi_0, \phi_1, \dots, \phi_M$ to generate an output signal with a different, mostly higher frequency $f_{\text{out}} > f_{\text{ref}}$. Digital programming allows then the control of f_{out} .

As this architecture synthesizes periods by means of changing the output signal’s phase with a DTC, the relation between phase and frequency is worth a brief look before discussing the circuit architecture. For continuous time signals, a frequency offset f_{offset} is related to a phase change $\Delta\phi$ by

$$f_{\text{offset}} = -\frac{1}{2\pi} \frac{\Delta\phi}{dt}, \quad (1.2)$$

where $\Delta\phi$ is the phase change that needs to be applied in every clock cycle $dt = 1/(f_{\text{out}} + f_{\text{offset}})$ [3]. Vice versa $\Delta\phi$ is obtained by integrating (equivalent to accumulating in digital processing) f_{offset} .

The heart of the DDPS systems is a DTC as presented in the coarse tuning section. Most architectures use an M phase generator followed by an $(M:1)$ phase selector as shown in Fig. 1.4(a) [3, 8, 17–19, 34, 44, 45]. The phase generator is most often implemented as DLL, but phase signals can also be tapped directly from an oscillator. Its output phases are exemplary plotted for $M = 8$ in Fig. 1.4(b). The phase selector forwards one of these signals to its output, based on a digital control word. The DTC programming is derived

Table 1.1 – Accumulator output for $M = 8$ and $FCW = 3.8$.

Clock Cycle	FCW	d_{acc}	d_{sel}
1	3.8	0.0	0
2	3.8	3.8	4
3	3.8	7.6	0
4	3.8	3.4	3
5	3.8	7.2	7
6	3.8	3.0	3

from accumulating (or integrating) a frequency control word (FCW), which is equivalent to f_{offset} from (1.2). The accumulator is clocked by f_{out} , which satisfies the assumption that the period needs to change on rate of $f_{\text{ref}} + f_{\text{offset}}$.

The waveforms in Fig. 1.4(b) and the related Table 1.1 show an example for $FCW = 3.8$, leading to an average period of $f_{\text{out}} \sim 2.11f_{\text{ref}}$. The fractional FCW is accumulated to d_{acc} and then truncated to d_{sel} , which has a data width of l and is connected to the MUX control (the fractional MSB of d_{acc} is added to its integer part to be precise). This leads to periods of 3 – 4 $t_{\text{d,LSB}}$, with a total average of 3.8 $t_{\text{d,LSB}}$. The clock to output delay of the accumulator T_{acc} defines the duty cycle (which is not at 50%) and limits the maximum possible output frequency. As this programming scheme allows multiple code changes per reference cycle, f_{out} can be much higher than f_{ref} .

As single-stage phase selectors can only implement a coarse DTC resolution with reasonable design effort (max. of 5 bit reported in [8]), two-stage DTC architecture were presented [3, 8]. Here a subsequent switched capacitor based fine tuning [8] or a PI [3] is employed to increase the resolution. Another two-stage DTC differs from the architectures described above and employs a multi-modulus divider (MMD) for coarse tuning and a DCDL for fine tuning [34]. This circuit omits high order phase selectors, however, it requires $f_{\text{ref}} > f_{\text{out}}$.

From system perspective, level and location of systematic spurs in the output spectrum can be related to the DTC's quantization noise or nonlinearity [14, 46, 47]. Therefore, DTCs with high resolution and low nonlinearity are preferred. Moreover, periodicity of the DTC code sequence is visible as spurs in the spectrum. The spurs can be reduced by randomizing the DTC programming through an accumulator implemented as 1st or 2nd order $\Delta\Sigma$ -modulator [17–19], or by applying random dithering [33].

As DDPS is an open loop system, it can change its output frequency in a single output clock cycle. This fact and the possibility of a wide frequency range are the main advantages compared to PLLs. In addition, multiple DDFSs can share the same reference or multi-phase generator. This enables the generation of multiple clocks at different frequencies from the same PLL [34] or DLL [8], thus reducing the number of on-chip synthesizers as well as moving clock generation to a fully digital domain.

1.2.2 Clock and Data Recovery Circuits (CDR)

Wireline inter-chip communication systems aim continuously at higher data rates. This imposes design challenges on CDR circuits, which are implemented on receiver (RX) side to recover the transmitted data sequence from the distorted input signal together with its

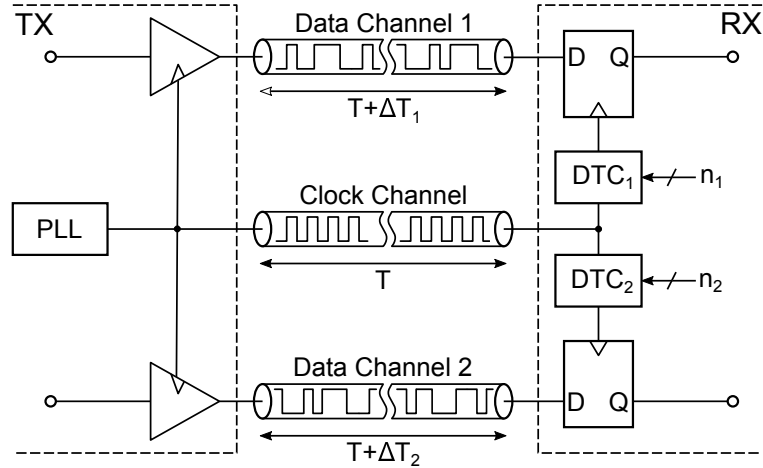


Figure 1.5 – Source-synchronous interface with DTC phase adjustment [16].

clock signal.

Most wireline transmission systems are source-synchronous or source-asynchronous systems (also called plesichronous systems). In the synchronous case, data is transferred together with the reference clock signal, while in the asynchronous case the RX and transmitter (TX) chips generate their own reference frequencies, leading to a possible frequency shift between transmitting and receiving clock.

Multi-channel source-synchronous interfaces transmit data on multiple channels and a clock signal in a separate channel, as shown in Fig. 1.5 [16]. The imperfect matching and spacial channel separation on RX and TX side lead to skew between the data and clock signals, labeled here as $\Delta T_{1/2}$. On the RX side, the CDR circuit needs to correct the clock signal's phase for the skew $\Delta T_{1/2}$ to sample the incoming data at the ideal time. For this purpose, each channel can shift the reference clock with a DTC [16].

Source-asynchronous systems need to adjust the frequency on top of a possible phase shift. Instead of using multiple PLLs at RX side to operate the CDR on several channels, a single PLL is used for reference clock generation and DTC can be used for phase and frequency correction [48]. As a slight frequency shift can be seen as continuous phase shift (see (1.2) for the relation between phase and frequency), it can also be corrected by the DTC. The DTC is required to allow modulo 2π operation, which enables continuous phase shifts without unwanted wrap-arounds. Attractive circuits for this purpose are PIs or quadrature PIs, where quadrature refers to four input signals, shifted against each other by 90° [48–50], such as generated by a DTC coarse tuning stages based on a QVCO with subsequent (4 : 2) MUX. As the CDR's PIs operate mostly on sinusoidal signals, their linearity is much higher than in systems with digital signals and steep edges, where nonlinearity is the major drawback of PIs. Apart from this application type, PIs usually operate on digital signals. On DTC side the design focus is especially on the PI, as it needs to operate at data rates in the multi gigahertz range for state-of-the-art wireline transmission.

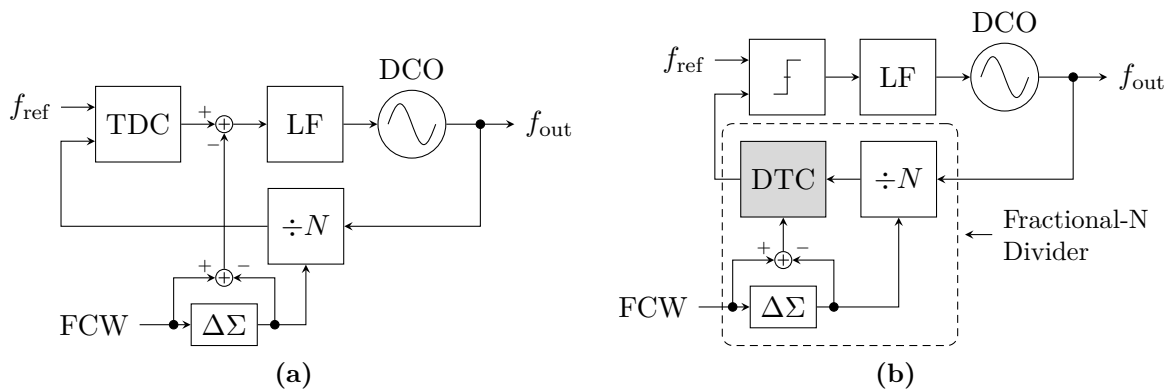


Figure 1.6 – Fractional-N ADPLL implemented with (a) integer-N divider and TDC, and (b) integer-N divider, DTC to realize fractional-N operation, and 1 bit TDC implemented as comparator.

1.2.3 DTC Assisted TDCs

Fig. 1.6(a) shows the well known all-digital PLL (ADPLL), where TDCs are used as phase detectors to allow the fully digital implementation of the LF and the use of a digitally controlled oscillator (DCO) [51–54]. The ADPLL enables fractional-N operation through $\Delta\Sigma$ -modulation of the integer-N divider in its feedback path. The divider control switches between different integer division ratios, resulting in an averaged output frequency through the low pass characteristic of the LF. The unavoidable error between the fractional FCW and the actual integer division ratio is subtracted from the TDC output to reduce the code activity in front of the LF [30]. The TDC is one of the key blocks in the ADPLL’s control loop. It requires high design effort and consumes a significant portion of the overall power. Furthermore, the generated fractional spurs depend mainly on its nonlinearity as well as its resolution.

If the integer-N divider is replaced by a fractional one, a bang-bang phase detector (BBPD) would suffice as TDC replacement. Fractional-N division can be realized by placing a DTC subsequently to the integer-N divider as depicted in Fig. 1.6(b), where the DTC adds the fractional part to the integer-N division. The phase error derived from the divider control word is fed to the DTC (phase error is obtained from frequency error through integration), which delays the signal accordingly. As DTCs have a certain quantization, the TDC could at least be relaxed in terms of detection range. The TDC range can now be in the domain of the DTC’s resolution instead of the DCO period, resulting in a significantly simplified design.

A first approach was introduced in [55], where a 4 bit DLL based DTC was connected in series to the integer-N divider of the feedback loop, allowing to forward an intermediate divider output to the TDC. This allows to reduce the TDC range by four MSBs, simplifying the design and reducing the power consumption. This approach was taken one step further in [56], where a 9 bit switched capacitor (DCDL) based DTC allows to reduce the TDC range to only 8 ps. In [30, 57, 58] this concept was finally extended to a 10 bit switched capacitor based DTC, enabling to reduce the TDC to a BBPD. This allows to use a simple comparator as 1 bit TDC, solving the issues of TDC nonlinearity and resolution [59].

Another solution to the same problem is the use of a DTC in the reference instead of

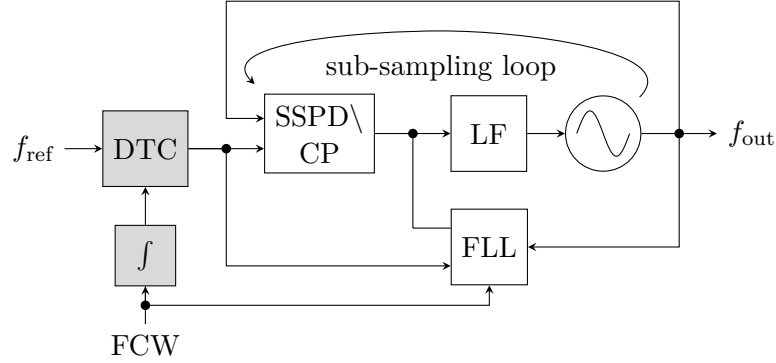


Figure 1.7 – DTC-based fractional-N sub-sampling PLL [65].

the feedback path. It was first introduced in [28], where a sample based counter is used as phase detector. The ADPLL is restricted to integer-N mode if the DTC is deactivated, fractional-N mode is enabled when the DTC is used to a-priori delay the reference edges according to an accumulated FCW. The DTC is realized as digitally controlled Vernier delay line (similar to a DCDL), which is in principle a series of switched capacitor DTC cells. In [22, 33, 60–62] the reference path DTC is used to reduce the detection range of the TDC, resulting in the advantages discussed above.

While the discussed approaches reduce the requirements of the TDC regarding range, resolution, and nonlinearity, the DTC design moves into focus. At DTC level, resolution and nonlinearity can be handled with less design effort and lower power overhead [59]. Adaptive digital pre-distortion is applied in order to reduce the nonlinearity and adjust the delay range over PVT [57]. To keep the power and phase noise advantage of the DTC-based approach, the correction is only applied in the digital domain [59]. In addition, $\Delta\Sigma$ -modulation can be used to overcome the limitations of the DTC resolution [63]. The full DTC range needs to cover the maximum expected error from $\Delta\Sigma$ -control at the divider output, plus a margin for PVT [57], which is in the order of multiple VCO periods. Overall, BBPD based ADPLLs can achieve identical spur/noise performance while reducing the power and complexity compared to TDC based ADPLLs [64].

1.2.4 Fractional-N Sub-Sampling PLLs (SSPLL) and Multiplying DLLs (MDLL)

One step further in the direction of TDC assistance in the reference path allows the DTC to generate a shift of the reference clock to enable fractional-N operation. However, this technique was not explored for ADPLLs, but to enable fractional-N operation in sub-sampling PLLs (SSPLL) and multiplying DLLs (MDLL). Both, SSPLLs and MDLLs, are attractive architectures for clock generation, as they offer low power and low noise. In the following, the working principle of their integer-N version is recapped briefly, followed by a discussion of the DTC extension that enables fractional-N operation for both architectures.

The first SSPLL was published in 2009 [66]. Its block diagram is shown in Fig. 1.7, where the DTC is assumed to be bypassed for now and some digital processing on the FCW is left out for simplicity. It has two control loops, a sub-sampling loop and a frequency-locked loop (FLL), that share the same LF. The FLL resembles a regular PLL control loop and

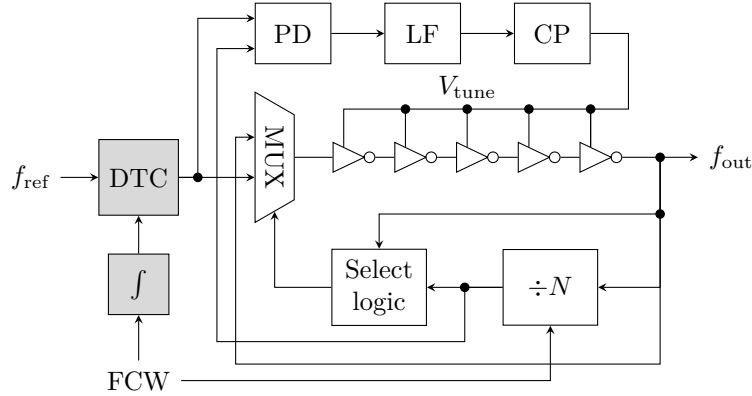


Figure 1.8 – DTC-based fractional-N MDLL [69].

consists of an integer- N divider, a PD, and a CP. It is used to lock the oscillator to the desired target frequency for start-up purposes. After locking, the FLL is disabled to save power, and the sub-sampling loop remains to be the active control loop. The sub-sampling phase detector (SSPD) compares the phase of f_{ref} and f_{out} at every rising edge of f_{ref} . As the divider is removed from the feedback loop, its power consumption and generated noise are removed from the system. However, as the output is compared directly to the reference, this system is limited to integer- N operation.

The MDLL was first published in 2002 [67], and its block diagram is shown in Fig. 1.8 where the DTC is assumed as bypassed and some digital processing on the FCW is left out for simplicity again. The MDLL consists of an odd number of subsequent inverters (in this example five) and one multiplexer (MUX), and its overall propagation delay can be tuned by the voltage V_{tune} . The tuning voltage is the output of the control loop consisting of PD, LF, and CP (optionally implemented digitally as DAC). It locks the output phase to the reference phase. Every N^{th} output cycle the select logic controls the MUX to forward a reference edge into the DLL, removing the accumulated jitter. While it brings the advantage of lower output noise up to a frequency offset of $f_{\text{ref}}/2$ (a much higher offset than the loop filters of PLLs usually provide), the MDLL suffers mainly from two problems [68]: first, f_{out} can only be changed in integer multiples of f_{ref} , and second, the timing of the phase injection needs to be very accurate (low phase offset in the phase detector) as else strong reference spurs occur in the output spectrum.

While both architectures provide advantages in terms of noise and power, they face the same limitation of only integer- N operation. In 2014, three groups of authors explored the DTC as a means to extend the SSPLLs / MDLLs to fractional- N subsampling systems: the first fractional- N SSPLL was published by [65, 70], followed only a few months later by [26, 29, 71]; the fractional- N MDLL was published simultaneously with the first PLL publication [68, 69]. All methods use a DTC as in Fig. 1.7 and 1.8 for a frequency shift on the reference clock f_{ref} , which effectively keeps the integer- N operation of the system, but alters the reference frequency to enable overall fractional- N operation. The operation resembles the DDPS with one main difference in the DTC architecture: while the DDPS architectures allow multiple code changes per reference cycle, only one phase change for each reference edge is allowed here. However, this is no limitation, as only the fractional frequencies need to be generated, whereby the required frequency shift is limited. For

MDLL systems, fractional-N operation was already published in 2012 [72], however, it is not DTC-based and allows only a coarse frequency resolution of ~ 1 MHz.

The DTC-based reference shift enables the tuning of the output frequency with fine resolution. As the DTCs operate at reference frequency, they can be implemented in a power efficient way. Depending on the digital control, the DTC should cover a wide delay range of multiple output cycles of the system (VCO/DCO or MDLL cycle) over PVT variations (e.g. 2-3 in [68], and 5 in [26]). As they operate directly on the reference clock, low jitter is desired. However, jitter degrades with increased DTC range as higher delay is related to higher jitter [11] (especially in the DTC implementations used for this application, as discussed in Section 1.1). As the DTC is not covering a full reference clock cycle, it generates an overflow in a periodic fashion. This is visible at the PLL output as spurs at frequency offsets of multiples of the reference frequency from the carrier, where the spur power is further increased through DTC nonlinearity.

Several authors further explored this synthesizer architectures for PLLs [14, 23, 27, 32, 37, 73, 74] and MDLLs [25]. Ongoing effort is spent in linearizing the DTC to reduce reference spur power levels in the output spectrum, for example through digital pre-distortion of the DTC codes [27] or development of DTCs with intrinsically higher linearity [14, 37]. In addition, system level design effort is spent in investigating DTC-based frequency synthesis at the reference clock [75].

1.2.5 Polar and Outphasing Transmitters

Similar to frequency synthesis in the DDPS architecture, which already allows for generation of spread spectrum clocks [44], DTCs can apply phase modulation on a reference LO signal. This allows to implement all digital polar transmitters [13, 76–78] and outphasing transmitters [7, 10, 31, 42, 79, 80]. Both types employ a DTC to apply phase modulation on an LO signal, enabling a wide modulation bandwidth (up to 400 MHz presented in [31]) due to the open-loop nature of the DTC [81].

Conventional polar transmitters apply phase modulation directly at the PLL and the amplitude modulation at the power amplifier (PA). Two-point modulation at the PLL enables a wider modulation bandwidth [81], where an ADPLL is a favorable implementation, as the digital nature of the control loop allows direct application of the modulation data at the FCW and the DCO [52]. As the phase modulated PLL output signal has then a constant amplitude, it allows the use of very efficient PAs, giving a power advantage compared to I/Q modulators [82]. A COordinate Rotation DIgital Computer (CORDIC) block is used to convert the I/Q data stream of the baseband chip from Cartesian to Polar coordinates [83]. This conversion however widens the bandwidth of the phase data significantly, imposing high requirements on the two-point modulation in the PLL control loop and the frequency range of the oscillator. Therefore, a separation of LO synthesis and phase modulation, which is united in the modulation PLL for the conventional case, is desirable.

A DTC-based polar transmitter as depicted in Fig. 1.9(a) applies phase modulation directly on an LO signal (generated by a PLL), removing the two-point modulation from the PLL regulation loop. The LO acts as reference signal for the DTC, and the phase information from the CORDIC is the digital data input. The remaining blocks operate identical to the conventional polar transmitter. As the PLL does not need to be tailored

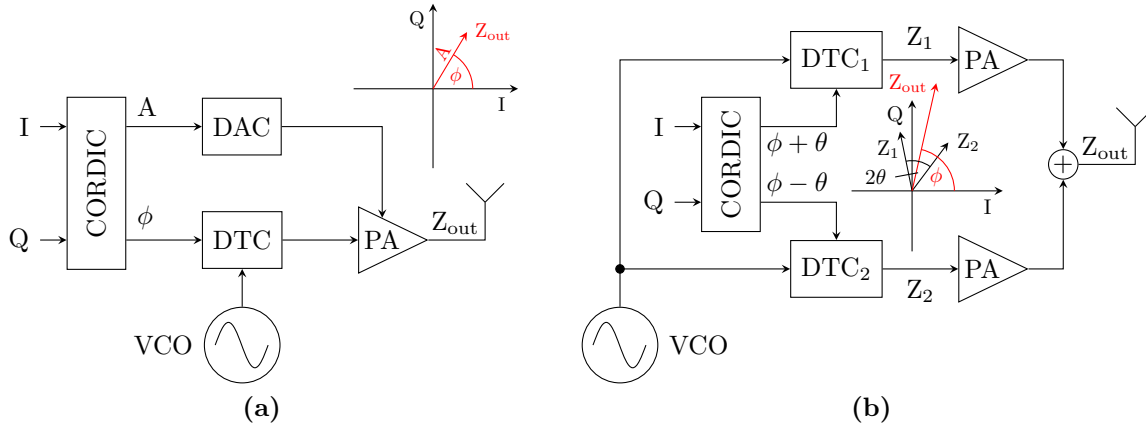


Figure 1.9 – DTC-based transmitters: (a) polar transmitter, and (b) outphasing transmitter.

to a certain modulation scheme, this topology allows a higher order of reconfigurability, making it attractive for software-defined radio (SDR) applications. Recently, a DTC for a polar transmitter was implemented in a digital design flow [13], leveraging the advantages of the digital DTC circuit topology for faster system integration.

Another transmitter architecture in which DTCs have been implemented are outphasing transmitters. A block diagram of the outphasing transmitter is shown in Fig. 1.9(b). Here, two constant envelope signals Z_1 and Z_2 are generated, shifted against each other by 2θ in the phase domain. Both signals are combined in a PA, enabling to control the combined output power of Z_{out} by adjusting the phase shift 2θ . The larger the phase shift, the lower the output power. The common phase shift ϕ of both signals determines then the phase of the output signal. The signal vector diagram in Fig. 1.9(b) visualizes how θ and ϕ are used to generate Z_{out} .

The all digital nature of these phase modulator architectures imposes new design challenges with regard to DTC quantization and nonlinearity. As the modulation data sequence is of random nature, the DTC quantization leads to a quantization noise floor similar to TDC quantization noise [54, pp. 21-22]. This leads to high requirements on the DTC resolution. Furthermore, DTC nonlinearity is corrected digitally with look-up tables (LUT), that are filled by measuring the nonlinearity with external equipment [7, 42] or on-chip with a TDC [13, 76]. As the design and control of the DTC-based phase modulator can be fully digital, it is a scaling friendly architecture for future multi-mode and multi-band transceivers. On the other hand, high frequency operation in the gigahertz domain makes the DTC and its digital data path a significant contributor to the power consumption of the modulated LO generation.

1.3 Motivation and Objectives

Phase interpolators (PI) are favorable DTC fine tuning implementations, as they provide a defined tuning range without need for further calibration. However, their high systematic nonlinearity makes them unattractive for many applications. As there are already known approaches for PI linearization, they may impose an attractive alternative to the switched

capacitor based fine tuning, if the linearization could be applied while providing a high resolution and low power consumption. The present dissertation focuses on these aspects of PI design. The nonlinearity as discussed in [16] is modeled and analyzed with a high accuracy, missing in publications so far. Furthermore, a linearized PI achieving a linearity in the domain of known switched capacitor based fine tuning implementations is presented. As this imposes an increased power consumption, further architectures are explored that increase the linearity of conventional PIs, while providing a competitive power consumption. To further increase PI competitiveness compared to the switched capacitor based fine tuning, splitting the PI into thermometrically and binary controlled parts is investigated to enhance the resolution. This technique is well known in conventional DAC design (also for switched capacitor based DTC fine tuning), but a correct implementation for PIs differs from conventional DACs and has not been presented so far.

In order to have a reference for the newly developed circuits, Chapter 2 introduces an existing DTC design with a divider based coarse tuning and a PI based fine tuning architecture, operating at 2 GHz with a resolution of $t_{d,LSB} = 244$ fs. The coarse tuning architecture is used as a framework for integrating and testing different PI architectures. The fine tuning serves then as reference for the newly developed PIs. To create a solid foundation for DTC comparison, all important circuit measures are defined in this chapter.

Chapter 3 focuses on the implementation of reference and newly developed PIs. A total of three test chips were fabricated for the present thesis in 28 nm standard CMOS technology. These test chips include one test chip for a high linearity PI design, aiming at theoretically ideal linearity, and two test chips for a second, more conventional PI design, focusing on low power consumption and enhanced linearity. The implemented DTCs operate in a frequency range of 2–3 GHz and provide resolutions of 48.8–244.1 fs, surpassing all previously published architectures in this frequency range. The circuit implementations are presented together with simulation results, and the static nonlinearity of each PI is modeled and analyzed in detail. The developed models are used to calculate the PI's nonlinearity based on certain design parameters. The model allows a quick evaluation of initial design parameters, and helps identifying trade-offs between them.

Up to Chapter 3 only the static nonlinearity of DTCs is investigated and compared. In Chapter 4 the dynamic nonlinearity is defined, which is especially important in applications that show high code activity, as DDPS or transmitter DTCs. The root causes of dynamic nonlinearity are analyzed and their effects on the DTC is quantified. Circuit simulations show the impact on different DTC operation modes, namely DDPS and transmitters like operation. Finally, an extension to the low drop-out (LDO) voltage regulator supply used for the DTC is presented, that compensates for dynamic errors at supply voltage level. This compensation is implemented in one of the test chips.

Afterwards, Chapter 5 presents test chip measurements of the discussed PI based DTCs. After a brief review on known methods for DTC verification, a novel measurement method is presented that allows for linearity measurements with femtosecond accuracy. The measurement results are then compared to circuit simulations and model calculations. Configurability of the PI's interpolation range Δt in the test chips allows to validate the models in a wide operation range. Furthermore, the correct operation of the dynamic effects compensation circuit is verified.

Finally, Chapter 6 concludes this thesis and gives an outlook on future challenges in DTC design.

2 DTC Architecture and Characterization

The investigation of phase interpolators as DTC fine tuning requires a surrounding system in which the PI is embedded. While it is possible to generate the required input signals by means of external signal generators, an actual PI based DTC implementation requires an on-chip PLL for generation of the reference signal and a coarse tuning block to generate both PI input signals. As an on-chip PLL and coarse tuning stage influence the overall DTC performance and linearity, they are required to give a full and realistic picture of the circuit. This chapter introduces the multistage DTC architecture in which context the PI circuits are investigated. The presented architecture was used for a previously developed reference design, to which all newly developed PIs are compared. As frequency synthesis with a PLL is well known, the PLL implementation is omitted from the following discussion.

The present chapter introduces the multistage DTC architecture and its implementation in Section 2.1. All presented DTC blocks are discussed in detail, as they differ from state of the art implementations as reviewed in Section 1.1 and introduce new concepts, especially to DTC coarse tuning. Afterwards, Section 2.2 presents the DTC performance measures that are used throughout this work to compare different architectures. This includes typical D/A converter metrics regarding linearity, as well as measures for noise performance. Finally, Section 2.3 gives an overview on the DTC configuration that is discussed in the subsequent chapters.

2.1 Investigated Multistage DTC Architecture

The investigation of PI circuits requires a coarse tuning block which provides two signals of the same frequency, shifted against each other by Δt in time domain or $\Delta\phi$ in phase domain. The PI takes these signals as inputs, and produces an output signal of the same frequency with a phase according to its programming.

This section presents an existing PI based three-stage DTC reference design. An MMD as ultra coarse tuning stage provides two output signals at f_{out} with 3 bit resolution, which are shifted against each other by Δt_{uc} . As this spacing is usually too wide for a linear phase interpolation, a subsequent novel coarse tuning stage, implemented as multiplexer and delay element stage (MUX+DEL), reduces this spacing with 1 bit resolution to $\Delta t_c = \Delta t_{\text{uc}}/2$. Finally, a PI takes the two coarse tuning output signals as input, and produces a single DTC output with a phase enclosed between the two input signals, controlled by its digital input code.

The number of bits for the three stages are $k_{\text{MMD}} = 3$, $k_{\text{MUX+DEL}} = 1$, and $k_{\text{PI}} = 7 \dots 10$. The PI resolution k_{PI} depends on its architecture and implementation. The reference design

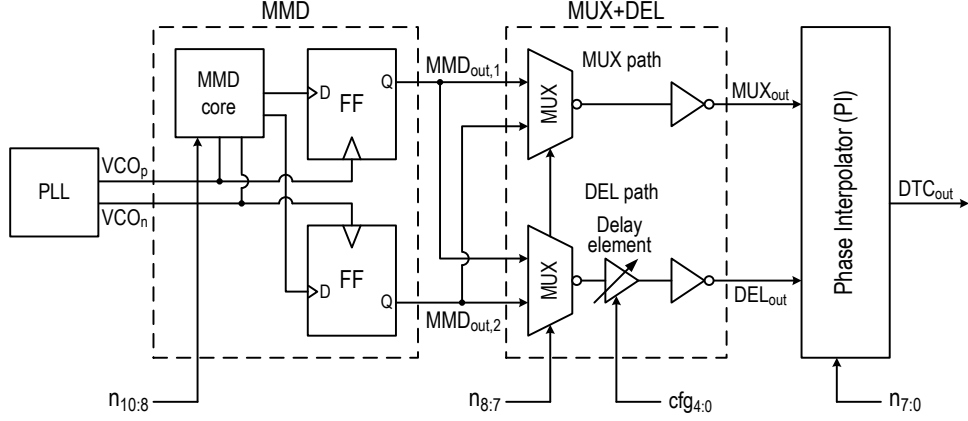


Figure 2.1 – Architecture overview of the three-stage DTC.

is implemented with $k_{PI} = 7$. For the investigated PIs that are discussed in Chapter 3, two different DTC configurations are used: 1) a three-stage DTC design as described above, and 2) a two-stage DTC design consisting only of an MMD and a PI. The stages are configured in a manner that leads to a total DTC resolution of

$$\begin{aligned} k_{DTC,1} &= k_{MMD} + k_{MUX+DEL} + k_{PI} \\ &= 11, \text{ and} \end{aligned} \quad (2.1)$$

$$\begin{aligned} k_{DTC,2} &= k_{MMD} + k_{PI} \\ &= 12 \dots 13, \end{aligned} \quad (2.2)$$

where the number of bits k_{MMD} , $k_{MUX+DEL}$, and k_{PI} relate to the resolution of the MMD, MUX+DEL, and PI stage, respectively. The digital DTC code n is in the range of $0 \leq n \leq N$ for $N = 2^{k_{DTC,1/2}} - 1$. The maximum code for all discussed DTCs and tuning stages will be denoted with N .

Fig. 2.1 shows the top block diagram of the DTC. The upcoming sections describe the detailed operation of the single blocks at the example configuration of a three-stage design with $f_{ref} = 8$ GHz and $f_{out} = 2$ GHz. The operation of the two-stage DTC design can be explained in an analogous manner.

2.1.1 Multi-Modulus Divider

The MMD as depicted in Fig. 2.1 is split in two consecutive parts: first, the divider core generates two signals with the desired frequency and phase relation according to the digital programming, and second, the subsequent flip-flops (FF) re-sample the divider core for low noise on $VCO_{p/n}$. The MMD produces two signals $MMD_{out,1}$ and $MMD_{out,2}$ at $f_{out} = 2$ GHz from a differential VCO signal at $f_{ref} = 8$ GHz, provided by a PLL. It has a nominal division ratio of 4, and the two additional division modi 3 and 5. The outputs $MMD_{out,1/2}$ are aligned with the pseudo-differential signals $VCO_{p/n}$, enabling the generation of two signals with a temporal spacing of half a VCO period $\Delta t_{uc} = T_{VCO}/2$. The total number of control bits $k_{MMD} = 3$ is split in least significant bits (LSB) and most significant bits (MSB): LSB bit n_8 controls the temporal order of $MMD_{out,1/2}$, and the MSB bits $n_{10:9}$ determine the divider's division modi. In the following, the influence of

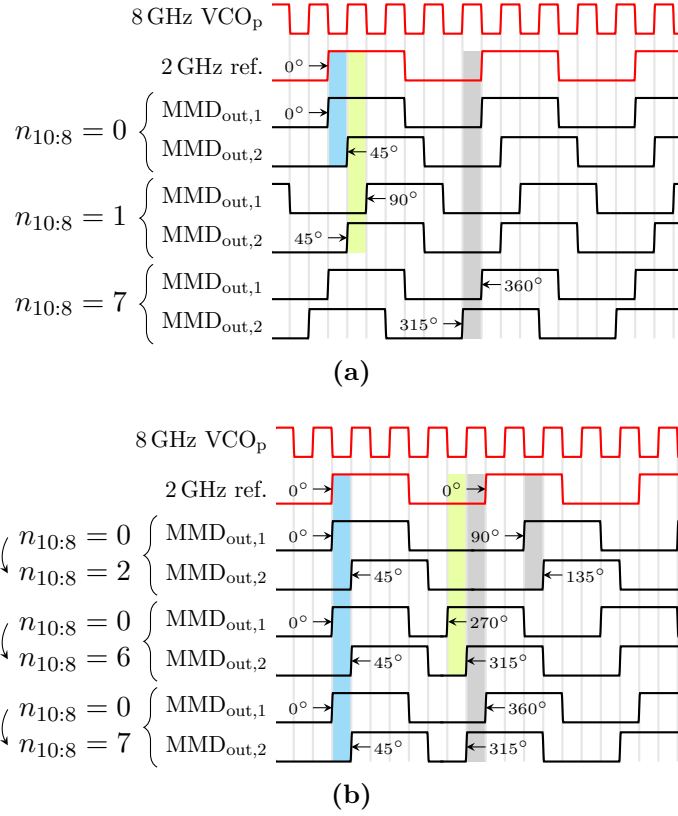


Figure 2.2 – MMD output waveforms for (a) different static digital codes $n_{10:8}$, and (b) different dynamic code changes triggering division modes 3 and 5.

LSB and MSB programming on the MMD output signals, plotted in Fig. 2.2, is discussed in detail.

MMD_{out,2} is aligned with a rising edge of VCO_p, whereas MMD_{out,1} is, depending on n_8 , aligned with the rising edge of VCO_n either directly leading or lagging VCO_p of MMD_{out,2}. This enables a temporal spacing of $\Delta t_{uc} = T_{VCO}/2 = 62.5$ ps between MMD_{out,1/2}, which is equivalent to a phase spacing of $\Delta\phi_{uc} = 45^\circ$. Figure 2.2(a) shows how the LSB affects the relation of MMD_{out,1/2} if the code transitions from $n_{10:8} = 0$ to $n_{10:8} = 1$. The 2 GHz reference signal is identical to MMD_{out,1} for $n_{10:8} = 0$ and is the 0° reference to which the phases of MMD_{out,1/2} are related. While for $n_{10:8} = 0$ MMD_{out,1} has a phase of 0° and for $n_{10:8} = 1$ a phase of 90°, MMD_{out,2} stays at 45° during this code transition. This leads to the enclosure of the phase range 0–45° for $n_{10:8} = 0$ and 45–90° for $n_{10:8} = 1$ between MMD_{out,1/2}.

The division modi 3 and 5 are controlled by $n_{10:9}$ and can re-align MMD_{out,2} with a different rising edge of VCO_p by changing the instantaneous division ratio to 3 or 5 for a single 2 GHz output cycle. If $n_{10:9}$ is increased (including a wrap around $n_{10:9} : 3 \rightarrow 0$) or decreased (including a wrap around $n_{10:9} : 0 \rightarrow 3$) a division-by-5 or division-by-3 is triggered, respectively. For a single division-by-5 MMD_{out,2} shifts by +125 ps ($\cong +90^\circ$), and for a single division-by-3 by -125 ps ($\cong -90^\circ$). After the division MMD_{out,1} is still aligned relative to MMD_{out,2}, determined by n_8 . Subsequent divisions shift the DTC output multiple times, enabling a wrap around of the DTC output phase. This leads to the enclosure of the full 2 GHz 2π range by MMD_{out,1/2} over code, which is visualized for

static $n_{10:8}$ in Fig. 2.2(a). The examples of $n_{10:8} = 0$, $n_{10:8} = 1$, and $n_{10:8} = 7$ highlight which phase part of the 2 GHz reference signal is enclosed between $\text{MMD}_{\text{out},1/2}$ depending on $n_{10:8}$.

Figure 2.2(b) illustrates $\text{MMD}_{\text{out},1/2}$ for application of different division modi, which requires MMD code transitions. For the first two code transitions only the MSB is triggered, $\text{MMD}_{\text{out},2}$ shifts according to the division while $\text{MMD}_{\text{out},1}$ stays aligned to it in the same way as before the code change. The last code transition shows an example for $n_{10:8} = 0 \rightarrow n_{10:8} = 7$, where a division-by-3 plus a LSB change is executed in parallel, leading to a wrap around of the 2π range in phase domain. As the current implementation of a 3/4/5 divider allows only for a single division per output cycle, the programming is limited to phase changes of $\pm 90^\circ$ due to a division plus a possible LSB change, leading to a maximum phase change of $\pm 135^\circ$.

The MMD output flip-flops are implemented as low noise flip-flops to achieve low jitter for $\text{MMD}_{\text{out},1/2}$. They re-sample the outputs of the divider core, which allows to design the core in a more power efficient way as it does not need to provide good phase noise performance. Depending on the noise specifications, the flip-flops can dominate the overall MMD layout area.

This architecture can also be operated at other frequencies than 2 GHz. While operation at lower frequencies is easily possible, higher frequencies are limited by the internal timing of the MMD, which needs to be designed accordingly. All above mentioned frequency and time values change according to the new input reference frequency and the new DTC output frequency. Independent of the frequency f_{ref} , the dynamic range of the MMD is always 2π .

2.1.2 Multiplexer and Delay Element

The coarse-tuning stage reduces the spacing Δt_{uc} with 1 bit resolution to $\Delta t_c = \Delta t_{\text{uc}}/2 = 31.25$ ps and is implemented as illustrated in Fig. 2.1. It has a 2 bit interface, using the MMD's LSB n_8 to determine whether $\text{MMD}_{\text{out},1}$ is leading or lagging $\text{MMD}_{\text{out},2}$. The upper path, called MUX path, selects with a MUX (controlled by $n_{7:8}$) either $\text{MMD}_{\text{out},1}$ or $\text{MMD}_{\text{out},2}$ for output MUX_{out} . With $n_7 = 0$ the temporally "early" signal is selected, and with $n_7 = 1$ the "late" one. The lower path, called DEL path (controlled by n_8), automatically selects the "earlier" signal of $\text{MMD}_{\text{out},1/2}$ for DEL_{out} . The delay element in this path delays the signal ideally by $t_{\text{d,DEL}} = 31.25$ ps. As the delay varies over process, voltage and temperature (PVT), it can be adjusted with 5 bit resolution via control input $\text{cfg}_{4:0}$. The propagation delay t_d of the two paths is the sum of the propagation delay of the single elements:

$$t_{\text{d,MUX}} = t_{\text{d,MUX}} + t_{\text{d,inv}} \quad (2.3)$$

$$t_{\text{d,DEL}} = t_{\text{d,MUX}} + t_{\text{d,DEL}} + t_{\text{d,inv}} \quad (2.4)$$

The resulting delay difference between them is only determined by the delay element:

$$t_{\text{d,DEL}} - t_{\text{d,MUX}} = t_{\text{d,DEL}} = \Delta t_c \quad (2.5)$$

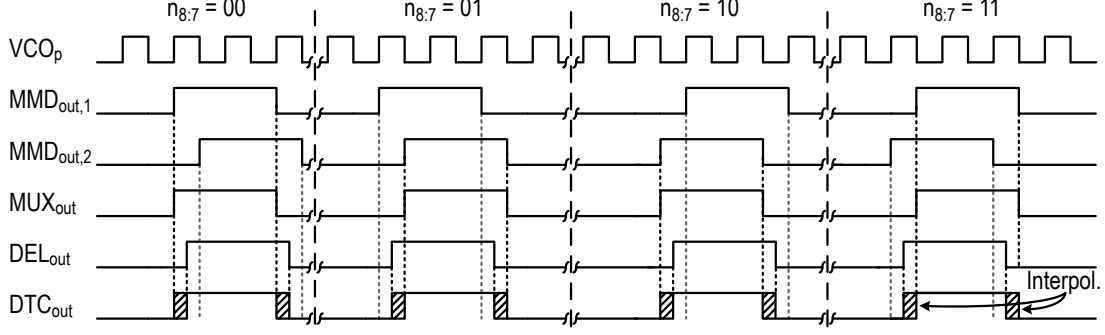


Figure 2.3 – Signal alignment between VCO_p and the single DTC blocks.

As $\Delta t_c \triangleq \Delta\phi_c$, the phases of the two outputs can be expressed as

$$\phi(\text{MUX}_{\text{out}}) = \phi(\text{MMD}_{\text{out},1/2}), \text{ and} \quad (2.6)$$

$$\phi(\text{DEL}_{\text{out}}) = \phi(\text{MMD}_{\text{out},1}) + \Delta\phi_c, \quad (2.7)$$

for the example of $n_8 = 0$, meaning that $\text{MMD}_{\text{out},1}$ is early. A phase shift due to the constant propagation delay of both paths is neglected in (2.6) and (2.7). The waveforms in Fig. 2.3 show the outputs of MMD and MUX+DEL stage, depending on $n_{8:7}$.

The resulting signal spacing Δt for the PI depends on the control of the MMD and MUX+DEL stages. As the resolution of the configuration input is finite, $t_{d,\text{DEL}}$ is unlikely to be at its ideal value of 31.25 ps. The actual phase spacing at the PI input can either be $\Delta t = t_{d,\text{DEL}}$ or $\Delta t = \Delta t_{\text{uc}} - t_{d,\text{DEL}}$.

2.1.3 Phase Interpolator

The PI is a key building block in this architecture and provides high time resolution. This section discusses the fundamental behavior of PIs. The actual implemented designs are presented in Chapter 3. The PI has two input signals In_1 and In_2 , which are connected to the outputs MUX_{out} and DEL_{out} of the coarse tuning stage as shown in Fig. 2.1. A certain phase spacing $\Delta\phi$, equivalent to a time spacing Δt , is enclosed between $\text{In}_{1/2}$. The PI output signal has the same frequency as the input signals and allows phase tuning in a range of $\Delta\phi$. During the interpolation process $\text{In}_{1/2}$ are weighted in phase domain, controlled by a digital code word. The phase of the output signal of an ideal and linear PI can be described by

$$\phi(\text{DTC}_{\text{out}})[n] = \frac{N-n}{N}\phi(\text{In}_1) + \frac{n}{N}\phi(\text{In}_2) + \phi_{\text{pd}}, \quad (2.8)$$

where the phase of the input signals is weighted to produce an output signal with a desired phase. The maximum digital code N results from a PI with k_{PI} bit resolution according to $N = 2^{k_{\text{PI}}}$. Note the difference between PIs and the overall DTC or DACs in general: A DTC or a general DAC with k bit resolution have programming codes in the range of $0 \leq n \leq 2^k - 1$, whereas the digital code word n for PIs is in a range of $0 \leq n \leq 2^{k_{\text{PI}}}$. This leads to one additional programming code and accounts for the possibility to weight the phase either fully to one or the other input. The difference between PIs and general DACs is discussed in detail in Section 3.4. In addition to the

weighting, the output signal's phase contains a constant phase shift ϕ_{pd} equivalent to the code independent propagation delay t_{pd} of the PI. Further derivation of DTC performance metrics in Section 2.2 will show that ϕ_{pd} has no influence on the static DTC nonlinearity. Equation (2.8) shows that DTC_{out} is aligned with In_1 for $n = 0$ and with In_2 for $n = N$. This means that the PI covers exactly $\Delta\phi$ over its whole code range, which is one of the most important advantages compared to other fine tuning architectures with an undefined range (e.g. switched capacitor based delay cells).

Another advantage is the interpolation on rising and falling edges, which is not given in all PI implementations [1, 36, 84–86]. This enables a duty cycle of 50% for a constant DTC code, which is a mandatory for some applications. Both DTC coarse tuning stages, MMD and MUX+DEL, apply the DTC code on both edges to keep the duty cycle of 50% for constant DTC code.

2.1.4 Digital Data Path

The digital data for the DTC can be fed by two different data paths. For exact lab validation of the DTC's linearity, a ramp generator is built in. All other digital input data sequences can be stored in an on-chip static random-access memory (SRAM) and then be programmed to the DTC.

The ramp generator can be configured for a minimum and maximum code, as well as a code step size. The code is then swept continuously between minimum and maximum in a triangular ramp with the programmed step size. The digital part is clocked by a divided DTC output. Each code stays active for a time determined by a counter, enabling to adjust the programming to speed and bandwidth of external measurement devices.

The SRAM provides the possibility to store an arbitrary sequence of DTC codes. The codes are then read from the memory at f_{out} and programmed to the DTC. The DTC latches the data in with each falling output edge and applies it to the subsequent rising edge. As the analog domain is phase, the converted information is stored in the edge position. Code and select signal changes for the single DTC blocks need to be applied during logic low or high level of the analog signals. This means that a new code can be applied to each rising edge, while the falling edges have the same code as their previous rising edge. As the investigated DTCs operates at rates of ≥ 2 GHz, correct timing of the clock and data signals for each block is crucial. A single clock is not sufficient to latch in new data to the DTC, as the system has only $t = 0.5/f_{2GHz}$ to acquire new data, which is a delay in the order of the overall DTC propagation delay. Therefore each block latches in new data with different clock signals $f_{clk,MMD}$, $f_{clk,MUX+DEL}$, and $f_{clk,PI}$, derived from the respective block's output signals. The digital data has to be synchronized for each of these clock signals.

2.2 DTC Performance Characteristics

As the DTC is a D/A converter, the performances of interest cover mostly performances typically evaluated for D/A converters and include: 1) the transfer function (TF) of digital input code n vs. output phase (or delay); 2) integral nonlinearity (INL); 3) differential nonlinearity (DNL); and 4) monotonicity [87, pp. 614-618]. As the DTC generates an

RF signal, also the generated jitter, which can be measured as phase noise, is of interest. Furthermore, power consumption is a key performance parameter, as typical applications can be embedded in systems that run from battery. Usually glitches are an additional concern, especially in high speed D/A converters as discussed here, originating from timing mismatch of the digital programming signals that control the analog D/A converter sections [87, pp. 633-634]. This concern can be discarded for DTCs, as the toggling of the control signals has no direct influence on the output signal's phase. The control state of the analog DTC sections is only evaluated at each rising and falling edge of the processed signal, in-sensitizing the DTC regarding glitches.

The TF is the initial point to evaluate the systematic nonlinearity of this circuit. It is the code n dependent part of the DTC's propagation delay t_d or output phase ϕ . From the TF, INL as well as DNL can be derived. As phase shift and time delay are equivalent, TF, INL, and DNL can be defined in phase and time domain. The TF is the code n dependent part of the DTC's propagation delay t_d or the DTC's output phase ϕ :

$$\begin{aligned} \text{TF}[n] &= t_d[n] - t_d[0] \\ &= \phi[n] - \phi[0] \end{aligned} \quad (2.9)$$

This definition acknowledges that the DTC output signals' phase $\phi[n]$ cannot be measured in an absolute manner, but only relative to a reference. It also implies that an absolute delay in the signal path is not of relevance for the static nonlinearity, but only the delay difference between different DTC codes (it may be relevant for the system the DTC is embedded in). The 0° reference is chosen as the phase of code $n = 0$, to which the phase of all other codes is aligned.

While the TF contains all nonlinearity information, it is not illustrating them very well. Therefore, INL and DNL are defined to check different characteristics of the nonlinearity. The INL is the absolute difference between the TF of the implemented and an ideal DTC. As an ideal k bit DTC has a dynamic range of exactly $2\pi \hat{=} T_{\text{out}}$ and wraps around 2π , it holds $\text{TF}[n = 2^k] - \text{TF}[n = 0] = T_{\text{out}}$. This leads to the ideal linear TF and INL of

$$\text{TF}_{\text{ideal}}[n] = \frac{n}{2^k} T_{\text{out}}, \text{ and} \quad (2.10)$$

$$\begin{aligned} \text{INL}[n] &= \text{TF}[n] - \text{TF}[n]_{\text{ideal}} \\ &= t_d[n] - t_d[0] - \frac{n}{2^k} T_{\text{out}}, \end{aligned} \quad (2.11)$$

where $0 \leq n \leq (2^k - 1)$. The DNL is the step size between two neighboring DTC codes compared to the ideal step size t_{LSB} , which is given as

$$t_{\text{LSB}} = \frac{T_{\text{out}}}{2^k}. \quad (2.12)$$

Thus, the DNL is defined as

$$\text{DNL}[n] = \text{TF}[n] - \text{TF}[n - 1] - t_{\text{LSB}}, n \in \{1, 2, \dots, (2^k - 1)\}. \quad (2.13)$$

Statements about monotonicity are made based on the DNL. If the DNL holds $\text{DNL}[n] \geq -t_{\text{LSB}}$, then all positive DTC code steps provide a positive additional output delay, meaning

a fully monotonic DTC. To quantify the nonlinearity in a single number, the DNL's root mean square (rms) values is calculated:

$$\text{DNL}_{\text{rms}} = \sqrt{\frac{1}{2^k - 2} \sum_{j=1}^{(2^k-1)} (\text{DNL}[j])^2}. \quad (2.14)$$

As INL and DNL are calculated based on the TF, each of them can be measured either in time or in phase domain. The measurement methods used in the present thesis capture either the TF or the DNL. The conversion from DNL to TF and INL is done by

$$\text{TF}[n] = \sum_{j=0}^n (\text{DNL}[j] + t_{\text{LSB}}), n \in \{1, 2, \dots, (2^k - 1)\}, \text{ and} \quad (2.15)$$

$$\text{INL}[n] = \sum_{j=0}^n \text{DNL}[j], n \in \{1, 2, \dots, (2^k - 1)\}. \quad (2.16)$$

According to (2.9), TF and INL are equal to zero for $n = 0$. For this DTC topology with k bit resolution the full scale (FS) and full scale range (FSR) definitions from [88, p. 501] are modified to

$$\begin{aligned} \text{FS} &= 2\pi - t_{\text{LSB}} \\ &= 2\pi \left(1 - \frac{1}{2^k}\right), \text{ and} \end{aligned} \quad (2.17)$$

$$\begin{aligned} \text{FSR} &= \lim_{2^k \rightarrow \infty} \text{FS} \\ &= 2\pi, \end{aligned} \quad (2.18)$$

as the DTC covers a full output period. Unlike most D/A converters, this DTC supports a wrap around of the code due to the MMD coarse tuning, enabling to directly measure the FSR. The 2π range is covered by design, as the circuit is again in the state of $n = 0$ after the code wrap around. As the present thesis focuses on PIs, the FSR of this particular DTC block is of special interest. For example, a PI design with $\text{FSR} = \Delta t$ is not easily achieved due to layout asymmetry of $\text{In}_{1/2}$ or layout coupling between $\text{In}_{1/2}$. Each distortion of $\text{In}_{1/2}$'s symmetry leads to an effectively smaller/larger Δt : if In_1 has for example a different capacitive loading as In_2 (e.g. due to layout asymmetry), the two paths deviate in their code independent propagation delay, effectively changing Δt . A range of $\text{FSR} > \Delta t$ and $\text{FSR} < \Delta t$ is possible and is referred to as range extension and range compression, respectively. All above discussed equations can also be evaluated for sub-ranges of the DTC code in an analogous manner, e.g. only for the PI code range $0 \leq n \leq 2^{k_{\text{PI}}}$ with an FSR of Δt . Then, the equations for TF_{ideal} and t_{LSB} ((2.10) and (2.12)) are re-defined with $N = 2^{k_{\text{PI}}}$ to

$$\text{TF}_{\text{ideal}}[n] = \frac{n}{N} \Delta t, \text{ and} \quad (2.19)$$

$$t_{\text{LSB}} = \frac{\Delta t}{N}. \quad (2.20)$$

They can now be used in (2.11) and (2.13) for INL and DNL calculation accordingly.

As DTCs are often used in communication systems, this section closes with a discussion about jitter and phase noise. Different types of RF circuits require special circuit simulation techniques to extract the relevant and correct noise information, which is extensively discussed in the literature [89–91]. Depending on the application, either phase noise at a specific frequency offset f_{offset} from the LO signal at f_{out} , or overall jitter are of interest. The DTCs investigated in the present thesis are designed to meet certain phase noise targets. As the DTC is a purely digital circuit, it accumulates jitter as the signal propagates through, which can be quantified in circuit simulations. An analysis of DTC generated jitter in dependence of the total DTC delay is discussed in [14]. To be able to compare it to a measurement, jitter needs to be converted to phase noise. The relation between both metrics is discussed in detail in [11, 92]. Circuit simulations yield into the spectral density of jitter $S_{\tau,s}(f)$ with the unit $[\text{s}/\sqrt{\text{Hz}}]$ [93], that needs to be converted to the jitter PSD $S_{\tau}(f)$ with unit $[\text{rad}^2/\text{Hz}]$:

$$S_{\tau}(f_{\text{offset}}) = \left(2\pi f \sqrt{S_{\tau,s}(f_{\text{offset}})}\right)^2. \quad (2.21)$$

The single-sideband (SSB) phase noise, as measured with a spectrum analyzer, is given in the unit $[\text{dBc}/\text{Hz}]$ [94, p. 26]:

$$\mathcal{L}(f_{\text{offset}}) = 10 \log_{10} \left(\frac{P_{\text{noise}}(f_{\text{out}} + f_{\text{offset}})}{P_{\text{LO}}(f_{\text{out}})} \right). \quad (2.22)$$

The simulation result from (2.21) can be directly converted to phase noise with

$$\mathcal{L}(f_{\text{offset}}) = 10 \log_{10} \left(\frac{S_{\tau}(f_{\text{offset}})}{2} \right), \quad (2.23)$$

enabling a direct comparison to measurements. As the DTC reference is generated by a chip internal PLL, the phase noise of the output spectrum is dominated by typical PLL phase noise characteristics for small offset frequencies from the output signal. Therefore, the far-off noise at $f_{\text{offset}} = 100 \text{ MHz}$ is measured to determine the DTC phase noise performance. However, all circuits in the signal path contribute to the noise level, meaning the DTC is not the sole contributor to the measured phase noise.

2.3 Summary

This chapter discussed the multistage DTC architecture that is the basis for investigation of PI circuits. The MMD and the MUX+DEL stage generate the two PI input signals from a pseudo-differential VCO signal with a frequency range of 8–12 GHz. The discussed three-stage DTC architecture contains all important building blocks for further investigation of PIs. Different PIs are integrated with identical coarse tuning blocks and benchmarked against each other. As one of the presented PI architectures enables operation directly on the MMD output spacing Δt_{uc} , the use of the MUX+DEL stage is optional. This leads to a two-stage configuration with only MMD and PI in addition to the discussed three-stage design. The investigated configurations are listed in Table 2.1 together with the targeted resolution. While the first two PIs are designed for a specific operation frequency, the

Table 2.1 – DTC configurations for the investigated PI designs.

Frequency	DTC config.	Resolution [bit]	Resolution [°]	Resolution [s]
2 GHz	MMD	3 bit	45°	62.50 ps
	MUX+DEL	1 bit	22.5°	31.25 ps
	PI	7 bit	175.8 m°	244.14 fs
2.5 GHz	MMD	3 bit	45°	50.00 ps
	PI	10 bit	43.9 m°	48.83 fs
2.2–3 GHz	MMD	3 bit	45°	41.67–56.82 ps
	PI	9 bit	87.9 m°	81.38–110.97 fs

third design focuses on operation in a wide frequency range, which translates to a range of Δt at the PI input.

All important DTC performances that need to be evaluated in modeling, simulation, and lab verification were introduced. As DTCs are DACs, their performance depends mainly on their linearity, next to key parameters as operation frequency, resolution, monotonicity, phase noise, and power consumption. Therefore, this chapter introduced TF, INL, and DNL for evaluation of DTC nonlinearity.

The upcoming chapter first introduces the implementation of the reference PI and afterwards the implementation of the newly developed PIs in the DTC configurations according to Table 2.1.

3 Phase Interpolator Design and Modeling

The key component of the DTC presented in the last section is the PI, which provides high time resolution. This chapter analyzes different PI architectures that were developed during this dissertation project and compares them to the existing reference design.

First, the digitally controlled edge interpolator (DCEI) reference PI design is discussed and analyzed in Section 3.1. It is implemented in an 11 bit DTC, generating a 2 GHz output signal. A DCEI circuit model is developed that identifies the key design parameters influencing its nonlinearity. It is validated against simulation results and, in Chapter 5, also against test chip measurements.

In Section 3.2 a linearization concept for the DCEI is discussed, leading to the contention-free DCEI (CF-DCEI). A fully analytical CF-DCEI model identifies key design parameters that lead to theoretically perfect linearity. A developed test chip enables further investigation of the linearized interpolation and validation of the analytical model. However, the linearization of the interpolation comes at the cost of extra current consumption, which is the main drawback of the CF-DCEI compared to the DCEI.

Therefore, another topology is developed, which enables a reduction in power consumption compared to both, DCEI and CF-DCEI. The concept of two-point interpolation is explored in Section 3.3, resulting in the two-points digitally controlled edge interpolator (DCEI²). The DTC architecture is simplified from a three-stage to a two-stage architecture, as the MUX+DEL stage is no longer needed for the DCEI². This leads to an overall reduced complexity of the architecture, and enables significant savings in area and power consumption compared to the other DTC designs. Again, a circuit model is used to

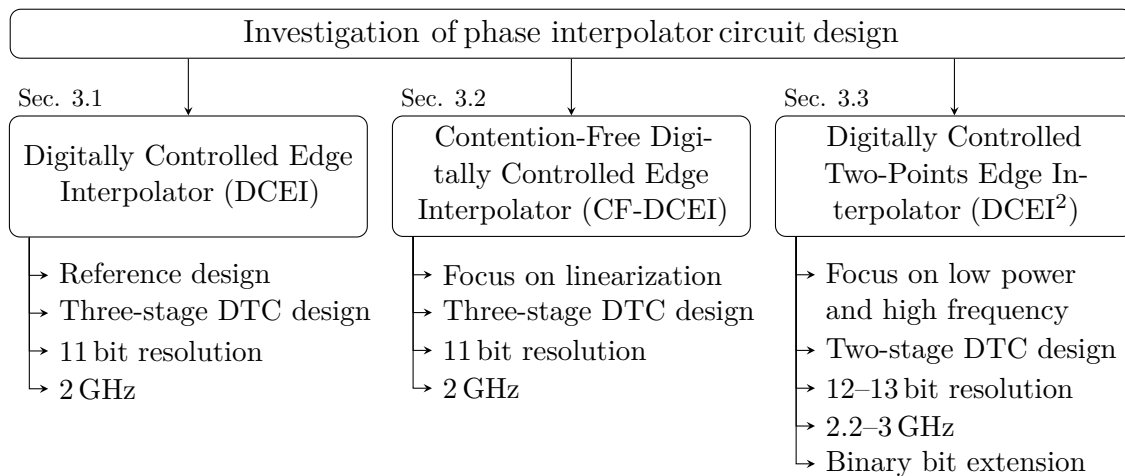


Figure 3.1 – Overview on investigated PI architectures.

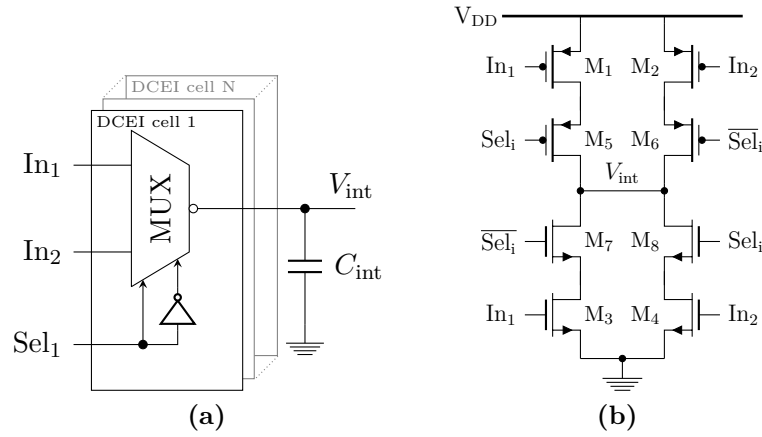


Figure 3.2 – (a) Implementation and interconnection of the DCEI unit cells, and (b) transistor level implementation of the analog MUX core.

describe the interpolation process. Two evolutionary test chips were developed, the first one as a prove of concept and a second improved re-design. Figure 3.1 gives an overview on the three topologies and summarizes the focus of the investigation and the key design parameters resolution and operation frequency.

Last, resolution enhancement with binary bits is explored on the example of the DCEI² PI architecture. Hybrid DAC architectures, splitted into thermometrically and binary coded parts, are well known [87, pp. 640-642]. It is an attractive concept, as it enables increased resolution with only slight increase in area and power consumption. As the conventional approach of binary extension for DACs cannot be applied to PIs, differences between PIs and conventional DACs are discussed in Section 3.4 and the correct implementation is presented. Most interesting is the monotonicity of the binary bits, which should be given over PVT variations. Different architectures of the binary interpolation cells are explored and tested in the two evolutionary test chips.

Finally, all above mentioned designs are directly compared in Section 3.5, which concludes this chapter.

3.1 Digitally Controlled Edge Interpolator

Fig. 1.3(b) already illustrated a high level PI implementation, where two buffers, driven by In_1 and In_2 , are weighted to provide interpolation in the phase domain. The network that combines both signals is kept generic in this figure.

The DCEI concept as shown in Fig. 3.2(a) implements N identical parallel MUX cells that select either In_1 or In_2 to weight them in the interpolation process. Each cell is controlled individually by its select signal Sel_i . The generic combination network from Fig. 1.3(b) is implemented as its simplest case: a short. This means the outputs of all DCEI cells are connected to the common interpolation node V_{int} with its associated capacitance C_{int} . Fig. 3.2(b) shows the implementation of the analog MUX core. Each MUX consists of two tristate inverters, one connected to In_1 and the other to In_2 . The Sel_i signal selects one of the tristate inverters to drive the common interpolation node V_{int} . The input signals

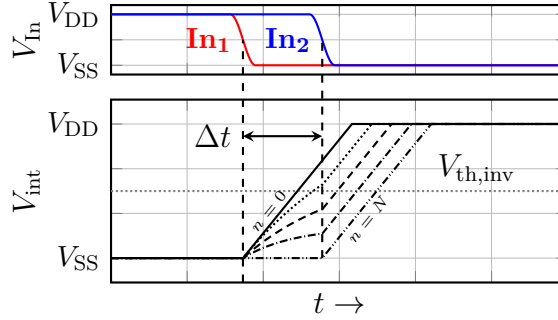


Figure 3.3 – Waveforms of the ideal DCEI interpolation process for different codes n .

$In_{1/2}$ are identical for all cells.

An exemplary ideal interpolation on the rising edge is plotted in Fig. 3.3 for different digital codes n . The falling edge of In_1 triggers the start of the interpolation, as it is an inverting stage. Now $(N - n)$ DCEI cells are configured to charge the net V_{int} to V_{DD} , while the remaining n DCEI cells select In_2 and are configured to pull V_{int} to V_{SS} . This contentious condition between the two groups of DCEI cells leads to the nonlinear charging during the denoted time interval Δt . With the falling edge of In_2 , all cells are now configured to charge V_{int} to V_{DD} , leading to a linear charging with constant slope in this region. Section 3.1.1 analyzes this process in detail and provides realistic waveforms of the interpolation process.

The DCEI based DTC operates at a reference and output frequency of $f_{\text{ref}} = 8$ GHz and $f_{\text{out}} = 2$ GHz, respectively. The DCEI is designed for an input signal spacing of $\Delta t = 31.25$ ps and a digital resolution of $k_{\text{PI}} = 7$ bit, resulting in a time resolution of 244 fs. Therefore, $N = 2^{k_{\text{PI}}} = 128$ identical unit interpolation cells are implemented, arranged in an array of the dimension 16×8 as shown in Fig. 3.4. A set of array input drivers provides sufficient driving strength for steep signal slopes at all DCEI cell inputs. The array output drivers evaluate the interpolation when V_{int} crosses their threshold voltage $V_{\text{th,inv}}$ and drive the DCEI output, which is identical to the DTC output.

Two measures were taken to ensure a monotonic behavior over the full interpolation range. First, the control of the array is thermometrically coded. Each cell implements a selection logic that generates Sel_i from row and column select signals, which are routed vertically and horizontally over the cell array. The digital decoder is split into a row and a column decoder with 3 and 4 bit control, respectively. Each decoder implements a binary to thermometric conversion to generate the 16 row and 8 column signals. Second, the decoder is designed in a fashion that the DCEI cells are selected in a meander form according to the cell numbering in Fig. 3.4. This ensures that only neighboring cells switch their state for code changes of 1 LSB during row or column transitions.

The ideal DCEI covers exactly the phase difference enclosed between $In_{1/2}$ at its output over code. However, the actual physical implementation can introduce parasitic resistances and capacitances that degrade this ideal behavior and lead to range compression or extension. Even if the DCEI is a circuit that is built completely from digital blocks, it has a high sensitivity towards parasitic layout capacitors/resistors and needs to be integrated with an analog design flow. Especially the input nets are critical, as the spacing Δt between $In_{1/2}$ defines the FSR of the DCEI. Therefore, following layout-induced effects should be

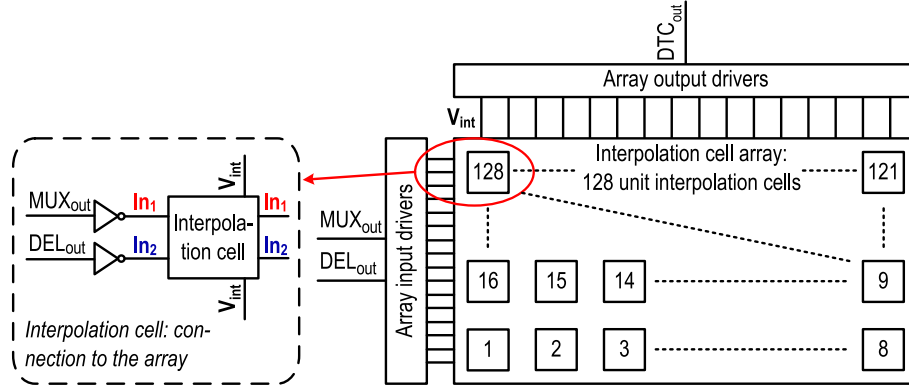


Figure 3.4 – DCEI cell array topology.

prevented:

1. Mismatch between the overall net parasitics of In_1 and In_2
2. Coupling between In_1 and In_2
3. Coupling between $In_{1/2}$ and V_{int}

The first effect leads effectively to a distortion of Δt from the inputs to the outputs of the DCEI's input buffers. This can, depending on temporal order of the input signals, extend or reduce the DCEI's effective Δt . It is addressed by a symmetrical design of the nets $In_{1/2}$ to ensure identical capacitive loading. The second effect reduces Δt at $In_{1/2}$, independent of the input signals' temporal order. This reduces the DCEI's effective Δt and leads to positive spikes in the DNL, for code transitions that include switching of the input signals' temporal order. Therefore, the close parallel routing of $In_{1/2}$ should be prevented to reduce the parasitic capacitance between them. The third and last effect is least critical and, again, reduces Δt at $In_{1/2}$. It is addressed by horizontal routing of $In_{1/2}$ and vertical routing of V_{int} in different metal layers, as indicated in the interpolation cell connection to the array of Fig. 3.4. Furthermore, the unit cell transistors directly coupled to V_{int} are connected to the select signals instead of $In_{1/2}$ to increase the isolates between $In_{1/2}$ and V_{int} (see Fig. 3.2(b)).

The upcoming section recaps the contributors to nonlinearity in DCEI type PIs according to the literature, and provides a circuit model that includes them to describe the interpolation process.

3.1.1 DCEI Model

According to the literature, the main contributors to PI nonlinearity are: (a) the ratio $\Delta t/\tau_{int}$ [16, 95], where τ_{int} is the RC time constant of V_{int} , (b) shoot-through currents between V_{DD} and V_{SS} during the interpolation [16, 84], and (c) limited rise/fall-time $t_{r,f}$ of the input signals [95]. The following evaluation of a DCEI circuit model shows the influence of $\Delta t/\tau_{int}$ and $t_{r,f}$ on the interpolation nonlinearity. The shoot-through current is given by the circuit topology, and is not discussed in the DCEI analysis. It is addressed in Section 3.2, which analyzes the interpolation in absence of shoot-through current.

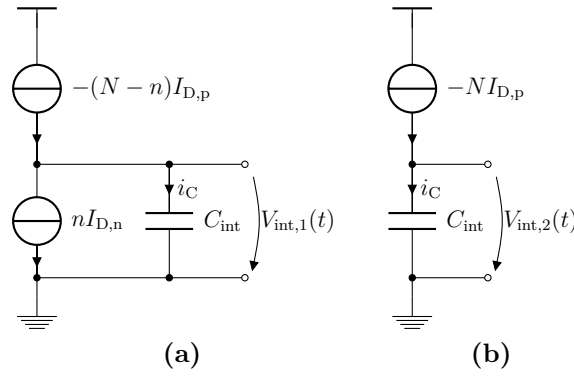


Figure 3.5 – DCEI equivalent circuit for rising interpolation in region (a) $0 \leq t < \Delta t$ with initial condition $V_{\text{int},1}(0) = 0$, and (b) $\Delta t \leq t$ with initial condition $V_{\text{int},2}(\Delta t) = V_{\text{int},1}(\Delta t)$.

The interpolation is analyzed exemplary for the case of the rising interpolation. The model calculates the TF of the DCEI, from which all other linearity measures are derived. According to (2.9), this requires to calculate the time $t_d[n]$ for each code n .

The interpolation is defined piecewise for the two regions $0 \leq t < \Delta t$, when only In_1 has switched from V_{DD} to V_{SS} , and $\Delta t \leq t$, where both input signals have switched. The equivalent circuit of the DCEI for both regions is depicted in Fig. 3.5(a) and (b). The PMOS and NMOS branches of the DCEI unit cell from Fig. 3.2(b) are modeled as current sources. Each of these sources accounts for the drain current through the two stacked transistors, that are connected to the select signals and $\text{In}_{1/2}$. The drain current $I_{\text{D},n/p}$ can be described by the Shichman-Hodges transistor model [96]. This model is not valid for modern deep sub-micron devices. However, for digitally controlled transistors only the on-case with $V_{\text{GS}} = -V_{\text{DD}}$ and the off-case with $V_{\text{GS}} = 0$ are of interest. A $V_{\text{DS}} \rightarrow I_{\text{D}}$ transfer function according to the Shichman-Hodges model is fitted to simulation results of the on-case, while $I_{\text{D},\text{off}} = 0$ is assumed for the off-case.

Appendix A derives the following model equations for two cases: (a) for ideal input signals with $t_f = 0$, and (b) for input signals with finite slope $t_f > 0$. As the resulting ordinary differential equation (ODE) for node V_{int} is a Riccati equation of the form

$$\frac{dV_{\text{int}}(t)}{dt} = aV_{\text{int}}(t)^2 + bV_{\text{int}}(t) + c, \{a, b, c \in \mathbb{R}\}, \quad (3.1)$$

it is not possible to give an analytical solution on which the influence of the important parameters is easily observed. The constants a , b , and c abbreviate technology parameter expressions. Therefore, the ODE is evaluated numerically and the linearity influencing parameters from the literature are varied to check their influence on the INL.

One measure that can be described analytically, and that is also important for model investigations of the upcoming PIs, is the minimum rise time $t_{\text{int},0}$ of the interpolation node. It is defined as the rise time $V_{\text{int}} : 0 \rightarrow V_{\text{th,inv}}$ for $n = 0$ and $t_f = 0$:

$$t_{\text{int},0} = \frac{C_{\text{int}} V_{\text{th,inv}}}{NI_{\text{D,sat},0}}. \quad (3.2)$$

This measure is more convenient to use than τ_{int} , as it can be directly extracted from the waveforms of a simulation. However, $t_{\text{int},0}$ is also influenced by t_f . The actually measured

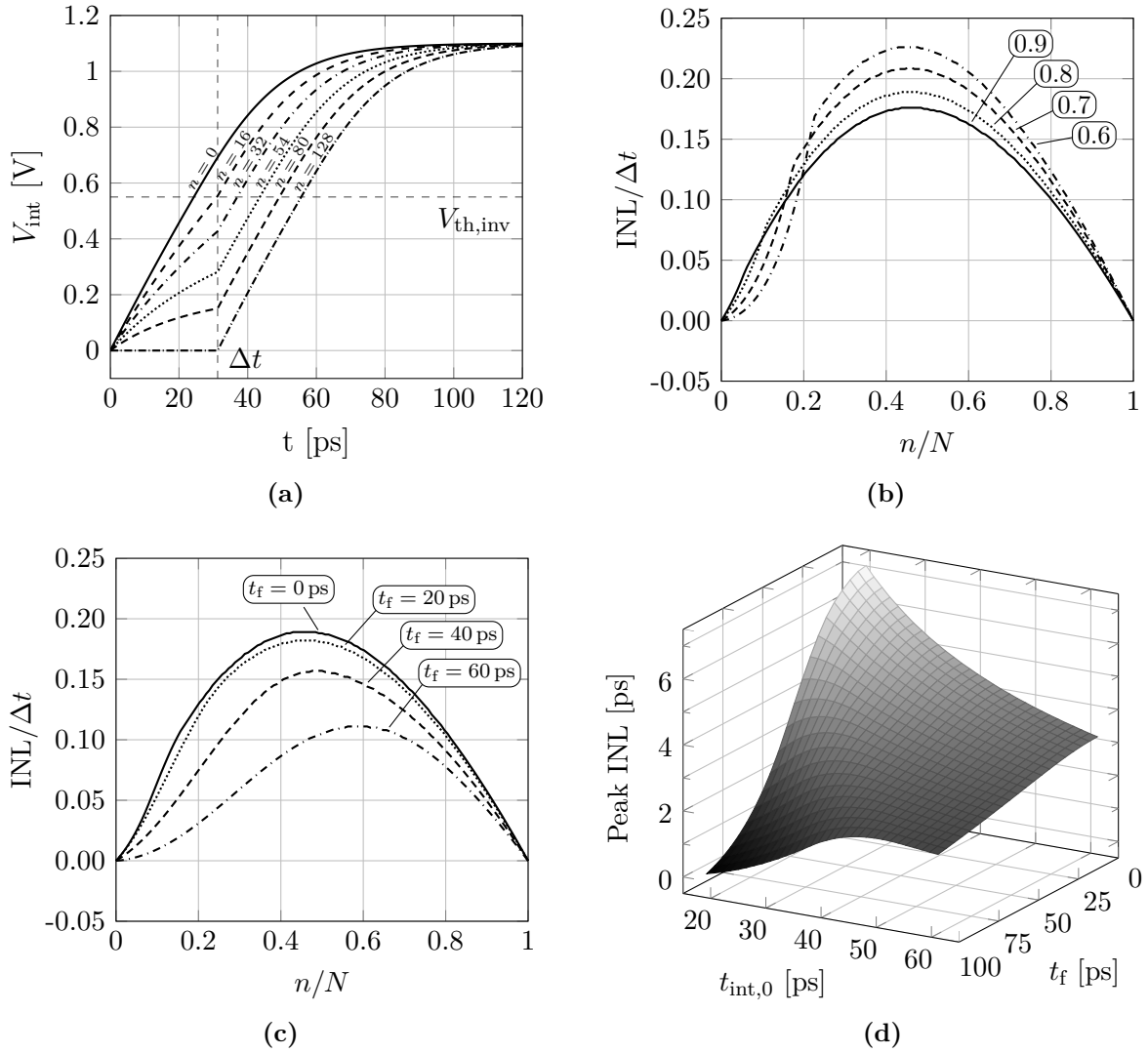


Figure 3.6 – Results of DCEI model evaluation:

- (a) waveforms at the interpolation node for different codes n and $t_f = 0$,
- (b) calculated INL for different ratios $t_{\text{int},0}/\Delta t$ and $t_f = 0$,
- (c) calculated INL for different t_f at $t_{\text{int},0}/\Delta t = 0.8$, and
- (d) peak INL for variation of $t_{\text{int},0}$ and t_f for $\Delta t = 31.25$ ps.

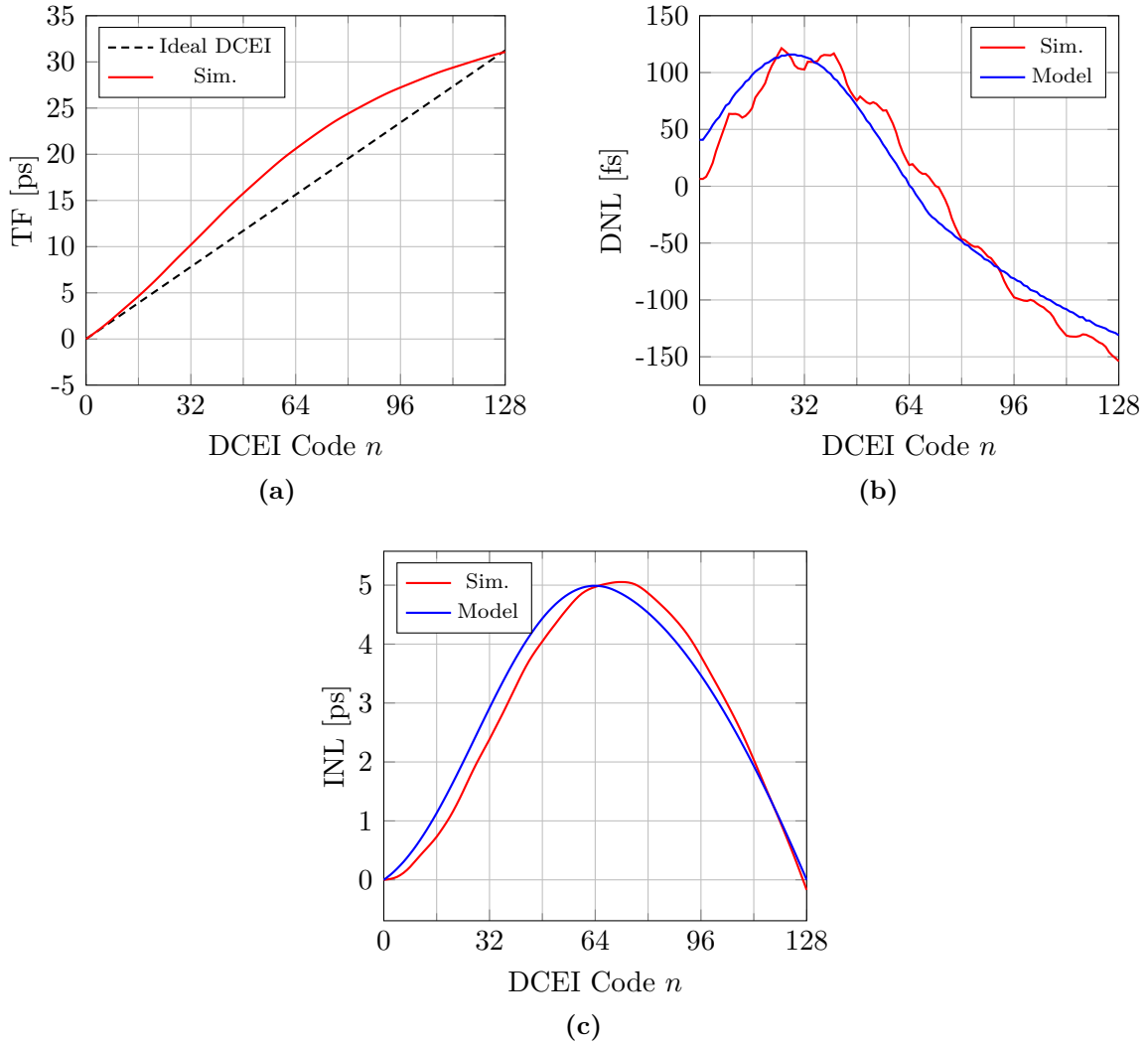


Figure 3.7 – Simulated and modeled nonlinearity of the DCEI: (a) TF, (b) DNL, and (c) INL. Model parameters: $t_{\text{int},0} = 26$ ps, $t_f = 32$ ps, $\Delta t = 31.25$ ps.

waveform for $n = 0$ and $t_f > 0$ leads to

$$\begin{aligned}
 t_{\text{int}} &= \frac{C_{\text{int}} V_{\text{th,inv}}}{N I_{\text{D,sat},0}} + \frac{t_f}{2} \\
 &= t_{\text{int},0} + \frac{t_f}{2},
 \end{aligned} \tag{3.3}$$

from which $t_{\text{int},0}$ can be calculated. The fall time t_f does not measure the 90% \rightarrow 10% fall time, but the time from when a current starts to flow in the DCEI branches until it reaches the maximum current, which is approximately the fall time $V_{\text{In}_{1/2}} : V_{\text{th,inv}} \rightarrow V_{\text{SS}}$.

First, the model equations for case (a) with $t_f = 0$ are investigated. Fig. 3.6(a) plots the waveforms at the interpolation node V_{int} for different digital codes n . For $t < \Delta t$ V_{int} is charged nonlinearly by $(n - N)$ cells, and afterwards for $t \geq \Delta t$ linearly by all N cells. If these plots are evaluated for their threshold crossing times at each code, the INL can be calculated according to (2.11). The INL plots in Fig. 3.6(b) show that the rise time

at the interpolation node can then be traded-off against overall linearity. This is done by e.g. increasing the capacitance C_{int} , which influences $t_{\text{int},0}$ linearly according to (3.2). However, for reduced nonlinearity also the slopes of the signals degrade. This has two consequences: first, jitter generated in the signal path increases with degrading slopes [11], thus increasing the generated phase noise, and second, implementation for high frequencies can only support signals up to a certain rise time.

For finite slopes with $t_f > 0$ in case (b) the INL's peak and shape changes. Fig. 3.6(c) plots the INL for a ratio of $\Delta t/t_{\text{int},0} = 0.8$ and variation of t_f . It shows a strong sensitivity towards the slopes of $\text{In}_{1/2}$. As the peak INL is an important measure, the two influencing parameters $\Delta t/t_{\text{int},0}$ and t_f are varied to produce the surface plot from Fig. 3.6(d). This shows that a small $t_{\text{int},0}$ makes the INL sensitive to t_f , while a large one leads to an insensitivity. The DCEI produces a relatively low INL for small $t_{\text{int},0}$ and large t_f . This, however, is a poor design point according to the reasons mentioned in the previous paragraph. The operating frequency of 2 GHz leads to a signal period of $T_{2\text{GHz}} = 500$ ps, so that $t_f = 100$ ps would degrade the rectangular waveforms severely (as discussed earlier: t_f only measures $\sim 50\%$ of the actual fall time).

Simulation results of the discussed DCEI are compared to a model evaluation in Fig. 3.7. The model parameters $t_{\text{int},0} = 26$ ps, $t_f = 32$ ps, and $\Delta t = 31.25$ ps are extracted from circuit simulation. First, Fig. 3.7(a) shows the simulated TF compared to the ideal one. As discussed in Section 2.2, the TF is used to extract the FSR, while DNL and INL are utilized for a more detailed evaluation of the nonlinearity. Therefore, the TF is left out for the remaining discussion in this chapter and is only used to proof the 2π DTC range in measurements. The DNL in Fig. 3.7(b) shows a good matching between simulation and measurement. The ripple on the simulated DNL can be related to the columns of the DCEI array and is caused by parasitic resistors that are not part of the model. Fig. 3.7(c) shows that the overall INL shape as well as its peak matches the model well.

To summarize these results, if the DCEI is kept in a reasonable design point with $\sim t_f \leq 50$ ps, a low INL can be achieved for large $t_{\text{int},0}$, large t_f , and small Δt . Consequently, the peak INL needs to be traded-off against jitter and the maximum achievable frequency. Assuming Δt is given by the DTC coarse tuning stages, the peak INL sensitivity translates to following physical design parameters:

$$|\text{INL}_{\text{max}}| \downarrow \Leftrightarrow C_{\text{int}} \uparrow, V_{\text{th,inv}} \uparrow, |I_{\text{D,sat},0}| \downarrow, t_f \uparrow \quad (3.4)$$

The substantial part of the INL is systematic and cannot be reduced below a certain level for reasonable design parameters. As a low INL is desirable, the mechanisms that lead to high nonlinearity are analyzed in the upcoming section and a linearized design is presented.

3.2 Contention-Free Digitally Controlled Edge Interpolator

The DCEI modeling showed the nonlinearity of a PI exhibiting shoot-through currents. This section proves through further modeling, that the shoot-through current is the major contributor to nonlinearity and clearly dominates over the ratio $\tau_{\text{int}}/\Delta t$ and the rise/fall

time of the input signals $t_{r/f}$. This makes the prevention of shoot-through current imperative for a high linearity PI design.

Two different PI architectures are reported in the literature, that solve the problem of shoot-through currents and provide high linearity. First, a short-circuit-current-suppression (SCCS) PI with 1 bit resolution is used to generate an intermediate event between the two rising edges of the input signals in [84] (50% interpolation). Resolutions above 1 bit are achieved by cascading several 1 bit SCCS cells. The drawback of this approach is a long cascade of interpolation stages if high resolution is targeted, and hence accumulated jitter. More recent publications [14, 37] implement a PI similar to [84] as a cascade of 1 bit interpolation cells, proving the value of this concept for linear phase interpolation.

Second, current integrating PIs are reported that use current steering DACs to integrate a current on a capacitor. An initial current $0 \leq I_1 \leq I_{\max}$ is triggered by the rising edge of In_1 , and with the rising edge of In_2 the current is raised to I_{\max} . The crossing of a following inverter's threshold voltage $V_{th,inv}$ evaluates the interpolation. The crossing time t_{cross} depends linearly on the charging current in the first interpolation period. This concept has been implemented as a 1 bit and a 3 bit current integrating PI in [86], and as a 5 bit PI in [85]. The advantage is a single stage interpolation. However, it requires the design of a current steering DAC, whose nonlinearity directly influences the linearity of the phase interpolation.

A third approach uses a very similar concept to implement a constant slope charging fine tuning [1]. However, the implementation is a fine delay with a single input and not a PI. Here an internal node (comparable to the interpolation node of the other designs) is pre-charged by a current-steering DAC (I-DAC), and then a constant current charges it further until the threshold of a subsequent inverter is crossed. The pre-charging is triggered by a reset mechanism, while in [85, 86] one of the input signals triggers the pre-charging of the interpolation node by their current steering DAC, and therefore implement an interpolation. This work presents an impressive peak INL of only 328 fs for a FSR of 189 ps, but measurements were only presented with an external DAC instead of the chip internal one.

The drawback of all discussed PIs is that the programmed code is only applied on the rising output edge. The falling edge is used to reset the circuit and prepare the internal

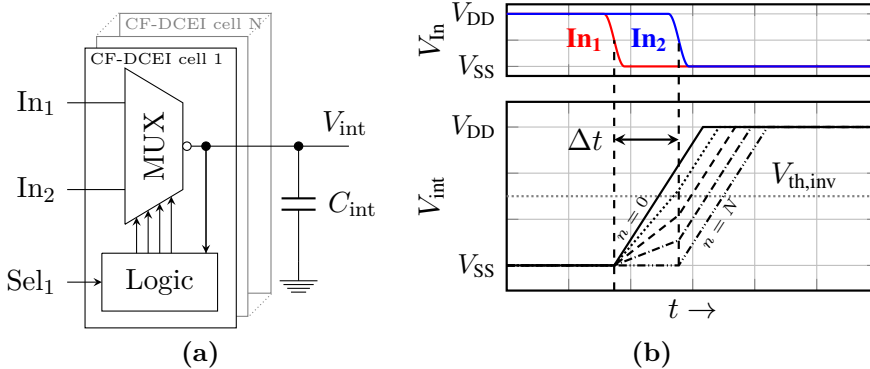


Figure 3.8 – (a) Implementation of CF-DCEI unit cells, and (b) waveforms of the ideal linearized interpolation process for different codes n .

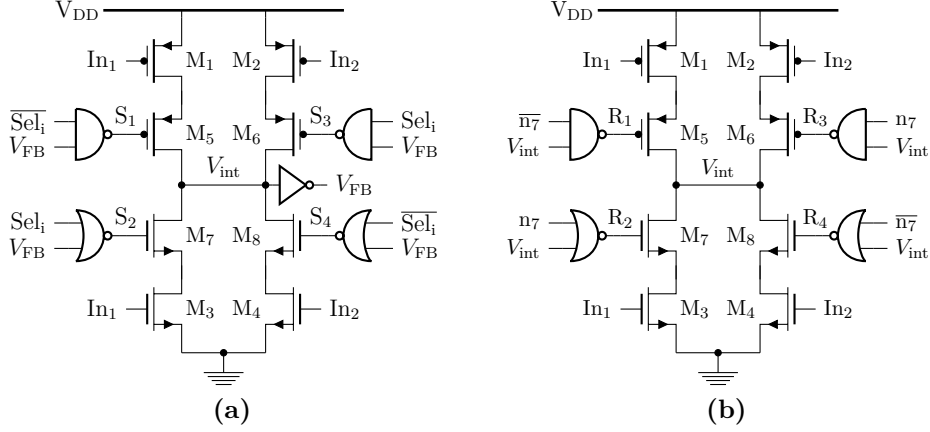


Figure 3.9 – Implementation of (a) i^{th} interpolation cell, and (b) retention cells.

nodes for the next interpolation.

This section presents the CF-DCEI, which prevents the shoot-through currents by adding a more complex control logic to the DCEI's analog MUX core. The CF-DCEI keeps the advantage of steep slopes throughout the signal path, and applies the programmed codes on both, rising and falling output edge. First, the circuit architecture that enables contention free interpolation is presented. Afterwards, an analytical model of the CF-DCEI is derived to quantify the improvements through contention-free operation on the linearity. This model also evaluates the influence of the ratio $\tau_{\text{int}}/\Delta t$ and the input signals slopes $t_{r/f}$ on the CF-DCEI linearity. As the above described PIs operate in principle very similar, the findings of the model can also be applied to these designs. Based on the model, key design parameters that influence the linearity of the CF-DCEI are identified.

3.2.1 Design and Implementation

The CF-DCEI design is based on the DCEI design. The analog MUX core implementation is identical, with a modified, more complex control logic. Fig. 3.8(a) shows the newly placed control logic, that provides four control signals to the analog MUX core, derived from Sel_i and a logical feedback from the interpolation node. This logic enables the prevention of shoot-through current during the phase interpolation period. The plots in Fig. 3.8(b) show that this leads to a linear charging of V_{int} during the denoted time interval Δt (the DCEI showed nonlinear charging in this region).

The interpolation cell (INTC) core structure is implemented similar to the DCEI as a MUX, based on two tristate inverters, and is shown in Fig. 3.9(a). The logic block from Fig. 3.8(a) is implemented with the four logic gates, two NAND and two NOR gates, and controls the select transistors $M_5 - M_8$. Each PMOS and NMOS branch has its own control signal $S_1 - S_4$, generated by the individual selection state of each INTC's Sel_i and a logical feedback signal V_{FB} from V_{int} , which indicates a finished interpolation. Only one branch per cell can be active at a time, preventing shoot-through currents between V_{DD} and V_{SS} during the time when In_1 and In_2 have different logic levels. Fig. 3.10(a) illustrates the selection of the INTC branches and how the changes of the logic control signals are triggered. A finished interpolation triggers a change in the selection signals

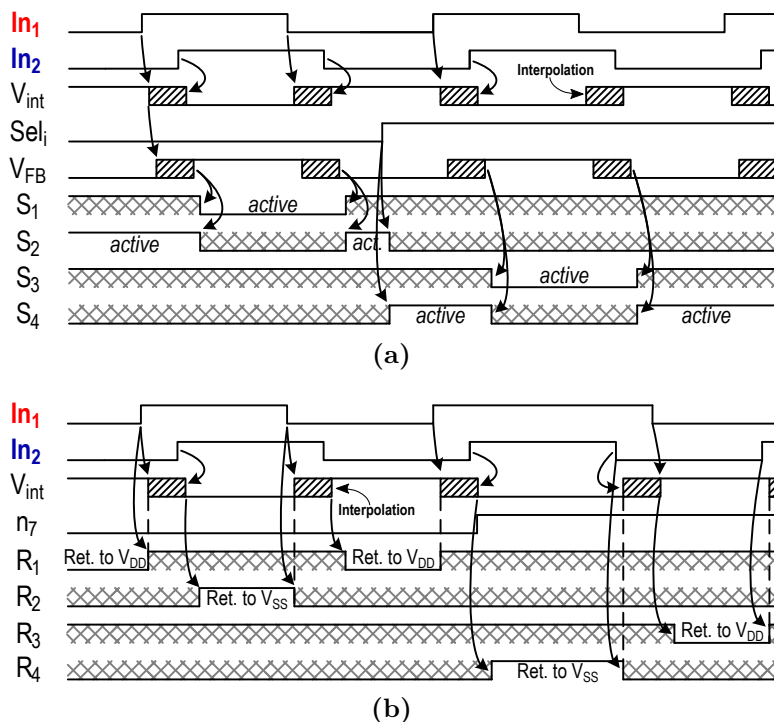


Figure 3.10 – Logic timing diagram of (a) interpolation cells, and (b) retention cells.

and prepares the cell for the subsequent interpolation. As falling and rising interpolation are altering, the select signals switch even without a change of Sel_i twice per cycle. A rising interpolation requires one PMOS branch to be active in each INTC, and a falling interpolation one NMOS branch. New values for Sel_i are latched into the cell array with the falling edge of DTC_{out} , which is the inverted V_{int} . As the CF-DCEI interpolates on rising and falling edges, a duty cycle of 50% is maintained for constant code n .

However, CF-DCEI cells with this logic leave V_{int} floating between interpolations, as the INTCs provide no conducting path between V_{DD}/V_{SS} and V_{int} during this time. This makes V_{int} susceptible to noise, generating a non-deterministic source for nonlinearity as each interpolation can have a different initial voltage at V_{int} . Therefore, retention cells (RETC) are introduced, that hold the state of V_{int} during this period. Fig. 3.9(b) shows their design, which is similar to the INTCs, but with different logic generating the control signals R_1 - R_4 . Bit n_7 of the coarse-tuning stage indicates whether In_1 is leading or lagging In_2 and is used to select the "earlier" input. Fig. 3.10(b) shows the timing diagram of the RETCs. The retention branches are turned off by the "earlier" input signal shortly before the interpolation starts, and turned on again shortly after V_{int} switched, indicating a finished interpolation. This control ensures, that RETCs provide only a conducting path between V_{DD}/V_{SS} and V_{int} during the period when the INTCs leave V_{int} floating. Branches in INTCs and RETCs cannot be active at the same time, preventing shoot-through currents between both cell types. Fig. 3.11 shows simulations of the interpolation process, illustrating retention and interpolation periods. The RETCs recover V_{int} to V_{DD}/V_{SS} before the subsequent interpolation starts.

As the same resolution and operating frequency as for the DCEI are targeted, the array structure and the digital decoder are kept almost identical. Fig. 3.12 shows the cell array

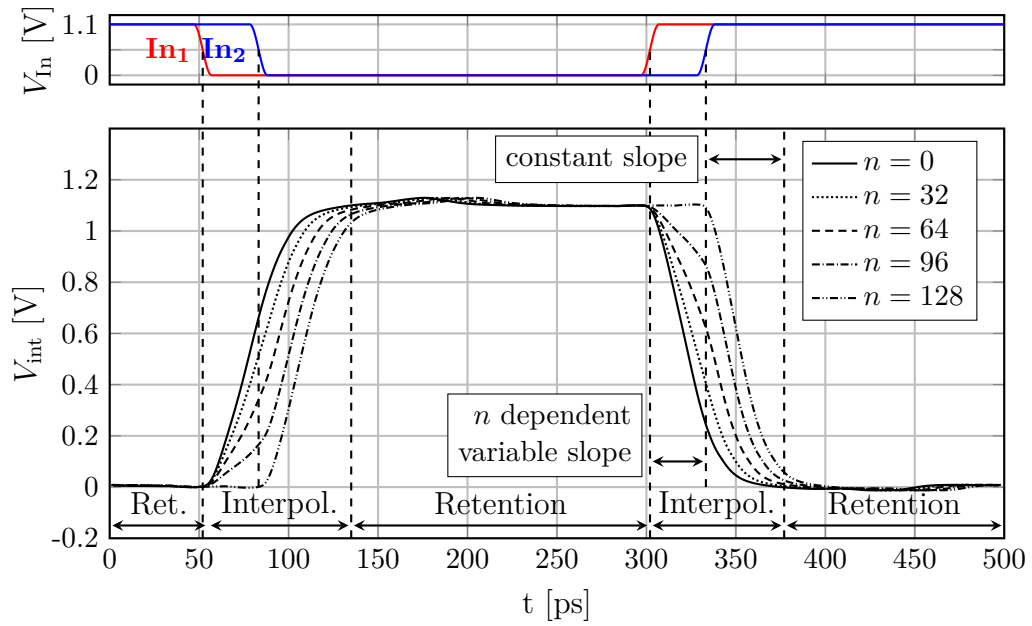


Figure 3.11 – Simulated waveforms of the interpolation process for different code words n .

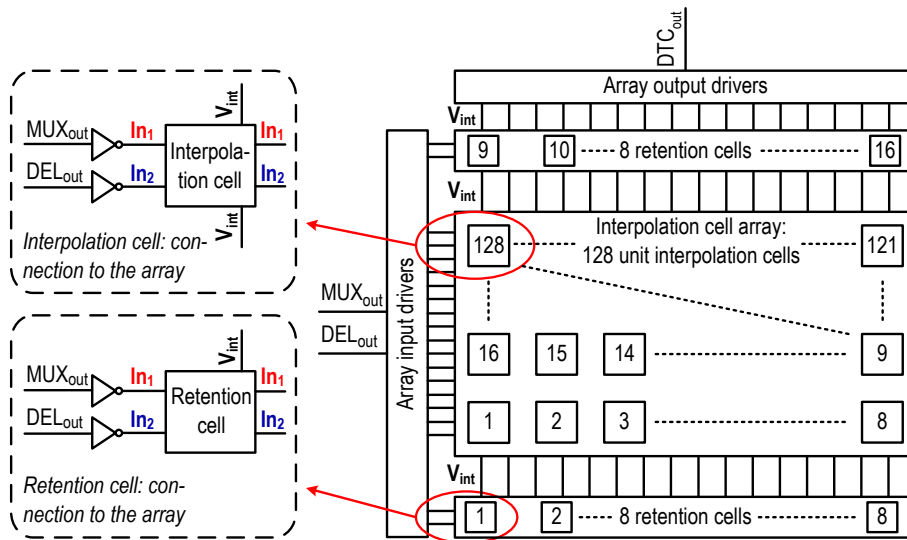


Figure 3.12 – CF-DCEI cell array topology.

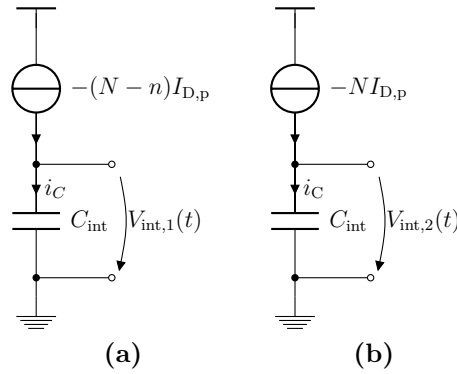


Figure 3.13 – CF-DCEI equivalent circuit for rising interpolation in region (a) $0 \leq t < \Delta t$ with initial condition $V_{\text{int},1}(0) = 0$, and (b) $\Delta t \leq t$ with initial condition $V_{\text{int},2}(\Delta t) = V_{\text{int},1}(\Delta t)$.

architecture of the CF-DCEI. The 7 bit resolution is achieved by a total of 128 identical INTCs placed in an array of 16x8 cells, thermometrically coded to ensure monotonic behavior. All INTCs have identical input signals $\text{In}_{1/2}$, and their outputs are connected to the common interpolation node V_{int} . At the top and the bottom of the array, a row of eight RETCs each is placed. The total number of 16 RETCs is as a compromise between a minimum driving strength, that is needed to recover V_{int} to $V_{\text{DD}}/V_{\text{SS}}$, and a reasonable integration into the cell array, which leads to a multiple of eight cells.

The static current consumption is equal for all codes, as the same capacitance C_{int} is charged and discharged during each cycle. For simplicity, the DTC architecture was discussed single-ended, but the presented description applies to a differential design accordingly. The actual DTC is implemented with pseudo-differential signals. This accounts for the problem of differences in rising and falling interpolation for cross corner process variations, i.e. slow NMOS and fast PMOS devices, or fast NMOS and slow PMOS devices.

3.2.2 CF-DCEI Model

The following analysis of the CF-DCEI's linearity is on the example of a rising interpolation, the falling interpolation can be analyzed in an analogous manner.

The CF-DCEI's equivalent circuit from Fig. 3.13(a) and (b) describes the piecewise defined charging of C_{int} by the PMOS devices of the INTCs. The difference to the DCEI model is the absence of the NMOS current source for region $0 \leq t < \Delta t$. For $n = 0$ the first region is identical, as the NMOS current source in the DCEI draws no current then. The second region $\Delta t \leq t$ is identical to the DCEI model for all n . In the CF-DCEI architecture, $(N - n)$ cells start charging with switching of In_1 and the remaining n cells delayed by Δt with switching of In_2 .

As the assumption of ideal input signals with fall time $t_f = 0$ and no channel length modulation $\lambda = 0$ simplifies the analysis for a start, the equations are derived in three steps with increasing accuracy: (a) $t_f = 0$, $\lambda = 0$, (b) $t_f = 0$, $\lambda \neq 0$, and (c) $t_f > 0$ with arbitrary λ . The detailed derivation on how $t_d[n]$, $\text{TF}[n]$, and $\text{INL}[n]$ are obtained through a 1st order model for (a) and (b) can be found in Appendix B.1, and through a 2nd order model for (c) in Appendix B.2.

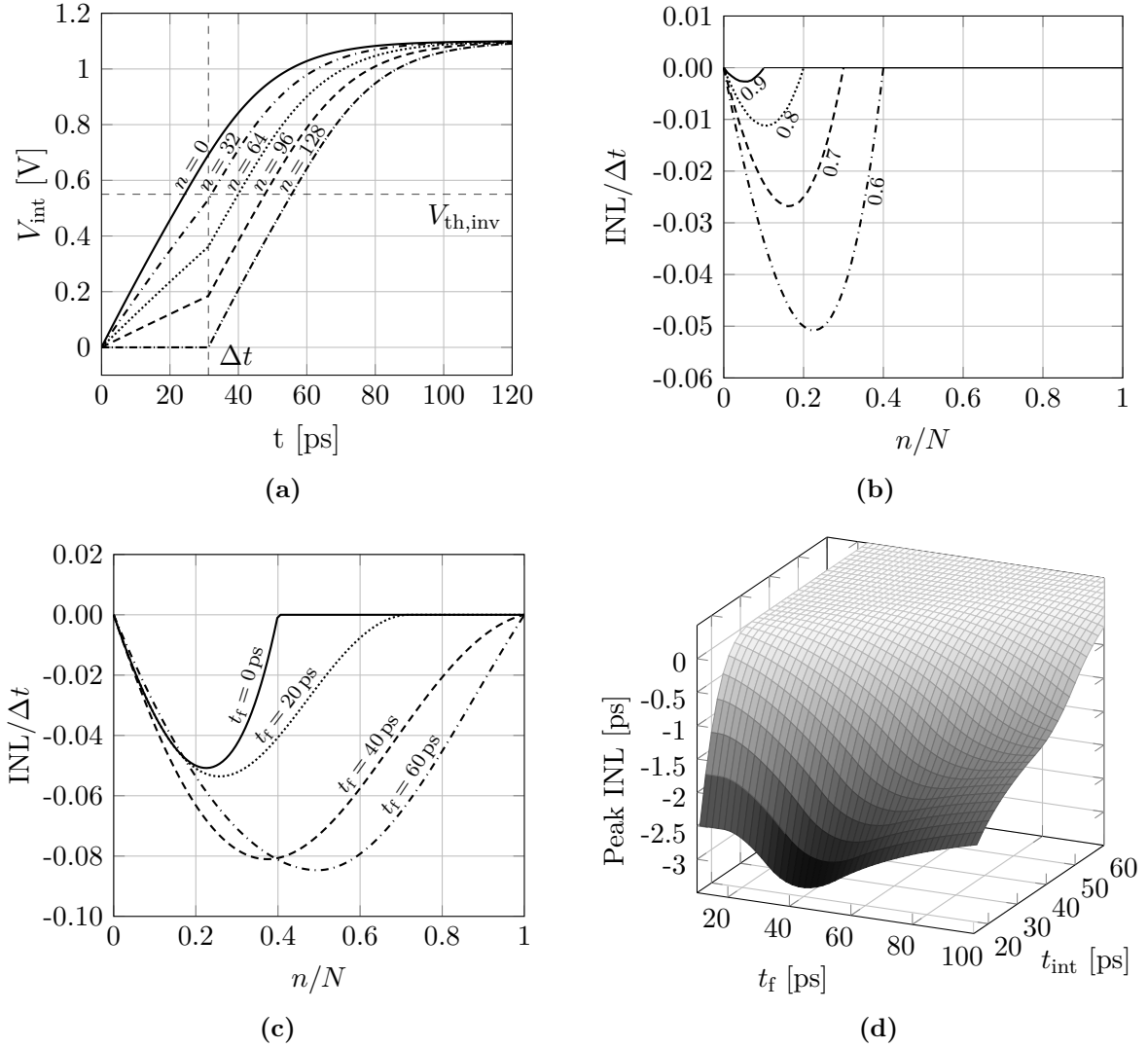


Figure 3.14 – Results of CF-DCEI model evaluation:

- (a) waveforms at the interpolation node from numerical model evaluation for different codes n and $t_f = 0$,
- (b) INL calculated from 1st order model for different ratios $t_{\text{int},0}/\Delta t$,
- (c) INL calculated from 2nd order model for different t_f at $t_{\text{int},0}/\Delta t = 0.8$, and
- (d) peak INL calculated from 2nd order model for variation of $t_{\text{int},0}$ and t_f with $\Delta t = 31.25$ ps.

A numerical evaluation of the ODEs from case (a) leads to the waveforms at V_{int} as shown in Fig. 3.14(a). The interpolation node is charged linearly with an n -dependent slope in the first region, which is the major improvement compared to the DCEI.

Case (a) shows that the INL only depends on the design parameters t_{int} and Δt , which can be easily obtained from simulation without extracting technology parameters. The INL derived in Appendix B.1 is piecewise defined as:

$$\text{INL}_1[n] = \left(\frac{N}{N-n} - 1 \right) t_{\text{int},0} - \frac{n}{N} \Delta t \quad , \text{ where } 0 \leq n \leq \left\lfloor N \left(1 - \frac{t_{\text{int},0}}{\Delta t} \right) \right\rfloor, \text{ and} \quad (3.5)$$

$$\text{INL}_2[n] = 0 \quad , \text{ where } \left\lfloor N \left(1 - \frac{t_{\text{int},0}}{\Delta t} \right) \right\rfloor < n \leq N. \quad (3.6)$$

The equations are plotted in Fig. 3.14(b) for different ratios $t_{\text{int},0}/\Delta t$. The peak INL is then given as

$$|\text{INL}_{\text{max}}| = \left(\sqrt{t_{\text{int},0}} - \sqrt{\Delta t} \right)^2, \text{ where } 0 \leq t_{\text{int},0} \leq \Delta t. \quad (3.7)$$

Ideal linearity can be achieved for $t_{\text{int},0} \geq \Delta t$, as increasing t_{int} beyond Δt leads to $\text{INL}[n] = 0$. Case (b) with $\lambda \neq 0$ yields into an effectively smaller charging time

$$t_{\text{int},0,\lambda \neq 0} \approx t_{\text{int},0}/(1 + 0.5\lambda_p V_{\text{th,p}}), \quad (3.8)$$

leaving the general INL shape almost untouched. The peak INL can be reduced by either reducing Δt or increasing $t_{\text{int},0}$, which leads to a variation of the following physical design parameters:

$$|\text{INL}_{\text{max}}| \downarrow \Leftrightarrow C_{\text{int}} \uparrow, V_{\text{th,inv}} \uparrow, |I_{D,\text{sat},0}| \downarrow \quad (3.9)$$

Case (c) expands the model to include realistic input slopes $t_f > 0$, as they show an influence on the interpolation [95]. The analytical INL equations derived in Appendix B.2 are plotted in Fig. 3.14(c) for a constant ratio $t_{\text{int},0}/\Delta t = 0.8$ and different t_f (all equations are listed in Table B.4). Unlike for the DCEI, the CF-DCEI's peak INL reduces for smaller t_f , leading to a good design point as steep slopes and low INL are compatible. The parameters t_f and $t_{\text{int},0}$ can now be varied and the resulting INL is evaluated for its peak, which is plotted in Fig. 3.14(d). As the equations do not allow a normalized scaling of all parameters to Δt , the implemented case of $\Delta t = 31.25$ ps is plotted. Following trend is observed from equations and plot:

$$|\text{INL}_{\text{max}}| \downarrow \Leftrightarrow t_f \downarrow, t_{\text{int}} \uparrow, \Delta t \downarrow, \sqrt{t_f t_{\text{int}}} \downarrow, \sqrt{t_f \Delta t} \downarrow \quad (3.10)$$

Fig. 3.14(d) shows that even if the INL would be ideal according to sufficiently large t_{int} , input signals with large t_f limit the linearity. As a rule of thumb (with some exceptions according to Fig. 3.14(d)) the peak INL reduces for increasing t_{int} , e.g. through additional C_{int} , or for reducing t_f , e.g. through stronger input drivers. This has to be traded off against phase noise or jitter, which increases for degrading slopes in the signal path.

Fig. 3.15 compares circuit simulation with extracted layout parasitics to the INL calculated with the model. The model estimates the peak INL well, but it does not cover effects introduced through parasitic resistances. Overall the model enables a quick evaluation of the peak INL. For steep slopes of the input signals the simple equation (3.7) can be used and delivers sufficient accuracy to identify key parameters in circuit design. As the

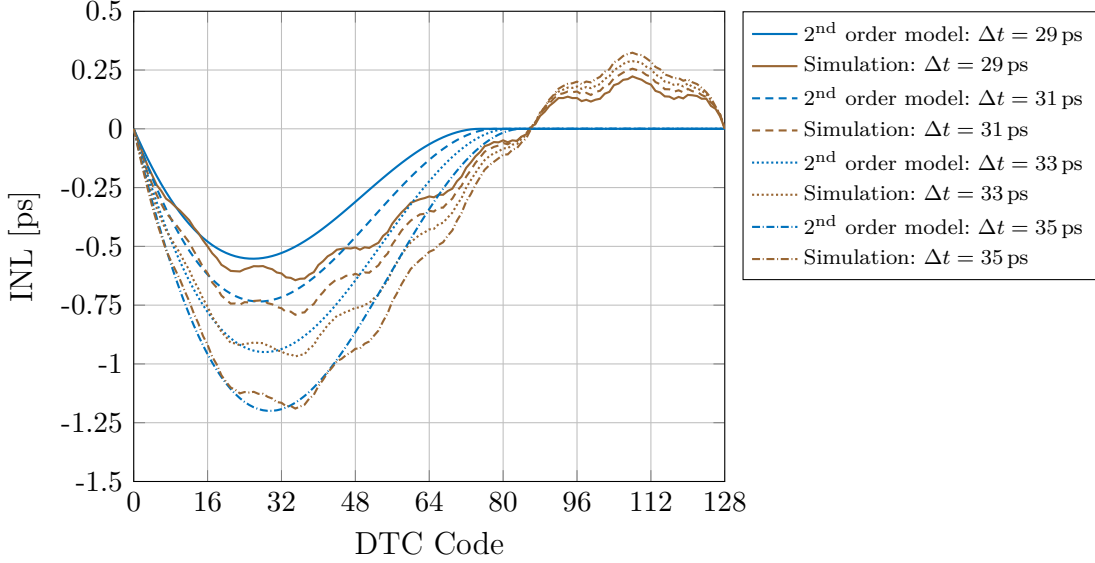


Figure 3.15 – Comparison between 2nd order model and simulation with layout extracted parasitics. Model parameters: $t_{\text{int}} = 24.5$ ps, $t_f = 25$ ps, and $N = 128$.

developed equations are purely analytical, a surface plot as in Fig. 3.14(d) that allows a more complex overview on the design parameters can be calculated in the order of seconds.

While the CF-DCEI delivers superior linearity compared to the DCEI, it increases the current consumption significantly, as the newly introduced control logic operates on a rate of $2f_{\text{out}} = 4$ GHz to generate the control signals for contention free operation on rising and falling edges. A comparison table with all key performances of the discussed designs is presented in the summary of this chapter (Section 3.5).

Comparison to Switched Capacitor Based Fine Tuning

How does the CF-DCEI’s linearity compare to the highly linear switched capacitor based fine tuning now? The major benefit of the switched capacitor delay cells is a high linearity, and its major drawbacks are the undefined range and the code dependent current consumption [1, 7, 26]. As the CF-DCEI is not suffering from these drawbacks, a comparison of the nonlinearity shows whether the delay cells achieve better linearity than the CF-DCEI, or if the CF-DCEI is also the favorable implementation regarding nonlinearity. Circuit simulations results of the simple single stage switched capacitor fine tuning from Fig. 3.16(a) are plotted in Fig. 3.17. Key parameter of the design is the range of the tuning capacitance $C_{\text{min}} \leq C_{\text{tune}} \leq C_{\text{max}}$, which is varied in linear steps of $(C_{\text{max}} - C_{\text{min}})/N$ over code n .

Appendix C derives a model for the switched capacitor circuit to analyze its nonlinearity. The following discussion is on the example of a rising edge at node $V_{\text{out}}^{\text{sc}}$, and can be extended to the falling edge in an analogous manner. Next to the range of C_{tune} , the model is based on three time intervals as denoted on the idealized waveforms in Fig. 3.16(b): 1) The delay $t_{\text{int},1}^{\text{sc}}[n]$ until the threshold of the output inverter is crossed at $V_{\text{int}}^{\text{sc}}$ and the output net starts switching ($V_{\text{int}}^{\text{sc}}(t) : V_{\text{DD}} \rightarrow V_{\text{th,inv}}$); 2) the relevant fall time $t_{\text{int},2}^{\text{sc}}[n]$ at $V_{\text{int}}^{\text{sc}}$ that influences the turn-on time of the output inverter ($V_{\text{int}}^{\text{sc}}(t) : V_{\text{th,inv}} \rightarrow \sim 0.1V_{\text{SS}}$); and 3) the delay $t_{\text{out}}^{\text{sc}}[n]$ until the output crosses the threshold of the subsequent stage ($V_{\text{out}}^{\text{sc}}(t) : V_{\text{SS}} \rightarrow V_{\text{th,inv}}$).

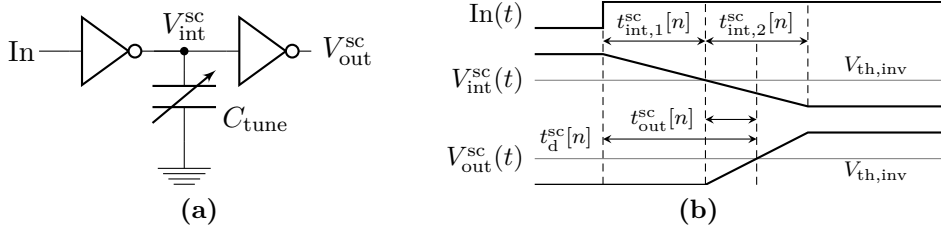


Figure 3.16 – Switched capacitor based fine tuning: (a) circuit implementation, and (b) idealized waveforms for ideal input signal.

The minimum delays at $V_{\text{int}}^{\text{sc}}$ are defined as $t_{\text{int},1,0/2,0}^{\text{sc}} = t_{\text{int},1/2}^{\text{sc}}[0]$, and the minimum delay $t_{\text{out},0}^{\text{sc}}$ at $V_{\text{out}}^{\text{sc}}$ is measured for an ideal falling edge at $V_{\text{int}}^{\text{sc}}$. As in the case of the CF-DCEI, the time delays are easily extracted from simulation. According to the derived model, the overall delay from switching of the input to threshold crossing of the output signal is given as

$$\begin{aligned} t_d^{\text{sc}}[n] &= t_{\text{int},1}^{\text{sc}}[n] + t_{\text{d,out}}^{\text{sc}}[n] \\ &= \left(1 + \frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}}\right) t_{\text{int},1,0}^{\text{sc}} + \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[n]}, \text{ with} \end{aligned} \quad (3.11)$$

$$t_{\text{int},2}^{\text{sc}}[n] = \left(1 + \frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}}\right) t_{\text{int},2,0}^{\text{sc}}. \quad (3.12)$$

Note that the overall delay depends only on the minimum delays $t_{\text{int},1,0}^{\text{sc}}$, $t_{\text{int},2,0}^{\text{sc}}$, and $t_{\text{out},0}^{\text{sc}}$, the range of C_{tune} , and the programming. It is a superposition of the delays until $V_{\text{int}}^{\text{sc}}$ and $V_{\text{out}}^{\text{sc}}$ cross the threshold of their respective subsequent stage. The former delay depends linearly on the code, and on the minimum delay $t_{\text{int},1,0}^{\text{sc}}$ at this net. The latter one depends nonlinearly on the fall time $t_{\text{int},2}^{\text{sc}}[n]$ of $V_{\text{int}}^{\text{sc}}$, which introduces the nonlinearity to the fine

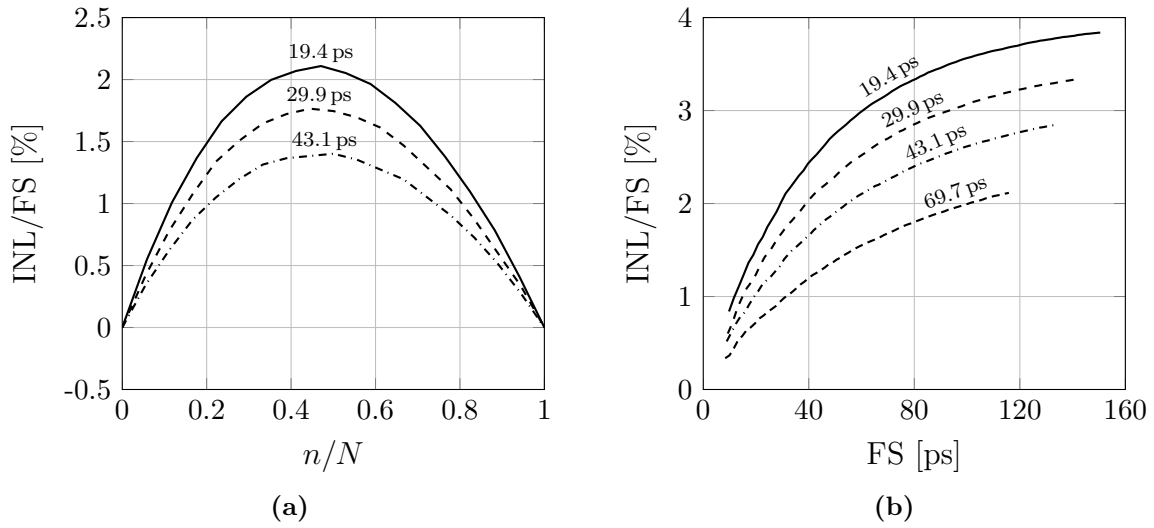


Figure 3.17 – Simulated nonlinearity of the switched capacitor based fine tuning for different $t_{\text{int},2,0}$: (a) INL for FS = 31.25 ps, and (b) peak INL plotted against FS.

tuning block. This proves the assumption from [1], that the code-dependent slope is the main nonlinearity source in this fine tuning architecture. Further comparison of model and simulated INL are available in Appendix C.

Depending on the ratio of the CF-DCEI's interpolation node rise time and the switched capacitor fine tuning's minimum fall time $t_{\text{int}}/t_{\text{int},2.0}^{\text{sc}}$, either one or the other circuit has an advantage in terms of nonlinearity. The CF-DCEI's simulation results from Fig. 3.15 show $|\text{INL}_{\text{max}}| \approx 2\%$ for $\text{FS} = \Delta t = 29.49 \text{ ps}$, which is very close to $|\text{INL}_{\text{max}}^{\text{sc}}| \approx 1.7\%$ for $\text{FS}^{\text{sc}} = 29.9 \text{ ps}$ from Fig. 3.17(a). From linearity point of view, the CF-DCEI architecture can outperform the switched capacitor based fine tuning in the FS ranges compared here, if additional capacitance is added to the CF-DCEI's interpolation node (the presented design uses only the parasitic layout capacitance). Additionally, the CF-DCEI has all the inherent advantages (e.g. well defined tuning range) that the switched capacitor fine tuning lacks. However, as discussed in Chapter 1, for wide FS of multiple VCO periods the CF-DCEI has excessive nonlinearity and the switched capacitor fine tuning has a clear advantage.

3.3 Digitally Controlled Two-Points Edge Interpolator

The DCEI² topology aims at reduced power consumption and higher operation frequency compared to the DTCs based on the DCEI and CF-DCEI. While the higher frequency can be achieved by proper dimensioning of the signal path and the digital logic, a meaningful power reduction requires innovations in the general PI design. The key improvement in the DCEI² are two subsequent interpolations, the first responsible to extend the PI range by a factor of two, and the second to provide high resolution. The extended range allows the removal of the MUX+DEL stage, simplifying the overall DTC design significantly and reducing the current consumption. However, analysis of DCEI and CF-DCEI showed that the peak INL depends heavily on the interpolation range, which should not increase compared to the other designs. The DCEI and CF-DCEI are also capable of operating at twice the input phase spacing, only at the expense of extensive INL increase.

The reduction from a three-stage to a two-stage design has three major benefits: 1) the current and area of the MUX+DEL stage are no longer accounting to the DTC; 2) the multiplexers, inverters, and the delay element in the MUX+DEL stage no longer contribute to jitter, relaxing the requirements for the MMD and the DCEI², which now have a higher jitter budget that can be used to reduce power consumption (e.g. by reducing the low-noise flip-flop transistor width); and 3) each DTC stage generates its own clock signal to latch in the new code words, allowing a reduction from three to two clock signals. One clock signal less simplifies the design of the synchronization logic, that distributes the incoming data words to the single DTC stages.

To take advantage of these benefits, the DCEI² should be comparable to the DCEI and CF-DCEI in terms of area, current consumption, load to the MMD, and generated jitter. This section shows, that the DCEI² design can compete in all mentioned points and furthermore allows the implementation of k_{PI} bits (leading to $N = 2^{k_{\text{PI}}}$ codes) with only $K = 2^{(k_{\text{PI}}-1)}$ DCEI² unit cells, which is important as the removal of the MUX+DEL stage reduces the DTC by 1 bit of resolution that needs to be re-gained by the PI.

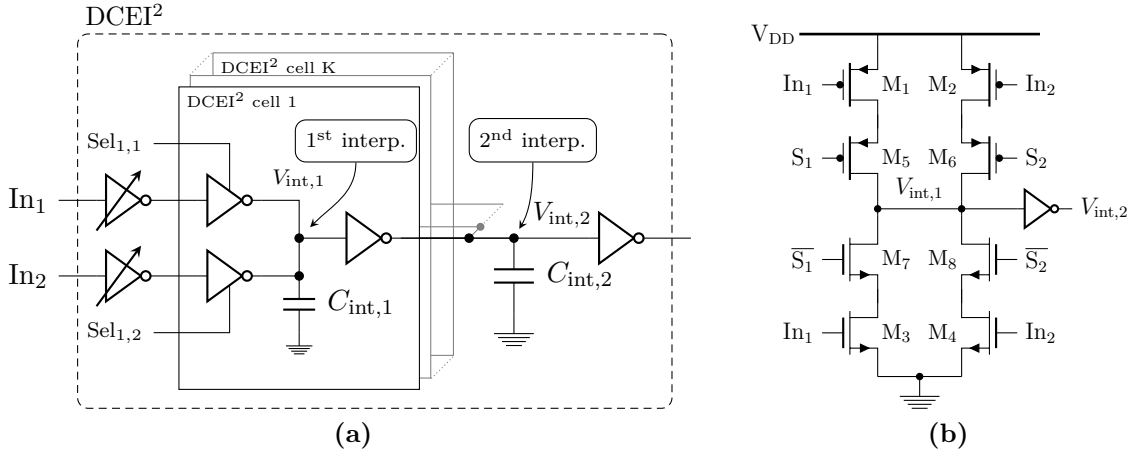


Figure 3.18 – DCEI² schematics: (a) interconnection of DCEI² unit cells and location of interpolations, and (b) unit cell transistor implementation.

3.3.1 Design and Implementation

Fig. 3.18(a) shows an overview of the DCEI² implementation. The input signals $In_{1/2}$ are shifted against each other by Δt_1 , and a subsequent set of input drivers provides sufficient driving strength for the DCEI² cell array. Each DCEI² unit cell consists of two stages: First, an input stage of two tristate inverters, which is controlled by two select signals $Sel_{i,1}$ and $Sel_{i,2}$ (for the i^{th} cell), and second, an inverter as output stage. The internal net $V_{\text{int},1}$ is not accessible from outside the unit cell. Compared to the analog core of the DCEI unit cell from Fig. 3.2(b), an inverter is added to the output and an additional select signal is needed for control. The first interpolation is local to each DCEI² unit cell at net $V_{\text{int},1}$, while the second interpolation at the common node $V_{\text{int},2}$ is driven by all unit cells. The inverter between $V_{\text{int},1}$ and $V_{\text{int},2}$ separates both nets, enabling two subsequent interpolation points. Finally, an inverter re-gains the slopes from $V_{\text{int},2}$ and drives the subsequent circuit. The transistor level implementation of the unit cells is shown in Fig. 3.18(b).

First Interpolation

The first interpolation is local to each DCEI² unit cell and can be configured individually for each unit cell. The tristate inverters can either act as a MUX and select In_1 or In_2 , or interpolate between In_1 and In_2 by selecting both, to generate a signal located temporally between these events. This implements the functionality of the MUX+DEL stage, only that the additional event is generated by interpolation and not by a delay element.

The signals S_1 , S_2 , \bar{S}_1 , and \bar{S}_2 are derived from $Sel_{i,1/2}$ (for the i^{th} cell) and control the select transistors $M_5 - M_8$. The allowed configurations are: (1) $S_1 = 1$ and $S_2 = 0$, selecting In_1 ; (2) $S_1 = 0$ and $S_2 = 1$, selecting In_2 ; and (3) $S_1 = 1$ and $S_2 = 1$, selecting $In_1 + In_2$. In the cases (1) and (2) the input stage operates similar to a CMOS multiplexer. The selected tristate device drives $V_{\text{int},1}$, and the threshold crossing is aligned with either In_1 or In_2 , which leads to the red and blue waveform at $V_{\text{int},1}$ as plotted in Fig. 3.19(a). Case (3) activates both tristate stages, leading to the constant phase interpolation on $V_{\text{int},1}$ plotted in green. The case of $S_1 = 0$ and $S_2 = 0$ is not allowed, as it leaves $V_{\text{int},1}$ floating.

The interpolation is constant, as it cannot be influenced by digital programming. Ideally,

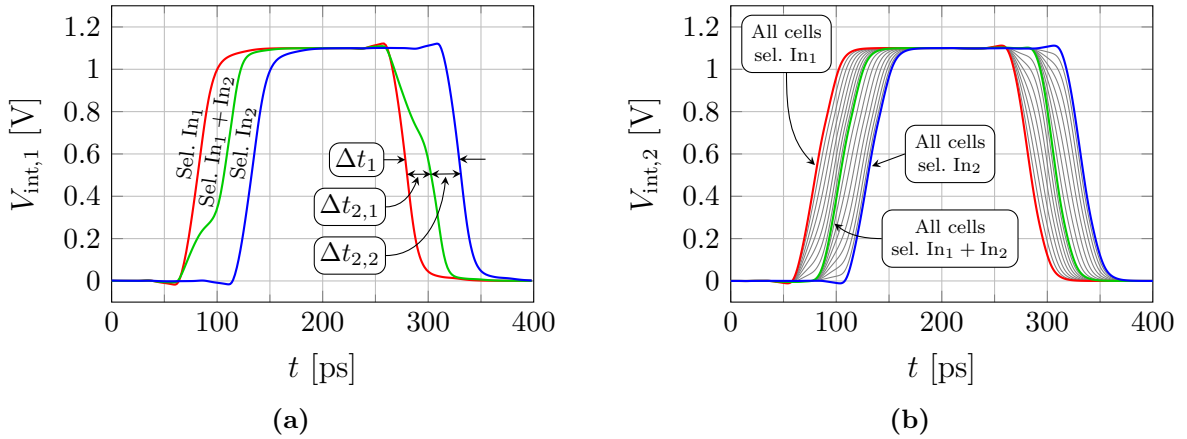


Figure 3.19 – Simulated waveforms at both DCEI² interpolation nodes: (a) local interpolation at $V_{\text{int},1}$ for different select signal configurations, and (b) passive second interpolation at $V_{\text{int},2}$ for different codes, colored plots highlight the special cases when all cells have an identical configuration.

the interpolated edge at $V_{\text{int},1}$ is placed temporally between the cases (1) and (2), so that Δt_1 is sliced in two equal intervals $\Delta t_{2,1} = \Delta t_{2,2} = \Delta t_1/2$. These intervals are then the interpolation range for the second interpolation. In Fig. 3.19(a), the intervals Δt_1 , $\Delta t_{2,1}$, and $\Delta t_{2,2}$ are annotated for the example of the falling edges.

Second Interpolation

For the second interpolation, the outputs of all DCEI² unit cells are connected to the common interpolation node $V_{\text{int},2}$. Depending on the configuration of each unit cell, their output stages start to drive $V_{\text{int},2}$ at different times, triggered by the threshold crossing of $V_{\text{int},1}$. The interpolation is plotted for different codes in Fig. 3.19(b). The colored plots highlight the cases where all cells have identical configuration. This interpolation cannot be actively controlled as in the case of the DCEI or CF-DCEI. While the other two PIs have identical input signals and the unit cells select either one of them, the "unit cell" for the second interpolation is only a simple inverter. The weighting is done by providing different input signals to these inverters, thus weighting them in the second interpolation. It is a passive interpolation and purely depends on the configuration of the first interpolation in each DCEI² unit cell.

The DCEI² control logic ensures that only two distinct region are allowed for the second interpolation: interpolation between unit cells configured for (a) In_1 or $\text{In}_1 + \text{In}_2$, or for (b) $\text{In}_1 + \text{In}_2$ or In_2 . Other configurations, such as part of the unit cells are configured with In_1 and the remaining ones for In_2 , effectively neglect the local interpolation and increase the nonlinearity due to higher Δt_2 . Codes from 0 to $N/2$ and codes from $N/2$ to N lead to interpolation in region (a) and (b), respectively. Table 3.1 shows in which state all unit cells are configured for different codes. The waveforms on the right hand side indicate the interpolation region. The effective range Δt_2 at the input of the second interpolation is then either $\Delta t_2 = \Delta t_{2,1}$, or $\Delta t_2 = \Delta t_{2,2}$.

Table 3.1 – Control of a DCEI² cell array with $K = 2^{(k_{PI}-1)}$ cells: number of DCEI² unit cells in each select state for different codes.

DCEI ² Code	Sel. In ₁	Sel. In ₁ + In ₂	Sel. In ₂
0	K	0	0
1	$K - 1$	1	0
2	$K - 2$	2	0
\vdots	\vdots	\vdots	\vdots
$N/2 - 2$	2	$K - 2$	0
$N/2 - 1$	1	$K - 1$	0
$N/2$	0	K	0
$N/2 + 1$	0	$K - 1$	1
$N/2 + 2$	0	$K - 2$	2
\vdots	\vdots	\vdots	\vdots
$N - 2$	0	2	$K - 2$
$N - 1$	0	1	$K - 1$
N	0	0	K

Table 3.2 – Key specifications of DCEI² test chip configurations.

	Frequency		Resolution [bit]			Resolution [time]		
	f_{ref}	f_{out}	k_{therm}	k_{bin}	k_{total}	Δt_1	Δt_2	$t_{d,\text{LSB}}$
DCEI ² V1	10 GHz	2.5 GHz	8	2	10	50.0 ps	25.0 ps	48.8 fs
DCEI ² V2	8.8 GHz	2.2 GHz	7	2	9	56.8 ps	28.4 ps	111.0 fs
	12 GHz	3 GHz	7	2	9	41.7 ps	20.8 ps	81.4 fs

Implementation

The DCEI² is implemented in two evolutionary test chips configured as stated in Table 3.2. The first version V1 implements an array with $16 \times 8 = 128$ unit cells, and the second version V2 with $8 \times 8 = 64$ unit cells. The smaller array aims at reduction of area and power consumption. Both designs implement a segmented architecture with a thermometrically controlled array and additional binary controlled cells for resolution enhancement. Their 2 bit binary extension is discussed separately in Section 3.4.

The array and decoder implementation is similar to the DCEI with one additional control line per column to account for the additional state of local interpolation. Further differences between both chip versions include improvements in DCEI² decoder design, MMD design, and the power supply concept. Chapter 4 discusses the DTC's supply voltage sensitivity and the new power supply concept in detail.

The following analysis on the DCEI²'s nonlinearity and current consumption is based on the second version with 9 bit resolution, where only the thermometrically controlled unit cell array is investigated and the two binary bits are left out of the discussion for now.

DCEI² Nonlinearity

The overall nonlinearity of the DCEI² is determined by both interpolation points. The requirement towards the first interpolation is the generation of the interpolated edge

exactly 50% between In_1 and In_2 , which leads to ideal intervals for Δt_2 of

$$\Delta t_{2,1} : 0 \longrightarrow \Delta t_1/2, \text{ and} \quad (3.13)$$

$$\Delta t_{2,2} : \Delta t_1/2 \longrightarrow \Delta t_1. \quad (3.14)$$

However, a 50% interpolation is hard to achieve over PVT. As discussed before, the phase interpolation depends on three major parameters: $t_{r,f}$, t_{int} , and Δt . The input spacing Δt_1 is given by design and cannot be influenced. There is also no built-in mechanism to control t_{int} , as the unit cells lacks further control. Actually, t_{int} is sensitive to all PVT factors, (a) voltage variations, (b) temperature variations, and (c) process variations. The designer cannot influence (b) and (c) during chip operation, and (a) can be ruled out as the supply voltage also supplies other blocks that rely on a specified voltage level. This leaves t_f as candidate for tuning of the first interpolation. The drivers for In_1 and In_2 can be designed in a way, that allows to tune their driving strength with a digital control signal. They are implemented as tristate inverters, as indicated in Fig. 3.18(a).

The second interpolation lacks the possibility of any configuration. However, it is intrinsically more robust against (a)-(c), as $\Delta t_2 = \Delta t_1/2$. The overall nonlinearity is similar to the one of the DCEI, which makes all findings from the DCEI analysis, summarized in (3.4), also applicable on this interpolation.

The simulated INL is plotted in Fig. 3.20. The red plot shows the ideally tuned DCEI², leading to two similar DCEI like INL shapes in the code regions 0 – 256 and 256 – 512. Tuning of t_f influences especially the INL at code 256, where all cells are configured for $In_1 + In_2$. The INL here relates then directly to the time intervals:

$$\text{INL}[n = 256] = \Delta t_{2,1} - \Delta t_1/2 = \Delta t_1/2 - \Delta t_{2,2}. \quad (3.15)$$

The optimum setting can be expressed as the optimization problem

$$\min_{t_f} |\text{INL}[n = 256]|. \quad (3.16)$$

A certain tuning range is needed to enable configuration of an ideal INL (such as the red plotted one) over PVT, which is discussed in detail in Section 3.3.2.

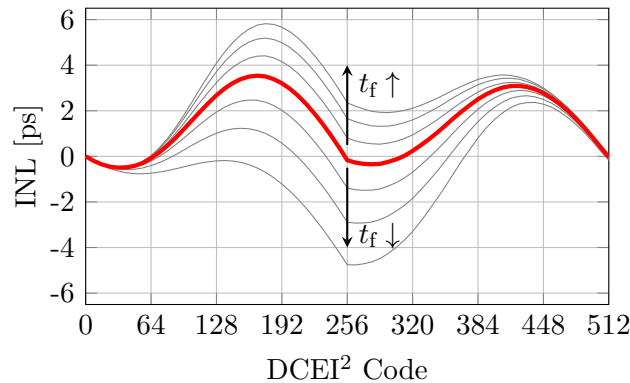


Figure 3.20 – DCEI² INL for variation of t_f through the tunable input buffers. The red plot highlights an ideally tuned DCEI², as it delivers the smallest peak-to-peak INL.

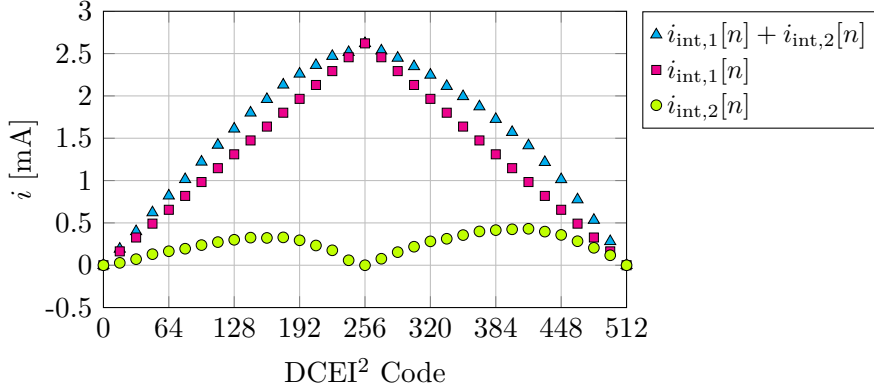


Figure 3.21 – Simulated static code-dependent DCEI² current consumption (i_{base} is not included).

DCEI² Current Consumption

While the DCEI² architecture provides superior power consumption compared to the DCEI and CF-DCEI, it has the major drawback of a code-dependent current consumption. This leads to a code-dependent modulation of the supply voltage and affects not only the DCEI², but all circuits that share the same supply. Unit cells configured for $I_{n_1/2}$ act as a multiplexer and have a different current consumption than cells configured for $I_{n_1} + I_{n_2}$, which have additional shoot-through current during the interpolation period. Furthermore, the second interpolation node shows a code-dependent current, which is, however, small compared to the current variation of the first interpolation. The described effects contribute to the static current consumption, which measures the current for constant DCEI² code n :

$$i_{\text{static,DCEI}^2}[n] = i_{\text{base}} + i_{\text{int},1}[n] + i_{\text{int},2}[n]. \quad (3.17)$$

It is separated into a code independent base current consumption i_{base} , which is invariant to n , and a code-dependent part. The current i_{base} contains contributors such as decoder clocking, (dis)charging of the net capacitances that toggle on RF rate, or leakage current (bias currents are not listed here, as the DCEI² is a fully digital circuit). Fig. 3.21 shows a simulation of the code-dependent part of the DCEI²'s static current consumption.

The static code-dependent current is plotted in blue for different codes. From these simulation results the code-dependent current parts of both interpolations, $i_{\text{int},1}[n]$ and $i_{\text{int},2}[n]$, can be calculated. The current of the unit cells is expected to depend only on the number of cells configured for local interpolation (as listed in Table 3.1). This results in a linear dependency over code with minimum at code 0 and 512 (unit cells act only as MUX), and maximum at code 256, where all DCEI² unit cells are configured for local interpolation. The current difference between MUX and interpolation state, which is the shoot-through current, can be calculated from the simulation results as

$$i_{\text{diff}} = \frac{i_{\text{int},1}[N/2]}{\#\text{cells}} = \frac{i_{\text{int},1}[256]}{256}. \quad (3.18)$$

The remaining difference between $i_{\text{static,DCEI}^2}[n]$ and $i_{\text{int},1}[n]$ is the code-dependent current of the second interpolation, which is plotted in green. It has no linear code dependency,

and its peak is > 6 times smaller than the peak variable current of the first interpolation. If the DCEI² is programmed with a random code sequence, the average shoot-through current due to the first interpolation is ~ 1.3 mA. Even with this amount of current spent into this mechanism, this PI topology is the most power saving one among the PIs investigated in this dissertation project (a direct comparison table is given in Section 3.5).

The impact of the code-dependent current on the DTC system and its accuracy is analyzed in Chapter 4. There the dynamic performance is discussed, which takes changing DTC codes into account.

3.3.2 DCEI² Model

The two subsequent interpolation are modeled independently with the equations developed for the DCEI model. The first interpolation provides signals with a time spacing Δt_2 to the second stage. As the first interpolation is not perfectly linear, Δt_2 varies in the two regions and has a value of $\Delta t_{2,1}$ or $\Delta t_{2,2}$ as described in the previous section. The overall linearity depends on the linearity of the first interpolation, plus the linearity of the second interpolation.

For the first interpolation, the three configurations listed in Table 3.3 determine the interpolation process. The threshold crossing time $t_{d,int,1}$ for $V_{int,1}$ is calculated with help of the already developed DCEI model. Evaluating the DCEI model with $N = 1$ for both codes, $n = 0$ and $n = N = 1$, results in $t_{d,int,1}$ for the first and third configuration. The second configuration needs a different model with $N = 2$. Here $t_{d,int,1}$ is only extracted for $n = 1$, which is effectively an interpolation between two DCEI cells. From model point of view this is the exact operation of the first interpolation, as two tristate inverter branches are driving a common node, triggered by $In_{1/2}$. All other model parameters are identical for both evaluations. Table 3.3 lists the configurations of the DCEI model to extract the DCEI² delays for all configurations of the first interpolation. The design goal is ideal linearity in the first interpolation, which can be expressed as

$$t_{d,int,1}[1] - t_{d,int,1}[0] = t_{d,int,1}[2] - t_{d,int,1}[1] = \Delta t_1/2. \quad (3.19)$$

As the local interpolation lacks any further configuration mechanisms, t_{int} is defined by design. However, PVT variations lead to an influence on t_{int} , especially the process variations. This needs to be considered during design and is evaluated in the model.

Assuming that Δt_1 , or a range of Δt_1 due to a range of f_{ref} , is given by the design specifications, the parameter influencing the interpolation is $t_{r/f}$. For this DTC topology Δt_1 directly relates to the period of the reference signal: $\Delta t_1 = T_{VCO}/2$. The model

Table 3.3 – Extraction of DCEI² delays for the first interpoalction with equivalent DCEI models.

#	DCEI ² Conf.	DCEI ² delay	Model param.	
			N	n
1)	Sel. In_1	$t_{d,int,1}[0]$	1	0
2)	Sel. $In_1 + In_2$	$t_{d,int,1}[1]$	2	1
3)	Sel. In_2	$t_{d,int,1}[2]$	1	1

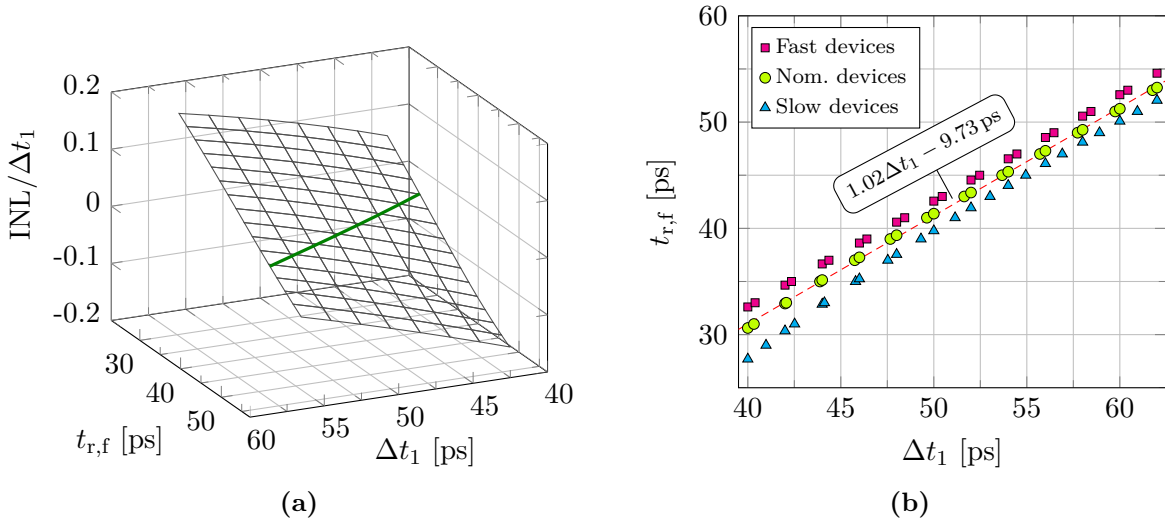


Figure 3.22 – Evaluation of the INL at node $V_{\text{int},1}$: (a) INL for variation of $t_{r/f}$ and Δt , the colored contour line highlighting the region of ideal linearity, and (b) evaluation of the contour line for different process corners, indicating $t_{r/f}$ in dependency of Δt to achieve ideal linearity.

is evaluated for reference frequencies in the range of 8.8–12 GHz (according to output frequencies in the range of 2.2–3 GHz), leading to Δt_1 in the range of 41.7–56.8 ps. The INL consists only of three points, one for each configuration, of which the first and third point have zero INL by definition:

$$\text{INL}_{\text{int},1}[0] = \text{INL}_{\text{int},1}[2] = 0 \quad (3.20)$$

$$\begin{aligned} \text{INL}_{\text{int},1}[1] &= (t_{d,\text{int},1}[1] - t_{d,\text{int},1}[0]) - \Delta t_1/2 \\ &= \Delta t_1/2 - (t_{d,\text{int},1}[2] - t_{d,\text{int},1}[1]) \end{aligned} \quad (3.21)$$

The resulting INL is equal to $\text{INL}_{\text{int},1}[1]$. It is plotted in Fig. 3.22(a) for variation of Δt_1 and $t_{r/f}$. The green contour line marks the region, in which zero INL, thus a perfectly tuned first interpolation, is achieved.

The design parameter of interest, $t_{r/f}$, can now be extracted from the relation of $t_{r/f}$ and Δt at zero INL (at the green contour line), which is plotted in Fig. 3.22(b). To be able to tune the INL to zero over the full $\Delta t_1 \propto 1/(2f_{\text{ref}})$ range, $t_{r/f}$ has to be adjusted in the range covered by the plotted marks, which is approximately 30–54 ps. The dependency between $t_{r/f}$ and Δt_1 is almost linear, with one example fit annotated in the graph. A small tuning range is desirable, as large tuning range requires more tristate input inverters, which can take a significant portion of the overall DCEI² area.

The second interpolation can be calculated with the same model as discussed for the DCEI in Section 3.1.1. The major difference are input signals with different $t_{r/f}$. If the first interpolation is configured for $\text{In}_{1/2}$, the signals that trigger the second interpolation have a different slope as if the first interpolation is configured for $\text{In}_1 + \text{In}_2$. However, the findings regarding design parameters from Section 3.1.1 are still valid for this case. Therefore, the nonlinearity model investigation for the DCEI² focuses only on the first interpolation.

3.4 Binary Bit Resolution Enhancement

If PIs as discussed so far are extended by 1 bit in resolution, it goes hand in hand with doubling the size of the unit cell array, and with this a significant increase in power consumption. The thermometric control of the array ensures monotonicity, which is the major benefit of this architecture. A better trade-off for resolution enhancement is the placement of additional unit cells, which are controlled in a binary fashion and generate a delay smaller than the delay of the thermometrically controlled unit cells. The PI is then segmented into a thermometrically and binary controlled part. Its corresponding interpolation cells are further referred to as thermometer cells (or T_{cell}) and binary cells (or B_x). With the thermometer cells as reference, the binary cells have for instance half ($B_{1/2}$), or quarter ($B_{1/4}$) of their "size", which is further discussed in Section 3.4.2. This technique is used in D/A converters for several decades now and brings advantages as high accuracy and monotonicity for the thermometrically controlled MSBs, while saving power and area for the binary controlled LSBs [87, pp. 640-642].

However, hybrid array implementations for DTC PIs have not been published so far and differ from implementations in a conventional DAC. The DTC example with a 1 bit PI from Table 3.4 shows how a continuous output phase change is achieved for a thermometrically controlled array, and how the coarse and fine tuning stage are synchronized. Two cells are needed for a 1 bit interpolation, as a single cell would implement only a MUX and no PI. The bold highlighted bit controls the PI and the other bits control the coarse tuning stage (e.g. the MMD) that changes $In_{1/2}$. $T_{\text{cell},1}$ is directly controlled by the coarse tuning's LSB, while $T_{\text{cell},2}$ is controlled by the PI's digital control line. For rising codes, a "forward" interpolation from In_1 to In_2 is followed by a "backward" interpolation from In_2 to In_1 , for which the order of the input signals is changed. This symmetrical "forward" and "backward" control is required to change only the programming of a single cell when the input signals are switching their temporal order, and is implemented in this fashion for all discussed PIs.

A generalization of this example leads to four major differences to conventional DACs: 1) PIs cover a range of $0 - FS$ over code, while conventional DACs cover $0 - (FS - 1\text{LSB})$; 2) to achieve a continuous phase change in a DTC, the number of states in a k bit PI is $2^k + 1$ (code $0 - 2^k$), while it is 2^k (code $0 - (2^k - 1)$) in a conventional k bit DAC; 3) the unit cells cannot be disconnected from the interpolation (by e.g. putting them in a high-Z state) but are always involved in the interpolation process by selecting either In_1 or In_2 ; and 4) the converter's gain is only defined by Δt , and not by the total amount of cells.

Table 3.4 – Logic states of a thermometrically controlled 1 bit PI including coarse tuning.

	Code	Selected Input		Weighting	
		$T_{\text{cell},1}$	$T_{\text{cell},2}$	In_1	In_2
	000	In_1	In_1	2	0
	001	In_1	In_2	1	1
	010	In_2	In_2	0	2
	011	In_2	In_1	1	1
	100	In_1	In_1	2	0

First, Section 3.4.1 discusses the conceptual segmentation architecture of the PI array and presents two possible control schemes. Afterwards the implementation of the binary cells is presented in Section 3.4.2 on example of the DCEI².

3.4.1 Architecture of a Binary Extended Cell Array

To highlight the issues of binary bit implementation in the PI array, the 1 bit array from Table 3.4 is extended by a single $B_{1/2}$ cell to $k_{PI} = 2$ in the same fashion as for a conventional DAC. Table 3.5(a) shows a list of all logical states in the newly formed array. The bold highlighted bits are the new 2 bit PI code. The problem of this extension is one additional logic state, that is introduced due to the DAC-like binary bit extension and that leads to a larger number of logic states than available digital codes. This results in DNL spikes, as there are three steps of 0.5, and one step of 1.0 for the code transition **011** \rightarrow **100**.

To overcome this problem, a thermometer cell needs to be removed from the array and has to be replaced by an equivalent amount of binary cells. The correct implementation where one T_{cell} is replaced by two $B_{1/2}$ cells is shown in Table 3.5(b). The main difference is the weighting sum, which stays now at 2.0 as in the original PI from Table 3.4. This results in one less state and correctly implements the binary cells for the given number of codes, as all DNL steps are of 0.5 now. One of the newly added $B_{1/2}$ cells is now controlled by the MSB, instead of one T_{cell} as in Table 3.4 and 3.5(a).

While the 1 bit binary extension leaves only one possible implementation, there are two different control schemes for higher order binary extensions: a) additional binary cells that are controlled in a binary fashion, or b) additional binary cells that are controlled in a thermometric fashion. For the DCEI² 2 bit binary extension, control scheme a) requires to replace one thermometer cell by four $B_{1/4}$ cells, while b) requires the replacement by one $B_{1/2}$ cell and two $B_{1/4}$ cells. In both cases the driving strength of all added unit cells adds up to the driving strength of the removed thermometer cell. Fig. 3.23 shows the implementation of the unit cells into the 7 bit DCEI² array with 8x8 thermometer cells, as done for DCEI² test chip V2. Cell 64 is removed and replaced by an equivalent number of binary cells according to control scheme b). In general, any cell Z (binary or thermometric)

Table 3.5 – Binary bit extension of a 1 bit PI: (a) conventional extension leading to a missing programming code, and (b) proposed extension.

(a)						(b)					
Code	Selected Input			Weighting		Code	Selected Input			Weighting	
	T_{cell}	T_{cell}	$B_{1/2}$	In_1	In_2		T_{cell}	$B_{1/2}$	$B_{1/2}$	In_1	In_2
000	In_1	In_1	In_1	2.5	0.0	000	In_1	In_1	In_1	2.0	0.0
001	In_1	In_1	In_2	2.0	0.5	001	In_1	In_2	In_1	1.5	0.5
010	In_1	In_2	In_1	1.5	1.0	010	In_2	In_1	In_1	1.0	1.0
011	In_1	In_2	In_2	1.0	1.5	011	In_2	In_2	In_1	0.5	1.5
?	In_2	In_2	In_1	0.5	2.0	100	In_2	In_2	In_2	0.0	2.0
100	In_2	In_2	In_2	0.0	2.5						

8	9	24	25	40	41	56	57	
7	10	23	26	39	42	55	58	
6	11	22	27	38	43	54	59	
5	12	21	28	37	44	53	60	
4	13	20	29	36	45	52	61	
3	14	19	30	35	46	51	62	$B_{1/4}$
2	15	18	31	34	47	50	63	$B_{1/4}$
1	16	17	32	33	48	49	64	$B_{1/2}$

Figure 3.23 – 7 bit DCEI² unit cell array with 2 bit binary extension according to control scheme (b).

can be replaced by two cells $Z_{1/2}$ with half the size. Recursive application of this method leads to both extension examples, a) and b).

As example for both control schemes, the codes around the transition from first to second interpolation region in the DCEI² ($In_1 \rightarrow In_1 + In_2$, and $In_1 + In_2 \rightarrow In_2$) are listed in Table 3.6. This highlights how both control schemes achieve continuous phase change. The DCEI² control determines with the bold highlighted n_8 , if the interpolation is in the first or second region (compare Table 3.1). The control bit of each cell selects $In_{1/2}$ for 0, and $In_1 + In_2$ for 1. Due to this fact, the 2 bit control word $b_{1:0}$ for the binary cells needs to be "aware" of the interpolation region and is therefore generated by the logic expression

$$b_{1:0} = n_{1:0} \oplus n_8. \quad (3.22)$$

Control scheme (a) uses a thermometric-to-binary conversion on $b_{1:0}$ and controls three of the $B_{1/4}$ cells directly with it, while in (b) the $B_{1/2}$ cell is directly connected to b_1 , and

Table 3.6 – Binary cell control for a 2 bit binary extension to a 7 bit DCEI² cell array.

DCEI ² Code	Control scheme (a)				Control scheme (b)			Cell sum		
	$B_{1/4}$	$B_{1/4}$	$B_{1/4}$	$B_{1/4}$	$B_{1/2}$	$B_{1/4}$	$B_{1/4}$	In_1	$In_1 + In_2$	In_2
0 111110 11	1	1	1	0	1	1	0	1.25	62.75	0.00
0 111111 00	0	0	0	0	0	0	0	1.00	63.00	0.00
0 111111 01	1	0	0	0	0	1	0	0.75	63.25	0.00
0 111111 10	1	1	0	0	1	0	0	0.50	63.50	0.00
0 111111 11	1	1	1	0	1	1	0	0.25	63.75	0.00
1 000000 00	1	1	1	1	1	1	1	0.00	64.00	0.00
1 000000 01	0	1	1	1	1	0	1	0.00	63.75	0.25
1 000000 10	0	0	1	1	0	1	1	0.00	63.50	0.50
1 000000 11	0	0	0	1	0	0	1	0.00	63.25	0.75
<u>1000001</u> <u>00</u>	1	1	1	1	1	1	1	0.00	63.00	1.00
	therm.		bin.							

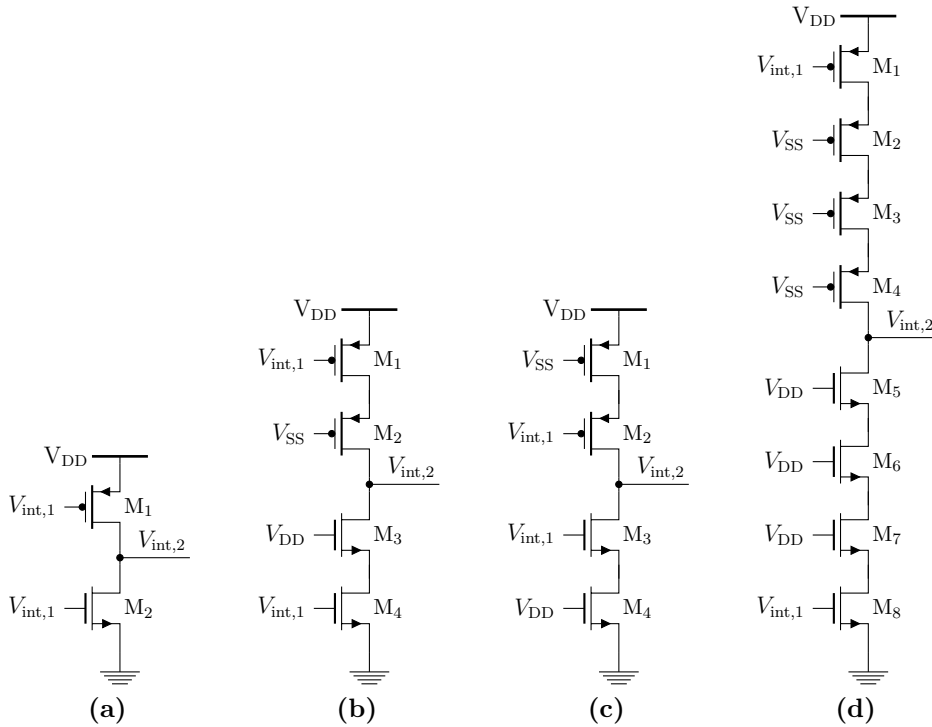


Figure 3.24 – Possible implementation of the output stages of (a) the thermometric DCEI² unit cell, (b) a first version of the $B_{1/2}$ or $B_{1/4}$ cell, (c) a second version of the $B_{1/2}$ or $B_{1/4}$ cell, and (d) the $B_{1/4}$ cell.

the $B_{1/4}$ cell to b_0 . In both cases, the remaining $B_{1/4}$ cell is controlled directly by n_8 , as it was implemented in the initially discussed Tables 3.4 and 3.5(b).

To understand the advantages of one control scheme over the other, first the implementation of the binary cell has to be discussed.

3.4.2 Binary Unit Cell Implementation

The interpolated signal at V_{int} is generated by weighting the NMOS and PMOS current sources shown in Fig. 3.5. A higher resolution is achieved by a higher number of control states. For the code range $0 \leq n \leq N$, this means either increasing N (which has the drawbacks described above), or making n fractional. By looking at the current source model in Fig. 3.5(a), the latter one can be achieved by replacing one thermometric controlled current source by an equivalent of binary controlled sources. For the example of two binary bits, cell designs with half and quarter of the nominal driving strength are needed. As the resolution enhancement is targeted for the second DCEI² interpolation, the driving strength of the unit cell's output inverter has to be modified for the binary cells.

Fig. 3.24 shows different implementations of the thermometer cell and the binary cell. The thermometer cell's output inverter is depicted in Fig. 3.24(a). The straight forward way to reduce the current of a digitally controlled transistor by a factor of two is a modification of the transistor's length or width. As the DCEI² output inverter is already close to the minimum feature size of the 28 nm technology, it cannot be reduced in width. An increase

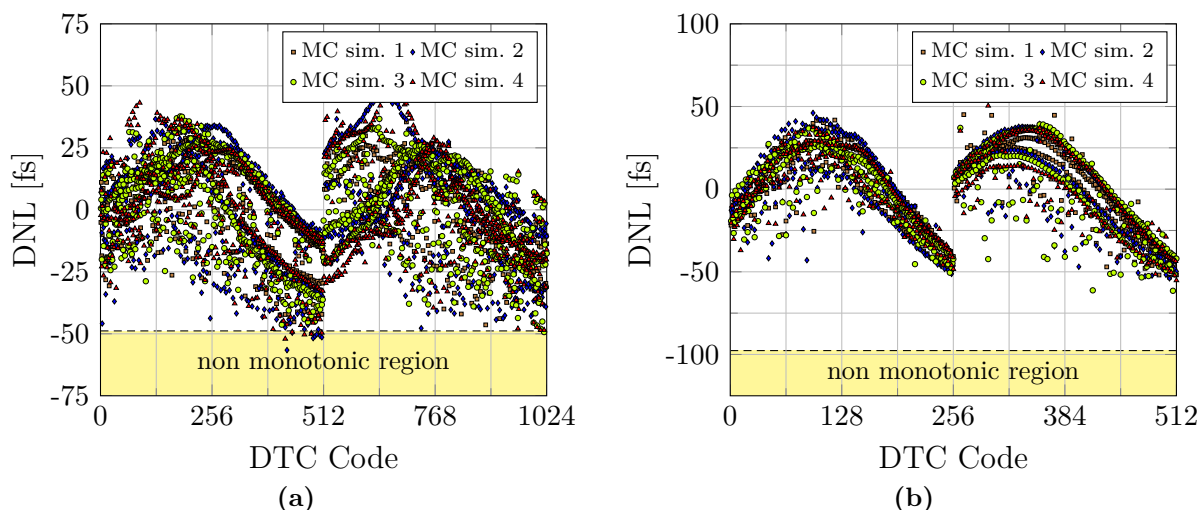


Figure 3.25 – Monte Carlo simulations of the DNL for binary bit implementation of (a) DCEI² test chip V1, and (b) DCEI² test chip V2.

in length has two impacts on the cell: first, the capacitance of $V_{\text{int},1}$ would change due to the different load, resulting in a possibly different temporal position of the edges at $V_{\text{int},1}$; second, different wells need to be placed for devices with different length, leading to additional constraints on physical cell layout that increase the cell dimensions.

The $B_{1/2}$ cells is therefore implemented as a stack of transistors as shown in Fig. 3.24(b) and (c). The new transistors are always turned on and their sole purpose is to double the resistance of the branches. This ensures that the local interpolation is as identical as possible in thermometer and binary cells. Of these two cases, Fig. 3.24(b) shows the favorable implementation, as the physical layout of net V_{int} can be kept identical to the thermometer cell. If the same scheme is applied to the $B_{1/4}$ cell, it would lead to an even larger stack as shown in Fig. 3.24(d). Parasitic layout effects as well as effects of the device itself lead to a different PVT and mismatch behavior compared to the stack of two transistors.

While the general INL shape and peak is determined by the thermometer cell array, the monotonicity of the design depends on the binary cell implementation. A single $B_{1/2}$ cell yields easily into a binary extended and monotonic DCEI², but the implementation of $B_{1/2}$ and $B_{1/4}$ together can cause non-monotonic behavior. For a single $B_{1/2}$ cell, the only requirement for monotonic behavior is a smaller current than the thermometer cell. If $B_{1/2}$ and $B_{1/4}$ are implemented, their combined driving strength should not exceed the thermometer cell's driving strength over PVT, mismatch, and DTC code. Therefore, the monotonicity of a 2 bit binary extension depends on the implementation of the $B_{1/4}$ cell and its relation to the $B_{1/2}$. Two different $B_{1/4}$ implementations were tested in the two DCEI² test chips:

- 1) DCEI² test chip V1: $B_{1/2}$ implemented with two stacked devices (Fig. 3.24(b)), and $B_{1/4}$ with four stacked devices (Fig. 3.24(d))
- 2) DCEI² test chip V2: $B_{1/2}$ implemented with two stacked devices (Fig. 3.24(b)), and $B_{1/4}$ with two stacked devices with increased transistor length (Fig. 3.24(b))

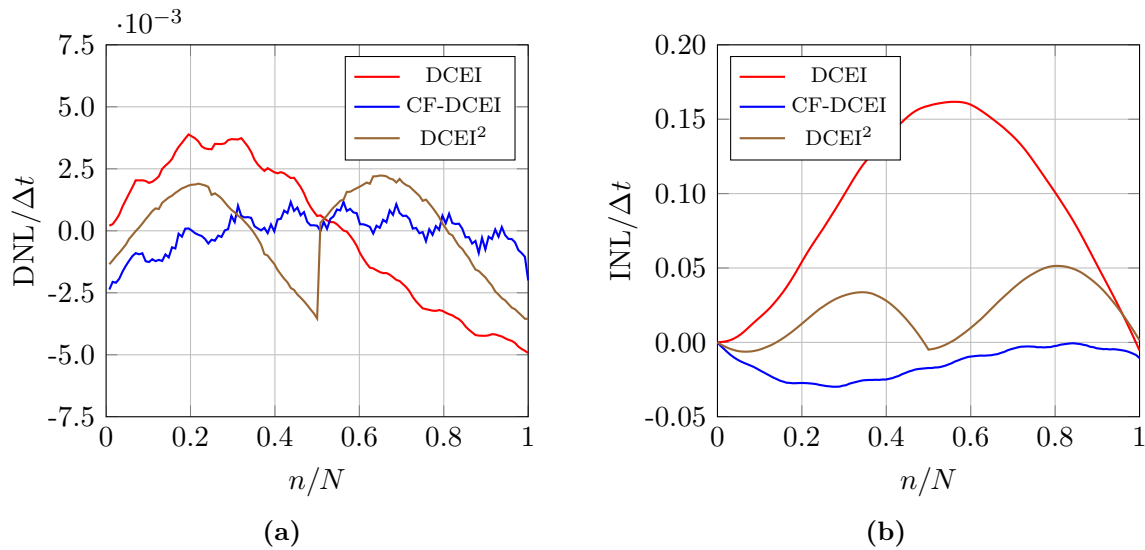


Figure 3.26 – Comparison of DCEI ($f_{\text{out}} = 2 \text{ GHz}$), CF-DCEI ($f_{\text{out}} = 2 \text{ GHz}$), and DCEI² ($f_{\text{out}} = 2.5 \text{ GHz}$) in terms of (a) DNL, and (b) INL, both normalized for N and Δt .

While 1) keeps identical layout of V_{int} for $B_{1/2}$ and $B_{1/4}$, 2) trades off slight differences at V_{int} due to different loading against a stronger similarity between $B_{1/2}$ and $B_{1/4}$ in terms of the transistor stack. Monte Carlo (MC) circuit simulations with extracted parasitics prove which concept gives advantages in terms of monotonicity. For the MC DNL plots from Fig. 3.25(a) the $B_{1/2}$ and $B_{1/4}$ cell are implemented with a stack of two and four transistors, respectively. While most of the DCEI² codes shows a monotonic behavior, some of the DNL points are only marginally monotonic or already in the non-monotonic region. If both binary cell are implemented with a two transistor stack, the DNL as plotted in Fig. 3.25(b) is clearly monotonic and shows sufficient margin to the non-monotonic region.

Simulations show a clear advantage of the binary cell implementation of test chip V2 over test chip V1. While both implement a 2 bit binary extension, test chip V1 has 1 bit higher resolution and is, therefore, not directly comparable to test chip V2. The T_{cell} and $B_{1/2}$ implementations are almost identical in terms of absolute device sizing in both test chip versions. A further comparison between both implementations based on the test chip measurements is presented in Chapter 5.

3.5 Summary and Conclusion

This chapter introduced the DCEI reference PI design, and presented two novel PI architectures developed in the present thesis: the CF-DCEI and DCEI². The design target for the CF-DCEI was an increased linearity for identical operation conditions as the DCEI, while the DCEI² focused on low power consumption, high resolution and high operation frequency. The key performance values of all three PIs are compared in Table 3.7.

The CF-DCEI design reduced the peak nonlinearity by 82% at the expense of 29% higher DTC power consumption due to the additional control logic inside the unit cells.

The area increased by 65 %, which is, however, still 85 % smaller compared to the smallest DTC in the gigahertz range published so far [2]. The DCEI² was designed for low power consumption. Compared to the DCEI, the first and the second version reduced the power consumption by 17 % and 25 %, respectively. The major benefit is the two-stage instead of a three-stage DTC design, as the MUX+DEL stage can be left out due to a larger interpolation range. Furthermore, the segmentation of the DCEI² array into thermometrically and binary controlled parts allows the reduction of the array size, while increasing the resolution.

Fig. 3.26 compares the simulated DNL and INL of the DCEI, CF-DCEI, and DCEI² (V2). To enable direct comparison, INL and DNL are normalized to Δt , and the code range is normalized to N . For the DCEI², only the thermometrically controlled cells are evaluated (every 4th code), as they determine the shape and the peak of the overall INL. The INL plots in Fig. 3.26(b) proof the significant advantages of the CF-DCEI over the DCEI in terms of nonlinearity: the peak INL is reduced by $> 80\%$. The DCEI² allows an extension of the interpolation range Δt , while reducing the absolute nonlinearity. This reflects in the normalized plot as significant improvement compared to the DCEI. One key design parameter that enables this is the reduction of the total interpolation range from 50 ps to two times 25 ps, compared to the DCEI and CF-DCEI with 31.25 ps.

For all designs analytical and numerical models were developed. Simulation and measurement results were compared to the models and key design parameters were identified, which influence the linearity of the interpolation. All models are based on the physical Shichman-Hodges transistor model and are evaluated with help of a fitted NMOS and PMOS $V_{DS} \rightarrow I_{DS}$ transfer function. The technology parameters are related to the intuitive key parameters $t_{r/f}$ and t_{int} , which describe rise and fall times at certain nodes inside the PIs. These two time delays are extracted easily from transient circuit simulations. The models allow a fast evaluation of all important design parameters and give a good estimation on the peak nonlinearity and its shape.

Simulation and modeling of a simple switched capacitor fine tuning stage allowed a direct comparison to the CF-DCEI. It out-performs the high linearity switched capacitor fine tuning in the ranges of Δt investigated in this work. The INL can theoretically be reduced to zero, it has a defined tuning range, and a constant current consumption prevents re-modulation of the supply voltage. Circuit simulations with non-zero INL show comparable peak INL for both designs. For large scale fine tuning ranges of multiple VCO periods, such as used in PLLs, the CF-DCEI is no option as it cannot compete in terms of linearity due to extensive INL for large $\Delta t/t_{int}$ ratio.

To enhance the PIs resolution with only a minimum power penalty, an extension of the PI cell array with 2 binary bits was explored. Based on the DCEI² architecture, different transistor level implementations were designed and implemented in test chips. This allows to reduce the size of the DCEI² array compared to an implementation that is controlled only thermometrically, thus saving power and area.

Besides the presented area and power consumption numbers, this chapter investigated and compared only the PI's static nonlinearity. To draw a complete picture, the full DTC architectures have to be compared, which is done in Chapter 5 based on test chip measurements. In addition to the static nonlinearity, also dynamic effects influence the DTC performance. They are discussed in the upcoming chapter.

Table 3.7 – Comparison of DCEI, CF-DCEI, and DCEI² by simulation data.

		DCEI	CF-DCEI	DCEI ² V1	DCEI ² V2
Frequency		2 GHz	2 GHz	2.5 GHz	2.2–3 GHz
Resolution	$t_{d,LSB}$	244 fs	244 fs	48.8 fs	81.4–111.0 fs
	[bit]	11 bit	11 bit	13 bit	12 bit
DNL _{rms}		89.82 fs	23.09 fs	48.25 fs ¹⁾ 19.81 fs ³⁾	86.50 fs ²⁾ 24.13 fs ⁴⁾
INL _{max}		5.05 ps	0.93 ps	2.89 ps	2.57 ps
Power	MMD	5.5 mW	5.5 mW	5.0 mW	5.0 mW ⁶⁾
	MUX+DEL	2.2 mW	2.2 mW	-	-
	PI	7.7 mW	12.1 mW	11.1 mW	9.7 mW ⁶⁾
	Total	15.4 mW	19.8 mW	16.1 mW	14.6 mW ⁶⁾
	Total/ f_{out}	7.7 μ W MHz ⁻¹	9.9 μ W MHz ⁻¹	6.4 μ W MHz ⁻¹	5.8 μ W MHz ⁻¹
Phase noise ⁵⁾		-161.8 dBc/Hz	-161.6 dBc/Hz	-161.4 dBc/Hz	-160.6 dBc/Hz
Area		0.0055 mm ²	0.0091 mm ²	0.0052 mm ²	0.0046 mm ²

¹⁾ only therm. controlled array, $t_{d,LSB} = 195.3$ fs

²⁾ only therm. controlled array, $t_{d,LSB} = 390.6$ fs, $f_{out} = 2.5$ GHz

³⁾ incl. binary bits, $t_{d,LSB} = 48.8$ fs

⁴⁾ incl. binary bits, $t_{d,LSB} = 97.7$ fs, $f_{out} = 2.5$ GHz

⁵⁾ measured at $f_{offset} = 100$ MHz

⁶⁾ measured at $f_{out} = 2.5$ GHz

4 Dynamic Effects in DTCs

While the static nonlinearity is an important measure for DTC performance, it is not the only effect that determines the overall nonlinearity. For the static nonlinearity characterization, the DTC's output phase ϕ_{static} is measured for each code in its periodic steady state. For code activity on the digital data input, the DTC is reconfigured to change the phase of the output signal. This activity causes dynamic effects, which can be measured as dynamic errors ϕ_{dyn} on the phase of the output signal. The output signal's phase is then a superposition of the targeted static phase ϕ_{static} and the dynamic error ϕ_{dyn} :

$$\phi = \phi_{\text{static}} + \phi_{\text{dyn}}. \quad (4.1)$$

The dynamic errors on the DTC's output phase and their related dynamic nonlinearity are defined in Section 4.1. Afterwards Section 4.2 reviews and analyzes the mechanisms behind the dynamic effects and discusses how to mitigate them during the circuit design phase. Dynamic errors are simulated exemplarily for the DCEI² V2 based DTC for different test cases, and the resulting dynamic nonlinearity performance figures are discussed in Section 4.3. To reduce the dynamic errors, two novel compensation circuits are presented in Section 4.4. Their impact on the dynamic nonlinearity is evaluated and compared to the simulations from Section 4.3. Finally, Section 4.5 summarizes this chapter with an overview on dynamic effect sources and design measures to mitigate them.

4.1 Definition of Dynamic Errors and Dynamic INL

To analyze the nature of dynamic errors, the simplified case of a single DTC code jump serves as example. The DTC is initially in its periodic steady state for code k , and then a code transition $k \rightarrow k + j$ is triggered. In the ideal case, the output delay change is only determined by the static DTC output and given by

$$\Delta\phi_{\text{static}}[k, k + j] = \phi_{\text{static}}[k + j] - \phi_{\text{static}}[k]. \quad (4.2)$$

Figure 4.1(a) illustrates that the target phase $\phi_{\text{static}}[k + j]$ is reached for the first edge after the code transition, which reflects the ideal DTC response. However, dynamic effects introduce dynamic errors ϕ_{dyn} on the DTC phase. Figure 4.1(b) shows, that the first output cycles after the code change can deviate from $\phi_{\text{static}}[k + j]$ and that ϕ then converges towards its static value, where it is in the periodic steady state for code $k + j$. The dynamic error depends on the cycle count n_{cycle} after the code change was applied, which is in a range of $1 \leq n_{\text{cycle}} \leq M$. This means dynamic errors can last for M cycles after the DTC code transition until they decay. The total delay change for the code transition is a superposition of static and dynamic DTC delay in dependency of n_{cycle} :

$$\Delta\phi[k, k + j, n_{\text{cycle}}] = \Delta\phi_{\text{static}}[k, k + j] + \phi_{\text{dyn}}[k, k + j, n_{\text{cycle}}]. \quad (4.3)$$

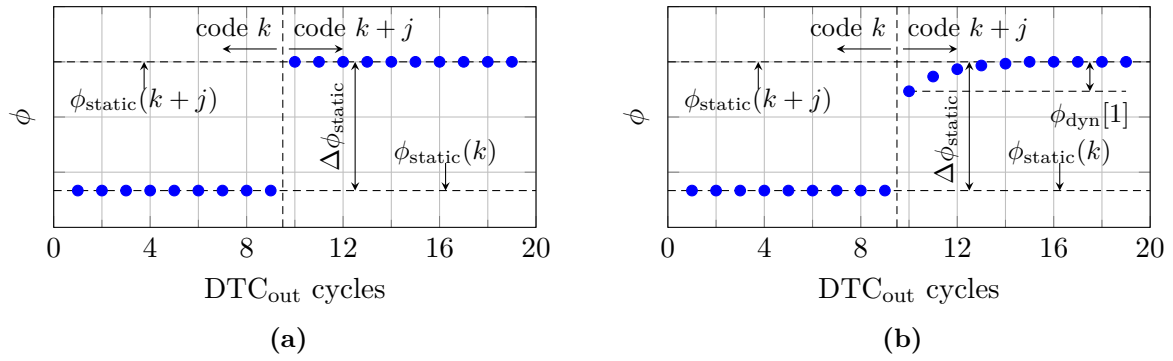


Figure 4.1 – DTC output phase ϕ for (a) only static nonlinearity, and (b) static and dynamic nonlinearity.

In the example from Fig. 4.1(b) the dynamic error decays after $M = 5$ DTC output cycles. Dynamic errors always converge to zero, as else the system would not have a periodic steady state and thus be unstable:

$$\lim_{n_{\text{cycle}} \rightarrow \infty} \phi_{\text{dyn}}[k, k + j, n_{\text{cycle}}] = 0. \quad (4.4)$$

Dynamic error can be further distinguished by their decay time. Due to the nature of their generation, which is discussed in the upcoming section, some effects are only visible in the cycle directly after the code transition, while others take several cycles to decay.

The equations above only analyzed a single code transition. Realistic DTC programming can include code changes on each clock cycle, leaving $\phi_{\text{dyn}}[k, k + j]$ no time to settle. Dynamic errors depend then not only on the single code transition, but on the whole code history, which is a code-dependent memory effect on ϕ . The INL can now be re-defined as a superposition of static and dynamic INL to

$$\text{INL}[n, n_{-1}, \dots, n_{-M}] = \text{INL}_{\text{static}}[n] + \text{INL}_{\text{dyn}}[n, n_{-1}, \dots, n_{-M}], \quad (4.5)$$

which is a function not only depending on the currently applied code, but also on the code history. At each code n the INL depends now on the previously programmed code sequence, which can be of a random nature, such as modulation, or of a more deterministic nature, such as frequency shift generated by periodic code ramps (compare the DDPS application from Section 1.2.1).

4.2 Root Causes of Dynamic Errors

Several mechanisms dynamically affect the generated DTC output signal. This section neglects the possibility of externally induced dynamic effects, for example coupling of neighbouring circuits of the same SoC into the DTC, and focuses only on dynamic effects generated within the DTC itself. The dynamic effects that contribute to dynamic errors are classified in two major groups:

1. Supply-induced effects with a decay time of multiple output cycles ($M > 1$)

- a) Code-dependent current consumption
 - b) Instantaneous change of average current
 - c) Logic current consumption
2. Logic-induced effects with a decay time of a single output cycle ($M = 1$)
- a) Digital control signal timing
 - b) Digital control signal coupling

Supply-Induced Effects

The first group covers all effects which introduce dynamic errors via the DTC's supply voltage V_{sup} . One general problem of DTCs is the poor power supply rejection. As DTCs target to generate a certain absolute delay, they can be easily disturbed by supply voltage distortions, as the propagation delay of most CMOS circuits is directly influenced by it. The overall output delay of an inverter for instance, which is in its simple or tristate form a basic building block of all DTC coarse and fine tuning blocks introduced in Section 1.1, is directly influenced by the supply voltage. For the example of an inverter that charges its output net with the capacitance C_{out} to the threshold voltage of a subsequent inverter $V_{\text{th,inv}}$, the delay is given as

$$\begin{aligned}
 t_d &= \frac{V_{\text{th,inv}} C_{\text{out}}}{I_{\text{D,sat}}} \\
 &= \frac{V_{\text{th,inv}} C_{\text{out}}}{0.5 K_p \frac{W_{\text{eff}}}{L_{\text{eff}}} (V_{\text{DD}} - V_{\text{th,n}})},
 \end{aligned} \tag{4.6}$$

where the drain current $I_{\text{D,sat}}$ of the charging PMOS device is described by the Shichman-Hodges transistor model [96]. Equation (4.6) is in accordance to equations used for CMOS inverter delay estimations [97, pp. 199-202]. Deriving (4.6) for V_{DD} yields in

$$\frac{dt_d}{dV_{\text{DD}}} = \frac{-V_{\text{th,inv}} C_{\text{out}}}{0.5 K_p \frac{W_{\text{eff}}}{L_{\text{eff}}} (V_{\text{DD}} - V_{\text{th,p}})^2}, \tag{4.7}$$

which shows that there are no easy options to reduce the supply sensitivity. The supply sensitivity dt_d/dV_{DD} can be determined from static simulations of the DTC by measuring t_d for different supply voltage levels. An example for the DCEI² V2 based DTC is plotted in Fig. 4.2 for the code range of the DCEI². The sensitivity per 1 mV supply change is already in the dimensions of $t_{\text{d,LSB}}$, and, furthermore, the sensitivity is not constant for all DCEI² codes.

Thus, ϕ has a strong sensitivity towards supply voltage variations ΔV_{sup} . As V_{sup} is usually generated by an internal supply regulator, such as a LDO [87, pp. 324-325], all deviations from its average load current Δi_{load} disturb V_{sup} by ΔV_{sup} , which directly translates to propagation delay changes in the DTC's signal chain and therefore dynamic errors:

$$\Delta i_{\text{load}} \rightarrow \Delta V_{\text{sup}} \rightarrow \phi_{\text{dyn}}. \tag{4.8}$$

The supply regulator then recovers V_{sup} to its steady-state value, which can take several output cycles (depending on the regulator's bandwidth) and is the major contributor to

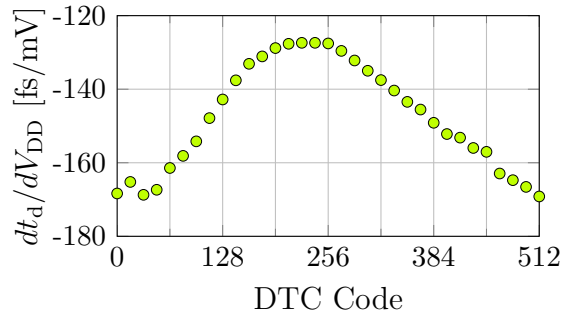


Figure 4.2 – Voltage sensitivity of DTC delay.

the decay time. All DTC operations that introduce a Δi_{load} to the load current lead to dynamic errors.

Logic Induced Effects

The second group covers effects that are caused by the interaction between digital signals and the DTC’s analog signal path. These are mainly direct coupling between digital clock or control signals into the analog signals, and effects caused by improper timing of digital signals directly interacting with the analog signal path. While the supply correlated dynamic effects usually take several cycles to settle, dynamic errors due to digital control signals are only visible in the cycle directly after the code change.

Each effect listed above is described in the following on the example of a single code transition. As one type of dynamic error lasts several output cycles while the other one lasts only a single output cycle, a superposition of different dynamic effects and memory effects influence the overall dynamic error.

4.2.1 Code-Dependent Current Consumption

Some of the DTC topologies presented in Section 1.1 have an inherent code-dependent current consumption. While the presented coarse tuning concepts show equal or almost equal current consumption for different codes in their periodic steady state, the two most commonly used fine tuning blocks have a deterministic code/current dependency: (a) switched capacitor based delay cells, and (b) PIs. The switched capacitor cells modify their capacitance to alter the overall delay, and consequently the current to charge/discharge the internal node. PIs have a shoot-through current during the phase interpolation process, which depends on the code. This leads to a more complex current/code dependency as discussed on the example of the DCEI² in Section 3.3.1. Only the CF-DCEI topology shows a constant current/code behavior, as the shoot-through current is prevented by additional control logic.

Many published DTC designs face this issue and compensate for it: full dummy delay cell arrays which are controlled by the inverse code to equalize the current consumption for all codes are placed [22–24, 26, 29]. This proves the severity of this problem, as the current consumption is willingly doubled to mitigate the resulting dynamic errors.

The impact of this dynamic effect on the DTC supply voltage is analyzed in Section 4.4 in detail. Simulations on the example of the DCEI² V2 based DTC including an implemented

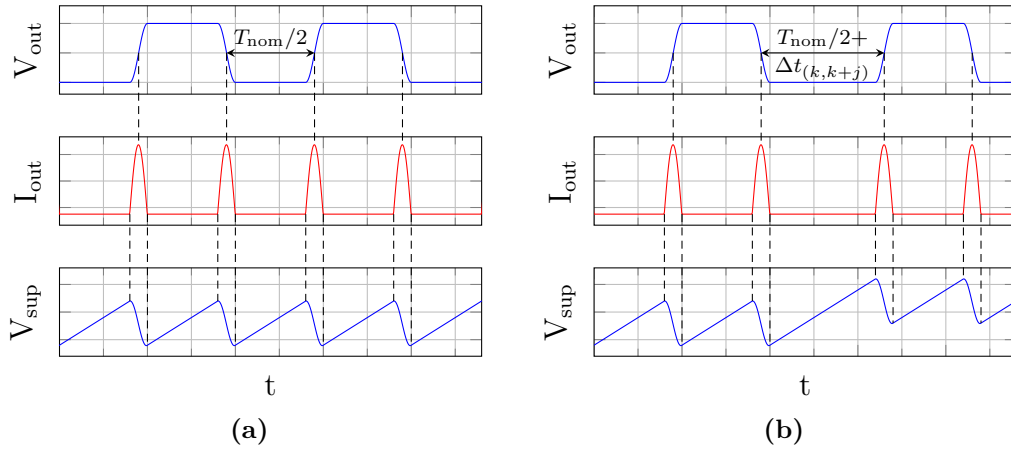


Figure 4.3 – Supply glitch caused by instantaneous change of average current: DTC output voltage V_{out} , current I_{out} , and supply voltage V_{sup} for (a) constant DTC output period, and (b) DTC output period stretch due to code transition $k \rightarrow k + j$.

LDO show the influence on the supply caused by different DCEI² code transitions.

4.2.2 Instantaneous Change of Average Current

When the DTC transitions between different codes, the output signal's period changes for the cycle in which the code transition is applied. This is according to (1.2) equal to an instantaneous frequency change. Corresponding to

$$P_{nom} = 0.5CV^2 f_{nom} \Leftrightarrow i_{nom} = 0.5CV f_{nom}, \quad (4.9)$$

the nominal frequency of a signal directly relates to its current consumption, which can be rewritten as a dependency from the nominal period as

$$i_{nom} = 0.5CV \frac{1}{T_{nom}}. \quad (4.10)$$

Figure 4.3(a) shows the periodic steady state of a DTC with constant code. Neglecting leakage, current is only drawn at signal transitions. In Fig. 4.3(b) a code change $k \rightarrow k + j$ exemplarily stretches the period for a single cycle to $T_{(k,k+j)} = T_{nom} + \Delta t_{(k,k+j)}$, leading to a different average current consumption for this cycle of

$$i_{(k,k+j)} = 0.5CV \frac{1}{T_{(k,k+j)}} \stackrel{(4.10)}{=} i_{nom} \frac{T_{nom}}{T_{(k,k+j)}}. \quad (4.11)$$

Assuming a linear DTC which covers 2π of the generated output signal over code n , the period after a code transition relates directly to the code change:

$$T_{(k,k+j)} = (N + j)t_{d,LSB}. \quad (4.12)$$

The current difference $\Delta i_{(k,k+j)}$ from the nominal current i_{nom} can be given from (4.10), (4.11), and (4.12) for the cycle of the DTC code change as

$$\begin{aligned}
 \Delta i_{(k,k+j)} &= i_{(k,k+j)} - i_{\text{nom}} \\
 &= i_{\text{nom}} \left(\frac{T_{\text{nom}}}{T_{(k,k+j)}} - 1 \right) \\
 &= i_{\text{nom}} \left(\frac{N t_{\text{d,LSB}}}{(N+j)t_{\text{d,LSB}}} - 1 \right) \\
 &= \frac{-j}{N+j} i_{\text{nom}}.
 \end{aligned} \tag{4.13}$$

This shows that $\Delta i_{(k,k+j)}$ does not depend on k , but only on the magnitude of the code step j . However, (4.13) assumes that the signal period changes equally throughout the whole DTC, which is not true. To be precise, the DTC needs to be split in coarse and fine tuning part. If a coarse tuning transition is triggered, the period changes for the involved coarse tuning block differently than for the DTC output. The total current sum of the DTC can be given as

$$i_{\text{nom}} = i_{\text{nom},0} + i_{\text{nom,coarse}} + i_{\text{nom,fine}}, \tag{4.14}$$

where $i_{\text{nom},0}$ is the current not influenced by code changes, $i_{\text{nom,coarse}}$ the current of all blocks processing the coarse tuned signals, and $i_{\text{nom,fine}}$ the current of all blocks processing the fine tuned signal. To extract the coarse tuning activity, the coarse tuning related code change j_{coarse} has to be extracted from the total code change by removing the k_{fine} bit fine tuning related code activity:

$$j_{\text{coarse}} = \left((k+j) - \left((k+j) \bmod 2^{k_{\text{fine}}} \right) \right) - \left(k - \left(k \bmod 2^{k_{\text{fine}}} \right) \right), \text{ with} \tag{4.15}$$

$$j = j_{\text{coarse}} + j_{\text{fine}}, \text{ and} \tag{4.16}$$

$$k = k_{\text{coarse}} + k_{\text{fine}}. \tag{4.17}$$

Then $\Delta i_{(k,k+j)}$ can be split into coarse and fine tuning related current changes:

$$\Delta i_{(k,k+j)} = \frac{-j_{\text{coarse}}}{N+j_{\text{coarse}}} i_{\text{avg,coarse}} + \frac{-j}{N+j} i_{\text{avg,fine}}. \tag{4.18}$$

Now k is also involved in the solution, as the coarse tuning block only changes its current if the code transition $k \rightarrow k+j$ triggers a coarse tuning code transition. Figure 4.4 visualizes the discontinuous nature of this function for code transitions that leave the output phase ϕ of the DTC in the range of $0^\circ \leq \phi < 90^\circ$. If more stages are involved the equation needs to be extended in an analogous manner.

If the DTC is supplied by a LDO supply regulator, the LDO drives a constant current into the supply net, which is equal to the average current consumption. The small change in the average current leads to a glitch on the supply voltage as shown in Fig. 4.3(b), as the bandwidth of the LDO control loop is not wide enough to react to such fast transitions. Depending on signal frequency and LDO bandwidth, the supply takes multiple cycles to recover its output voltage. Simulation results of the LDO output voltage for the example of the DCEI² V2 based DTC are presented in Section 4.4.

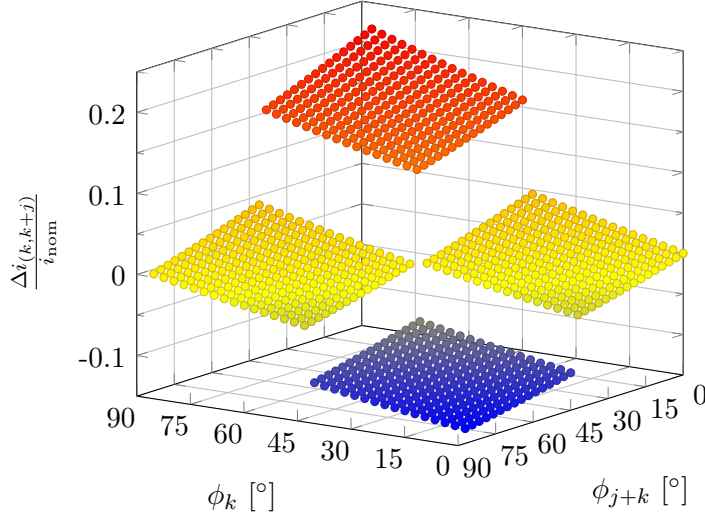


Figure 4.4 – Instantaneous current change for code transition $k \rightarrow k + j$ for the ranges $0^\circ \leq \phi_k < 90^\circ$ and $0^\circ \leq \phi_{k+j} < 90^\circ$. The DTC resolution is $k = 7$ bit with $k_{\text{coarse}} = 3$ bit and $k_{\text{fine}} = 4$ bit, and $i_{\text{nom},0} = i_{\text{nom,coarse}} = i_{\text{nom,fine}}$.

4.2.3 Logic Current Consumption

While the analog current consumption of the DTC is deterministic for static codes as well as code transitions, the current consumption of the digital logic is not. For the analog signal path, the code transition $k \rightarrow k + j$ determines the load current. It is given by the static current consumption of code k and $k + j$, and by the instantaneous duty cycle change introduced by the code transition. The digital current consumption i_{dig} can be split in two components:

$$i_{\text{dig}} = i_{\text{clk}} + i_{\text{logic}}. \quad (4.19)$$

The clock related current i_{clk} is as deterministic as the analog DTC current, as the clock is derived from the coarse and fine tuning outputs and has a fixed load. The digital logic, mostly consisting of decoders for coarse and fine tuning, consumes a current i_{logic} that depends on the code activity and on the clock rate, both additionally depending on code transition (the code sequence modulates the frequency). Furthermore, the magnitude of a code transition is not directly related to the decoder activity and strongly depends on the decoder implementation (e.g. in how many lines and columns the array is arranged).

To keep the related dynamic effects low, the digital current should be supplied by a different supply regulator than the analog current. It may not completely decouple both supply domains, as coupling via substrate is still possible, but this measures prevents supply voltage disturbances caused by digital activity.

4.2.4 Digital Control Signal Timing

Fine tuning architectures can be build as array of unit cells or as binary weighted cells. The binary weighted architecture has the advantage of lower area and power, while an array, controlled in thermometric fashion, is inherently monotonic and more robust against mismatch. Array structures are usually controlled by line and column select signals, which trigger the internal select signals of each unit cell. Process variation, mismatch, and

inherent delay lead to different switching times of the unit cell internal select signals. For example a select signal transition triggered by a column signal can have a different delay than the same transition triggered by a row signal. Also, binary unit cell extensions to a thermometric array can have column and line signals with different load, thus different delay time (compare Fig. 3.23: the column of binary cells has less load on its column signal, as five cells less are connected).

This problem is best visible for negative j , so that the code jump $k + j$ squeezes the DTC output edges closer together. All select signals have then less time to settle properly, and in extreme cases the programming just fails. This, however, should be taken into account during the design phase, as the expected code changes are known beforehand.

Fig. 4.5 shows the internal waveforms of all DCEI² unit cells for (1) a code change of $127 \rightarrow 719$, stretching the phase by $+52.03^\circ$, and (2) a code change of $719 \rightarrow 127$, squeezing the phase by -52.03° . The pseudo-differential signals of the first interpolation node $V_{\text{int},1}$ and the select signals of all binary and thermometric cells are plotted here. Case (1) shows a stretching of the DTC output waveform with changing select signals at ~ 0.15 ns, leading to relaxed timing requirements inside the DCEI² unit cells. Case (2) squeezes the rising edge closer to the previous falling edge with changing select signals at ~ 0.6 ns, reducing the time for proper settling of the select signals. Fig. 4.5(a) shows the ideal case, where the select signals switch shortly after the falling edge of DTC_{out} . In

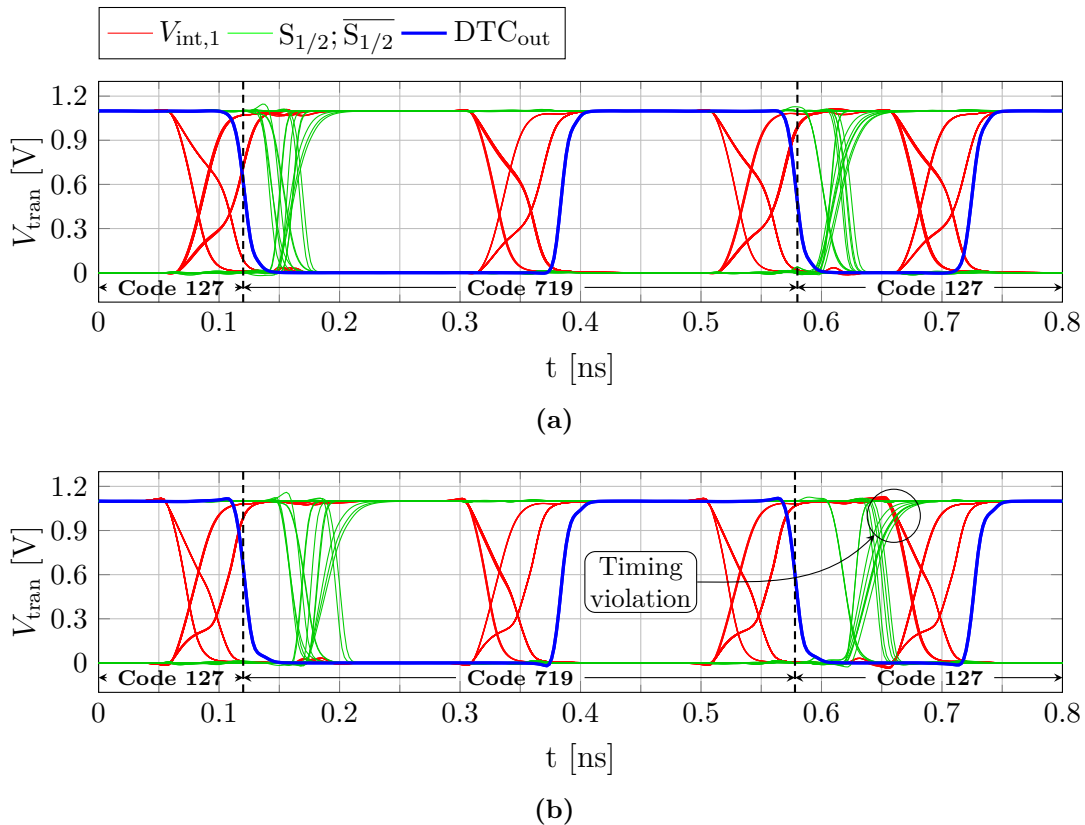


Figure 4.5 – Selection signals of the DCEI² V2 for a code sequence $127 \rightarrow 719 \rightarrow 127$ ($\pm 52.03^\circ$ phase step) at 2.5 GHz with (a) proper timing of the select signals, and (b) poor timing of the select signals.

Fig. 4.5(b) the select signals switch later, violating the timing as indicated on the plot, thus introducing dynamic errors.

The plot illustrates that the select signals should switch as close as possible after a falling DTC_{out} . This way they cannot disturb the internal signals anymore, as the output already toggled, and the time to the subsequent rising edge is maximized for better select signal settling. However, a proper timing is not easily achieved in circuit design, as the clock is derived from the DCEI²'s output, leading to a modulated clock. Strong clock drivers and a short logic path are imperative to get optimum timing and low dynamic errors.

4.2.5 Digital Control Signal Coupling

Last, activity on digital signals can directly couple into the analog signal. As the DTC evaluates only edge transitions, this is not in all cases critical. For example if a digital signal couples during its voltage level transition into the analog DTC signal, it does not have any effect on it while the analog signal is at high or low logic level. However, when the digital signal transition is close to the analog one, the digital signal can potentially introduce a dynamic error. As the analog signal transitions are shifted code-dependent by up to a full period, certain codes are affected while others are not.

Simulations demonstrated that this effect causes only minor dynamic errors. However, floor-planning of the physical layout should proactively reduce possible coupling between digital and analog signals through appropriate signal routing.

4.3 Dynamic Error Simulation

Dynamic errors were analyzed in simulation of the DCEI² V2 based DTC design introduced in Section 3.3. The previous section discussed dynamic effects under the assumption, that only a single code transitions is involved and that the DTC is in a periodic steady state before the transition. To quantify dynamic errors for real use-cases, a more complex test case needs to be evaluated.

Therefore, the DTC data path is excited with a code sequence $n_{dyn,err}$ that is a superposition of modulation data and a periodic code ramp

$$n_{dyn,err}[p] = n_{mod}[p] + n_{ramp}[p], \quad (4.20)$$

where $n_{mod}[p]$ is a code sequence describing a modulated signal, in the following test cases random phase modulation (RPM) data, and $n_{ramp}[p]$ is a constantly rising or falling periodic code ramp (PCR), adding a frequency shift component to the transmitted signal (compare the frequency synthesis with DDPS from Section 1.2.1).

This enables several test cases to prove the assumptions on dynamic errors from the beginning of this chapter. Two test groups are evaluated: first, the DTC is only excited with a periodic code ramp of constant step size ($n_{mod}[p] = 0$). Depending on the step size, this leads according to (1.2) to a certain frequency shift f_{offset} of the output signal from its center frequency f_{out} . If the simulation runs long enough to capture several samples per code, each sample for a given code is expected to have a similar dynamic error, as the code history is identical. A positive code ramp (leading to $f_{offset} < 0$) relaxes especially

Table 4.1 – Dynamic error test cases for $f_{\text{out}} = 2560$ MHz.

Test case	DTC step size/range		f_{offset}	Description
	n_{step}	ϕ_{step}		
PCR 1	−413	−36.3°	281 MHz	Periodic code ramp
PCR 2	−917	−80.6°	721 MHz	Periodic code ramp
PCR 3	917	80.6°	−457 MHz	Periodic code ramp
RPM 1	[−512, 512]	[−45°, 45°]	-	Random phase modulation
RPM 2	[−1024, 1024]	[−90°, 90°]	-	Random phase modulation

the select signal timing on the DTC, leading to lower dynamic errors, while negative code ramps (leading to $f_{\text{offset}} > 0$) tighten timings. Second, only phase modulation is applied ($n_{\text{ramp}}[p] = 0$). To limit the code activity, the allowed code step is limited by a minimum and maximum value. Larger code jumps are expected to lead to higher dynamic errors, therefore, different limits are investigated. Table 4.1 lists the simulated PCR and RPM test cases.

As accurate simulation requires layout extraction of the full DTC and LDO plus a package model, the simulation time for the ~ 3000 simulated RF periods is in the order of days. To speed up simulations, not every code is triggered, but the code sequence is forced on a grid of $0, 8, \dots, 4088$, setting the three LSBs to 0: $n_{2:0} = 0$. This ensures that a sufficient amount of samples is acquired for each code to do a statistical evaluation, while keeping the simulation time in a reasonable range.

From a transient simulation, the phase of each RF period is extracted and compared to the ideal expected value according to the programmed code. If a code is programmed multiple times during the code sequence, multiple INL values are measured that scatter according to the expected static INL and the dynamic errors. Fig. 4.6(a) and (b) show the extracted dynamic INL INL_{dyn} exemplary for test cases PCR 1 and RPM 1, respectively. Comparing both plots, the scattering of the PCR test case is much less than for the RPM test case.

The discontinuities at code transitions $88 \rightarrow 96$ and $600 \rightarrow 608$ in Fig. 4.6(a) originate from the nature of the applied code ramp. For example, codes in the range of 96 - 504 are preceded by codes in the range of 512 - 920, crossing always one MMD code transition. These codes are then most strongly influenced by MMD dynamic errors, which reflects in an deterministic dynamic error. Thus, code 96 is always preceded by a MMD transition, while code 88 is not, which leads to the discontinuity.

To evaluate and quantify the dynamic errors, INL_{dyn} needs to be compared to the static INL. The average dynamic INL $\overline{\text{INL}_{\text{dyn}}}$ is calculated as mean value of all measured INL points for a given code n and reveals differences depending on the test case. As the most prominent differences occur around code 256 and 768, mostly the first DCEI² interpolation seems to be affected by the high code activity. The reason behind the differences are variations in the supply voltage behavior (the first interpolation is sensitive to supply variations), as the modulation introduces strong load current fluctuation. Two effects, namely code-dependent current consumptions and instantaneous change of average current, are responsible. While the former effect is present in PCR and RPM test cases, the latter one differs for both test cases. The output frequency $f'_{\text{out}} = f_{\text{out}} + f_{\text{offset}}$ is constant

for the PCR case, while RPM introduces a frequency modulation with high bandwidth, resulting in a strong variation of the average current. A detailed analysis of this effect and its differences to supply voltage conditions for static INL measurements is presented in the upcoming section.

Not only $\overline{\text{INL}}_{\text{dyn}}$ is of interest, but also the distribution for each code in $\text{INL}_{\text{dyn}}[n]$, described by its standard deviation. The resulting $\sigma(\text{INL}_{\text{dyn}}[n])$ is a measure of INL_{dyn} 's scattering and, therefore, also of the dynamic error. It is plotted in Fig. 4.6(d) for three test cases. As expected the PCR test case shows the lowest dynamic errors, and the RPM test case with larger step sizes shows increased dynamic errors compared to the smaller step sizes. It is noteworthy that the dynamic errors, even for the PCR case, are multiple times higher than the DTC resolution $t_{\text{d,LSB}}$.

The following section discusses the reduction of dynamic errors through newly developed compensation circuits. While the current section only presented simulation results for three of five test cases, a full list of results from all test cases is presented in Section 4.4.3.

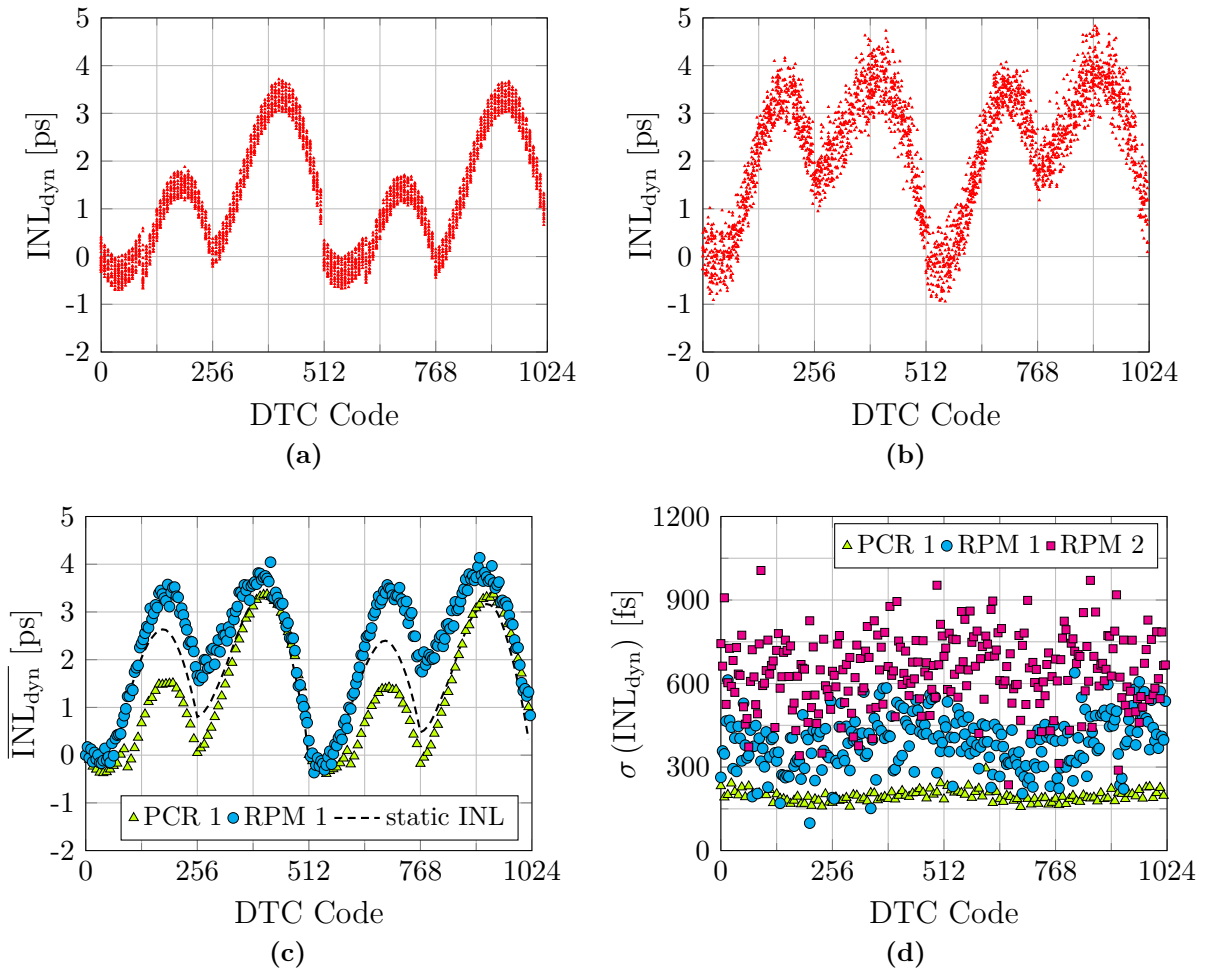


Figure 4.6 – Dynamic errors simulation results of the DCEI² V2 based DTC: dynamic INL of test cases (a) PCR 1 and (b) RPM 1, (c) corresponding average dynamic INLs, and (d) standard deviation of dynamic INLs.

4.4 Compensation for Load Current Variations at Supply Regulator Level

The most prominent contributors to dynamic errors were identified as LDO induced supply voltage distortions. They result from the fact that the DTC current changes with different codes and code transitions, imposing LDO regulation activity on the supply voltage. A solution for the logic current consumption was already proposed in Section 4.2. The remaining root causes are the code-dependent load current and the instantaneous load current change. Section 4.2 described how the load current conditions can be predicted from the programmed DTC code sequence. While the code-dependent current depends only on the currently applied DTC code, the instantaneous load current change depends also on the previously programmed code. This enables to extract the DTC load current from $n[z]$ and $n[z^{-1}]$. With this knowledge, a compensation mechanism can be operated that, for instance, either equalizes the LDO load current (e.g. a dummy fine tuning programmed with an inverted code), or that compensates for the load current change at supply regulator level. Both concepts rely on the fact, that the load current changes are known before they appear, enabling a compensation in a feed forward manner instead of a regulation as reaction to load current change.

For all discussed DTCs, the supply regulator is implemented as dual-loop LDO as shown in Fig. 4.7. A detailed analysis of a similar LDO architecture can be found in [98]. In the following, the LDO architecture is described briefly under negligence of the red highlighted components and pins, which are discussed afterwards. The LDO aims at controlling the pass gate device M_1 to regulate the supply voltage V_{out} that is connected to the DTC supply net V_{sup} . The green highlighted slow control loop senses the output voltage through the resistive divider consisting of $R_{1/2}$ and compares it to the band-gap reference voltage V_{ref} with the operational amplifier. The amplifier output signal V_{casc} controls then the gate of the cascode device M_2 . This ensures the correct control of the targeted output voltage level at V_{out} with high gain (usually > 40 dB [98]), that is provided by the operational amplifier.

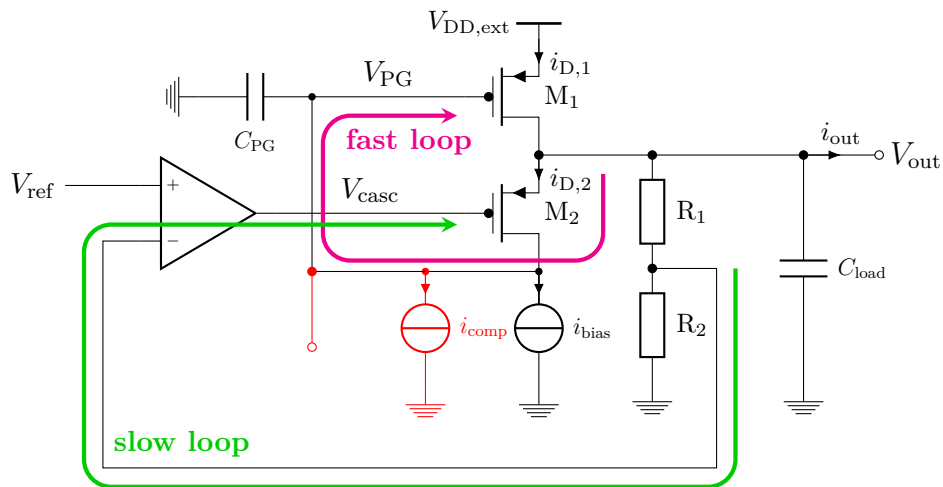


Figure 4.7 – LDO implementation for all discussed DTC variants. Red highlighted components mark the extension for the dynamic effects compensation.

The red highlighted fast-loop senses load current variation through M_2 and is biased by the current source i_{bias} . The voltage conditions at node V_{PG} are plotted in red in Fig. 4.9(a) for a settled slow control loop (fixed V_{casc}). If the load current increases, e.g. due to the DCEI² code change from $n : 0 \rightarrow 256$ at $t \sim 35$ ns, the conceptual fast-loop regulation operates as following:

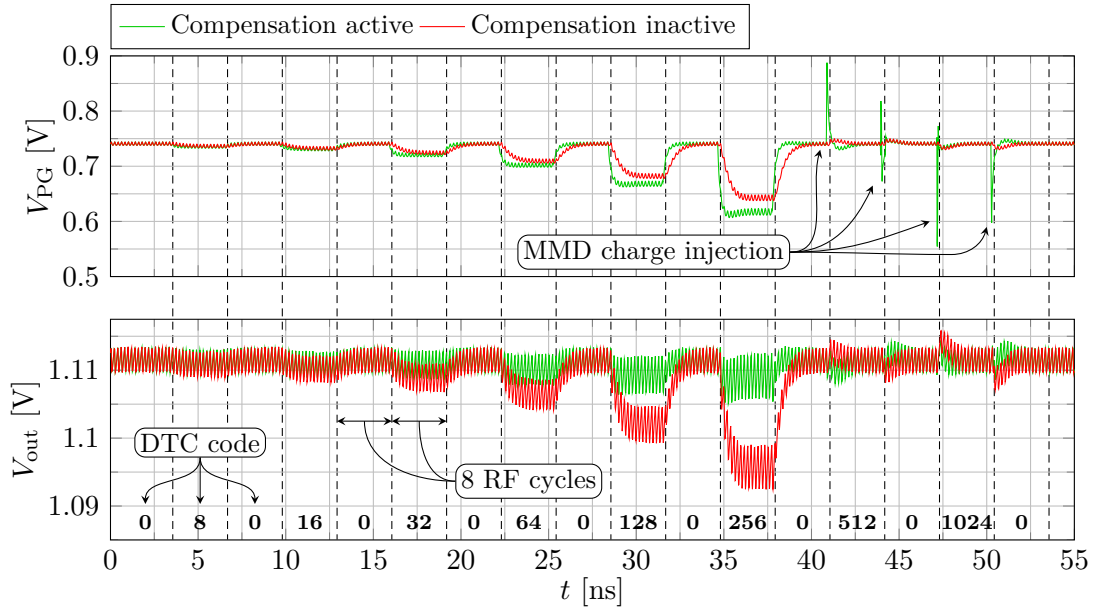
- i_{out} ($\cong i_{\text{load}}$ of the DTC) increases due to the code change, leading to a drop of V_{out}
- V_{GS} and V_{DS} of M_2 drop consequentially, reducing $i_{\text{D},2}$
- As i_{bias} is constant and $i_{\text{D},2}$ drops, V_{PG} is discharged by $\Delta i = i_{\text{bias}} - i_{\text{M}_2}$, thus opening M_1 and increasing $i_{\text{D},1}$ to adjust to changes of i_{out}
- The negative feedback fast-loop regulates V_{PG} until $\Delta i = 0$

As consequence from the fact, that i_{out} changes much faster than the slow-loop regulation, a remaining voltage error at V_{out} is observed (see the difference of V_{out} between code 0 and 256 in Fig. 4.9(a)). The magnitude of this drop is mostly influenced by the LDO's decoupling capacitor C_{load} and the fast-loop's gain. The slow-loop ensures that the average output voltage is regulated to the target defined by $R_{1/2}$ and V_{ref} , while the fast-loop reacts to fast transient current changes. However, the DTC changes its code and thus its current consumption on a rate of up to 3 GHz, which is much higher than the LDO's bandwidth in the megahertz domain.

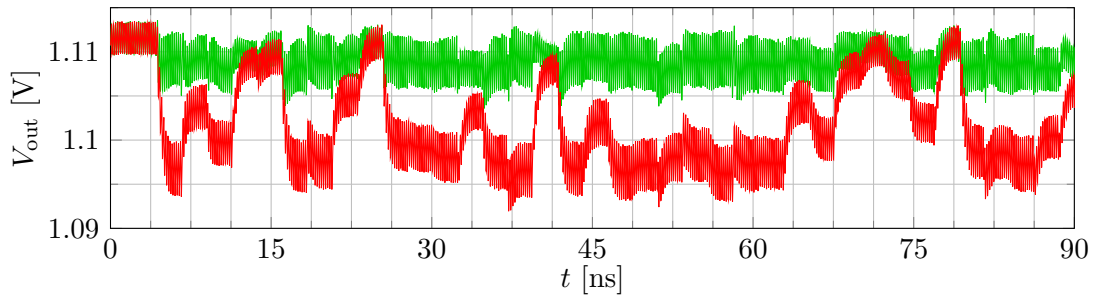
Therefore, the LDO from Fig. 4.7 is extended by a compensation current source i_{comp} in parallel to the biasing current, and an output pin that allows a charge injection on net V_{PG} through further compensation circuits. The former compensation accounts for code-dependent current due to the DCEI² programming, and the latter one for an instantaneous load current change due to MMD activity. According to (4.18) the instantaneous load current changes are related to coarse and fine tuning blocks, that are in this DTC design MMD and DCEI², respectively. As the MMD transitions dominate instantaneous load current change as pictured in Fig. 4.4, its compensation is implemented exemplary for the MMD case, but can be extended for compensation of arbitrary code jumps of coarse and fine tuning in an analogous manner. Both compensation types rely on the fact, that the DTC code changes much faster than the slow control loop. In the following, DCEI² and MMD compensation circuits are introduced and their impact on the supply voltage and on dynamic errors is discussed based on simulation results.

4.4.1 DCEI² Compensation

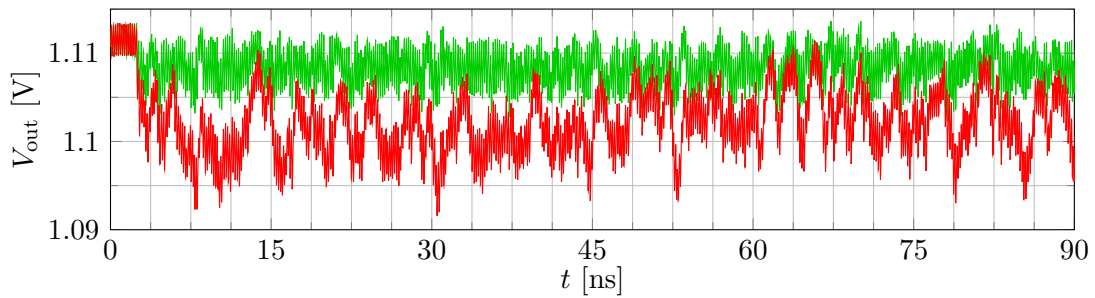
As the DTC current consumption depends almost linearly on the programmed code (compare Fig. 3.21), the LDO biasing can be adjusted in a feed-forward fashion to account for fast transient load current changes. The programmed DTC code enables the a-priori knowledge of i_{load} , which is encoded approximately linear in the DCEI² code. The LDO biasing is adjusted by placing a current source i_{comp} parallel to i_{bias} , as shown in Fig. 4.7. Five parallel current sources are used to implement i_{comp} , that are binary weighted and controlled by $n_{\text{DCEI}^2,7:3}$. This reflects the approximately linear current/code dependency of the DCEI² and excludes the LSB of the thermometrically controlled array ($n_{\text{DCEI}^2,2}$)



(a)



(b)



(c)

Figure 4.8 – DTC supply voltage with active and inactive compensation for (a) code toggling, and random code change at (b) every 6th and (c) every RF cycle.

and the binary bits ($n_{\text{DCEI}^2,1:0}$), as they influence the overall current consumption only negligibly.

The current source i_{comp} changes its value with the falling clock that latches the new DCEI² code word into the PI array. As i_{load} only changes with the subsequent rising DCEI² output edge, i_{bias} changes slightly before and adjusts $i_{\text{D},1}$ almost synchronous to the actual change Δi_{load} . The ideal LDO step of $\Delta i_{\text{D},1} = \Delta i_{\text{load}}$ would prevent V_{out} from changing its level, thus leading to a more constant V_{out} over code changes. The gate voltage V_{PG} of M_1 is plotted in Fig. 4.8(a). For active compensation, V_{PG} changes much faster to the steady state forced by i_{comp} . When i_{load} changes as predicted, V_{PG} is already half an RF cycle after the clock close to its steady state, thus regulation activity is reduced. As a consequence, the voltage drop at V_{out} is prevented, as $i_{\text{D},1}$ changes almost synchronous with i_{load} .

The resulting effect on the output voltage is shown in Fig. 4.8(a) in the time range of $0 \text{ ns} \leq t \leq 40 \text{ ns}$. In this time interval, the DTC code toggles as indicated on the plot (staying always constant for 8 RF clock cycles), leading to i_{load} variations due to the DCEI². While the different levels V_{out} are influencing only the static INL, the transient transitions between two codes introduces dynamic effects. With active DCEI² compensation, V_{out} has an overall more constant level, and more important, the transient changes due to regulation are reduced significantly.

To reduce the supply modulation effect for DTC blocks with code-dependent current, usually e.g. DCEI or switched capacitor dummy blocks are implemented that are programmed with an inverted code to equalize the overall current. Though this measure is effective, the excessive cost in terms of area and current consumption is not fit for a low power circuit design. While a dummy DCEI² would double the PI's current consumption, the current sources including their digital control add only $\sim 100 \mu\text{A}$. The prediction of i_{comp} cannot be accurate over PVT and is only an approximation of the load behavior, however, it is implemented to reduce and not remove dynamic errors.

4.4.2 MMD Compensation

The MMD distorts only one RF period through an instantaneous change of the average current (see Section 4.2.2). But even if the current distortion is short, the LDO needs several RF cycles to recover from it. To compensate for this effect, a charge is injected on the gate of M_1 , so that the device is slightly opened or closed just when the current change is expected. For a division-by-3 a higher current is expected, and for division-by-5 a lower current: M_1 needs to be opened (lower gate voltage) and closed (higher gate voltage) for division-by-3 and division-by-5, respectively. For LSBs changes of $n_{\text{MMD},0} : 0 \rightarrow 1$ and $n_{\text{MMD},0} : 1 \rightarrow 0$ the instantaneous current is lower and higher, respectively. In the following, first the LSB compensation and afterwards the division compensation is discussed.

The schematic of the LSB compensation circuitry is shown in Fig. 4.9(a). The LSB is identical to $n_{\text{MMD},0}$ and LSB changes are synchronized with the MMD clock to align the charge injection with the load current distortion. With the rising clock edge, $V_{\text{C,LSB}}$ is driven to the LSB state by the flip-flop, injecting a charge on V_{PG} via the compensation capacitor $C_{\text{C,LSB}}$. The injected charge leads to a voltage spike $\Delta V_{\text{PG,LSB}}$ on V_{PG} , that is

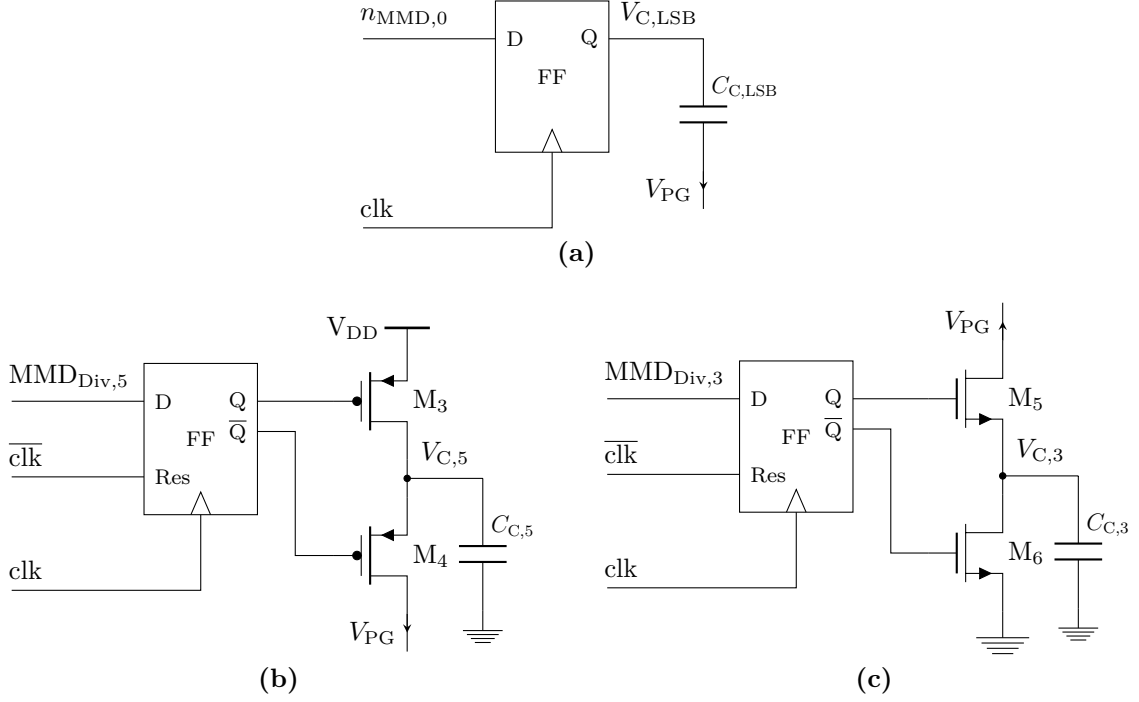


Figure 4.9 – MMD compensation circuits for charge injection on V_{PG} : compensation for (a) LSB changes, (b) division-by-5, and (c) division-by-3.

calculated to

$$\Delta V_{PG,LSB} = \pm \frac{C_{C,LSB}}{C_{PG}} V_{DD}, \quad (4.21)$$

where the positive and negative sign relate to $V_{C,LSB} : V_{SS} \rightarrow V_{DD}$ and $V_{C,LSB} : V_{DD} \rightarrow V_{SS}$, respectively.

For the division compensation, a logic needs to detect the division cases $MMD_{Div,3/5}$ from $n_{MMD,2:1}$, to indicate if the next code transitions triggers a division. The charge injection is then implemented as shown in Fig. 4.9(b) and (c). The compensation capacitors $C_{C,5}$ and $C_{C,3}$ are pre-charged to V_{DD} by M_3 and V_{SS} by M_6 , respectively. When $MMD_{Div,3/5}$ is active and the clock arrives (again, aligned with the MMD output clock), $M_{3/6}$ are turned off to end the pre-charging and $M_{4/5}$ are turned on to connect $C_{C,3/5}$ to V_{PG} . This initiates a charge sharing process between $C_{C,3/5}$ and V_{PG} 's capacitance C_{PG} (which is mainly M_1 's gate capacitance). Assuming ideal switches $M_{4,5}$, the charge sharing leads to a voltage spike $\Delta V_{PG,div-3/5}$ on V_{PG} of

$$\Delta V_{PG,div-3} = -V_{PG} \frac{C_{C,3}}{C_{PG} + C_{C,3}}, \text{ and} \quad (4.22)$$

$$\Delta V_{PG,div-5} = (V_{DD} - V_{PG}) \frac{C_{C,5}}{C_{PG} + C_{C,5}}. \quad (4.23)$$

Unlike the LSB transitions, the MMD divisions do not occur in alternating order. The charge injection needs additional control to be able to continuously inject a positive or negative charge for division-by-3 and division-by-5, respectively. Therefore, $C_{C,3/5}$ are pre-charged during the low clock period (flip-flop has active reset) and the active clock

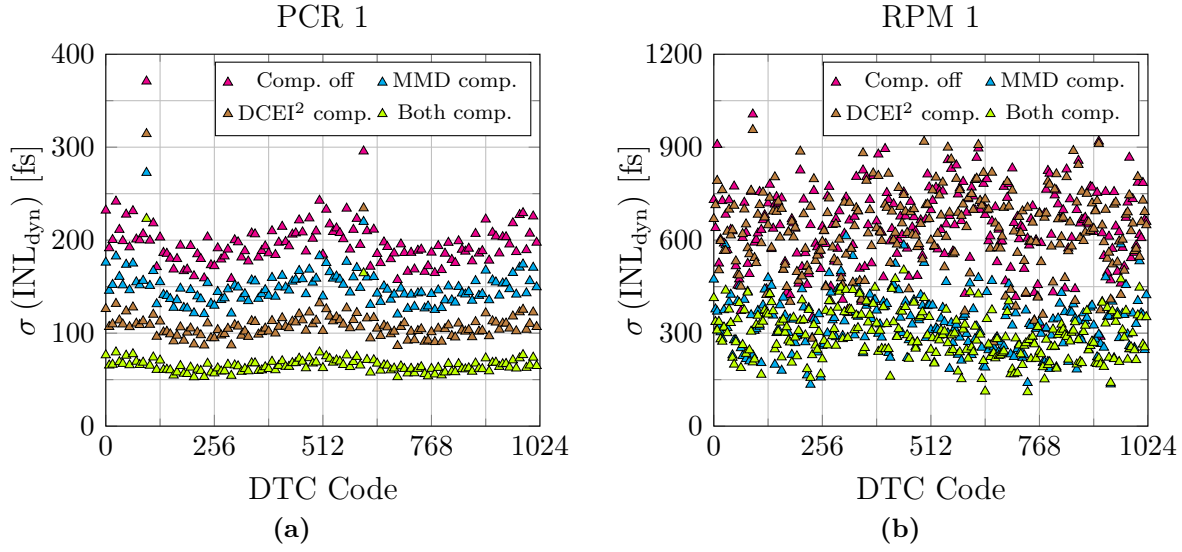


Figure 4.10 – Standard deviation of test case (a) PCR 1 and (b) RPM 1 for different compensation settings.

period closes the switch to V_{PG} (if $MMD_{Div,3/5}$ indicate to do so) to trigger the charge sharing process. The reset pin resets only the slave latch of the master-slave flip-flop, and proper signal timing of clk and \overline{clk} prevents timing violations.

Fig. 4.8(a) shows the LSB and MSB MMD compensation charge injection on net V_{PG} in the range of 40–55 ps. Half an RF cycle before the code changes, a charge injection leads to the visible voltage spike, resulting in a short opening or closing of M_1 that compensates for the instantaneous change of average current.

Either of the compensation circuits (DCEI² or MMD compensation) are deactivated with an active reset on the flip-flops in their data path. The remaining impact on the LDO is a slightly higher capacitance at net V_{PG} , as the additional compensation transistors are connected to it. Simulations proved stable LDO behavior for the minimum and maximum value of i_{comp} in AC simulation. Dynamic changes of i_{comp} cannot be verified in AC simulation, and transient simulations did not show any risk of unstable behavior. Both compensation types operate with RF rate and are validated in simulation for $f_{out} = 2.5$ GHz.

4.4.3 Compensation Impact on Dynamic Effects

To evaluate the influence of the compensation circuits on the dynamic errors, the simulations from Section 4.3 are repeated with active compensation. The main point of interest is the standard deviation of INL_{dyn} . Fig. 4.10(a) and (b) show exemplary for test cases PCR 1 and RPM 1 how $\sigma(INL_{dyn})$ changes for either only MMD or DCEI² compensation, and for both compensations.

To quantify the dynamic errors in a single number, the average standard deviation $\overline{\sigma(INL_{dyn})}$ is calculated. All five test cases are compared based on $\overline{\sigma(INL_{dyn})}$ for different compensation settings in Table 4.2. For test cases PCR 1/2 and RPM 1/2, all compensation settings reduce dynamic errors, proving the functionality of the LDO extension. If both

Table 4.2 – Average of $\sigma(\text{INL}_{\text{dyn}}[n])$ over code for different compensation settings and test cases.

Test case	$\overline{\sigma(\text{INL}_{\text{dyn}})}$			
	Comp. off	MMD comp.	DCEI ² comp.	Both comp.
PCR 1	197 fs	149 fs	109 fs	67 fs
PCR 2	292 fs	183 fs	217 fs	119 fs
PCR 3	36 fs	75 fs	123 fs	18 fs
RPM 1	395 fs	279 fs	361 fs	236 fs
RPM 2	645 fs	332 fs	634 fs	293 fs

compensation types are active, a total dynamic error reduction of 40–66% is observed. While the DCEI² compensation brings the highest benefit to test cases PCR 1/2, the MMD compensation has the strongest impact on RPM 1/2. As already discussed in Section 4.3, PCR test cases show very deterministic dynamic errors, for example certain codes are always preceded by either an LSB or MSB MMD transition. In the case of RPM, each code can be reached without MMD transition, or by positive or negative LSB/MSB MMD transitions. Therefore, the RPM test cases show stronger dynamic effects due to MMD activity, and the MMD compensation shows its strength here.

Test case PCR 3 has already outstanding low dynamic errors without any compensation. This results from the positive code ramp that relaxes DTC logic timings, as each period is stretched. Here the activation of only MMD or DCEI² compensation increases the dynamic error, and only the activation of both compensation achieves a lower dynamic error. Consequently, deactivation of the compensation is advised for operation conditions where low dynamic errors are expected.

4.5 Summary and Conclusion

This chapter reviewed and analyzed dynamic effects that occur on top of the static non-linearity in DTC's. Dynamic effects are defined based on code transitions and the code history. Low dynamic errors are expected from use cases with periodic code patterns, such as frequency synthesis, as each programmed code has identical code history. Use cases with random code activity, such as phase modulation, are expected to have higher dynamic errors as the code history is also of a random nature. The dynamic errors cannot be predicted or modeled easily due to the number of possible effects and involved circuits.

Table 4.3 summarizes the identified root causes for dynamic effects and shows countermeasures that can be taken into account during the DTC design phase to achieve a design with lower dynamic errors. On top of the design guidelines for the specific dynamic effect root causes, general countermeasures for supply induced dynamic effects include an LDO with high bandwidth, thus fast regulation response to transient load current changes, and a large decoupling capacitor C_{load} for reduced supply ripple due to load current changes.

Simulations of the DCEI² V2 based DTC proved the assumption, that periodic code patterns lead to lower dynamic errors than random phase modulation. For both test cases, the supply varies around an average value as the load current changes continuously. In

Table 4.3 – Overview on all identified dynamic effects, their root-causes, and possible countermeasures to reduce dynamic errors.

Effect	Root cause	Countermeasures
Code-dependent current consumption	Especially switched capacitor and PI fine tuning show a code-dependent current	Dummy load to equalize current
Instantaneous change of average current	Code change causes duty cycle change, altering the current consumption instantaneously	Supply regulator with high bandwidth and large decoupling capacitances
Logic current consumption	Digital current consumption distorts the supply voltage.	Using different supply regulators for the digital and analog supply domains.
Logic select signal timing	Long clock to select signal delay violates select timing inside the PI unit cells for negative code jumps	Reduce clock to select signal delay by appropriate design
Logic signal coupling	Digital signals couple in analog signals and disturb them	Physical separation of analog and digital signals in layout

contrast to this, static INL measurements leave the same DTC code active for multiple DTC output cycles to settle possible dynamic effects. This allows the supply to settle to a steady state value of the fast control loop for the static INL measurement. These facts about the different supply conditions for static and dynamic INL measurements introduce a difference between static and average dynamic INL, where the average dynamic INL additionally depends on the programmed code sequence.

In addition to the circuit design measures, a novel compensation circuit was presented to reduce the dynamic effects. The fast regulation loop of the dual loop LDO from Fig. 4.7 is extended by two compensation circuits that compensate for code-dependent current consumption of the DCEI² and instantaneous changes of the average current caused by the MMD. The expected DTC current consumption is extracted from $n[z]$ and $n[z^{-1}]$ and enables a feed-forward modification of the LDO pass-gate transistor's gate voltage to adjust the LDO for the expected current. This overcomes the bandwidth limitations of the LDO and reduced LDO control loop regulation activity on its output voltage. Dynamic error circuit simulations showed a reduction of dynamic errors by 40–66 % for active compensation.

5 DTC Measurements

This chapter presents and compares measurement results for the three test chips developed in the present thesis, plus measurements of the already existing DCEI based DTC reference test chip. This includes the quasi-static nonlinearity of the different DTCs, as well as measurements of dynamic effects and evaluation of the presented dynamic effects compensation. The measured and simulated DTC nonlinearity is quasi-static, as the DTC is excited in both cases with a slowly changing DTC code ramp to extract TF, DNL, and INL. Before discussing the measurement results, known DTC measurement techniques are recapped briefly and a method to capture quasi-static DTC nonlinearity with femtosecond accuracy is discussed.

The chapter is structured as following: First, Section 5.1 introduces the measurement techniques used for DTC verification. Afterwards, Section 5.2 and 5.3 present quasi-static nonlinearity measurement results of the CF-DCEI and DCEI² based DTCs, respectively, and compares the measured nonlinearity to the models discussed in Chapter 3. Section 5.4 discusses dynamic errors measurements and evaluates the functionality of the dynamic effects compensation circuits. Finally, Section 5.5 concludes this chapter with a comparison of the quasi-static DTC linearity to measurements of the reference DCEI based DTC design.

5.1 Measurement Setup

Most published DTCs are embedded in an application that does not allow for external measurements of the DTC's nonlinearity, thus limiting the presentable nonlinearity to circuit simulation results only. However, some authors present detailed DTC measurement results, using one of the three following measurement techniques: First, DTCs can be characterized by measuring the phase of the output signal with a high speed oscilloscope [12, 26, 29]. The output waveform is then evaluated for the time of its threshold crossings

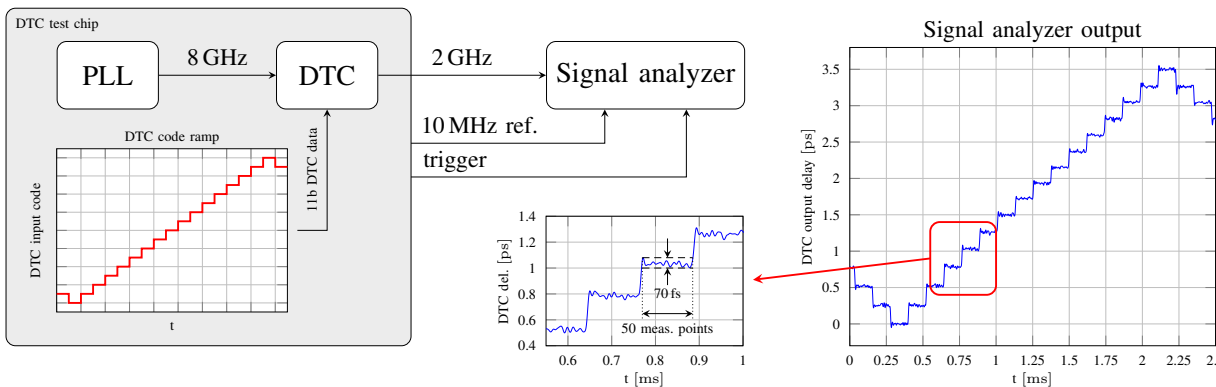


Figure 5.1 – Measurement setup and interconnection of test chip and spectrum analyzer.

to obtain the output signal’s delay. Second, on-chip phase detectors can be used to convert phase delay to a voltage that is measured externally [2, 16]. An XOR gate with inputs connected to a reference clock and the DTC output generates a square wave signal with a duty cycle proportional to the input signals’ phase difference. A subsequent low pass filter converts the duty cycle to a voltage, proportional to the DTC phase shift. Third, the DTC can be characterized by exciting it with periodic code transitions, and evaluating the generated spurs in the DTC output signal spectrum [1, 99]. The DTC toggles between two codes, generating spurs with a power level proportional to the code related phase difference. The extracted data is then the DTC’s DNL. While this method achieves high accuracy, each measurement captures a single code transition only.

The linearity measurement method used in the present thesis was published in [A3]¹. For evaluation of the quasi-static DTC nonlinearity, the DTC is excited with a triangular code ramp of digital data according to Section 2.1.4, applying a slowly varying phase change with a period T_{ramp} on the DTC output signal. The output signal is then demodulated by a Rohde & Schwarz FSV signal analyzer with *Analog Demod* software option, measuring the transient phase change in the baseband domain. As high accuracy requires averaging by the signal analyzer, the measurement is externally triggered by a signal with a period of $T_{\text{trigger}} = T_{\text{ramp}}$, provided by the DTC test chip platform. Furthermore, a 10 MHz signal synchronizes the signal analyzer to the reference clock of the PLL. To adjust the demodulation frequency of the signal analyzer exactly to f_{out} , an integer channel is chosen for the PLL, so that the output frequency is an integer multiple of the the chip reference.

As a signal’s phase is a relative measure, code $n = 0$ is defined as 0° reference and all measurements are aligned to it. Always 16 codes are measured at once with a code ramp of 16 steps. The full measurement is then split in code ramps with codes 0–15, 15–31, and so forth, and then aligned by their overlapping codes. Fig. 5.1 shows the interconnection of test chip and signal analyzer. The code ramp shown on the left hand side leads to the measurement plotted on the right hand side. The single phase steps have 50 measurement points each, which are distributed Gaussian with a peak-to-peak variation of ~ 80 fs as indicated in Fig. 5.1. Averaging them leads to femtosecond accuracy, which is needed for DTC resolutions down to $t_{\text{d,LSB}} = 48.8$ fs.

5.2 Quasi-Static CF-DCEI Nonlinearity

The CF-DCEI design focused on linearizing the interpolation process of the DCEI. The measured TF from Fig. 5.2(a) shows the coverage of the full 2π FSR. Fig. 5.2(b) shows the DNL. It is ≥ -63 fs ($-0.26 t_{\text{d,LSB}}$), which proves that it is a fully monotonic DTC as the absolute step size is always positive. While $|\text{DNL}| \leq 63$ fs for most codes, spikes of ~ 300 fs ($1.25 t_{\text{d,LSB}}$) are visible at codes that trigger MMD transitions, caused by physical layout coupling between $\text{MMD}_{\text{out},1/2}$. Fig. 5.2(c) shows the INL with a peak value of -1.2 ps, dominated by the CF-DCEI. As the MMD aligns the circuit with different VCO edges every 512 codes, the nonlinearity is repetitive in the range of one VCO cycle.

The simulated current consumption of the MMD, MUX+DEL and CF-DCEI from a 1.1 V supply is 5 mA, 2 mA and 11 mA, respectively (compare Table 3.7). The supply

¹ Pre-work for this measurement method was done in [100] and [101]. It was further modified for the present thesis, including the verification that it is capable of single digit femtosecond accuracy.

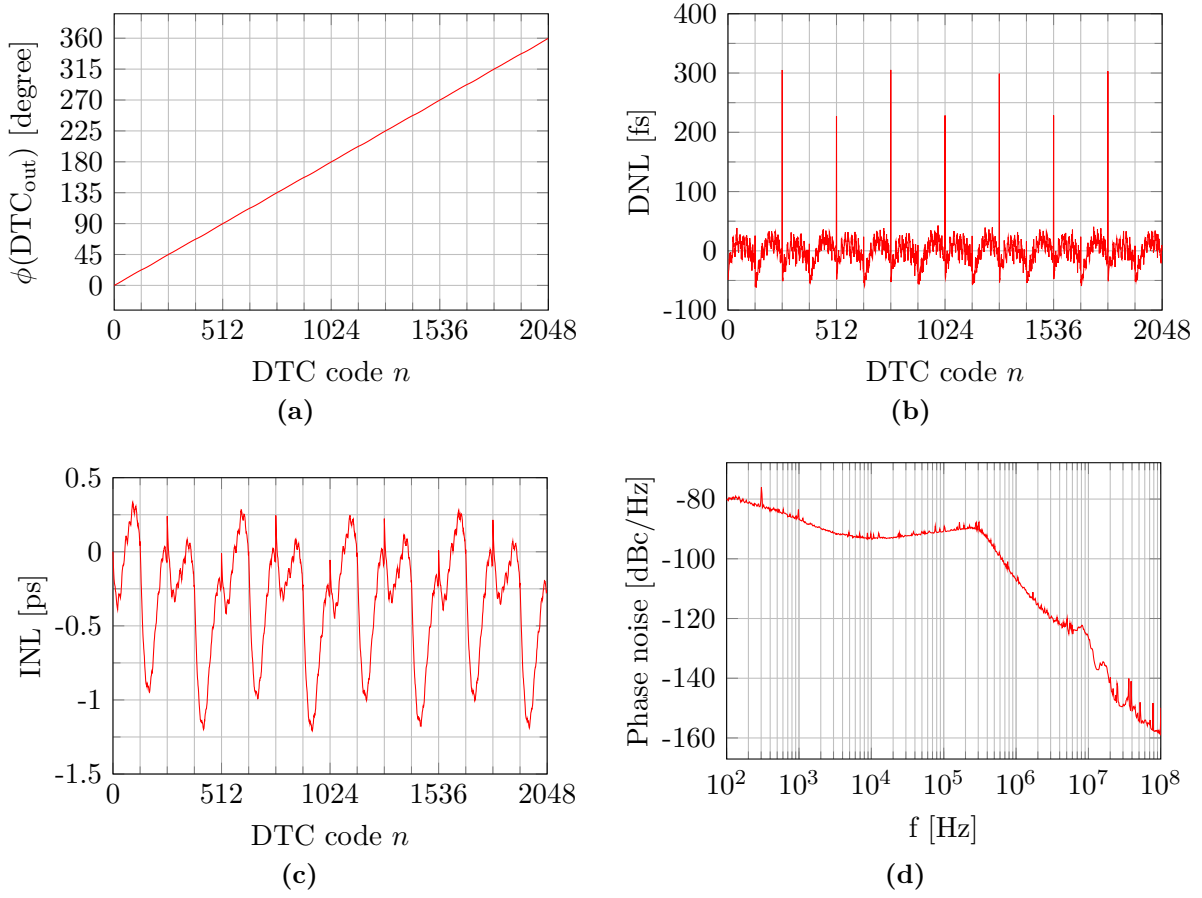


Figure 5.2 – CF-DCEI based DTC measurement results: (a) TF, (b) DNL, (c) INL, and (d) phase noise.

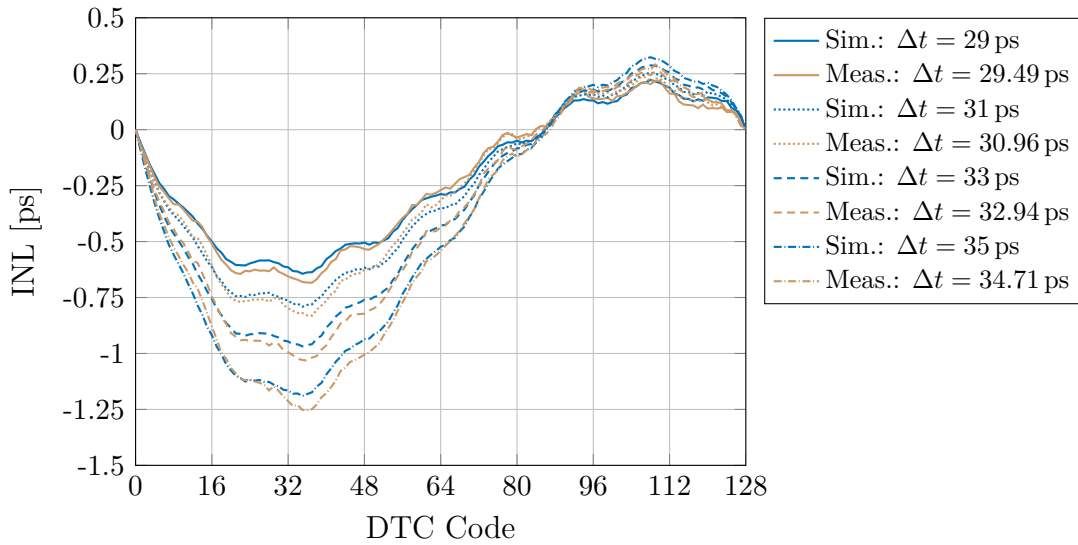


Figure 5.3 – INL comparison between measurement and circuit simulation of the CF-DCEI.

voltage is generated by a chip internal LDO, that has a measured current consumption of 18 mA from a 1.4 V supply, verifying the circuit simulations. Fig. 5.2(d) shows the measurement of the phase noise for a 2 GHz carrier signal. The close-in phase noise is dominated by typical PLL characteristics. As the DTC is a purely digital circuit, the generated jitter is of interest. Jitter limits the far-out phase noise floor, which is at -159 dBc/Hz for an offset of +100 MHz from the carrier. This value already includes contributors as the PLL or the 8 GHz signal path. The measurement noise prevents the evaluation of phase noise dependency on DTC codes, as possible differences are not visible. A difference is expected when the CF-DCEI weights either MUX or DEL output of the coarse tuning stage, as the delay element adds additional jitter to the signal compared to the MUX path, where this block is missing.

To prove a high measurement accuracy, the measured CF-DCEI's INL is compared to simulations in Fig. 5.3. Variation of Δt in measurements is enabled through variation of t_{DEL} , and the simulated and measured CF-DCEI INLs are aligned to the respective interpolation range. The comparison shows excellent matching with slight deviation caused by PVT variations, mismatch, and measurement noise.

The extracted peak INL ($|\text{INL}|_{\text{max}}$) of the CF-DCEI (not of the full DTC) is then compared to the 1st and 2nd order CF-DCEI model in Fig. 5.4. The 1st order model shows the correct peak INL trend for variation of Δt , however, lacks accuracy in the absolute INL value. When $t_{\text{r/f}}$ is sufficiently small compared to $t_{\text{int},\lambda \neq 0}$ ($t_{\text{r/f}} = 25$ ps), the 1st order model gives a good approximation of the peak INL, as the effects covered in this model dominates over the other INL contributor, namely the finite $t_{\text{r/f}}$. For example in the region of $\Delta t = 36 - 40$ ps, the 1st order model with $t_{\text{int},\lambda \neq 0}$ approximates the simulation results with good accuracy.

The 2nd order model gives an excellent estimation of the peak INL over a wide range of Δt . Considering the approximation that derives $t_{\text{int},\lambda \neq 0}$ from t_{int} , a comparison between a numerical and analytical estimation of the 2nd order model clarifies how it impacts the model accuracy. The negligible difference between both confirms the assumption made in Appendix B that lead to equation (3.8).

5.3 Quasi-Static DCEI² Nonlinearity

The design of the DCEI² based DTC focused on low power consumption and higher resolution. If not noted otherwise, the following measurement results are captured with the DCEI² V2 based DTC test chip. As the major differences between DCEI² V1 and V2 are the PI array dimensions and the binary bit implementation, a detailed comparison of both DCEI² designs by means of binary bit linearity is given in Section 5.3.2.

The quasi-static linearity characteristics of the DCEI² were measured similar to the CF-DCEI and are plotted in Fig. 5.5 for $f_{\text{out}} = 2496$ MHz (integer channel of the PLL). The TF from Fig. 5.5(a) proves full coverage of the 2π FSR over code. Fig. 5.5(b) shows the DNL. The major part of the DNL is in the range of -60 – 80 fs, with peaks of -193 fs ($-1.98 t_{\text{d,LSB}}$) and -276 fs ($-2.83 t_{\text{d,LSB}}$) at code transitions that trigger MMD LSB and MSB transitions, respectively. The spikes originate from coupling between $\text{MMD}_{\text{out},1/2}$ at the interconnection between MMD and DCEI², as well as coupling between their equivalents inside the DCEI² cell array. The spreading of the DNL in the ranges of $0 - 127$,

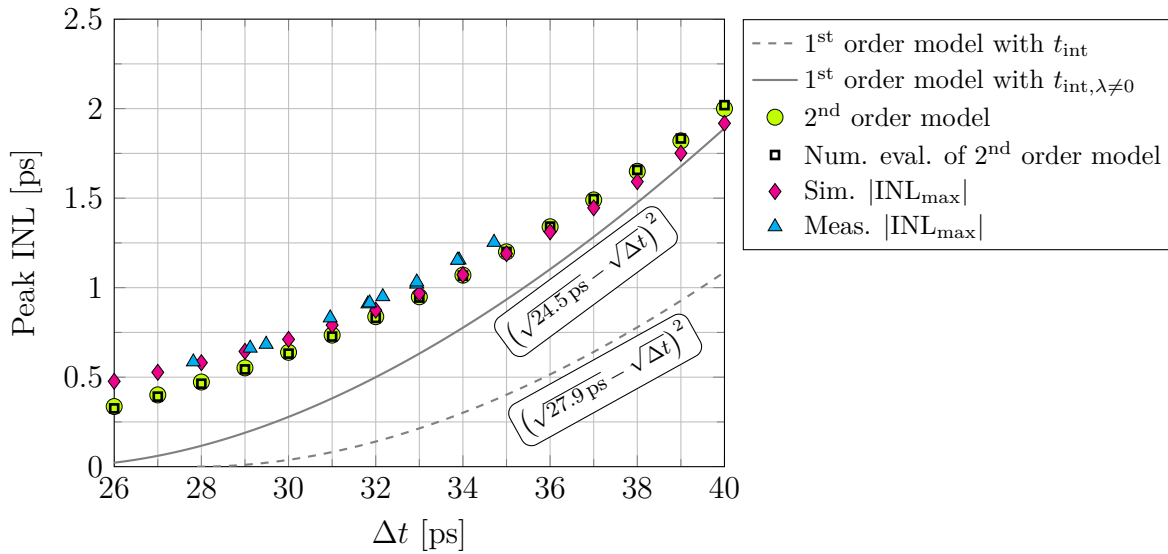


Figure 5.4 – Comparison of the CF-DCEI's peak INL ($|INL|_{\max}$) extracted from model, circuit simulation, and measurement.

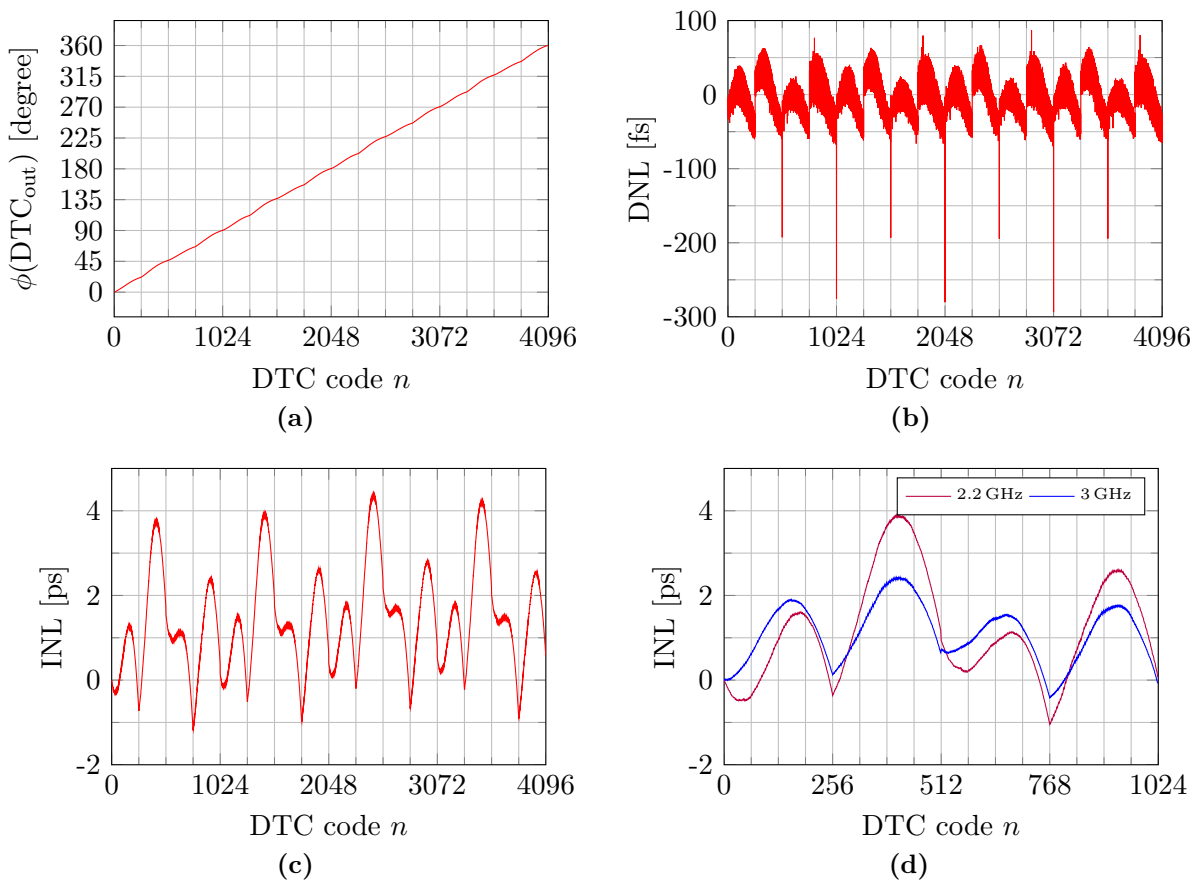


Figure 5.5 – Measurement results of the DCEI² V2 based DTC for $f_{\text{out}} = 2496$ MHz: (a) TF, (b) DNL, and (c) INL. (d) INL for $f_{\text{out}} = 2.2$ GHz and $f_{\text{out}} = 3$ GHz.

128 – 255, and so on, is caused by mismatch of the binary unit cells, which is analyzed in detail in Section 5.3.2. A visible effect of row or column transition as in the CF-DCEI is not visible here, as the resistance of the row and column interconnections were lowered to reduce it significantly.

The INL is plotted in Fig. 5.5(c). For the same reasons as in the CF-DCEI design, the INL is repetitive in a range of 1024 codes with deviation due to measurement noise. For the INL at codes $512 + h1024, h \in \{0, 1, 2, 3\}$, the DTC output is aligned with $\text{MMD}_{\text{out},2}$, that is triggered by VCO_n . The INL is at 1.25 ps at these points, indicating a mismatch between $\text{VCO}_{p/n}$. Simulations of the LO distribution network between PLL and DTC (including layout extracted parasitics) found 0.95 ps mismatch between $\text{VCO}_{p/n}$ introduced by this network. The remaining 300 fs are related to $\text{VCO}_{p/n}$ buffering and routing inside of the MMD, which is confirmed by circuit simulations of this block. This effect results in interpolation ranges of $\Delta t_1 = 26.25$ ps and $\Delta t_2 = 23.75$ ps for the DCEI² and explains the INL differences to circuit simulations.

The INL for the min. and max. frequency of $f_{\text{out}} = 2.2$ GHz and $f_{\text{out}} = 3$ GHz is plotted in Fig. 5.5(d) for the range of one VCO cycle. As expected, the smaller Δt related to the higher frequency leads to a lower peak INL, as the DCEI²'s nonlinearity depends on it.

5.3.1 INL Tuning

As the 1st interpolation in the DCEI² is sensitive to PVT, an implemented tuning at input buffer level influences $t_{r/f}$ to ensure an optimum INL over PVT. The 5 bit digital control word $d_{t_{r/f},\text{tune}}$ reduces $t_{r/f}$ for higher digital values.

Fig. 5.6(a) and (b) show INL measurements at $f_{\text{out}} = 2496$ MHz for all possible settings of $d_{t_{r/f},\text{tune},4:0}$ with different supply voltage of $V_{\text{DD}} = 1.08$ V and $V_{\text{DD}} = 1.13$ V, respectively. The red INL plots highlight the ideally tuned INL, as $|\text{INL}[256]|$ is minimum. The interpolation parameters t_{int} and $t_{r/f}$ are influenced through supply voltage variation, which explains the difference between both graphs.

To investigate the INL tuning, the nonlinearity of the DCEI² V2 is evaluated in the same fashion as for the model (see Section 3.3.2). Variation of Δt is achieved through variation of f_{ref} , and variation of $t_{r/f}$ through variation of $d_{t_{r/f},\text{tune},4:0}$. As $t_{r/f}$ cannot be measured directly and the digital control has a nonlinear relation to it, a mapping between control word $d_{t_{r/f},\text{tune},4:0}$ and $t_{r/f}$ is extracted from circuit simulations. Fig. 5.6(c) shows the measurement points $\text{INL}[256]$ that are extracted from INL measurements in the range of $f_{\text{out}} = 2.2\text{--}2.6$ GHz. An interpolation between them results in the plotted grid. Evaluation of this surface for $\text{INL} = 0$ results in the green contour line, which marks the points of ideal tuning. Looking at the relation between Δt_1 and $t_{r/f}$ along this contour results in the plot in Fig. 5.6(d), which shows a linear relation as expected from the DCEI² model evaluation. A linear regression leads to the ideal tuning of

$$t_{r/f}(\Delta t_1) = 1.10\Delta t_1 - 15.48 \text{ ps}. \quad (5.1)$$

Simulations show a tuning range of $27.17 \text{ ps} \leq t_{r/f} \leq 56.82 \text{ ps}$ for the DCEI² input signals. Evaluating (5.1) for the range of Δt_1 given in Table 3.2 according to the frequency range of 2.2–3 GHz leads to a required tuning range of $30.44 \text{ ps} \leq t_{r/f} \leq 47.13 \text{ ps}$. The simulated range of 29.65 ps for $t_{r/f}$ is then 12.96 ps larger compared to the required range of 16.69 ps. However, Fig. 5.6(a) and (b) show a strong dependency on supply voltage,

and simulations show also a dependency on process variations. Therefore, the range may not be necessary for the specifically measured test chip, but a headroom is needed if PVT is taken into account.

5.3.2 Binary Bit Implementation

To evaluate the different binary bit implementations, both DCEI² versions are compared in their code range for ideal tuning of the input buffers ($\Delta t_{2,1} \approx \Delta t_{2,2}$). Both test chips feature only a 12 bit DTC data path with control bits $n_{\text{data},11:0}$. For the DCEI² V1 based DTC with 13 physical bits $n_{12:0}$, a multiplexer can select the 12 data path bits either for the upper or lower 12 bits of the DTC control: $n_{\text{data},11:0} \rightarrow n_{12:1}$ or $n_{\text{data},11:0} \rightarrow n_{11:0}$. The

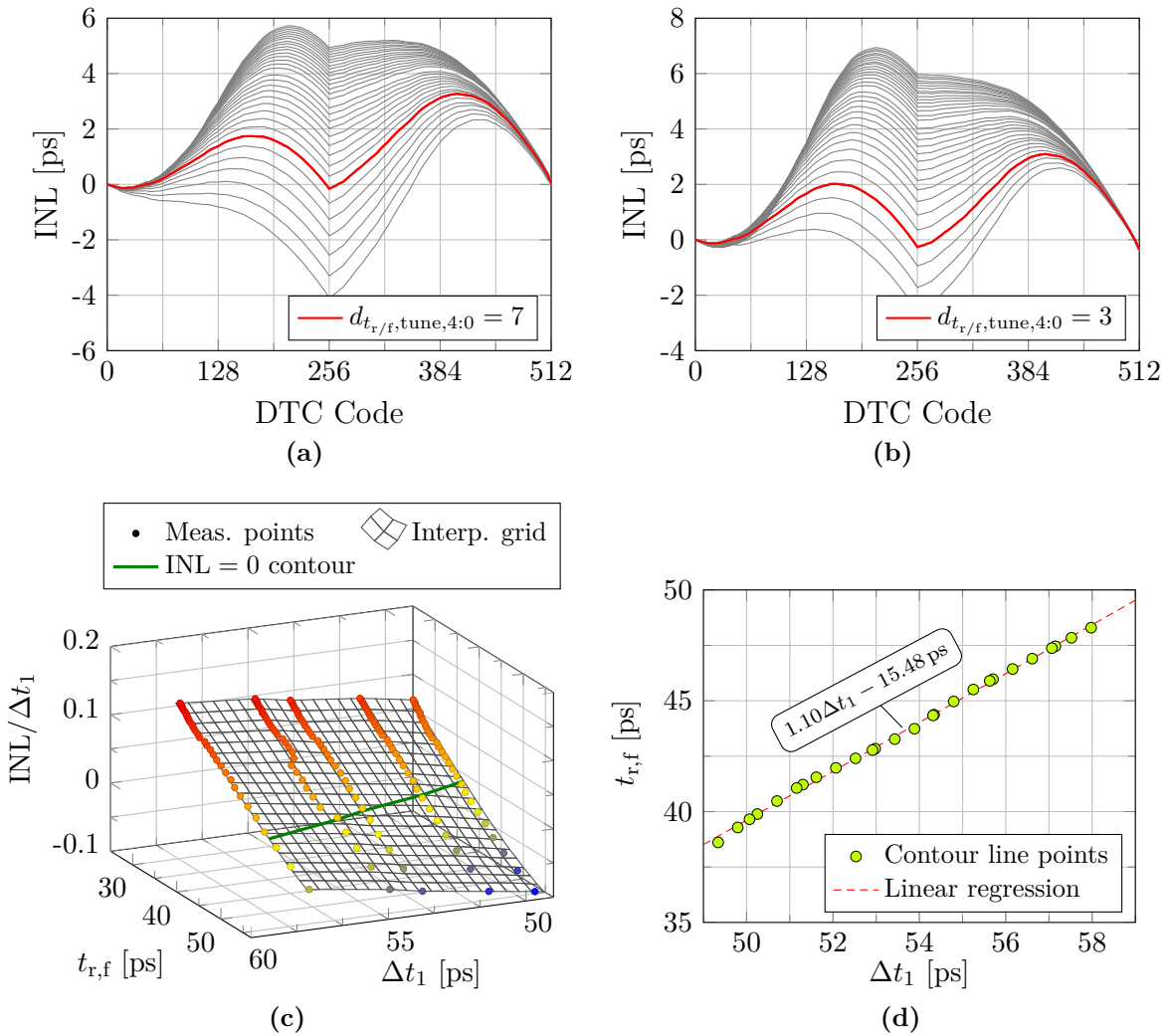


Figure 5.6 – Tuning of the INL with control word $d_{t_{r/f},tune}$ in the DCEI² V2 code range: (a) INL for $V_{DD} = 1.08$ V, (b) INL for $V_{DD} = 1.13$ V, (c) normalized INL[256] for $V_{DD} = 1.13$ V and variation of f_{ref} with highlighted contour line at $INL[256] = 0$, and (d) evaluation of relation between Δt_1 and $t_{r/f}$ at the contour line.

physical bit that is not connected to the digital data input is set to zero. The former case allows capturing of the INL over the full code range, while the latter one allows to control all binary bits, but not the MSB.

The measured DNL of both test chips is plotted in Fig. 5.7(a) and (b). For comparison to simulations, one of the Monte Carlo plots from Fig. 3.25 is added to each of the measurements. As expected from simulation, DCEI² V1 shows a DNL that is non-monotonic. The general DNL shape and distribution fits well to the simulation. DCEI² V2 measurements show slightly stronger deviations to circuit simulations, but the expectation of a monotonic behavior is confirmed by the test chip.

In the ideal case, the binary cells slice down the delay $t_{d,\text{therm}}$ between two thermometric steps in identical intervals: the $B_{1/2}$ should increase the overall delay by $t_{d,\text{therm}}/2$, and the $B_{1/4}$ cell by $t_{d,\text{therm}}/4$. As the DNL has a distinct shape defined by the thermometric cell array, $t_{d,\text{therm}}$ changes over code and the binary bits should adjust accordingly. Fig. 5.7(c) and (d) plot the ratio of each binary step to the corresponding step of the thermometric array for the full DCEI² code range. Ideally, the ratio for $B_{1/2}$ and $B_{1/4}$ cells is at 50% and 25%, respectively. For DCEI² V1 $B_{1/2}$ has $\pm \sim 15\%$ deviation from its ideal value. In DCEI² V2 this deviation was successfully reduced to $\pm \sim 7\%$, resulting in a close to ideal $B_{1/2}$ cell. However, the major difference between both version is the $B_{1/4}$ cell: while for DCEI² V1 the $B_{1/4}$ cell shows for certain codes (around code 500) almost the driving strength of the $B_{1/2}$ cell, it shows only a variation of $-3-37\%$ for DCEI² V2. Fig. 5.7(e) and (f) evaluate the binary bits for the minimum and maximum frequency of DCEI² V2, to check the behavior for different f_{out} and Δt . In both cases the DCEI² is monotonic, however, for $f_{\text{out}} = 3\text{ GHz}$ the deviation of the relative step size from its ideal value increases.

This leads to the conclusion, that binary cells should be implemented with a similar stack of transistors compared to the thermometric unit cell as done in DCEI² V2. The so implemented binary cells increase the resolution of the DTC with a low power overhead, while keeping the monotonic DTC behavior.

5.4 Dynamic DTC Performance

Next to measurements of quasi-static DTC nonlinearity, the dynamic DTC performance, including measurements of dynamic errors, is of interest. The DCEI² V2 based DTC test chip is used for all further dynamic error measurements. First, the principle functionality of the compensation circuits is verified. This requires the measurement of the DTC's output phase/delay for test cases that trigger the single compensation mechanisms for active and inactive compensation. An exemplary code sequence is plotted in Fig. 5.8(a) for code transitions that only modify DCEI² related codes for verification of the DCEI² compensation. Other code transitions, that for instance lead to MMD transitions, are measured in an analogous manner. The chip output signal is measured with a high speed sampling oscilloscope, and the resulting sinusoidal waveform is evaluated for its zero crossing times (extracting the phase of the sinus function). This information allows the calculation of the DTC output delay t_d . The deviations between the programmed code n and the measured t_d in Fig. 5.8(a) are according to the DTC's nonlinearity. Ideally, dynamic errors are extracted from the measurements by looking at the deviations between

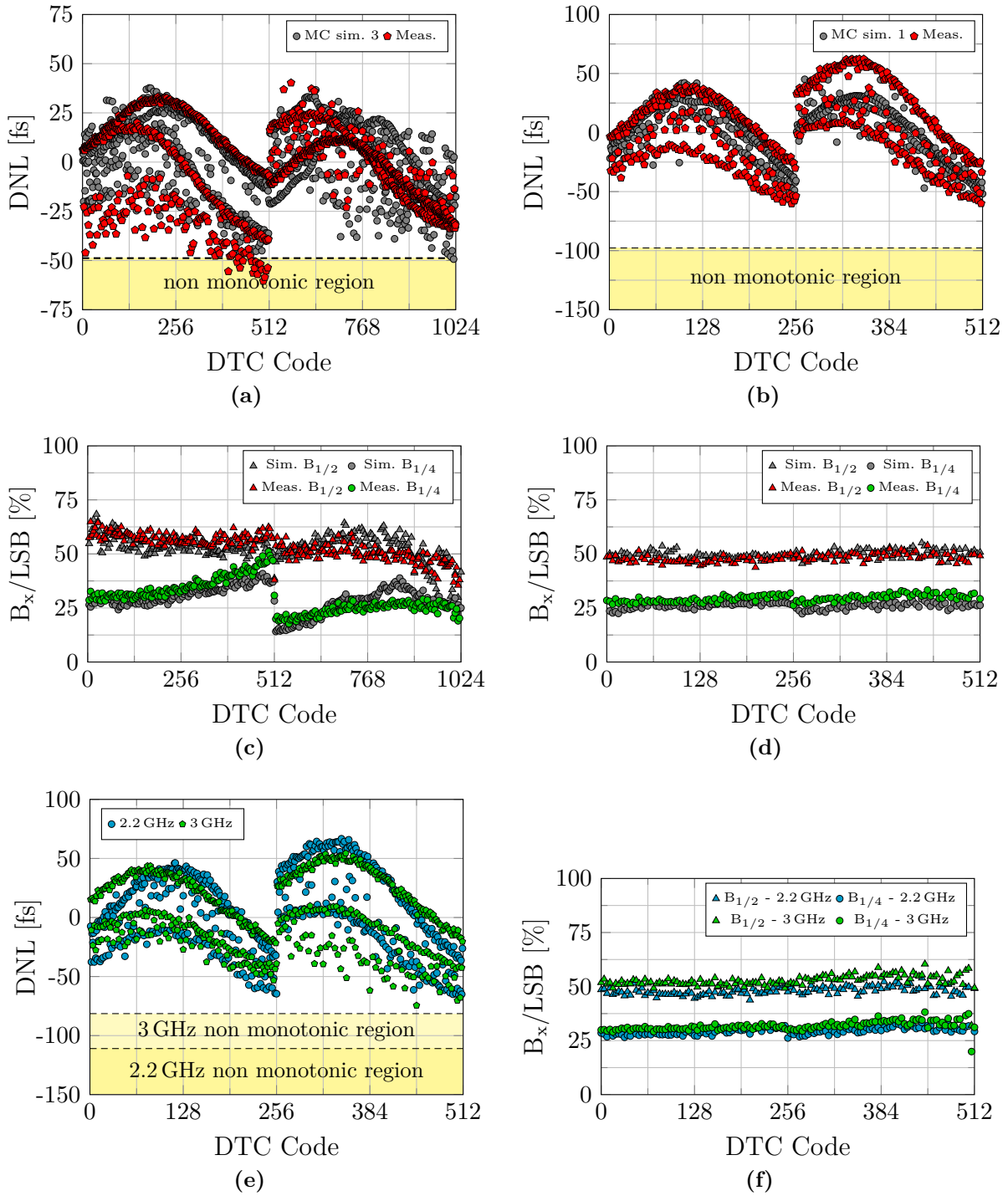


Figure 5.7 – Linearity of DCEI²'s binary bits:

- (a) DCEI² V1 with $t_{d,LSB} = 48.8$ fs at $f_{out} = 2.5$ GHz,
- (b) DCEI² V2 with $t_{d,LSB} = 97.7$ fs at $f_{out} = 2.5$ GHz,
- (c) binary step size related to respective LSB step for (a),
- (b) binary step size related to respective LSB step for (b),
- (e) measured DCEI² V2 at $f_{out} = 2.2$ GHz and $f_{out} = 3$ GHz, and
- (f) binary step size related to respective LSB step for (e).

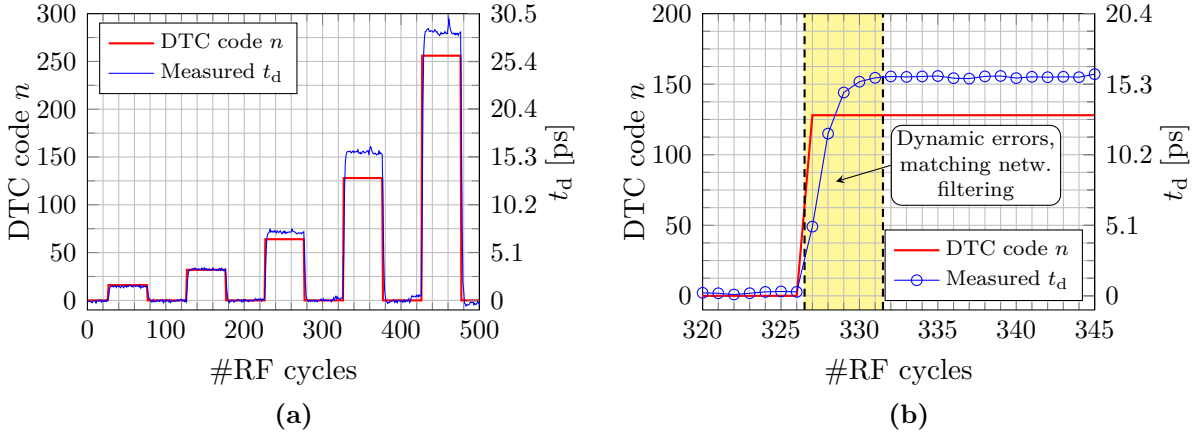


Figure 5.8 – DTC code sequence and measured DTC output delay at $f_{\text{out}} = 2.4 \text{ GHz}$: (a) Code sequence triggering the five DCEI² MSBs, and (b) zoom on a single code transition $n : 0 \rightarrow 128$.

the first cycles after a code transition and the average t_d of the target code. However, looking at the critical first five codes after a code transition, as plotted in Fig. 5.8(b), reveals dynamic errors that are magnitudes higher than expected from circuit simulation.

To understand their origin, the full signal chain needs to be investigated, including the interconnection of DTC and the measurement instruments. Fig. 5.9 shows the on-chip components that interconnect DTC and the chip output pin. As the DTC has comparably weak output drivers, the pseudo-differential signal is amplified at the chip output. A transformer is used as matching network to transform the amplifier’s output impedance to 50Ω , and to convert the pseudo-differential DTC output to a single ended chip output. This output pin is directly connected to the spectrum analyzers or oscilloscopes. The transformer is the key building block to understand the strong dynamic errors plotted in Fig. 5.8(b): the matching network is optimized for $f_{\text{out}} = 2.4 \text{ GHz}$ and shows a bandpass characteristic. A phase jump introduces high frequency components to the DTC output signal, which are attenuated in the matching network. Overall, the measured output is a superposition of quasi-static and dynamic DTC nonlinearity, plus the filtering effect the matching network introduces on the measured signal, where the filtering effect strongly dominates.

Before discussing the measurement results for the single compensation circuits, it should be noted that the instrument noise of the oscilloscope limits the overall measurement accuracy. A sampling rate of 40 GS/s leads to 16.67 samples per RF period, from which

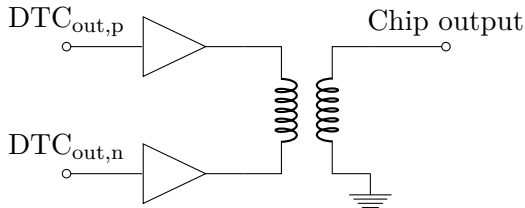


Figure 5.9 – Output stage of DTC test chip for 50Ω impedance matching .

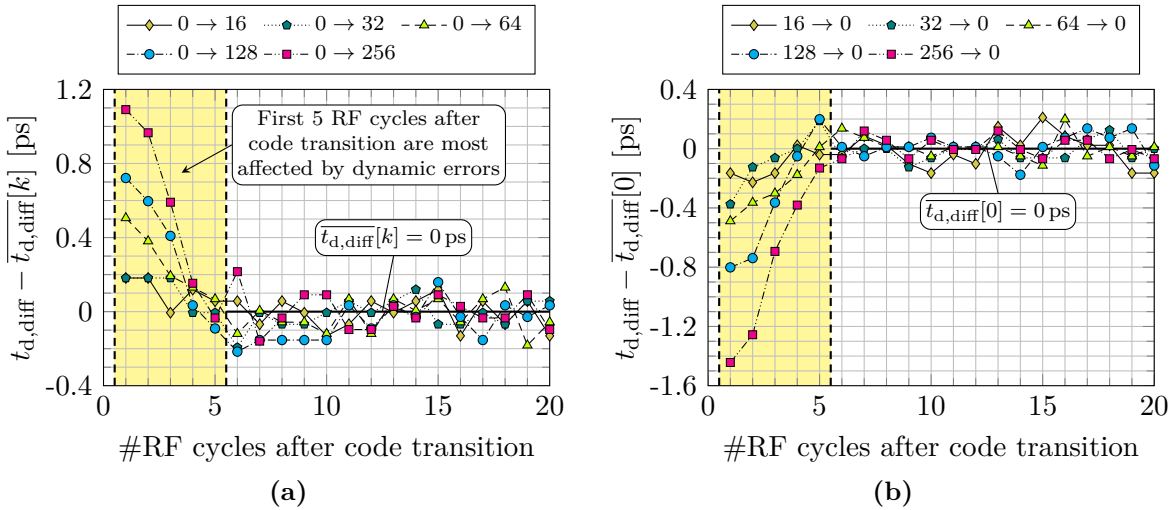


Figure 5.10 – Dynamic error difference due to active DCEI² dynamic effects compensation for code jumps (a) $n : 0 \rightarrow k$ (according to (5.2)) and (b) $n : k \rightarrow 0$ (according to (5.3)), for $k \in \{16, 32, 64, 128, 256\}$.

the exact zero crossing position is calculated. Even after averaging several measurements within the instrument, the noise floor of the DTC output delay is still in the domain of up to $\pm 2 t_{d,\text{LSB}}$ of the tested DCEI² V2 based DTC. The spectrum analyzer measurement method delivers high resolution well below $t_{d,\text{LSB}}$ of the 13 bit DCEI² V1 based DTC, but cannot be used to measure fast transient code changes due to the limited bandwidth (40 MHz) and sampling speed of the spectrum analyzer.

Section 5.4.1 and 5.4.2 prove the functionality of the compensation circuits for DCEI² and MMD code activity, respectively. Afterwards Section 5.4.3 discusses the limitations, that prevent further dynamic error measurements, as e.g. random phase modulation.

5.4.1 DCEI² Dynamic Effects Compensation

As mentioned in Section 4.4, the compensation current source i_{comp} from Fig. 4.7 is implemented as five parallel current sources, that are digitally enabled or disabled based on the DCEI² code $n_{\text{DCEI}^2,7:3}$. The code sequence plotted in Fig. 5.8(a) toggles each of these bits separately to test the individual current sources. Due to the above stated reasons, the dynamic errors cannot be measured directly. Therefore, measurements with active and inactive compensation are compared by calculating the delay difference of both cases. As the dynamic errors are expected to be in the order of multiple $t_{d,\text{LSB}}$, they should stand out regardless of the output matching network's filtering effect. The dominant component in the difference is expected to be the difference of dynamic errors due to active DCEI² compensation. The code-steps $n : 0 \rightarrow k$ and $n : k \rightarrow 0$ are tested for $k \in \{16, 32, 64, 128, 256\}$.

As dynamic errors are expected to decay in the order of a few RF cycles, the first 20 RF cycles after each code transition are compared. To enable a direct comparison of different code jumps, all measurements are aligned to their average delay $\bar{t}_d[k]$ and $\bar{t}_d[0]$ for the code steps of $n : 0 \rightarrow k$ and $n : k \rightarrow 0$, respectively. The average is calculated as mean

value of RF cycle 6-20 after the code transition to reduce the measurement noise. Then, the difference between both aligned measurements with active and inactive compensation is calculated to

$$t_{d,\text{diff}} - \overline{t_{d,\text{diff}}}[k] = (t_d|_{\text{comp,on}} - \overline{t_d}[k]|_{\text{comp,on}}) - (t_d|_{\text{comp,off}} - \overline{t_d}[k]|_{\text{comp,off}}), \text{ and} \quad (5.2)$$

$$t_{d,\text{diff}} - \overline{t_{d,\text{diff}}}[0] = (t_d|_{\text{comp,on}} - \overline{t_d}[0]|_{\text{comp,on}}) - (t_d|_{\text{comp,off}} - \overline{t_d}[0]|_{\text{comp,off}}), \quad (5.3)$$

which is automatically aligned to $\overline{t_{d,\text{diff}}}[k] = 0$ s and $\overline{t_{d,\text{diff}}}[0] = 0$ s. The comparison for positive and negative code jumps is plotted in Fig. 5.10(a) and (b), respectively. The yellow highlighted region indicates the RF cycles that are mostly affected by dynamic errors. The test case in Fig. 5.10(a) transitions from a code with low to a code with higher current consumption. According to the theory from Chapter 4, the DTC's supply voltage drops due to the higher current, and the LDO regulates the initially higher voltage to the lower one according to the bandwidth of the fast LDO loop (compare the LDO output voltage for these code changes, plotted in Fig. 4.8(a)). Due to the regulation activity, the DTC has a higher supply voltage for the first RF cycles after the code transition, leading to a shorter delay t_d and introducing a negative dynamic error. Thus, the active compensation targets to reduce the dynamic effect, leading to a positive difference according to (5.2), which is visible in the measurements plotted in Fig. 5.10(a), proving the functionality of the DCEI² compensation circuit. Vice versa, the dynamic errors for the test cases in Fig. 5.10(b) are expected to be positive, leading to an expected negative difference due to active compensation according to (5.3), which is again visible in the measurement results. It has to be noted, that a dynamic error reduction due to this graph can also mean an over-compensation, as the dynamic effects cannot be distinguished from the filtering effect of the matching network. As the measurements prove that the compensation influences the DTC in the correct direction, an optimum setting for the compensation is found with the optimum current for the current sources. Therefore, a digital control word allows tuning of the current sources, to increase or decrease their compensation strength. However, the optimum tuning for this control word cannot be determined by the discussed test.

5.4.2 MMD Dynamic Effects Compensation

The correct operation of the MMD compensation circuit is verified by triggering code transitions that lead to MMD activity. The dynamic errors expected from circuit simulations are above the level of $t_{d,\text{LSB}}$, leading to the assumption that they stand out when the code transition, that excites the MMD, is only of the magnitude of 1 LSB. To test all implemented MMD compensations circuits, the code transitions $n : 511 \rightarrow 512$, $n : 512 \rightarrow 511$, $n : 1023 \rightarrow 1024$, and $n : 1024 \rightarrow 1023$ are tested. Other possible code transitions, such as $n : 2047 \rightarrow 2048$, lead to identical effects on MMD and compensation and do not need to be tested (the DTC nonlinearity is periodic with 1024 codes). As introduced in the previous section, all measurements are aligned to the average delay $\overline{t_{d,\text{diff}}}[k]$ of the code k after the code transition.

A reduction of dynamic errors is clearly visible in Fig. 5.11(a)-(c). The uncompensated DTC shows clearly visible dynamic errors especially in the first two RF cycles after the code change. However, it is not possible to quantify the reduction of dynamic errors, as the measured delay of the compensated DTC is below the instrument noise floor. All three

cases support the theory of dynamic effect mechanisms stated in Chapter 4. A positive code step (Fig. 5.11(a) and (c)) leads to an instantaneously larger duty cycle, resulting in smaller LDO load current and a positive distortion on the LDO output voltage. This reduces the DTC delay for a short time period, and leads, therefore, to negative dynamic errors. The same effect leads to positive dynamic errors due to an instantaneously shorter duty cycle for the negative code step in Fig. 5.11(b). It is not possible to make any statement about MMD division-by-5 (Fig. 5.11(d)) based on this measurement data, as both measurements, with active and inactive compensation, are below the instrument noise floor.

As positive code steps lead to negative delay, a look on the DNL clarifies the non-monotonicity at these points. The DNL of $n : 511 \rightarrow 512$ as plotted in Fig. 5.11(a) and (b) is according to (2.13) $\text{DNL}[512] = -204 \text{ fs}$ and $\text{DNL}[512] = -200 \text{ fs}$, respectively. This is in accordance with $\text{DNL}[512] = -193 \text{ fs}$ measured in the quasi-static DTC linearity measurements from Fig. 5.5(b). The DNL of $n : 1023 \rightarrow 1024$ according to Fig. 5.11(c) and (d) is $\text{DNL}[1024] = -235 \text{ fs}$ and $\text{DNL}[1024] = -230 \text{ fs}$, respectively, and slightly deviates

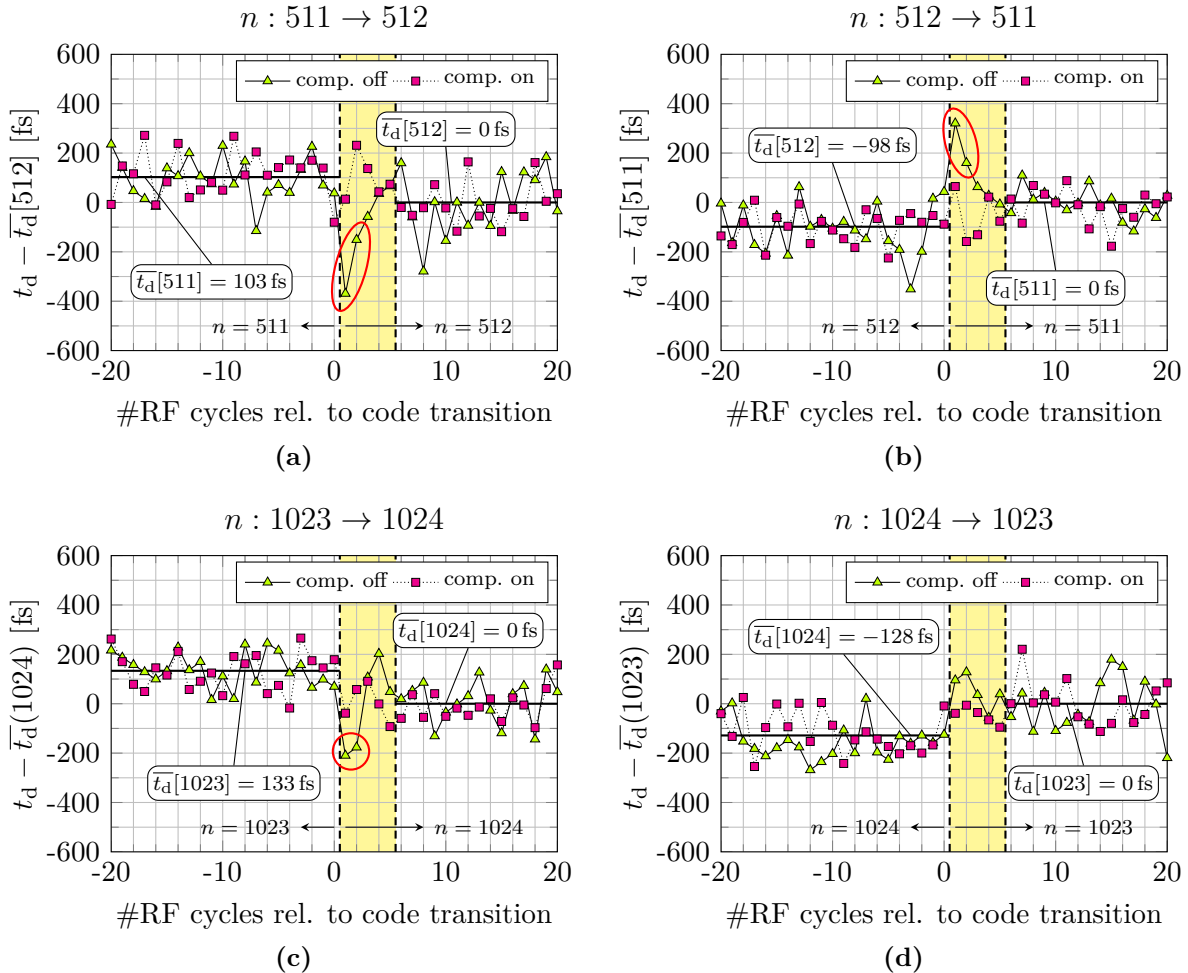


Figure 5.11 – Dynamic error comparison for active/inactive MMD dynamic effects compensation for (a) MMD LSB transition $0 \rightarrow 1$, (b) MMD LSB transition $1 \rightarrow 0$, (c) division-by-3, and (d) division-by-5.

from the quasi-static measurements result of $\text{DNL}[1024] = -276$ fs. The difference is explained by the instrument noise of the oscilloscope and the different output frequency of $f_{\text{out}} = 2.4$ GHz for the dynamic error measurements and $f_{\text{out}} = 2.496$ GHz for the quasi-static nonlinearity measurement.

5.4.3 Dynamic Error Measurement Limitations

The test cases of discrete code transitions presented so far prove the correct operation of the compensation circuits, but have only limited significance for realistic DTC applications. The LDO's fast control loop is settled prior to the code transition, which is not the case for realistic DTC programming scenarios. For the frequency synthesis test cases, the oscilloscope measurements have too high noise floor, even after averaging of several measurements. The effect of the dynamic error compensation vanishes below, leaving no indicator on dynamic errors. On the positive side, this confirms the assumption of low dynamic errors in frequency synthesis DTC applications, as discussed in Section 4.1. The random phase modulation test cases can be measured with this method, but the strong filtering effect of the on-chip matching network strongly dominates the measured DTC output delay. Even the extraction of the INL from the randomly modulated DTC output is, unlike for the frequency synthesis test cases, not possible. Thus, a verification of the simulated test cases from Section 4.4 with measurements is not possible for the currently implemented matching network.

5.5 Summary and Conclusion

The measurements presented in this chapter prove that all three test chips are fully functional and free of design bugs. A spectrum analyzer based DTC measurement method with femtosecond accuracy is the basis for all quasi-static DTC nonlinearity measurements. The DTC is excited with a code ramp, where each code is active for a sufficiently long time to let all dynamic effects in the system settle. When the chip output signal is demodulated to the baseband domain and evaluated for its phase with a Rohde & Schwarz FSV signal analyzer with *Analog Demod* software option, the programmed phase ramp can be directly measured. Based on this measurement, the DTC nonlinearity is extracted through post-processing of the raw measurement data.

A direct comparison of the quasi-static DNL and INL of the DCEI, CF-DCEI, and DCEI² V2 based DTCs is plotted in Fig. 5.12(a) and (b), respectively. The measurements are done for the full DTC code range $n \in [0, N]$. The DNL shows, that MMD transitions are critical in all of the PI designs. The DCEI shows DNL spikes of up to 810 fs at MMD and MUX+DEL transitions. These spikes could be reduced for the CF-DCEI and the DCEI² V2 to 305 fs and -275 fs, respectively. Through layout optimizations, the CF-DCEI shows the critical spikes only at MMD transitions and not additionally at MUX+DEL transitions, as in the DCEI (e.g. the DCEI spike at $n/N = 0.0625$ is gone for the CF-DCEI). Through the increased interpolation range of the DCEI², the MUX+DEL stage could be removed from its DTC architecture, leaving the only possible critical transitions those of the MMD. However, the DNL spikes of the DCEI² V2 based DTC are still well above the targeted DTC resolution, and lead in this specific case to non-monotonicity, even if the DCEI² V2

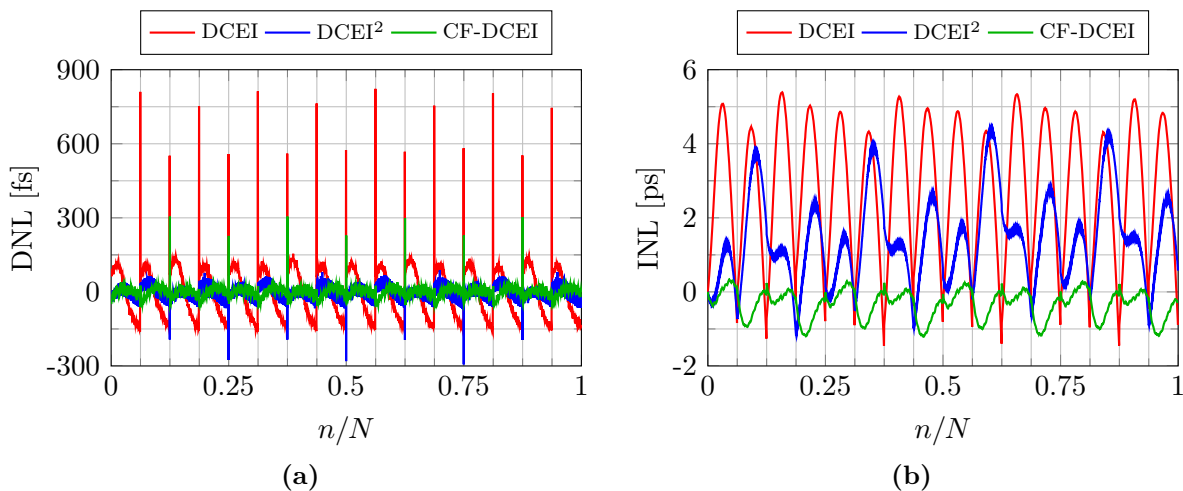


Figure 5.12 – Comparison of (a) DNL and (b) INL between: DCEI ($f_{\text{out}} = 2$ GHz), CF-DCEI ($f_{\text{out}} = 2$ GHz), and DCEI² V2 ($f_{\text{out}} = 2.496$ GHz).

itself shows a monotonic behavior. The INL comparison in Fig. 5.12(b) shows a significant INL reduction of CF-DCEI compared to the DCEI. The DCEI² V2 shows a peak INL at almost the same level as the DCEI. The expectations from circuit simulations were a lower INL, due to higher RC time constant and smaller Δt at the 2nd interpolation node. However, p/n mismatch of the 10 GHz LO-distribution network from PLL to DTC leads to an increased INL of ~ 1.2 ps at e.g. $n/N = 0.125$, where it should be ideally close to 0 ps due to the MMD design. Table 5.1 summarizes key performance parameters of all three designs to facilitate further comparison, and compares the measurements of the full DTC to the simulation results of the PIs.

A comparison of simulated and measured binary bit nonlinearity revealed the high accuracy of the introduced DTC measurement method. Measurements of DTC phase steps with single digit femtosecond precision matched well to circuit simulation results, with deviations expected from PVT variations and mismatch. The DCEI² V2's design target of a monotonic binary bit behavior was successfully validated for the full frequency range of 2.2–3 GHz. As expected from circuit simulations, the DCEI² V1 showed non-monotonicity at binary bit level. However, with an outstanding resolution of $t_{\text{d,LSB}} = 48.8$ fs at 2.5 GHz it is the DTC with the highest resolution published so far.

A comparison of measured, simulated and modeled CF-DCEI linearity shows excellent matching, proving the high accuracy of the analytical CF-DCEI model. Also the tuning of the 1st DCEI² interpolation is well approximated with the numerical model. The accuracy of both models proves, that no significant INL contributor was missed during circuit modeling.

Measurements of the dynamic error compensation circuit of the DCEI² V2 based DTC proves that the DCEI² and MMD compensation are both functional and that they reduce dynamic errors. Due to the measurement noise floor of the used 40 GS/s high speed sampling oscilloscope, a quantification of dynamic errors is not directly possible. Dynamic errors related to MMD transition are clearly visible above the instrument noise floor, while they are below the floor for active MMD dynamic effects compensation. Furthermore,

Table 5.1 – Comparison of measurement results from DCEI, CF-DCEI, and DCEI² based DTC, and comparison to PI simulation results (without further DTC stages).

		DCEI	CF-DCEI	DCEI ² V1	DCEI ² V2
Frequency		2 GHz	2 GHz	2.5 GHz	2.2–3 GHz
Resolution	$t_{d,LSB}$	244 fs	244 fs	48.8 fs	81.4–111.0 fs
	[bit]	11 bit	11 bit	13 bit	12 bit
DNL _{rms}	Sim. PI	89.82 fs	23.09 fs	19.81 fs	24.13 fs ¹⁾
	Meas. DTC	110.34 fs	25.32 fs	30.96 fs	31.61 fs ¹⁾
DNL _{peak}	Meas. DTC	810 fs	305 fs	209 fs	–275 fs ¹⁾
INL _{max}	Sim. PI	5.05 ps	0.93 ps	2.89 ps	2.57 ps ¹⁾
	Meas. DTC	5.39 ps	1.21 ps	3.62 ps	4.38 ps ¹⁾

¹⁾ $f_{out} = 2.5$ GHz

especially the polarity of the measured dynamic errors support the theory of the dynamic effects mechanisms introduced in Chapter 4. However, the dynamic error measurements are limited by filtering effects introduced by the on-chip output matching network. While the filtering effects prohibit the re-production of the simulated random phase modulation test cases from Chapter 4, the instrument noise floor prohibits the re-production of the frequency synthesis test cases.

6 Conclusion and Outlook

The research on design and implementation of DTC architectures gained popularity during the last decade, especially in the application field of frequency synthesis. Fig. 6.1 shows the number of DTC related publications over the last two decades, either presenting a DTC architecture or an application that incorporates a DTC, that were cited in the present thesis. The increasing number reflects that DTC enabled applications are attractive alternatives to their conventionally implemented counterparts, and that research on DTC architectures and applications is a hot topic with increasing attention from academia and industry. One of the major advantages is the possibility of a fully digital circuit architecture that benefits from process technology scaling, thus following the common trend in semiconductor industry.

The present thesis focused on PI based DTCs, as PIs are attractive DTC fine tuning circuits that bring advantages such as a well defined fine tuning range and low power consumption. The motivation of advancing PI circuit architecture and implementation for gigahertz domain DTC designs led to the following research objectives: 1) PI linearization and analysis of the mechanisms behind PI nonlinearity, as the high nonlinearity is the major drawback of conventional PIs; 2) increase of PI resolution; 3) increased operation frequency for high resolution PIs; and 4) the investigation of dynamic effects in DTCs, and mitigation of the resulting dynamic errors on the DTC output signal's phase. Two novel PI architectures, the CF-DCEI and DCEI², were developed and implemented in three test chips. The implementations base on an existing three-stage reference DTC design, consisting of an MMD as ultra coarse and a DCEI as fine tuning, plus an intermediate MUX+DEL coarse tuning stage to reduce the phase spacing at the DCEI input. The newly developed PI based DTCs re-use the MMD and MUX+DEL stage as DTC coarse tuning to enable direct comparison of CF-DCEI and DCEI² performance to the DCEI.

The reference DCEI has the PI advantages mentioned above, but shows high nonlinearity.

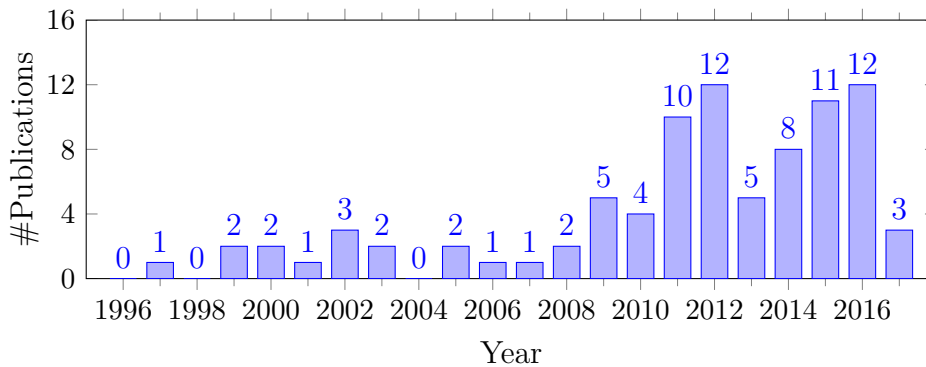


Figure 6.1 – Number of cited DTC publications per year over the last two decades until 02/2017.

PI models developed in the present thesis identified the shoot-through current as major nonlinearity source in conventional PI designs (e.g. the DCEI), which is in accordance with the literature. Therefore, the novel CF-DCEI architecture was developed to prevent shoot-through currents with additional control logic inside the unit interpolation cells, while still providing a high resolution of 7 bit at 2 GHz operation frequency. The interpolation cells are complemented with retention cells, allowing a linearized interpolation on rising and falling edges, thus rendering a PI reset unnecessary which needs to be used in linearized phase interpolators published to date. The linearization allows to reduce the peak INL by a factor of five, compared to the DCEI. Analytical circuit modeling revealed that the ratio of phase spacing Δt to the minimum rise time at the interpolation node $t_{\text{int},0}$ is the remaining major source of nonlinearity in contention-free PIs. Theoretically, $t_{\text{int},0} \geq \Delta t$ can lead to ideal linearity. Due to high nonlinearity of PIs, switched capacitor based fine tuning circuits are usually the preferred fine tuning implementation. As the CF-DCEI brings several advantages of PI architectures over switched capacitor fine tuning, such as a well defined tuning range and a constant current consumption for different codes, it is the preferred implementation for the investigated interpolation range of 31.25 ps due to its competitive nonlinearity. The drawback of the CF-DCEI is the high current consumption, originating from the linearization logic that needs to operate at $2f_{\text{out}}$, which increased the current consumption by 57.1% compared to the DCEI.

To provide a PI with low power consumption and wide interpolation range, which operates in the frequency range of 2.2–3 GHz, the novel DCEI² architecture was developed. A two-points interpolation allows a doubling of the interpolation range compared to the DCEI, and consequently the removal of the intermediate 1 bit MUX+DEL stage. The loss of 1 bit resolution is re-gained by the intrinsic architecture of the DCEI² unit cells, that allows to implement $(k_{\text{PI}} + 1)$ bit resolution with an array of $2^{k_{\text{PI}}}$ unit cells. The advantage is a lower phase noise, as the noise contributor MUX+DEL is removed from the signal path, and lower complexity and power consumption due to the reduction to a two-stage DTC design. The power consumption is reduced by 26.3% compared to the reference DCEI based DTC. To increase the resolution while providing a competitive current consumption, binary weighted DCEI² unit cells were added to the PI array to extend the resolution by 2 bit. The resulting DCEI² designs have a total resolution of 9–10 bit, surpassing all previously published PIs, which show only resolutions of up to 5 bit [2, 85]. As PIs have slight differences to conventional DACs, the novel and correct implementation of binary bits for phase tuning without discontinuities in the phase steps was presented and implemented. Test chip measurements proved monotonous phase interpolation with a resolution of 81.4 fs at 3 GHz.

The main challenge for further increase of the DCEI²'s operation frequency is the timing of the select signals. The select signals need to be applied to the unit cells in the time between two subsequent edges of the analog signal ($\cong 0.5T_{\text{out}} = \frac{1}{2f_{\text{out}}}$), which reduces for higher frequency. The major drawback of the DCEI² architecture is the dependency of the current consumption on the programmed DTC code. The resulting re-modulation of the supply voltage distorts the DTC's output phase, and introduces dynamic errors on the phase which occur in addition to the static DTC nonlinearity.

To understand the dynamic effects that lead to dynamic errors, the underlying mechanisms were elaborately analyzed. In general, dynamic errors originate either from poor timing of the control logic or from supply voltage induced effects as response to DTC load

Table 6.1 – Comparison of developed DTCs to recently published gigahertz domain DTC designs.

	CF-DCEI	DCEI ² V1	DCEI ² V2	JSSC 2013 [2]	JSSC 2012 [7]
Technology	28 nm CMOS	28 nm CMOS	28 nm CMOS	65 nm CMOS	32 nm CMOS
Supply	1.1 V	1.1 V	1.1 V	1.2 V	1.05 V
Frequency	2.0 GHz	2.5 GHz	2.2–3.0 GHz	0.1–1.5 GHz	2.4 GHz
Resolution	11 bit	13 bit	12 bit	8 bit	8 bit
$t_{d,LSB}$	244 fs	48.8 fs	81.4–111 fs	2.60–39.06 ps	1.63 ps
Range	2π (500 ps) (self-aligned)	2π (400 ps) (self-aligned)	2π (333–455 ps) (self-aligned)	2π (0.67–10 ns) (self-aligned)	2π (416.67 ps) (calibrated)
$ DNL_{max} $	305 fs	209 fs	275 fs	4.06 ps at 0.5 GHz	N/A
$ INL_{max} $	1.21 ps	3.62 ps	4.38 ps	10.4 ps at 0.5 GHz	2.93 ps
Power	19.8 mW	16.1 mW	14.6 mW (at 2.5 GHz)	4.3 mW (at 1.5 GHz)	N/A
Area	0.0091 mm ²	0.0052 mm ²	0.0046 mm ²	0.060 mm ²	0.100 mm ²

current variations from its periodic steady state. The latter effect leads to the most prominent dynamic errors, either introduced through the code dependent current consumption of the DCEI², or by DTC code changes that lead to modification of the output signal’s period and thus to an instantaneous change of the average current for a single DTC output cycle. The insights of this analysis enabled the design of a dynamic error compensation on supply voltage level. The fast control loop of an existing dual-loop LDO was extended with a dynamic effects compensation, that adapts the biasing to the DTC current consumption and injects a charge into the control loop for MMD transitions. Both compensation mechanisms use the knowledge of the expected DTC current from the programmed code sequence and operate on a logic rate of up to 3 GHz. Simulations show a reduction of dynamic errors by 40–66 % due to active compensation.

Table 6.1 compares key DTC performance metrics of the developed DTC designs to other recently published designs that operate in the gigahertz domain.

Based on the research of the present thesis and of the literature so far, two major topics for future DTC research can be derived: 1) DTC linearization through calibration; and 2) understanding and mitigation of dynamic errors.

1) *DTC linearization through calibration*: Many of the DTC applications introduced in Chapter 1 require a more linear DTC operation than most architectures can deliver by design. Therefore, digital pre-distortion of the DTC code is used to linearize the system [27, 59, 76]. However, measurement of the DTC nonlinearity is a challenge. Several authors already investigated the extraction of DTC nonlinearity from regulation loop activity of a closed loop-system (e.g. a PLL control loop) [27, 59], or measurement of the DTC nonlinearity with a calibration TDC for open-loop systems [13, 76]. However, the trend to increased DTC resolution imposes strict specifications on the calibration system, as it should have a similar or higher resolution than the implemented DTC. For the example of the DCEI² V1 DTC with $t_{d,LSB} = 48.8$ fs, no TDC in this resolution domain has been reported so far. Therefore, the targeted DTC resolution needs to be traded off between the physically implementable resolution and the performance of the integrated calibration engine.

2) *Understanding and mitigation of dynamic errors*: The dynamic error analysis from

Chapter 4 showed that dynamic errors can have levels of multiple times the DTC resolution $t_{d,LSB}$. Most applications program the DTC with constant code or a periodic code ramp, and thus generate low dynamic errors. However, applications as outphasing or polar transmitters apply random codes to the DTC, and are therefore more susceptible to dynamic effects. Future developments should focus also on lowering dynamic effects, for instance through implementation of designs with constant current consumption for all codes. The DCEI² compensation method proved that a more constant supply significantly reduces dynamic errors. Additionally, modulation data can lead to an effective INL that differs from a statically measured one, which needs to be considered during DTC calibration.

In conclusion, the present thesis proved feasible DTC designs for sub 100 fs resolution with small area, wide operation frequency range, and all advantages of divider and phase interpolator based DTC tuning stages. Further research on calibration and compensation of static and dynamic nonlinearity can lead to successful integration in application that benefit from high frequency and high accuracy, such as all digital frequency synthesizers.

Appendices

A DCEI Nonlinearity Model

The DCEI nonlinearity model is based on the piecewise defined equivalent circuit from Fig. 3.5. The current sources in this circuit relate to the drain current $I_{D,n/p}$ of the NMOS and PMOS devices in the unit interpolation cells and are described by the Shichman-Hodges transistor model [96]. Two equations describe $I_{D,n/p}$ for the linear and saturation region of the transistors:

$$I_{D,n/p,\text{lin}} = \pm \mu_{n/p} C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} \left((V_{\text{GS}} - V_{\text{th},n/p}) V_{\text{DS}} - \frac{V_{\text{DS}}^2}{2} \right), \text{ and} \quad (\text{A.1})$$

$$I_{D,n/p,\text{sat}} = \pm \frac{1}{2} \mu_{n/p} C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} (V_{\text{GS}} - V_{\text{th},n/p})^2 (1 + \lambda_{n/p} V_{\text{DS}}), \quad (\text{A.2})$$

where the NMOS and PMOS drain currents have the positive and negative sign, respectively. If the channel length modulation factor $\lambda_{n/p}$ is neglected, thus set to zero, the saturation drain current only depends on constant parameters. For simplification of further calculation, this current is abbreviated as

$$I_{D,\text{sat},0} = \pm \frac{1}{2} \mu_{n/p} C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} (V_{\text{GS}} - V_{\text{th},n/p})^2. \quad (\text{A.3})$$

The drain current in the saturation region can then be re-written as

$$I_{D,\text{sat}} = I_{D,\text{sat},0} (1 + \lambda_{n/p} V_{\text{DS}}). \quad (\text{A.4})$$

The Shichman-Hodges transistor model is not valid for modern deep sub-micron devices. However, for digitally controlled transistors only the on-case with $V_{\text{GS}} = \pm V_{\text{DD}}$ and the off-case with $V_{\text{GS}} = 0$ are of interest. A $V_{\text{DS}} \rightarrow I_{\text{D}}$ transfer function is fitted to simulation results of the on-case, while $I_{D,\text{off}} = 0$ is assumed for the off-case. If the technology parameters of (A.1) and (A.2) are not known beforehand, they can be extracted from the simulated transistor transfer function according to [88, pp. 744-754].

To determine the DCEI nonlinearity, the threshold crossing time $t_d[n]$ of $V_{\text{int}}(t_d[n]) = V_{\text{th,inv}}$ needs to be extracted for the piecewise defined equivalent circuit of Fig. 3.5. This means the charging waveform of C_{int} from $V_{\text{int}} = 0 \rightarrow V_{\text{DD}}$ has to be calculated and evaluated for the threshold crossing point $V_{\text{th,inv}}$. If $t_d[n]$ is known for $0 \leq n \leq N$, TF, DNL, and INL can be calculated. The following calculations are on the example of a rising interpolation. The falling interpolation can be analyzed in an analogous manner.

The capacitance C_{int} at the interpolation node is charged with the current i_{C} . The charging current is a superposition of the transistor currents and depends on the DCEI code word n . It is piecewise defined according to the model regions in Fig. 3.5(a) and (b) as

$$i_{\text{C},t < \Delta t} = (N - n)I_{D,p} - nI_{D,n}, \text{ and} \quad (\text{A.5})$$

$$i_{\text{C},t \geq \Delta t} = NI_{D,p}. \quad (\text{A.6})$$

The voltage at this node is then determined from the ordinary differential equation (ODE) of the charged capacitor

$$\frac{dV_{\text{int}}(t)}{dt} = \frac{1}{C_{\text{int}}} i_{\text{C}}(t). \quad (\text{A.7})$$

As variation of $V_{\text{int}}(t)$ leads to variation of V_{DS} for NMOS and PMOS devices, i_{C} is also $V_{\text{int}}(t)$ dependent, thus t dependent. According to (A.5), $i_{\text{C}}(t)$ has components of NMOS and PMOS drain current. For $V_{\text{int}}(t) \leq V_{\text{th,n}}$, the NMOS devices are in the linear region, while the PMOS devices are in saturation. Consequently, $i_{\text{C}}(t)$ has a linear component $V_{\text{int}}(t)$ due to the PMOS devices, and a quadratic component $V_{\text{int}}(t)^2$ due to the NMOS devices. The resulting ODE for node V_{int} is a Riccati equation of the form

$$\frac{dV_{\text{int}}(t)}{dt} = aV_{\text{int}}(t)^2 + bV_{\text{int}}(t) + c, \{a, b, c \in \mathbb{R}\} \quad (\text{A.8})$$

and can be solved analytically. Even though an analytical solution is possible, several reasons speak against it: first, solutions of (A.8) are of a complicated form; second, a , b , and c depend on CMOS technology parameters and are piecewise defined, depending on the operation region of the transistors; and third, the overall solution with inserted technology parameters provides no intuitive insight in the interpolation process. Therefore, a numerical solution is preferred, as it simplifies the model implementation and leads to identical results.

So far, the model equations assume that devices switch from zero drain current directly to their saturation drain current at $t = 0$ and $t = \Delta t$, triggered by the input signals. However, limited rise/fall-time $t_{\text{r,f}}$ of the input signals influences PI nonlinearity [95]. To extend the model for $t_{\text{f}} > 0$, the PMOS drain current is assumed to rise linear during the fall time of the input signals t_{f} as proposed in [95]. Concurrently, the NMOS current is falling linear, as the NMOS device is turned off while the PMOS device is turned on. An exemplary extension of (A.5) for $t_{\text{f}} > 0$ leads to

$$i_{\text{C},t < \Delta t} = (N - n) \left(I_{\text{D,p}} \frac{t}{t_{\text{f}}} - I_{\text{D,n}} \left(1 - \frac{t}{t_{\text{f}}} \right) \right) - nI_{\text{D,n}}. \quad (\text{A.9})$$

For a complete set of equations for $i_{\text{C}}(t)$, three major cases need to be distinguished: $t_{\text{f}} = 0$, $t_{\text{f}} < \Delta t$, and $t_{\text{f}} \geq \Delta t$. The piecewise defined equations for all of them are collected in Table A.1. Numerical integration of (A.7) with the piecewise defined i_{C} from Table A.1 leads to the charging waveforms at the interpolation node as plotted in Fig. 3.6(a), which are then evaluate for their crossing of $V_{\text{th,inv}}$ to extract $t_{\text{d}}[n]$.

A.1 Interpolation Time Constant $t_{\text{int},0}$

According to the literature, one main contributor to PI nonlinearity is the ratio $\Delta t / \tau_{\text{int}}$ [16, 95], where τ_{int} is the RC time constant of V_{int} . As it is not straight forward to extract τ_{int} from circuit simulations, the minimum rise time $t_{\text{int},0}$ of $V_{\text{int}} : 0 \rightarrow V_{\text{th,inv}}$ is used as an equivalent measure. The main advantage is an easy extraction from transient circuit simulation. It is extracted from the model equations by solving (A.7) for $n = 0$. Assuming

$V_{\text{int}}(t = 0) = 0 \text{ V}$, $\lambda_p = 0$, and $t_f = 0$, the resulting ODE

$$\frac{dV_{\text{int}}(t)}{dt} = \frac{NI_{\text{D,sat},0}}{C_{\text{int}}} \quad (\text{A.10})$$

is solved to

$$V_{\text{int}}(t) = \frac{NI_{\text{D,sat},0}}{C_{\text{int}}}t. \quad (\text{A.11})$$

Inserting $V_{\text{int}}(t_{\text{int},0}) = V_{\text{th,inv}}$ and solving for $t_{\text{int},0}$, which is the minimum time needed to charge V_{int} to $V_{\text{th,inv}}$, results in the final solution

$$t_{\text{int},0} = \frac{C_{\text{int}}V_{\text{th,inv}}}{NI_{\text{D,sat},0}}, \quad (\text{A.12})$$

which is similar to equations used to estimate CMOS inverter delays [97, pp. 199-202]. Eq. (A.12) gives a first overview on the parameters that eventually influence the time constant related nonlinearity. While N relates to the number of interpolation cells, determined by the DCEI's targeted resolution, $I_{\text{D,sat},0}$ is determined by the sizing of their transistors. As the DCEI is designed without a dedicated load capacitor, C_{int} relates to the parasitic capacitance of the interpolation node. However, C_{int} can be increased by adding a capacitor to V_{int} . The last influencing parameter is $V_{\text{th,inv}}$, relating to the threshold voltage of the DCEI output buffers, which evaluate the interpolation. It depends on the chosen devices, such as low- V_{th} or high- V_{th} transistors as available in many technologies, and the topology of the output buffer, which is implemented as inverter. The output buffer could also be implemented as a Schmitt-Trigger to shift $V_{\text{th,inv}}$. As the calculation assumed $\lambda_p = 0$ and $t_f = 0$, ODE (A.10) needs to be extended to cover the influence of both parameters.

A comparison between the DCEI's and CF-DCEI's equivalent circuits in Fig. 3.5 and Fig. 3.13, as well as their equations for $t_{\text{int},0}$ (A.12) and (B.5), shows that both circuits are equivalent for $n = 0$. Therefore, the calculation of the case $n = 0$ for $\lambda_p \neq 0$ and $t_f > 0$ are identical. The following equations are derived in Appendix B in the context of the CF-DCEI, as the CF-DCEI's nonlinearity has a higher sensitivity towards these effects. The results are cited here to discuss the correct extraction of $t_{\text{int},0}$ from transient circuit simulations

For $\lambda_p = 0$, $I_{\text{D,p,sat}} = I_{\text{D,sat},0}$ is at its minimum value, as it is not increasing for increasing V_{DS} . Approximating the ODE solution for $\lambda_p \neq 0$ with a Taylor extension, derived in Section B.1.2, leads to

$$t_{\text{int},0,\lambda \neq 0} \approx \frac{1}{(1 + 0.5\lambda_p V_{\text{th,p}})} t_{\text{int},0}, \quad (\text{A.13})$$

resulting in an effective smaller minimum rise time as λ_p and $V_{\text{th,p}}$ are both negative.

An interpolation model for $t_f > 0$ is derived in Section B.2, using a modified ODE that accounts for t_f . Assuming $0 < t_f < \Delta t$ and calculating t_{int} for $n = 0$ shows that it consists of two contributors:

$$\begin{aligned} t_{\text{int}}[n = 0] &= \frac{C_{\text{int}}V_{\text{th,inv}}}{NI_{\text{D,sat},0}} + \frac{t_f}{2} \\ &= t_{\text{int},0} + \frac{t_f}{2}. \end{aligned} \quad (\text{A.14})$$

Table A.1 – Piecewise defined DCEI ODEs.

Region of t_f	Region of t	i_C	
$t_f = 0$	$0 \leq t < \Delta t$	$(N - n)I_{D,p} - nI_{D,n}$	(A.15)
	$t \geq \Delta t$	$NI_{D,p}$	(A.16)
$t_f < \Delta t$	$0 \leq t \leq t_f$	$(N - n) \left(I_{D,p} \frac{t}{t_f} - I_{D,n} \left(1 - \frac{t}{t_f} \right) \right) - nI_{D,n}$	(A.17)
	$t_f < t < \Delta t$	$(N - n)I_{D,p} - nI_{D,n}$	(A.18)
	$\Delta t \leq t \leq (t_f + \Delta t)$	$(N - n)I_{D,p} - n \left(I_{D,p} \frac{t - \Delta t}{t_f} - I_{D,n} \left(1 - \frac{t - \Delta t}{t_f} \right) \right)$	(A.19)
	$t > (t_f + \Delta t)$	$NI_{D,p}$	(A.20)
$t_f \geq \Delta t$	$0 \leq t < \Delta t$	$(N - n) \left(I_{D,p} \frac{t}{t_f} - I_{D,n} \left(1 - \frac{t}{t_f} \right) \right) - nI_{D,n}$	(A.21)
	$\Delta t \leq t \leq t_f$	$(N - n) \left(I_{D,p} \frac{t}{t_f} - I_{D,n} \left(1 - \frac{t}{t_f} \right) \right) -$ $n \left(I_{D,p} \frac{t - \Delta t}{t_f} - I_{D,n} \left(1 - \frac{t - \Delta t}{t_f} \right) \right)$	(A.22)
	$t_f < t \leq (t_f + \Delta t)$	$(N - n)I_{D,p} - n \left(I_{D,p} \frac{t - \Delta t}{t_f} - I_{D,n} \left(1 - \frac{t - \Delta t}{t_f} \right) \right)$	(A.23)
	$t > (t_f + \Delta t)$	$NI_{D,p}$	(A.24)

While the effects of the λ factor cannot (or only insignificantly) be influenced by circuit design, t_f shows a major influence on t_{int} . Therefore, the extraction of $t_{\text{int},0}$ from transient circuit simulations requires to measure t_{int} and t_f .

A.2 DCEI Model Summary

Though a fully analytical solution of the DCEI interpolation ODE (A.7) is possible, its several drawbacks lead to a favorable solution with numerical methods. The interpolation waveforms are calculated numerically for all different code words n , and the resulting waveform is evaluated for its crossing time $t_d[n]$ of the threshold voltage $V_{\text{th,inv}}$. Afterwards, the TF, INL, and DNL are calculated with (2.9), (2.11), and (2.13), respectively.

For the waveform calculation, (A.7) is integrated numerically. The corresponding charging current $i_C(t)$ is piecewise defined for the different model configurations, summarized in Table A.1. The transistor drain currents are only denoted in their general form $I_{D,n/p}$, which does not indicate if the devices operate in their linear or saturation region. The detection of the region, and thus if (A.1) or (A.2) needs to be used for $I_{D,n/p}$ in the current time step of the numerical integration, can be implemented dynamically with the Shichman-Hodges model equations at each time step of the calculation. The full model for the discussed DTC is calculated in the order of seconds, enabling a fast evaluation of the INL for different DCEI design parameters.

B CF-DCEI Nonlinearity Model

B.1 First Order CF-DCEI Model Derivation

This appendix derives the CF-DCEI's nonlinearity from the circuit model of Fig. 3.13 for an exemplary interpolation on the rising edge, assuming ideal input signals with $t_f = 0$. The interpolation begins when the first input switches from V_{DD} to V_{SS} . Assuming it finishes when V_{int} crosses the subsequent inverters threshold voltage $V_{\text{th,inv}}$, the transistors are only in saturation. First order effects on the nonlinearity are investigated, which include the influence of Δt and t_{int} . As the channel length modulation factor λ complicates calculation, first the case of $\lambda = 0$ is analyzed, followed by an extension to $\lambda \neq 0$.

B.1.1 Simplified transistor model: $\lambda = 0$

The Shichman-Hodges drain current equation for the saturation region of a PMOS transistor is given as

$$I_{D,\text{sat}} = - \underbrace{\frac{1}{2} K_p \frac{W_{\text{eff}}}{L_{\text{eff}}}}_{I_{D,\text{sat},0}} (V_{\text{gs}} - V_{\text{th,p}})^2 (1 + \lambda_p (V_{\text{int}}(t) + V_{\text{th,p}})). \quad (\text{B.1})$$

Using Kirchhoffs Current Law and setting $\lambda_p = 0$, the ordinary differential equation (ODE) for $V_{\text{int}}(t)$ can be given with $k = (N - n)$ or $k = N$ according to Fig. 3.13 as

$$\frac{dV_{\text{int}}(t)}{dt} = \frac{k(-I_{D,\text{sat}})}{C_{\text{int}}}, \quad (\text{B.2})$$

and is solved with the initial conditions from Fig. 3.13 to (B.3) and (B.4) from Table B.1. Now the time $t_d[n]$, when $V_{\text{int}}(t)$ crosses $V_{\text{th,inv}}$, needs to be determined. $V_{\text{int},1/2}(t_d[n]) = V_{\text{th,inv}}$ has to be inserted in (B.3) and (B.4) and solved for $t_d[n]$, resulting in (B.5) and (B.7) from Table B.1. The min. rise time of $V_{\text{int}}(t)$ until it crosses $V_{\text{th,inv}}$ is defined as t_{int} and results from inserting $n = 0$ in (B.5) or (B.7). Depending on the circuit parameters, (B.5) or (B.7) are used for certain regions of n . The max. n for which (B.5) is valid can be determined by setting $t_{d,1/2}[n] = \Delta t$, inserting it into (B.5) or (B.7), and solving for n , which results in (B.6).

Calculating the INL according to (2.11) leads to (B.9) and (B.10), showing that all generated nonlinearity lays in the first region with $t \leq \Delta t$. The peak INL is a measure of overall nonlinearity. It is the global minimum of (B.9) and can be calculated to

$$|\text{INL}_{\text{max}}| = \left(\sqrt{t_{\text{int}}} - \sqrt{\Delta t} \right)^2, \text{ where } 0 \leq t_{\text{int}} \leq \Delta t. \quad (\text{B.11})$$

Table B.1 – Overview on 1st order model equations for $\lambda = 0$.

Equation		Range of validity
$V_{\text{int},1}(t) = \frac{(N-n)I_{\text{D,sat},0}}{C_{\text{int}}}t$	(B.3)	$0 \leq t \leq \Delta t$
$V_{\text{int},2}(t) = V_{\text{int},1}(\Delta t) + \frac{NI_{\text{D,sat},0}}{C_{\text{int}}}(t - \Delta t)$	(B.4)	$\Delta t < t$
$t_{\text{d},1}[n] = \frac{V_{\text{th,inv}}C_{\text{int}}}{(N-n)I_{\text{D,sat},0}}$	(B.5)	$n_{\text{max},1} = \lfloor N(1 - \frac{t_{\text{int}}}{\Delta t}) \rfloor$, where $0 \leq n_{\text{max}} < N$
$t_{\text{d},2}[n] = \underbrace{\frac{V_{\text{th,inv}}C_{\text{int}}}{NI_{\text{D,sat},0}}}_{t_{\text{int}}} + \frac{n}{N}\Delta t$	(B.7)	$n_{\text{max},2} = N$ (B.8)
$\text{INL}_1[n] = \left(\frac{N}{N-n} - 1\right)t_{\text{int}} - \frac{n}{N}\Delta t$	(B.9)	$0 \leq n \leq n_{\text{max},1}$
$\text{INL}_2[n] = 0$	(B.10)	$n_{\text{max},1} < n \leq N$

B.1.2 Complete transistor model: $\lambda \neq 0$

The last section made the assumption that the current sources have a constant current. In real systems this source is a transistor, which has different drain current for varying V_{DS} . Taking (B.1) with $\lambda_{\text{p}} \neq 0$ and inserting it in the ODE from (B.2), yields in a new ODE which can be solved with the initial condition $V_{\text{int}}(0) = 0$ to

$$V_{\text{int},1}(t) = \left(-V_{\text{th,p}} - \frac{1}{\lambda_{\text{p}}}\right) \left(1 - e^{\frac{(N-n)I_{\text{D,sat},0}\lambda_{\text{p}}}{C_{\text{int}}}t}\right). \quad (\text{B.12})$$

A Taylor expansion $V_{\text{int},1,\text{T}}(t)$ of (B.12), cut off after the linear term, provides an approximation of the charging process:

$$V_{\text{int},1,\text{T}}(t) = (1 + \lambda_{\text{p}}V_{\text{th,p}})\frac{(N-n)I_{\text{D,sat},0}}{C_{\text{int}}}t. \quad (\text{B.13})$$

This equation is equal to (B.3) for $\lambda_{\text{p}} = 0$. The three different ODE solutions (B.3), (B.12), and (B.13) are plotted in Fig. B.1 to visualize the charging of V_{int} . While in (B.3) the current is not increasing with V_{DS} and, therefore, underestimated, (B.13) overestimates it, as the linear Taylor expansion pins the transistor current to its highest level according to the channel length modulation given by the Lambda factor. With the knowledge of both extrema, only minimum and only maximum current, a final solution can be approximated with $\lambda'_{\text{p}} = 0.5\lambda_{\text{p}}$.

Calculating $t_{\text{d}}[n]$ from (B.13) at $n = 0$ (which is the min. rise time t_{int}), in the same fashion as for (B.5), shows how the minimum delay is affected:

$$t_{\text{d},1}[n = 0] = \underbrace{\frac{1}{(1 + \lambda'_{\text{p}}V_{\text{th,p}})} \frac{V_{\text{th,inv}}C_{\text{int}}}{NI_{\text{D,sat},0}}}_{t_{\text{int},\lambda \neq 0}} = \frac{1}{(1 + \lambda'_{\text{p}}V_{\text{th,p}})}t_{\text{int}}. \quad (\text{B.14})$$

As λ_{p} and $V_{\text{th,p}}$ are both negative, $t_{\text{int},\lambda \neq 0}$ is always smaller than approximated by the simplified model. However, the general INL shape stays untouched for the Taylor approximation.

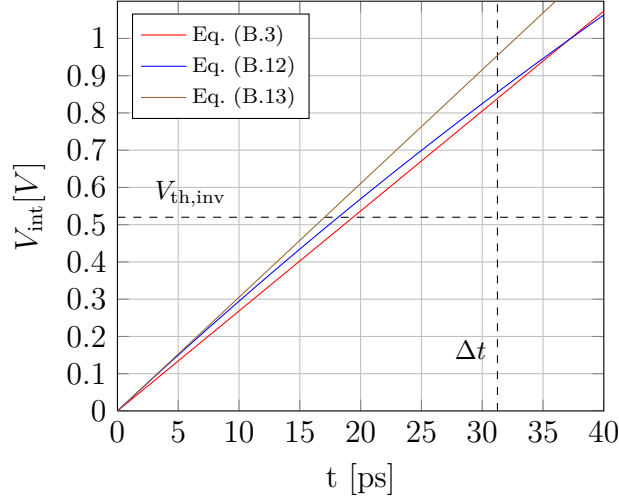


Figure B.1 – Voltage at $V_{\text{int}}(t)$ for $\lambda_p = 0$ (red), $\lambda_p \neq 0$ (blue), and Taylor approximated $\lambda_p \neq 0$ (brown).

Table B.2 – Overview on 2nd order model regions.

Region a): $t_f \leq \Delta t$	Region b): $\Delta t < t_f$
$V_{\text{int},1a}(t): 0 \leq t < t_f$	$V_{\text{int},1b}(t): 0 \leq t < \Delta t$
$V_{\text{int},2a}(t): t_f \leq t \leq \Delta t$	$V_{\text{int},2b}(t): \Delta t \leq t < t_f$
$V_{\text{int},3a}(t): \Delta t \leq t < \Delta t + t_f$	$V_{\text{int},3b}(t): t_f \leq t < \Delta t + t_f$
$V_{\text{int},4a}(t): \Delta t + t_f \leq t$	$V_{\text{int},4b}(t): \Delta t + t_f \leq t$

B.2 Second Order CF-DCEI Model Derivation

All calculations from Appendix B.1 assume ideal input signals with $t_f = 0$. At $t = 0$ and $t = \Delta t$ devices would switch from zero drain current directly to their saturation drain current. To extend the model for $t_f > 0$, the drain current is assumed to rise linear during the fall time of the input signals t_f as proposed in [95]. Furthermore, $\lambda = 0$ is assumed to simplify calculations. This can be modeled into (B.2) by adding a time dependency to the drain current of the charging transistor

$$I_{D,\text{sat},0} \rightarrow I_{D,\text{sat},0} \frac{t}{t_f}, \quad (\text{B.15})$$

which is valid for $0 \leq t < t_f$. Modifying the ODE from (B.2) with (B.15) and solving it with the initial condition $V_{\text{int}}(0) = 0$ yields into

$$V_{\text{int}}(t < t_f) = \frac{(N - n)I_{D,\text{sat},0}}{C_{\text{int}}} \frac{t^2}{2t_f}. \quad (\text{B.16})$$

Depending on the magnitude of t_f , the interpolation can now be described by two different sets of equations, each containing four piecewise functions, as listed in Table B.2.

B.2.1 Region a): $t_f \leq \Delta t$

The ODE solutions (B.3) and (B.16) can now be used to describe the piecewise charging of V_{int} . For this the time limits from Table B.2 as well as the number of cells which are associated to each region and ODE, either N , $(N - n)$ or n , have to be taken into account. Following the same approach as for the 1st order model, the node voltage at V_{int} is defined piecewise. The first region is directly given by (B.16) as

$$V_{\text{int},1a}(t) = \frac{(N - n)I_{D,\text{sat},0}}{C_{\text{int}}} \frac{t^2}{2t_f}. \quad (\text{B.17})$$

For $t \geq t_f$ the ODE is given as in (B.2), where the pre-charging to the voltage V_{pre} , calculated with $V_{\text{int},1a}(t_f)$ from (B.17), determines the integration constant. It is solved to

$$V_{\text{int},2a}(t) = \underbrace{V_{\text{int},1a}(t_f)}_{V_{\text{pre}}} + \frac{(N - n)I_{D,\text{sat},0}}{C_{\text{int}}}(t - t_f). \quad (\text{B.18})$$

This concept can be extended to the third and fourth region, leading to

$$\begin{aligned} V_{\text{int},3a}(t) &= V_{\text{int},2a}(\Delta t) + \frac{(N - n)I_{D,\text{sat},0}}{C_{\text{int}}}(t - \Delta t) + \frac{nI_{D,\text{sat},0}}{C_{\text{int}}} \frac{(t - \Delta t)^2}{2t_f} \\ &= \frac{(N - n)I_{D,\text{sat},0}}{C_{\text{int}}} \left(t - \frac{t_f}{2} \right) + \frac{nI_{D,\text{sat},0}}{C_{\text{int}}} \frac{(t - \Delta t)^2}{2t_f}, \text{ and} \end{aligned} \quad (\text{B.19})$$

$$\begin{aligned} V_{\text{int},4a}(t) &= V_{\text{int},3a}(\Delta t + t_f) + \frac{NI_{D,\text{sat},0}}{C_{\text{int}}}(t - \Delta t - t_f) \\ &= \frac{NI_{D,\text{sat},0}}{C_{\text{int}}} \left(t - \frac{t_f}{2} - \frac{n}{N}\Delta t \right). \end{aligned} \quad (\text{B.20})$$

Similar to the derivation for the 1st order model, $V_{\text{int},1a-4a}(t_{d,1a-4a}[n])$ are set equal to $V_{\text{th,inv}}$ and are then solved for $t_{d,1a-4a}[n]$. The results can be found in Table B.3.

Table B.3 – Overview on 2nd order model propagation delay functions.

Code-delay transfer function $t_d[n]$	n_{\max}
$t_{d,1a}[n] = \sqrt{\frac{2Nt_f t_{\text{int}}}{(N-n)}}$ (B.21)	⁻¹⁾
$t_{d,2a}[n] = \frac{N}{(N-n)}t_{\text{int}} + \frac{t_f}{2}$ (B.22)	$n_{\max,2a} = \left\lfloor N \left(1 + \frac{t_{\text{int}}}{t_f/2 - \Delta t} \right) \right\rfloor$ (B.23)
$t_{d,3a}[n] = \Delta t + t_f - \frac{N}{n}t_f +$ $t_f \sqrt{\frac{N^2}{n^2} - \frac{N}{n} \left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1 \right) + \frac{2\Delta t}{t_f}}$ (B.24)	$n_{\max,3a} = \left\lfloor N \left(1 + \frac{t_f/2 - t_{\text{int}}}{\Delta t} \right) \right\rfloor$ (B.25)
$t_{d,4a}[n] = t_{\text{int}} + \frac{t_f}{2} + \frac{n}{N}\Delta t$ (B.26)	$n_{\max,4a} = N$ (B.27)
$t_{d,1b}[n] = \sqrt{\frac{2Nt_f t_{\text{int}}}{(N-n)}}$ (B.28)	⁻²⁾
$t_{d,2b}[n] = \frac{n}{N}\Delta t \pm \sqrt{\Delta t^2 \left(\frac{n^2}{N^2} - \frac{n}{N} \right) + 2t_f t_{\text{int}}}$ (B.29)	$n_{\max,2b} = \left\lfloor N \frac{t_f}{\Delta t} \frac{2t_{\text{int}} - t_f}{\Delta t - 2t_f} \right\rfloor$ (B.30)
$t_{d,3b}[n] = \frac{N-n}{N}\Delta t + t_f - \frac{N}{n}t_f - \sqrt{2t_f t_{\text{int}}} +$ $t_f \sqrt{\frac{N^2}{n^2} - \frac{N}{n} \left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1 \right) + \frac{2\Delta t}{t_f}}$ (B.31)	$n_{\max,3b} = \left\lfloor N \left(1 + \frac{t_f/2 - t_{\text{int}}}{\Delta t} \right) \right\rfloor$ (B.32)
$t_{d,4b}[n] = t_{\text{int}} + \frac{t_f}{2} + \frac{n}{N}\Delta t$ (B.33)	$n_{\max,4b} = N$ (B.34)

¹⁾ Threshold crossings in this region exist only for $t_f \geq 2t_{\text{int}}$, which is a poor design point as it leads to $\text{INL}[n_{\max}] \geq 0.171\Delta t$ (remember: $t_f \leq \Delta t$).

²⁾ Threshold crossings in this region exist only for $t_f \geq 2t_{\text{int}}$, which is a poor design point from jitter point of view.

Depending on design parameters, V_{int} crosses $V_{\text{th,inv}}$ in different regions 1a - 4a. The time domain limits for $V_{\text{int},1a-4a}(t)$ as defined in Table B.2 can be translated to limits in code domain for $t_{d,1a-4a}[n]$. To determine the upper code limit, the transfer function is set equal to the upper time limit, $t_{d,1a-4a}[n_{\text{max},1a-4a}] = t_{\text{max},1a-4a}$, and solved for n_{max} . The results can be found in Table B.3. The lower limits $n_{\text{min},1-4}$ are according to n_{max} of the previous region.

The delay $t_{d,1a}[n]$ shows only valid solutions for $t_f \geq 2t_{\text{int}}$, which is a poor design point according to (B.11), as it leads to $|\text{INL}_{\text{max}}| \geq 0.171\Delta t$ (remember: $t_f \leq \Delta t$). Further calculations assume $t_f < 2t_{\text{int}}$, leading to an identical minimum delay for all regions:

$$t_{d,2a/4a}[0] = \lim_{n \rightarrow 0} t_{d,3a}[n] = t_{\text{int}} + \frac{t_f}{2} \quad (\text{B.35})$$

This leaves $t_{d,4a}[n] = t_{d,\text{ideal}}[n]$, giving a piecewise defined INL as

$$\text{INL}_{2a/3a/4a}[n] = t_{d,2a/3a/4a}[n] - t_{d,\text{ideal}}[n]. \quad (\text{B.36})$$

Valid codes for region 2a) exist only for $n_{\text{max},2a} \geq 1$, meaning the first $V_{\text{th,inv}}$ crossing is in this region, which can be re-written from (B.23) as

$$t_f \leq 2\left(\Delta t - \frac{N}{N-1}t_{\text{int}}\right) \approx 2(\Delta t - t_{\text{int}}). \quad (\text{B.37})$$

Depending on (B.37), the INL equations are given in Table B.4, where $\text{INL}[0] = 0$ for each piecewise INL. To cover also the case of $\lambda \neq 0$, t_{int} has to be replaced by $t_{\text{int},\lambda \neq 0} = t_{\text{int}}/(1 + 0.5\lambda_p V_{\text{th,p}})$. The peak INL model evaluation from Fig. 5.4 compares the numerical and analytical evaluation of this model. As the results have only negligible differences, it proves the assumptions of the analytical model, especially that $t_{\text{int},\lambda \neq 0}$ is an accurate estimation.

B.2.2 Region b): $t_f > \Delta t$

The calculations can be done in an analogous manner to the last section. $t_{d,1b}[n]$ shows only valid solutions for $t_f > 2t_{\text{int}}$, which leads to a poor design from jitter point of view. Therefore, $t_f \leq 2t_{\text{int}}$ is assumed, meaning no crossing of $V_{\text{th,inv}}$ in region 1b.

The equations for $t_d[n]$ [(B.28), (B.29), (B.31), (B.33)], n_{max} [(B.30), (B.32), (B.34)] and $\text{INL}[n]$ [(B.43)-(B.46)] in this region are summarized in Table B.3 and B.4.

Table B.4 – Overview on 2nd order model INL equations.

Region	Range of t_f	Range of n	INL[n]
a) $t_f \leq \Delta t$	$t_f \leq 2(\Delta t - t_{\text{int}})$	$0 \leq n \leq n_{\text{max},2a}$	$\frac{n}{N-n}t_{\text{int}} - \frac{n}{N}\Delta t$ (B.38)
		$n_{\text{max},2a} < n \leq n_{\text{max},3a}$	$\frac{N-n}{N}\Delta t + \frac{t_f}{2} - \frac{N}{n}t_f - t_{\text{int}} + t_f\sqrt{\frac{N^2}{n^2} - \frac{N}{n}\left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1\right) + \frac{2\Delta t}{t_f}}$ (B.39)
		$n_{\text{max},3a} < n \leq N$	0 (B.40)
	$t_f > 2(\Delta t - t_{\text{int}})$	$0 < n \leq n_{\text{max},3a}$	$\frac{N-n}{N}\Delta t + \frac{t_f}{2} - \frac{N}{n}t_f - t_{\text{int}} + t_f\sqrt{\frac{N^2}{n^2} - \frac{N}{n}\left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1\right) + \frac{2\Delta t}{t_f}}$ (B.41)
		$n_{\text{max},3a} < n \leq N$	0 (B.42)
b) $t_f > \Delta t$	$t_f \geq 2t_{\text{int}}$	$0 \leq n \leq n_{\text{max},2b}$	$\sqrt{\Delta t^2\left(\frac{n^2}{N^2} - \frac{n}{N}\right) + 2t_ft_{\text{int}}} - \sqrt{2t_ft_{\text{int}}}$ (B.43)
		$n_{\text{max},2b} < n \leq N$	$\frac{N-n}{N}\Delta t + t_f - \frac{N}{n}t_f - \sqrt{2t_ft_{\text{int}}} + t_f\sqrt{\frac{N^2}{n^2} - \frac{N}{n}\left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1\right) + \frac{2\Delta t}{t_f}}$ (B.44)
	$t_f < 2t_{\text{int}}$	$0 < n \leq n_{\text{max},3b}$	$\frac{N-n}{N}\Delta t + \frac{t_f}{2} - \frac{N}{n}t_f - t_{\text{int}} + t_f\sqrt{\frac{N^2}{n^2} - \frac{N}{n}\left(\frac{2(\Delta t - t_{\text{int}})}{t_f} + 1\right) + \frac{2\Delta t}{t_f}}$ (B.45)
		$n_{\text{max},3b} < n \leq N$	0 (B.46)

C Switched Capacitor Fine Tuning Nonlinearity Model

This appendix calculates the nonlinearity of the switched capacitor based fine tuning. Fig. C.1(a) shows the investigated circuit. To determine the overall linearity of the switched capacitor based fine tuning, the overall propagation delay $t_{d,out}^{sc}$ of the signals needs to be calculated. The event of interest is the threshold crossing of node V_{out}^{sc} . Ideally, the output delay depends linearly on the tuning capacitance C_{tune} , and the capacitance is a linear function of the digital code word n given as

$$C_{tune}[n] = C_{min} + \frac{n}{N}(C_{max} - C_{min}). \quad (C.1)$$

The delay range depends on the ratio of C_{min}/C_{max} , and the resolution on $N = 2^{k_{sc}}$ (with k_{sc} the number of bits). As $t_{d,out}^{sc}$ is a function of C_{tune} , the delay range is defined as

$$\begin{aligned} t_{range} &= t_{d,out}^{sc}(C_{max}) - t_{d,out}^{sc}(C_{min}) \\ &= t_{d,out}^{sc}(C_{tune}[N]) - t_{d,out}^{sc}(C_{tune}[0]) \end{aligned} \quad (C.2)$$

The two main sources of nonlinearity can be: a) nonlinearity at net V_{int}^{sc} due to tuning of C_{tune} , and b) nonlinearity net V_{out}^{sc} due to a changing slope at the input of the output buffer. Section C.1 analyzes a) and derives analytical equations for the time delays of this net, annotated in the ideal waveform plot of Fig. C.1(b). Afterwards, Section C.2 calculates the overall nonlinearity of V_{out}^{sc} 's delay (also annotated in Fig. C.1(b)), and proves b) to be the major contributor to nonlinearity for this type of fine tuning circuit. The model derivation follows the methodology from Appendix B: enable a simple model calculation by the knowledge of easily measurable time delays and standard circuit parameters (in this case: min. and max. capacitance).

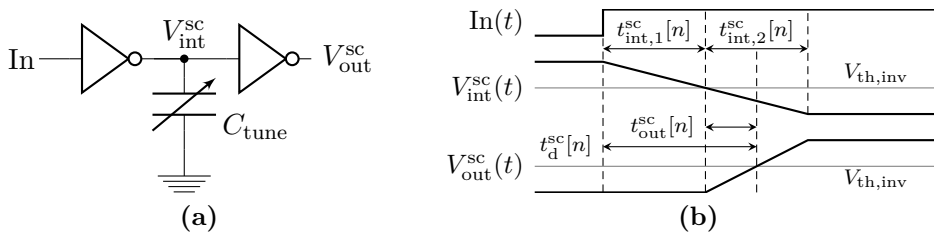


Figure C.1 – Switched capacitor based fine tuning: (a) circuit implementation, and (b) idealized waveforms for ideal input signal.

C.1 Linearity at Net $V_{\text{int}}^{\text{sc}}$

The overall nonlinearity is calculated for the case of a rising edge at $V_{\text{out}}^{\text{sc}}$. Therefore, first the discharging of $V_{\text{int}}^{\text{sc}}$ is analyzed, which leads to a charging of $V_{\text{out}}^{\text{sc}}$. A falling signal at $V_{\text{out}}^{\text{sc}}$ can be calculated in an analogous manner. The initial condition of the intermediate node $V_{\text{int}}^{\text{sc}}$ is

$$V_{\text{int}}^{\text{sc}}(t = 0) = V_{\text{DD}}. \quad (\text{C.3})$$

To describe the full discharging of this node, the calculation needs to be splitted in two regions: 1) $V_{\text{int}}^{\text{sc}}$ is discharged by the NMOS devices of the first inverter until $V_{\text{int}}^{\text{sc}} = V_{\text{th},n}$; and 2) $V_{\text{int}}^{\text{sc}}$ is further discharged by the same devices, that operate now in their linear instead of their saturation region.

Region 1: $V_{\text{int}}^{\text{sc}} \geq V_{\text{th},n}$

The ODE for net $V_{\text{int}}^{\text{sc}}$ is given by

$$\frac{dV_{\text{int}}^{\text{sc}}(t)}{dt} = -\frac{I_{\text{D,sat},0}}{C_{\text{tune}}[n]}, \quad (\text{C.4})$$

and with the initial condition $V_{\text{int}}^{\text{sc}}(0) = V_{\text{DD}}$ it is solved to

$$V_{\text{int}}^{\text{sc}}(t) = V_{\text{DD}} - \frac{I_{\text{D,sat},0}t}{C_{\text{tune}}[n]}. \quad (\text{C.5})$$

Solving (C.5) for t finally leads to

$$t = \frac{(V_{\text{DD}} - V_{\text{int}}^{\text{sc}}(t)) C_{\text{tune}}[n]}{I_{\text{D,sat},0}}. \quad (\text{C.6})$$

Two cases are of interest now: 1) the delay until the subsequent stage starts charging $V_{\text{out}}^{\text{sc}}$, and 2) the time from case 1) on, until the discharging NMOS device enters the triode region. The latter case marks the end of this region, as the NMOS device is assumed to be in saturation here. Case 1) is the discharging delay until $V_{\text{int}}^{\text{sc}}(t) : V_{\text{DD}} \rightarrow V_{\text{th,inv}}$, given as

$$t_{\text{int},1}^{\text{sc}}[n] = \frac{(V_{\text{DD}} - V_{\text{th,inv}}) C_{\text{tune}}[n]}{I_{\text{D,sat},0}}, \quad (\text{C.7})$$

and case 2) the additional discharging delay for $V_{\text{int}}^{\text{sc}}(t) : V_{\text{th,inv}} \rightarrow V_{\text{th},n}$, given as

$$\begin{aligned} t_{\text{int},2.1}^{\text{sc}}[n] &= \frac{(V_{\text{DD}} - V_{\text{th},n}) C_{\text{tune}}[n]}{I_{\text{D,sat},0}} - t_{\text{int},1}^{\text{sc}}[n] \\ &= \frac{(V_{\text{th,inv}} - V_{\text{th},n}) C_{\text{tune}}[n]}{I_{\text{D,sat},0}}. \end{aligned} \quad (\text{C.8})$$

To determine $t_{\text{int},1}^{\text{sc}}[n]$, it is sufficient to measure the fall time for $V_{\text{int}}^{\text{sc}}(t) : V_{\text{DD}} \rightarrow V_{\text{th,inv}}$ in transient simulation for the case of $C_{\text{tune}}[n = 0] = C_{\text{min}}$ which yields into

$$\begin{aligned} t_{\text{int},1.0}^{\text{sc}} &= t_{\text{int},1}^{\text{sc}}[n = 0] \\ &= \frac{(V_{\text{DD}} - V_{\text{th,inv}}) C_{\text{tune}}[0]}{I_{\text{D,sat},0}}. \end{aligned} \quad (\text{C.9})$$

Then $t_{\text{int},1}^{\text{sc}}[n]$ can be expressed by this time constant and the knowledge of $C_{\text{tune}}[n]$ as

$$t_{\text{int},1}^{\text{sc}}[n] = \left(1 + \frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}} \right) t_{\text{int},1,0}^{\text{sc}}. \quad (\text{C.10})$$

Region 2: $V_{\text{int}}^{\text{sc}} < V_{\text{th},n}$

The ODE for net $V_{\text{int}}^{\text{sc}}$ is now given by

$$\begin{aligned} \frac{dV_{\text{int}}^{\text{sc}}(t)}{dt} &= -\frac{I_{\text{D,lin}}}{C_{\text{tune}}[n]} \\ &= -\frac{\mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} \left((V_{\text{GS}} - V_{\text{th},n}) V_{\text{int}}^{\text{sc}}(t) - \frac{V_{\text{int}}^{\text{sc}}(t)^2}{2} \right)}{C_{\text{tune}}[n]}, \end{aligned} \quad (\text{C.11})$$

and is solved with the initial condition $V_{\text{int}}^{\text{sc}}(t=0) = V_{\text{th},n}$ to

$$V_{\text{int}}^{\text{sc}}(t) = -2(V_{\text{GS}} - V_{\text{th},n}) \frac{1}{\left(\frac{-2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{th},n}} + 1 \right) e^{2(V_{\text{GS}} - V_{\text{th},n}) \frac{\mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}} t}{2C_{\text{tune}}[n]}} - 1} \quad (\text{C.12})$$

$$\Leftrightarrow t = \frac{C_{\text{tune}}[n]}{(V_{\text{GS}} - V_{\text{th},n}) \mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}}} \ln \left(\frac{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{int}}^{\text{sc}}(t)}}{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{th},n}}} \right) \quad (\text{C.13})$$

The delay $t_{\text{int},2,2}^{\text{sc}}$ from $V_{\text{int}}^{\text{sc}}(t) : V_{\text{th},n} \rightarrow 0.1V_{\text{DD}}$ is now given as

$$t_{\text{int},2,2}^{\text{sc}} = \frac{C_{\text{tune}}[n]}{(V_{\text{GS}} - V_{\text{th},n}) \mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}}} \ln \left(\frac{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{0.1V_{\text{DD}}}}{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{th},n}}} \right) \quad (\text{C.14})$$

Note that in both regions, the fall times $t_{\text{int},2,1}^{\text{sc}}$ and $t_{\text{int},2,2}^{\text{sc}}$ depend linearly on $C_{\text{tune}}[n]$. The overall fall time for $V_{\text{int}}^{\text{sc}}(t) : V_{\text{th},\text{inv}} \rightarrow 0.1V_{\text{DD}}$ is then the sum of both regions:

$$\begin{aligned} t_{\text{int},2}^{\text{sc}}[n] &= t_{\text{int},2,1}^{\text{sc}}[n] + t_{\text{int},2,2}^{\text{sc}}[n] \\ &= \frac{(V_{\text{th,inv}} - V_{\text{th},n}) C_{\text{tune}}[n]}{I_{\text{D,sat},0}} + \\ &\quad \frac{C_{\text{tune}}[n]}{(V_{\text{GS}} - V_{\text{th},n}) \mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}}} \ln \left(\frac{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{0.1V_{\text{DD}}}}{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{th},n}}} \right). \end{aligned} \quad (\text{C.15})$$

As in case of the DCEI and CF-DCEI, the technology parameters determine the minimum fall time t_{int} at this node. The minimum value corresponds to minimum $C_{\text{tune}}[n=0]$:

$$\begin{aligned} t_{\text{int},2,0}^{\text{sc}} &= t_{\text{int},2}^{\text{sc}}[n=0] \\ t_{\text{int},2,0}^{\text{sc}} &= \frac{(V_{\text{DD}}/2 - V_{\text{th},n}) C_{\text{tune}}[0]}{I_{\text{D,sat},0}} + \\ &\quad \frac{C_{\text{tune}}[0]}{(V_{\text{GS}} - V_{\text{th},n}) \mu_n C_{\text{ox}} \frac{W_{\text{eff}}}{L_{\text{eff}}}} \ln \left(\frac{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{0.1V_{\text{DD}}}}{1 - \frac{2(V_{\text{GS}} - V_{\text{th},n})}{V_{\text{th},n}}} \right). \end{aligned} \quad (\text{C.16})$$

Expressing (C.15) with (C.16) leads to an intuitive formula that can be easily used with knowledge of $C_{\text{tune}}[n]$ and extracting $t_{\text{int},2,0}^{\text{sc}}$ from transient simulation:

$$\begin{aligned} t_{\text{int},2}^{\text{sc}}[n] &= \frac{C_{\text{tune}}[n]}{C_{\text{tune}}[0]} t_{\text{int},2,0}^{\text{sc}} \\ &= \left(1 + \frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}} \right) t_{\text{int},2,0}^{\text{sc}}. \end{aligned} \quad (\text{C.17})$$

C.2 Linearity at Net $V_{\text{out}}^{\text{sc}}$

Similar to the CF-DCEI analysis from Appendix B, where a delay is calculated for $t_f > 0$ in the 2nd order model, the charging of $V_{\text{out}}^{\text{sc}}$ (falling $V_{\text{int}}^{\text{sc}}$ leads to rising $V_{\text{out}}^{\text{sc}}$) needs to be divided in different regions. Compared to $V_{\text{int}}^{\text{sc}}$, where the switching delay depends linearly on C_{tune} , $V_{\text{out}}^{\text{sc}}$ is expected to switch fast. The assumption of $t_{f,\text{int}} \geq 2t_{\text{out}}$ (such as in Appendix B) simplifies the calculation, as a single equation instead of piecewise defined equations suffices. The delay of the $V_{\text{out}}^{\text{sc}}$ is according to (B.21) then given as

$$t_{\text{out}}^{\text{sc}}[n] = \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[n]}. \quad (\text{C.18})$$

The overall delay is then the sum from the delay of $V_{\text{int}}^{\text{sc}} : V_{\text{DD}} \rightarrow V_{\text{th,inv}}$, which determines when $V_{\text{out}}^{\text{sc}}$ starts switching, and the delay of $V_{\text{out}}^{\text{sc}}$ itself:

$$\begin{aligned} t_{\text{d}}^{\text{sc}}[n] &= t_{\text{int},1}^{\text{sc}}[n] + t_{\text{d,out}}^{\text{sc}}[n] \\ &= \left(1 + \frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}} \right) t_{\text{int},1,0}^{\text{sc}} + \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[n]} \end{aligned} \quad (\text{C.19})$$

Transfer function and INL are now calculated according to (2.9) and (2.11) (the full scale is this time not Δt , but the full scale of the fine tuning $\text{TF}[N]$):

$$\begin{aligned} \text{TF}[n] &= t_{\text{d}}^{\text{sc}}[n] - t_{\text{d}}^{\text{sc}}[0] \\ &= \left(\frac{n}{N} \frac{(C_{\text{max}} - C_{\text{min}})}{C_{\text{min}}} \right) t_{\text{int},1,0}^{\text{sc}} + \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[n]} - \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2,0}^{\text{sc}}}, \text{ and} \end{aligned} \quad (\text{C.20})$$

$$\begin{aligned} \text{INL}[n] &= \text{TF}[n] - \text{TF}_{\text{ideal}} \\ &= \text{TF}[n] - \frac{n}{N} \text{TF}[N] \\ &= \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[n]} - \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2,0}^{\text{sc}}} - \frac{n}{N} \left(\sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2}^{\text{sc}}[N]} - \sqrt{2t_{\text{out},0}^{\text{sc}} t_{\text{int},2,0}^{\text{sc}}} \right). \end{aligned} \quad (\text{C.21})$$

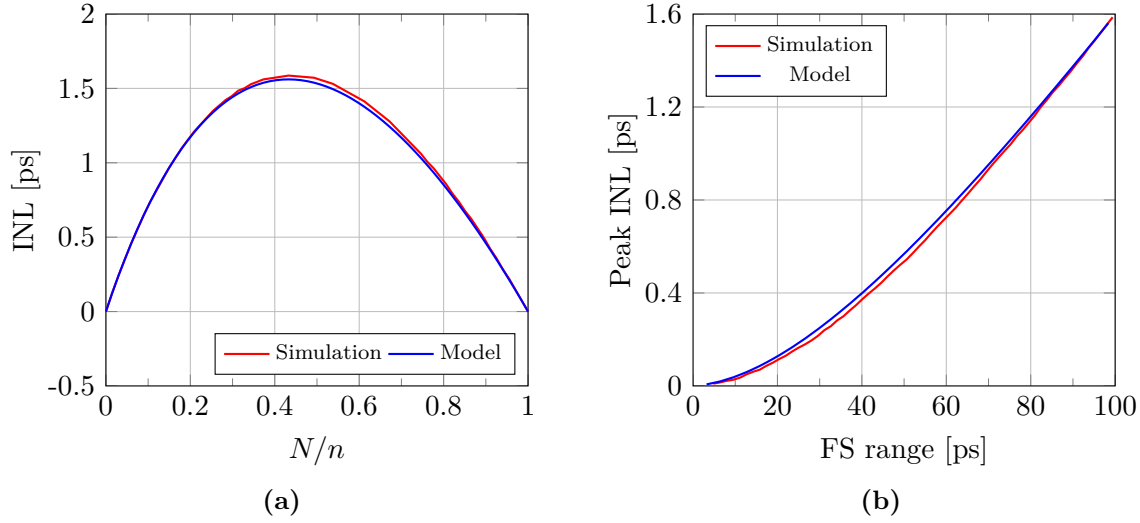
Note that the term related to $t_{\text{int},1}^{\text{sc}}[n]$ is not present in (C.21), as nonlinearity is generated in the discharging of $V_{\text{int}}^{\text{sc}}$. The global minimum of (C.21) depends then only on C_{min} and C_{max} and allows to express the capacitance value with the highest INL directly:

$$n_{\text{max}}^{\text{sc}} = \left\lceil N \left(\frac{C_{\text{max}} - C_{\text{min}}}{4(\sqrt{C_{\text{max}}} - \sqrt{C_{\text{min}}})^2} - \frac{C_{\text{min}}}{C_{\text{max}} - C_{\text{min}}} \right) \right\rceil \quad (\text{C.22})$$

$$C_{\text{tune}}[n_{\text{max}}^{\text{sc}}] = \frac{(C_{\text{max}} - C_{\text{min}})^2}{4(\sqrt{C_{\text{max}}} - \sqrt{C_{\text{min}}})^2} \quad (\text{C.23})$$

Table C.1 – Switched capacitor model parameters.

Parameter	Value	Description
C_{\min}	165 fF	Minimum capacitance at net $V_{\text{int}}^{\text{sc}}$
C_{\max}	500 fF	Maximum capacitance at net $V_{\text{int}}^{\text{sc}}$
N	70	Maximum fine tuning code
$t_{\text{int},1.0}^{\text{sc}}$	37.9 ps	Discharging time $V_{\text{int}}^{\text{sc}}(t) : V_{\text{DD}} \rightarrow V_{\text{th,inv}}$ for C_{\min}
$t_{\text{int},2.0}^{\text{sc}}$	48.6 ps	Discharging time $V_{\text{int}}^{\text{sc}}(t) : V_{\text{th,inv}} \rightarrow 0.1V_{\text{DD}}$ for C_{\min}
$t_{\text{out},0}^{\text{sc}}$	10 ps	Charging time $V_{\text{out}}^{\text{sc}}(t) : V_{\text{SS}} \rightarrow V_{\text{th,inv}}$ for ideal input signal

**Figure C.2** – Linearity of switched capacitor based fine tuning: (a) INL over code, and (b) peak INL for FS variation through variation of C_{\max} .

C.3 Comparison to Circuit Simulations

To verify the model, it is compared to circuit simulations. The input and output buffer from Fig. C.1 are implemented as CMOS inverters of equal sizing. The tuning capacitor is an ideal capacitor with variable value. The model parameters are listed in Table C.1.

With data from a single simulation, different configuration can be tested against the model. Variation of C_{\min} and C_{\max} in post-processing results in a different range, and variation of C_{\min} results in variation of $t_{\text{int},1.0}^{\text{sc}}$ and $t_{\text{int},2.0}^{\text{sc}}$ (fall times for varied C_{\min} are calculated with the model). To ensure high model accuracy, the covered range as well as the INL shape and peak are compared to simulations.

Fig. C.2(a) compares the simulated to the modeled INL, using the model parameters from Table C.1 that were extracted from simulations with $n = 0$. The range covered in simulation is 99.4 ps, and the range of the model calculation deviates only by -0.61% . This proves an excellent estimation of the INL, as only slight differences are visible in the plot. To validate the model for a wider range, C_{\max} is reduced in steps of 5 fF to modify the FS of the fine tuning. The peak INL of simulations and model is compared in Fig. C.2(b).

The results lead to the conclusion that key parameters that describe the switched

capacitor based fine tuning were identified correctly. This model can now be used to compare the linearity to the CF-DCEI. Furthermore, the rise and fall time parameters allow to make a statement about jitter, which relates to the slope steepness in a system.

List of Figures

1.1	Basic DTC operation: (a) top level overview on the DTC, and (b) example for relation between input reference signal and DTC output signal.	1
1.2	DTC coarse tuning architecture examples with k_{coarse} bit resolution: (a) DLL based coarse tuning, and (b) divider based coarse tuning [2].	3
1.3	DTC fine tuning architectures: (a) switched capacitor based delay cell or DCDL, (b) PI [2], and (c) DLL based phase filter [16].	4
1.4	DDPS frequency synthesis: (a) DDPS circuit architecture [18], and (b) example operation of a 3 bit DDPS block for generation of $f_{\text{out}} > f_{\text{ref}}$	6
1.5	Source-synchronous interface with DTC phase adjustment [16].	8
1.6	Fractional-N ADPLL implemented with (a) integer-N divider and TDC, and (b) integer-N divider, DTC to realize fractional-N operation, and 1 bit TDC implemented as comparator.	9
1.7	DTC-based fractional-N sub-sampling PLL [65].	10
1.8	DTC-based fractional-N MDLL [69].	11
1.9	DTC-based transmitters: (a) polar transmitter, and (b) outphasing transmitter.	13
2.1	Architecture overview of the three-stage DTC.	16
2.2	MMD output waveforms for (a) different static digital codes $n_{10:8}$, and (b) different dynamic code changes triggering division modes 3 and 5.	17
2.3	Signal alignment between VCO_p and the single DTC blocks.	19
3.1	Overview on investigated PI architectures.	25
3.2	(a) Implementation and interconnection of the DCEI unit cells, and (b) transistor level implementation of the analog MUX core.	26
3.3	Waveforms of the ideal DCEI interpolation process for different codes n	27
3.4	DCEI cell array topology.	28
3.5	DCEI equivalent circuit for rising interpolation in region (a) $0 \leq t < \Delta t$ with initial condition $V_{\text{int},1}(0) = 0$, and (b) $\Delta t \leq t$ with initial condition $V_{\text{int},2}(\Delta t) = V_{\text{int},1}(\Delta t)$	29
3.6	Results of DCEI model evaluation: (a) waveforms at the interpolation node for different codes n and $t_f = 0$, (b) calculated INL for different ratios $t_{\text{int},0}/\Delta t$ and $t_f = 0$, (c) calculated INL for different t_f at $t_{\text{int},0}/\Delta t = 0.8$, and (d) peak INL for variation of $t_{\text{int},0}$ and t_f for $\Delta t = 31.25$ ps.	30
3.7	Simulated and modeled nonlinearity of the DCEI: (a) TF, (b) DNL, and (c) INL. Model parameters: $t_{\text{int},0} = 26$ ps, $t_f = 32$ ps, $\Delta t = 31.25$ ps.	31
3.8	(a) Implementation of CF-DCEI unit cells, and (b) waveforms of the ideal linearized interpolation process for different codes n	33
3.9	Implementation of (a) i^{th} interpolation cell, and (b) retention cells.	34
3.10	Logic timing diagram of (a) interpolation cells, and (b) retention cells.	35

3.11	Simulated waveforms of the interpolation process for different code words n .	36
3.12	CF-DCEI cell array topology.	36
3.13	CF-DCEI equivalent circuit for rising interpolation in region (a) $0 \leq t < \Delta t$ with initial condition $V_{\text{int},1}(0) = 0$, and (b) $\Delta t \leq t$ with initial condition $V_{\text{int},2}(\Delta t) = V_{\text{int},1}(\Delta t)$.	37
3.14	Results of CF-DCEI model evaluation: (a) waveforms at the interpolation node from numerical model evaluation for different codes n and $t_f = 0$, (b) INL calculated from 1 st order model for different ratios $t_{\text{int},0}/\Delta t$, (c) INL calculated from 2 nd order model for different t_f at $t_{\text{int},0}/\Delta t = 0.8$, and (d) peak INL calculated from 2 nd order model for variation of $t_{\text{int},0}$ and t_f with $\Delta t = 31.25$ ps.	38
3.15	Comparison between 2 nd order model and simulation with layout extracted parasitics. Model parameters: $t_{\text{int}} = 24.5$ ps, $t_f = 25$ ps, and $N = 128$.	40
3.16	Switched capacitor based fine tuning: (a) circuit implementation, and (b) idealized waveforms for ideal input signal.	41
3.17	Simulated nonlinearity of the switched capacitor based fine tuning for different $t_{\text{int},2,0}$: (a) INL for FS = 31.25 ps, and (b) peak INL plotted against FS.	41
3.18	DCEI ² schematics: (a) interconnection of DCEI ² unit cells and location of interpolations, and (b) unit cell transistor implementation.	43
3.19	Simulated waveforms at both DCEI ² interpolation nodes: (a) local interpolation at $V_{\text{int},1}$ for different select signal configurations, and (b) passive second interpolation at $V_{\text{int},2}$ for different codes, colored plots highlight the special cases when all cells have an identical configuration.	44
3.20	DCEI ² INL for variation of t_f through the tunable input buffers. The red plot highlights an ideally tuned DCEI ² , as it delivers the smallest peak-to-peak INL.	46
3.21	Simulated static code-dependent DCEI ² current consumption (i_{base} is not included).	47
3.22	Evaluation of the INL at node $V_{\text{int},1}$: (a) INL for variation of $t_{r/f}$ and Δt , the colored contour line highlighting the region of ideal linearity, and (b) evaluation of the contour line for different process corners, indicating $t_{r/f}$ in dependency of Δt to achieve ideal linearity.	49
3.23	7 bit DCEI ² unit cell array with 2 bit binary extension according to control scheme (b).	52
3.24	Possible implementation of the output stages of (a) the thermometric DCEI ² unit cell, (b) a first version of the $B_{1/2}$ or $B_{1/4}$ cell, (c) a second version of the $B_{1/2}$ or $B_{1/4}$ cell, and (d) the $B_{1/4}$ cell.	53
3.25	Monte Carlo simulations of the DNL for binary bit implementation of (a) DCEI ² test chip V1, and (b) DCEI ² test chip V2.	54
3.26	Comparison of DCEI ($f_{\text{out}} = 2$ GHz), CF-DCEI ($f_{\text{out}} = 2$ GHz), and DCEI ² ($f_{\text{out}} = 2.5$ GHz) in terms of (a) DNL, and (b) INL, both normalized for N and Δt .	55
4.1	DTC output phase ϕ for (a) only static nonlinearity, and (b) static and dynamic nonlinearity.	60

4.2	Voltage sensitivity of DTC delay.	62
4.3	Supply glitch caused by instantaneous change of average current: DTC output voltage V_{out} , current I_{out} , and supply voltage V_{sup} for (a) constant DTC output period, and (b) DTC output period stretch due to code transition $k \rightarrow k + j$	63
4.4	Instantaneous current change for code transition $k \rightarrow k + j$ for the ranges $0^\circ \leq \phi_k < 90^\circ$ and $0^\circ \leq \phi_{k+j} < 90^\circ$. The DTC resolution is $k = 7$ bit with $k_{\text{coarse}} = 3$ bit and $k_{\text{fine}} = 4$ bit, and $i_{\text{nom},0} = i_{\text{nom,coarse}} = i_{\text{nom,fine}}$	65
4.5	Selection signals of the DCEI ² V2 for a code sequence $127 \rightarrow 719 \rightarrow 127$ ($\pm 52.03^\circ$ phase step) at 2.5 GHz with (a) proper timing of the select signals, and (b) poor timing of the select signals.	66
4.6	Dynamic errors simulation results of the DCEI ² V2 based DTC: dynamic INL of test cases (a) PCR 1 and (b) RPM 1, (c) corresponding average dynamic INLs, and (d) standard deviation of dynamic INLs.	69
4.7	LDO implementation for all discussed DTC variants. Red highlighted components mark the extension for the dynamic effects compensation.	70
4.8	DTC supply voltage with active and inactive compensation for (a) code toggling, and random code change at (b) every 6 th and (c) every RF cycle.	72
4.9	MMD compensation circuits for charge injection on V_{PG} : compensation for (a) LSB changes, (b) division-by-5, and (c) division-by-3.	74
4.10	Standard deviation of test case (a) PCR 1 and (b) RPM 1 for different compensation settings.	75
5.1	Measurement setup and interconnection of test chip and spectrum analyzer.	79
5.2	CF-DCEI based DTC measurement results: (a) TF, (b) DNL, (c) INL, and (d) phase noise.	81
5.3	INL comparison between measurement and circuit simulation of the CF-DCEI.	81
5.4	Comparison of the CF-DCEI's peak INL ($ \text{INL} _{\text{max}}$) extracted from model, circuit simulation, and measurement.	83
5.5	Measurement results of the DCEI ² V2 based DTC for $f_{\text{out}} = 2496$ MHz: (a) TF, (b) DNL, and (c) INL. (d) INL for $f_{\text{out}} = 2.2$ GHz and $f_{\text{out}} = 3$ GHz.	83
5.6	Tuning of the INL with control word $d_{t_{r/f},\text{tune}}$ in the DCEI ² V2 code range: (a) INL for $V_{\text{DD}} = 1.08$ V, (b) INL for $V_{\text{DD}} = 1.13$ V, (c) normalized INL[256] for $V_{\text{DD}} = 1.13$ V and variation of f_{ref} with highlighted contour line at INL[256] = 0, and (d) evaluation of relation between Δt_1 and $t_{r/f}$ at the contour line.	85
5.7	Linearity of DCEI ² 's binary bits: (a) DCEI ² V1 with $t_{\text{d,LSB}} = 48.8$ fs at $f_{\text{out}} = 2.5$ GHz, (b) DCEI ² V2 with $t_{\text{d,LSB}} = 97.7$ fs at $f_{\text{out}} = 2.5$ GHz, (c) binary step size related to respective LSB step for (a), (b) binary step size related to respective LSB step for (b), (e) measured DCEI ² V2 at $f_{\text{out}} = 2.2$ GHz and $f_{\text{out}} = 3$ GHz, and (f) binary step size related to respective LSB step for (e).	87
5.8	DTC code sequence and measured DTC output delay at $f_{\text{out}} = 2.4$ GHz: (a) Code sequence triggering the five DCEI ² MSBs, and (b) zoom on a single code transition $n : 0 \rightarrow 128$	88
5.9	Output stage of DTC test chip for 50Ω impedance matching	88

5.10	Dynamic error difference due to active DCEI ² dynamic effects compensation for code jumps (a) $n : 0 \rightarrow k$ (according to (5.2)) and (b) $n : k \rightarrow 0$ (according to (5.3)), for $k \in \{16, 32, 64, 128, 256\}$	89
5.11	Dynamic error comparison for active/inactive MMD dynamic effects compensation for (a) MMD LSB transition $0 \rightarrow 1$, (b) MMD LSB transition $1 \rightarrow 0$, (c) division-by-3, and (d) division-by-5.	91
5.12	Comparison of (a) DNL and (b) INL between: DCEI ($f_{\text{out}} = 2 \text{ GHz}$), CF-DCEI ($f_{\text{out}} = 2 \text{ GHz}$), and DCEI ² V2 ($f_{\text{out}} = 2.496 \text{ GHz}$).	93
6.1	Number of cited DTC publications per year over the last two decades until 02/2017.	95
B.1	Voltage at $V_{\text{int}}(t)$ for $\lambda_p = 0$ (red), $\lambda_p \neq 0$ (blue), and Taylor approximated $\lambda_p \neq 0$ (brown).	107
C.1	Switched capacitor based fine tuning: (a) circuit implementation, and (b) idealized waveforms for ideal input signal.	113
C.2	Linearity of switched capacitor based fine tuning: (a) INL over code, and (b) peak INL for FS variation through variation of C_{max}	117

List of Tables

1.1	Accumulator output for $M = 8$ and $FCW = 3.8$	7
2.1	DTC configurations for the investigated PI designs.	24
3.1	Control of a DCEI ² cell array with $K = 2^{(k_{PI}-1)}$ cells: number of DCEI ² unit cells in each select state for different codes.	45
3.2	Key specifications of DCEI ² test chip configurations.	45
3.3	Extraction of DCEI ² delays for the first interpoaltion with equivalent DCEI models.	48
3.4	Logic states of a thermometrically controlled 1 bit PI including coarse tuning.	50
3.5	Binary bit extension of a 1 bit PI: (a) conventional extension leading to a missing programming code, and (b) proposed extension.	51
3.6	Binary cell control for a 2 bit binary extension to a 7 bit DCEI ² cell array.	52
3.7	Comparison of DCEI, CF-DCEI, and DCEI ² by simulation data.	57
4.1	Dynamic error test cases for $f_{out} = 2560$ MHz.	68
4.2	Average of σ ($INL_{dyn}[n]$) over code for different compensation settings and test cases.	76
4.3	Overview on all identified dynamic effects, their root-causes, and possible countermeasures to reduce dynamic errors.	77
5.1	Comparison of measurement results from DCEI, CF-DCEI, and DCEI ² based DTC, and comparison to PI simulation results (without further DTC stages).	94
6.1	Comparison of developed DTCs to recently published gigahertz domain DTC designs.	97
A.1	Piecewise defined DCEI ODEs.	104
B.1	Overview on 1 st order model equations for $\lambda = 0$	106
B.2	Overview on 2 nd order model regions.	107
B.3	Overview on 2 nd order model propagation delay functions.	109
B.4	Overview on 2 nd order model INL equations.	111
C.1	Switched capacitor model parameters.	117

List of References

- [1] J. Z. Ru, C. Palattella, P. Geraedts, E. Klumperink, and B. Nauta, “A high-linearity digital-to-time converter technique: Constant-slope charging,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1412–1423, June 2015.
- [2] M.-S. Chen, A. Hafez, and C.-K. K. Yang, “A 0.1-1.5 GHz 8-bit inverter-based digital-to-phase converter using harmonic rejection,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 11, pp. 2681–2692, Nov. 2013.
- [3] F. Baronti, L. Fanucci, D. Lunardini, R. Roncella, and R. Saletti, “A high-resolution DLL-based digital-to-time converter for DDS applications,” in *IEEE Int. Frequency Control Symposium and PDA Exhibition*, May 2002, pp. 649–653.
- [4] S. Callender and A. Niknejad, “A phase-adjustable delay-locked loop utilizing embedded phase interpolation,” in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, June 2011, pp. 1–4.
- [5] J.-M. Chou, Y.-T. Hsieh, and J.-T. Wu, “A 125MHz 8b digital-to-phase converter,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2003, pp. 436–505.
- [6] S. Joshi, J. Liao, Y. Fan, S. Hyvonen, M. Nagarajan, J. Rizk, H.-J. Lee, and I. Young, “A 12-Gb/s transceiver in 32-nm bulk CMOS,” in *Symp. on VLSI Circuits*, June 2009, pp. 52–53.
- [7] A. Ravi, P. Madoglio, H. Xu, K. Chandrashekar, M. Verhelst, S. Pellerano, L. Cuellar, M. Aguirre-Hernandez, M. Sajadieh, J. Zarate-Roldan, O. Bochobza-Degani, H. Lakdawala, and Y. Palaskas, “A 2.4-GHz 20–40-MHz channel WLAN digital outphasing transmitter utilizing a delay-based wideband phase modulator in 32-nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 3184–3196, Dec. 2012.
- [8] T. Rapinoja, Y. Antonov, K. Stadius, and J. Ryyänen, “Fractional-N open-loop digital frequency synthesizer with a post-modulator for jitter reduction,” in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, May 2016, pp. 130–133.
- [9] K. Ryu, D. H. Jung, and S. O. Jung, “All-digital process-variation-calibrated timing generator for ATE with 1.95-ps resolution and a maximum 1.2-GHz test rate,” in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2013, pp. 41–44.
- [10] J. Lemberg, M. Kosunen, E. Roverato, M. Martelius, K. Stadius, L. Anttila, M. Valkama, and J. Ryyänen, “Digital interpolating phase modulator for wideband outphasing transmitters,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 5, pp. 705–715, May 2016.

- [11] A. A. Abidi, "Phase noise and jitter in CMOS ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, Aug. 2006.
- [12] P. Das and B. Amrutur, "An accurate fractional period delay generation system," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 7, pp. 1924–1932, July 2012.
- [13] P. Madoglio, H. Xu, K. Chandrashekar, L. Cuellar, M. Faisal, W. Y. Li, H. S. Kim, K. M. Nguyen, Y. Tan, B. Carlton, V. Vaidya, Y. Wang, T. Tetzlaff, S. Suzuki, A. Fahim, P. Seddighrad, J. Xie, Z. Zhang, D. S. Vemparala, A. Ravi, S. Pellerano, and Y. Palaskas, "A 2.4GHz WLAN digital polar transmitter with synthesized digital-to-time converter in 14nm Trigate/FinFET technology for IoT and wearable applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 226–227.
- [14] A. T. Narayanan, M. Katsuragi, K. Kimura, S. Kondo, K. K. Tokgoz, K. Nakata, W. Deng, K. Okada, and A. Matsuzawa, "A fractional-N sub-sampling PLL using a pipelined phase-interpolator with an FoM of -250 dB," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1630–1640, July 2016.
- [15] Y.-C. Choi, S.-S. Yoo, and H.-J. Yoo, "A fully digital polar transmitter using a digital-to-time converter for high data rate system," in *IEEE Int. Symp. on Radio-Frequency Integration Technology (RFIT)*, Jan. 2009, pp. 56–59.
- [16] P. Hanumolu, V. Kratyuk, G.-Y. Wei, and U.-K. Moon, "A sub-picosecond resolution 0.5–1.5 GHz digital-to-phase converter," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 414–424, Feb. 2008.
- [17] T. Rapinoja, K. Stadius, L. Xu, S. Lindfors, R. Kaunisto, A. Parssinen, and J. Ryyänen, "A digital frequency synthesizer for cognitive radio spectrum sensing applications," in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, June 2009, pp. 423–426.
- [18] T. Rapinoja, K. Stadius, L. Xu, S. Lindfors, R. Kaunisto, A. Parssinen, and J. Ryyänen, "A digital frequency synthesizer for cognitive radio spectrum sensing applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 5, pp. 1339–1348, May 2010.
- [19] T. Rapinoja, L. Xu, K. Stadius, and J. Ryyänen, "Implementation of all-digital wideband RF frequency synthesizers in 65-nm CMOS technology," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, May 2011, pp. 1948–1951.
- [20] S. Al-Ahdab, A. Mäntyniemi, and J. Kostamovaara, "A 12-bit digital-to-time converter (DTC) for time-to-digital converter (TDC) and other time domain signal processing applications," in *NORCHIP*, Nov. 2010, pp. 1–4.
- [21] S. Alahdab, A. Mäntyniemi, and J. Kostamovaara, "A 12-bit digital-to-time converter (DTC) with sub-ps-level resolution using current DAC and differential switch for time-to-digital converter (TDC)," in *IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC)*, May 2012, pp. 2668–2671.

-
- [22] A. Ba, Y.-H. Liu, J. van den Heuvel, P. Mateman, B. Busze, J. Gloudemans, P. Vis, J. Dijkhuis, C. Bachmann, G. Dolmans, K. Philips, and H. de Groot, "A 1.3nJ/b IEEE 802.11ah fully digital polar transmitter for IoE applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 440–441.
- [23] M. M. Bajestan, H. Attah, and K. Entesari, "A 2.8-4.3GHz wideband fractional-N sub-sampling synthesizer with -112.5dBc/Hz in-band phase noise," in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, May 2016, pp. 126–129.
- [24] X. Gao, O. Burg, H. Wang, W. Wu, C. T. Tu, K. Manetakis, F. Zhang, L. Tee, M. Yayla, S. Xiang, R. Tsang, and L. Lin, "A 2.7-to-4.3GHz, 0.16ps_{rms}-jitter, -246.8dB-FOM, digital fractional-N sampling PLL in 28nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 174–175.
- [25] S. Kundu, B. Kim, and C. H. Kim, "A 0.2-to-1.45GHz subsampling fractional-N all-digital MDLL with zero-offset aperture PD-based spur cancellation and in-situ timing mismatch detection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 326–327.
- [26] N. Markulic, K. Raczkowski, P. Wambacq, and J. Craninckx, "A 10-bit, 550-fs step digital-to-time converter in 28nm CMOS," in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2014, pp. 79–82.
- [27] N. Markulic, K. Raczkowski, E. Martens, P. E. P. Filho, B. Hershberg, P. Wambacq, and J. Craninckx, "A DTC-based subsampling PLL capable of self-calibrated fractional synthesis and two-point modulation," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 12, pp. 3078–3092, Dec. 2016.
- [28] N. Pavlovic and J. Bergervoet, "A 5.3GHz digital-to-time-converter-based fractional-N all-digital PLL," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011, pp. 54–56.
- [29] K. Raczkowski, N. Markulic, B. Hershberg, and J. Craninckx, "A 9.2-12.7 GHz wideband fractional-N subsampling PLL in 28 nm CMOS with 280 fs RMS jitter," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1203–1213, May 2015.
- [30] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. Lacaita, "A 2.9–4.0-GHz fractional-N digital PLL with bang-bang phase detector and 560-fs_{rms} integrated jitter at 4.5-mW power," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec. 2011.
- [31] M. Kosunen, J. Lemberg, M. Martelius, E. Roverato, T. Nieminen, M. Englund, K. Stadius, L. Anttila, J. Pallonen, M. Valkama, and J. Ryynänen, "A 0.35-to-2.6GHz multilevel outphasing transmitter with a digital interpolating phase modulator enabling up to 400MHz instantaneous bandwidth," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 224–225.
- [32] Y. Wu, M. Shahmohammadi, Y. Chen, P. Lu, and R. B. Staszewski, "A 3.5-6.8GHz wide-bandwidth DTC-assisted fractional-N all-digital PLL with a MASH $\Delta\Sigma$ TDC

- for low in-band phase noise,” in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2016, pp. 209–212.
- [33] Y. He, Y.-H. Liu, T. Kuramochi, J. van den Heuvel, B. Busze, N. Markulic, C. Bachmann, and K. Philips, “A $673\mu\text{W}$ 1.8-to-2.5GHz dividerless fractional-N digital PLL with an inherent frequency-capture capability and a phase-dithering spur mitigation for IoT applications,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 420–421.
- [34] A. Elkholy, A. Elshazly, S. Saxena, G. Shu, and P. K. Hanumolu, “A 20-to-1000MHz $\pm 14\text{ps}$ peak-to-peak jitter reconfigurable multi-output all-digital clock generator using open-loop fractional dividers in 65nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 272–273.
- [35] S. Kumaki, A. H. Johari, T. Matsubara, I. Hayashi, and H. Ishikuro, “A 0.5V 6-bit scalable phase interpolator,” in *IEEE Asia Pacific Conf. on Circuits and Systems (APCCAS)*, Dec. 2010, pp. 1019–1022.
- [36] R. Nandwana, T. Anand, S. Saxena, S.-J. Kim, M. Talegaonkar, A. Elkholy, W.-S. Choi, A. Elshazly, and P. Hanumolu, “A calibration-free fractional-N ring PLL using hybrid phase/current-mode phase interpolation method,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 882–895, April 2015.
- [37] A. T. Narayanan, M. Katsuragi, K. Kimura, S. Kondo, K. K. Tokgoz, K. Nakata, W. Deng, K. Okada, and A. Matsuzawa, “A fractional-N sub-sampling PLL using a pipelined phase-interpolator with a FOM of -246dB,” in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2015, pp. 380–383.
- [38] A. Nicholson, J. Jenkins, A. van Schaik, T. Hamilton, and T. Lehmann, “A 1.2V 2-bit phase interpolator for 65nm CMOS,” in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 2012, pp. 2039–2042.
- [39] S. Sidiropoulos and M. Horowitz, “A semidigital dual delay-locked loop,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1683–1692, Nov. 1997.
- [40] D. W. Jee, Y. Suh, B. Kim, H. J. Park, and J. Y. Sim, “A FIR-embedded phase interpolator based noise filtering for wide-bandwidth fractional-N PLL,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 11, pp. 2795–2804, Nov. 2013.
- [41] T. A. D. Riley, M. A. Copeland, and T. A. Kwasniewski, “Delta-sigma modulation in fractional-N frequency synthesis,” *IEEE Journal of Solid-State Circuits*, vol. 28, no. 5, pp. 553–559, May 1993.
- [42] A. Ravi, P. Madoglio, M. Verhelst, M. Sajadieh, M. Aguirre, H. Xu, S. Pellerano, I. Lomeli, J. Zarate, L. Cuellar, O. Degani, H. Lakdawala, K. Soumyanath, and Y. Palaskas, “A 2.5GHz delay-based wideband OFDM outphasing modulator in 45nm-LP CMOS,” in *Symp. on VLSI Circuits*, June 2011, pp. 26–27.

-
- [43] J. Sonntag and R. Leonowich, "A monolithic CMOS 10 MHz DPLL for burst-mode data retiming," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 1990, pp. 194–195.
- [44] H. Mair and L. Xiu, "An architecture of high-performance frequency and phase synthesis," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 835–846, June 2000.
- [45] D. E. Calbaza and Y. Savaria, "A direct digital period synthesis circuit," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1039–1045, Aug. 2002.
- [46] S. Talwalkar, "Quantization error spectra structure of a DTC synthesizer via the DFT axis scaling property," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 6, pp. 1242–1250, June 2012.
- [47] S. Talwalkar, "Digital-to-time synthesizers: Separating delay line error spurs and quantization error spurs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2597–2605, Oct. 2013.
- [48] R. Kreienkamp, U. Langmann, C. Zimmermann, T. Aoyama, and H. Siedhoff, "A 10-Gb/s CMOS clock and data recovery circuit with an analog phase interpolator," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 736–743, March 2005.
- [49] B. Abiri, R. Shivnaraine, A. Sheikholeslami, H. Tamura, and M. Kibune, "A 1-to-6Gb/s phase-interpolator-based burst-mode CDR in 65nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011, pp. 154–156.
- [50] S. Hu, C. Jia, K. Huang, C. Zhang, X. Zheng, and Z. Wang, "A 10Gbps CDR based on phase interpolator for source synchronous receiver in 65nm CMOS," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, May 2012, pp. 309–312.
- [51] R. B. Staszewski, K. Muhammad, D. Leipold, C.-M. Hung, Y.-C. Ho, J. L. Wallberg, C. Fernando, K. Maggio, R. Staszewski, T. Jung, J. Koh, S. John, I. Y. Deng, V. Sarda, O. Moreira-Tamayo, V. Mayega, R. Katz, O. Friedman, O. E. Eliezer, E. de Obaldia, and P. T. Balsara, "All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2278–2291, Dec. 2004.
- [52] R. B. Staszewski, J. L. Wallberg, S. Rezeq, C.-M. Hung, O. E. Eliezer, S. K. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phones," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.
- [53] K. Muhammad, Y. C. Ho, T. L. Mayhugh, C. M. Hung, T. Jung, I. Elahi, C. Lin, I. Deng, C. Fernando, J. L. Wallberg, S. K. Vemulapalli, S. Larson, T. Murphy, D. Leipold, P. Cruise, J. Jaehnig, M. C. Lee, R. B. Staszewski, R. Staszewski, and K. Maggio, "The first fully integrated quad-band GSM/GPRS receiver in a 90-nm digital CMOS process," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1772–1783, Aug. 2006.

- [54] S. Henzler, *Time-to-Digital Converters*. Springer, 2010.
- [55] M. Zanuso, S. Levantino, C. Samori, and A. L. Lacaïta, "A wideband 3.6 GHz digital $\Sigma\Delta$ fractional-N PLL with phase interpolation divider and digital spur cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 3, pp. 627–638, March 2011.
- [56] A. Elkholy, T. Anand, W. S. Choi, A. Elshazly, and P. K. Hanumolu, "A 3.7 mW low-noise wide-bandwidth 4.5 GHz digital fractional-N PLL using time amplifier-based TDC," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 867–881, April 2015.
- [57] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. Lacaïta, "A 2.9-to-4.0GHz fractional-N digital PLL with bang-bang phase detector and $560\text{fs}_{\text{rms}}$ integrated jitter at 4.5mW power," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011, pp. 88–90.
- [58] G. Marzin, S. Levantino, C. Samori, and A. Lacaïta, "A 20 Mb/s phase modulator based on a 3.6 GHz digital PLL with -36 dB EVM at 5 mW power," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 2974–2988, Oct. 2012.
- [59] S. Levantino, G. Marzin, and C. Samori, "An adaptive pre-distortion technique to mitigate the DTC nonlinearity in digital PLLs," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 8, pp. 1762–1772, Aug. 2014.
- [60] J. Zhuang and R. B. Staszewski, "A low-power all-digital PLL architecture based on phase prediction," in *IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS)*, Dec. 2012, pp. 797–800.
- [61] V. Chillara, Y.-H. Liu, B. Wang, A. Ba, M. Vidojkovic, K. Philips, H. de Groot, and R. Staszewski, "An $860\mu\text{W}$ 2.1-to-2.7GHz all-digital PLL-based frequency modulator with a DTC-assisted snapshot TDC for WPAN (Bluetooth smart and ZigBee) applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 172–173.
- [62] P. Chen, X. Huang, Y.-H. Liu, M. Ding, C. Zhou, A. Ba, K. Philips, H. De Groot, and R. Staszewski, "Design and built-in characterization of digital-to-time converters for ultra-low power ADPLLs," in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2015, pp. 283–286.
- [63] S. Levantino, "Advanced digital phase-locked loops," in *IEEE Custom Integrated Circuits Conf. (CICC)*, Sept. 2013, pp. 1–95.
- [64] S. Levantino, "Bang-bang digital PLLs," in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2016, pp. 329–334.
- [65] P. C. Huang, W. S. Chang, and T. C. Lee, "A 2.3GHz fractional-N dividerless phase-locked loop with -112dBc/Hz in-band phase noise," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 362–363.

-
- [66] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A 2.2GHz 7.6mW sub-sampling PLL with -126dBc/Hz in-band phase noise and 0.15ps_{rms} jitter in 0.18 μ m CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2009, pp. 392–393,393a.
- [67] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M. J. E. Lee, R. Rathi, and J. Poulton, "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, Dec. 2002.
- [68] S. Levantino, G. Marucci, G. Marzin, A. Fenaroli, C. Samori, and A. L. Lacaita, "A 1.7 GHz fractional-N frequency synthesizer based on a multiplying delay-locked loop," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 11, pp. 2678–2691, Nov. 2015.
- [69] G. Marucci, A. Fenaroli, G. Marzin, S. Levantino, C. Samori, and A. L. Lacaita, "A 1.7GHz MDLL-based fractional-N frequency synthesizer with 1.4ps RMS integrated jitter and 3mW power using a 1b TDC," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 360–361.
- [70] W. S. Chang, P. C. Huang, and T. C. Lee, "A fractional-N divider-less phase-locked loop with a subsampling phase detector," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2964–2975, Dec. 2014.
- [71] K. Raczkowski, N. Markulic, B. Hershberg, J. V. Driessche, and J. Craninckx, "A 9.2-12.7 GHz wideband fractional-N subsampling PLL in 28 nm CMOS with 280 fs RMS jitter," in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, June 2014, pp. 89–92.
- [72] P. Park, J. Park, H. Park, and S. Cho, "An all-digital clock generator using a fractionally injection-locked oscillator in 65nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012, pp. 336–337.
- [73] A. Elkholy, A. Elmallah, M. Elzeftawi, K. Chang, and P. K. Hanumolu, "A 6.75-to-8.25GHz, 250fs_{rms}-integrated-jitter 3.25mW rapid on/off PVT-insensitive fractional-N injection-locked clock multiplier in 65nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 192–193.
- [74] Y. H. Liu, J. van den Heuvel, T. Kuramochi, B. Busze, P. Mateman, V. K. Chillara, B. Wang, R. B. Staszewski, and K. Philips, "An ultra-low power 1.7-2.7 GHz fractional-N sub-sampling digital frequency synthesizer and modulator for IoT applications in 40 nm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–12, Dec. 2016, early access article.
- [75] H. Guo and T. Kwasniewski, "A DLL fractional M/N frequency synthesizer," in *IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE)*, May 2015, pp. 114–117.

- [76] N. Nidhi and S. Pamarti, "A 1.8GHz wideband open-loop phase modulator with TDC based non-linearity calibration in 0.13 μ m CMOS," in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, May 2015, pp. 91–94.
- [77] P. E. Su and S. Pamarti, "A 2.4 GHz wideband open-loop GFSK transmitter with phase quantization noise cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 3, pp. 615–626, March 2011.
- [78] S. Zheng and H. C. Luong, "A WCDMA/WLAN digital polar transmitter with low-noise ADPLL, wide-band PM/AM modulator and linearized PA in 65nm CMOS," in *European Solid-State Circuits Conf. (ESSCIRC)*, Sept. 2014, pp. 375–378.
- [79] M. E. Heidari, M. Lee, and A. A. Abidi, "All-digital outphasing modulator for a software-defined transmitter," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1260–1271, April 2009.
- [80] K.-W. Kim, S. Byun, K. Lim, C.-H. Lee, and J. Laskar, "A 600MHz CMOS OFDM LINC transmitter with a 7 bit digital phase modulator," in *IEEE Radio Frequency Integrated Circuits Symp. (RFIC)*, June 2008, pp. 677–680.
- [81] N. Nidhi, P.-E. Su, and S. Pamarti, "Open-loop wide-bandwidth phase modulation techniques," *IEEE Journal of Electrical and Computer Engineering*, vol. 2011, no. 507381, Aug. 2011.
- [82] J. Groe, "Polar transmitters for wireless communications," *IEEE Communications Magazine*, vol. 45, no. 9, pp. 58–63, Sept. 2007.
- [83] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sept. 1959.
- [84] T. Saeki, M. Mitsuishi, H. Iwaki, and M. Tagishi, "A 1.3-cycle lock time, non-PLL/DLL clock multiplier based on direct clock cycle interpolation for "clock on demand"," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1581–1590, Nov. 2000.
- [85] A. Agrawal, J. F. Bulzacchelli, T. O. Dickson, Y. Liu, J. A. Tierno, and D. J. Friedman, "A 19-Gb/s serial link receiver with both 4-tap FFE and 5-tap DFE functions in 45-nm SOI CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 3220–3231, Dec. 2012.
- [86] T. Dickson, Y. Liu, S. Rylov, A. Agrawal, S. Kim, P.-H. Hsieh, J. Bulzacchelli, M. Ferriss, H. Ainspan, A. Rylyakov, B. Parker, M. Beakes, C. Baks, L. Shan, Y. Kwark, J. Tierno, and D. Friedman, "A 1.4 pJ/bit, power-scalable 16 \times 12 Gb/s source-synchronous I/O with DFE receiver in 32 nm SOI CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 8, pp. 1917–1931, Aug. 2015.
- [87] T. Carusone, D. Johns, and K. Martin, *Analog Integrated Circuit Design*. Wiley, 2012.
- [88] P. Allen and D. Holberg, *CMOS Analog Circuit Design*, 3rd ed. OUP USA, 2012.

-
- [89] K. Kundert, “Predicting the phase noise and jitter of PLL based frequency synthesizers,” in *Phase-Locking in High-Performance Systems: From Devices to Architectures*. Wiley, 2003.
- [90] K. Kundert, “Modeling and simulation of jitter in phase-locked loops,” in *Analog Circuit Design*. Springer US, 1997.
- [91] K. Kundert, “Introduction to RF simulation and its application,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 9, pp. 1298–1319, Sept. 1999.
- [92] A. Hajimiri, S. Limotyrakis, and T. Lee, “Jitter and phase noise in ring oscillators,” *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, pp. 790–804, June 1999.
- [93] *Virtuoso Spectre Circuit Simulator RF Analysis Theory*, Cadence Design Systems, Inc., December 2009.
- [94] J. Rogers and C. Plett, *Radio Frequency Integrated Circuit Design*. Artech House, 2010.
- [95] D. Weinlader, “Precision CMOS receivers for VLSI testing applications,” Ph.D. dissertation, Stanford Univ., 2001.
- [96] H. Shichman and D. A. Hodges, “Modeling and simulation of insulated-gate field-effect transistor switching circuits,” *IEEE Journal of Solid-State Circuits*, vol. 3, no. 3, pp. 285–289, Sept. 1968.
- [97] J. Rabaey, A. Chandrakasan, and B. Carusone, *Digital Integrated Circuits: A Design Perspective*, ser. Prentice Hall electronics and VLSI series. Prentice Hall, 2003.
- [98] P. Hazucha, T. Karnik, B. A. Bloechel, C. Parsons, D. Finan, and S. Borkar, “Area-efficient linear regulator with ultra-fast load regulation,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 933–940, April 2005.
- [99] C. Palattella, E. Klumperink, J. Ru, and B. Nauta, “A sensitive method to measure the integral non-linearity of a digital-to-time converter based on phase modulation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 741–745, Aug. 2015.
- [100] C. Wendl, “Messautomatisierung zur Untersuchung eines digitalen Phasenschiebers für Mobilfunksysteme,” Diploma thesis, Kempten University of Applied Science, March 2013, unpublished.
- [101] S. Sievert, “Comparison of DTC imperfections in lab measurements and circuit simulations,” Master’s thesis, Technical University of Munich, Oct. 2013, unpublished.

List of Author Publications

- [A1] S. Sievert, A. Ben-Bassat, O. Degani, and R. Banin, “System for digitally controlled edge interpolator linearization,” U.S. Patent 9,407,245, Granted Aug. 2, 2016.
- [A2] S. Sievert, O. Degani, A. Ben-Bassat, R. Banin, A. Ravi, B. U. Klepser, Z. Boos, and D. Schmitt-Landsiedel, “A 2GHz 244fs-resolution 1.2ps-peak-INL edge-interpolator-based digital-to-time converter in 28nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016.
- [A3] S. Sievert, O. Degani, A. Ben-Bassat, R. Banin, A. Ravi, W. Thomann, B.-U. Klepser, Z. Boos, and D. Schmitt-Landsiedel, “A 2 GHz 244 fs-resolution 1.2 ps-peak-INL edge interpolator-based digital-to-time converter in 28 nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 12, pp. 2992 – 3004, Dec. 2016.
- [A4] S. Sievert, O. Degani, A. Ravi, and R. Banin, “An apparatus and a method for generating a radio frequency signal,” E.U. Patent Application PCT/IB2015/057 375, Filed Sept. 25, 2015.
- [A5] O. Degani, R. Banin, and S. Sievert, “Digitally controlled two-points edge interpolator,” U.S. Patent Application 14/868,834, Filed Sept. 29, 2015.
- [A6] S. Sievert, O. Degani, and E. Gordon, “Low drop-out compensation technique for reduced dynamic errors in digital-to-time converters,” U.S. Patent Application 15/200,495, Filed Aug. 8, 2016.
- [A7] S. Sievert, S. Zur, O. Degani, and R. Banin, “Phase interpolator, apparatus for phase interpolation, digital-to-time converter, and methods for phase interpolation,” E.U. Patent Application EP16 205 835.8, Filed Dec. 21, 2016.
- [A8] O. Degani, R. Banin, A. Ben-Bassat, and S. Sievert, “An apparatus for interpolating between a first signal edge and a second signal edge, a method for controlling such apparatus, and an interpolation cell for a digital-to-time converter,” E.U. Patent Application EP16 205 845.7, Filed Dec. 21, 2016.
- [A9] R. Banin, E. Nassar, I. Falkov, E. Fayneh, O. Degani, and S. Sievert, “Scalable interleaved digital-to-time converter circuit for clock generation,” U.S. Patent Application 15/391,575, Filed Dec. 27, 2016.

Acknowledgments

The research for the present thesis was conducted during my employment as doctoral candidate at Intel Deutschland GmbH in the RF Innovations department.

First, I want to thank Prof. Dr. rer. nat. Doris Schmitt-Landsiedel for the opportunity to conduct the research in cooperation with Intel Deutschland GmbH, and apl. Prof. Dr.-Ing. habil. Helmut Gräß for continuing the supervision of my dissertation after the retirement of Prof. Dr. rer. nat. Doris Schmitt-Landsiedel. Both provided me with valuable feedback on my work during its different stages and helped me to increase the quality of my dissertation.

Furthermore, I would like to thank Dr. Ofir Degani and Zdravko Boos for giving me the opportunity to work on such an interesting topic in their innovation teams at Intel. Dr. Degani supervised my research and provided continuous support and guidance. He encouraged me to think beyond the state-of-the-art technology and implement my own ideas, leading to several filed patent applications. Next to the technical challenging task, I highly appreciate the work experience in an intercultural team from Israel, Austria, USA and Germany. I got a very warm welcome during all my trips to different Intel sites worldwide.

Special thanks go to all my colleagues at Intel without whom my research would not have been possible. I had many discussions about DTC circuit design with Rotem Banin (who also organizes amazing team events), Assaf Ben-Bassat, Dr. Ashoke Ravi (special thanks also for his tour guiding competencies in all parts of Israel during our businesses trips together), Sarit Zur, Wolfgang Thomann, Dr. Bernd-Ulrich Klepser, Dr. Yorgos Palaskas and Dr. Stefano Pellerano. From DTC system perspective Peter Preyler, Elan Banin and Dr. Stefan Tertinek were never tired of discussions and explanations. Eshel Gordon provided support for circuit design of the integrated supply voltage regulator in which the dynamic effects compensation circuitry was implemented. Very special thanks go to Koren Solimani and Mosche Simchi, who were designing the physical layout for my circuit blocks. As the chip verification with high measurement accuracy was a challenging task, I also thank Yakov Gutkin, Thomas Maletz, Shaya Danziger, Roland Vuketich, Nabil Alomari and Christian Wendl for support in setting up the measurement environment and during bring-up of the test chips. Further thanks go to all my colleagues in Israel for pointing me to the best Falafel and Hummus places in the country.

Finally, I want to thank my wife Sarah for her encouragement to pursue the doctoral degree and her support during all times of my work.